HP-UX 시스템 관리 설명서: 보안 관리
HP-UX 11i v3

HP 제품 번호: 5992-3402
2008년 3월 발행
제4판

알림

이 설명서의 내용은 예고 없이 변경될 수 있습니다.

HP는 이 설명서에 대해 상업성이나 특정 목적에의 적합성에 대한 묵시적 보증 등 어떠한 종류의 보증도 하지 않습니다. HP는 이 설명서의 오류나 공급, 수행 또는 사용에 따른 직접적, 간접적, 특수적, 부수적 또는 파생적인 손해에 대해 책임을 지지 않습니다.

보증서 HP 제품에 적용되는 특정 보증서 사본과 교체 부품은 지역 대리점이나 서비스 센터에서 구할 수 있습니다.

미국 정부 라이센스 독점적인 컴퓨터 소프트웨어입니다. 소유, 사용 또는 복사를 위해서는 HP로부터 유효한 라이센스를 취득해야 합니다. FAR 12.211 및 12.212에 따라 상업용 컴퓨터 소프트웨어, 컴퓨터 소프트웨어 문서 및 상업용 품목의 기술 데이터는 공급업체의 표준 상업용 라이센스에 의거하여 미국 정부에 사용이 허가되었습니다.

상표권 UNIX®는 미국과 다른 국가에서 등록된 상표이며, The Open Group을 통해 독점적으로 라이센스를 받았습니다. VERITAS®는 Symantec Corporation의 등록 상표입니다.

승인 이 제품에는 Apache Software Foundation에서 개발한 소프트웨어가 포함되어 있습니다. 이 설명서는 Apache Software Foundation(http://www.apache.org)의 정보를 기반으로 합니다.

이 제품에는 OpenSSL Toolkit에 사용하기 위해 OpenSSL Toolkit(http://www.openssl.org)에서 개발한 소프트웨어가 포함되어 있습니다.

목차

설명서 정보.....13
I 시스템 보호.....19

1 안전하게 HP-UX 운영 환경 설치.....21
1.1 설치 보안 고려 사항.....21
1.2 부팅 프로세스 중 보안 문제 방지.....21
1.3 root에 대한 로그인 보안 활성화.....22
1.4 부팅 인증을 사용하여 권한 없는 액세스 방지.....22
1.5 설치 시간 보안 옵션 설정.....23
1.6 보안 패치 설치.....23
1.7 백업 및 복구에 관한 설치 후 보안 팁.....23

2 사용자 및 시스템 보안 관리.....25
2.1 사용자 액세스 관리.....25
2.1.1 사용자 계정 모니터링.....25
2.1.2 게스트 계정 모니터링.....26
2.1.3 응용 프로그램 사용자 계정 만들기.....26
2.1.4 그룹 계정 관리.....27
2.2 로그인 중 사용자 인증.....27
2.2.1 login 프로세스 설명.....28
2.2.2 로그인 추적 파일(btmp 및 wtmp) 확인.....29
2.2.2.1 last 명령 예제.....29
2.2.3 로그인한 사용자 확인.....30
2.3 PAM을 사용하여 사용자 인증.....30
2.3.1 개요.....30
2.3.2 PAM 라이브러리.....31
2.3.3 /etc/pam.conf를 사용한 시스템 범위 구성.....32
2.3.4 샘플 /etc/pam.conf 파일.....33
2.3.5 /etc/pam_user.conf 사용자 구성 파일.....34
2.3.6 예제: PAM이 로그인에 대해 작동하는 방식.....34
2.4 암호 관리.....35
2.4.1 시스템 관리자의 책임.....36
2.4.2 사용자 책임.....36
2.4.3 좋은 암호의 기준.....36
2.4.4 /etc/passwd 암호 파일 변경.....37
2.4.4.1 passwd 명령 예제.....37
2.4.4.2 /etc/passwd 파일 형식.....38
2.4.5 /etc/shadow 섀도 패스워드 파일.....38
2.4.6 /etc/passwd에서 의사 계정 제거 및 주요 하위 시스템 보호.....39

2.4.7 HP-UX Secure Shell을 사용한 로그인 보안 유지....40
2.4.8 NIS에 저장된 암호 보안 유지....40
2.4.9 LDAP 디렉토리 서버에 저장된 암호 보안 유지....40
2.5 시스템 보안 속성 정의....40
2.5.1 시스템 범위 속성 구성....41
2.5.2 사용자별 속성 구성....42
2.5.2.1 userdbset를 사용한 사용자별 속성 정의 예제....43
2.5.2.2 INACTIVITY_MAXDAYS 및 섀도 패스워드 파일....43
2.5.3 사용자 데이터베이스 문제 해결....43
2.6 setuid 및 setgid 프로그램 처리....43
2.6.1 setuid 및 setgid 프로그램이 위험할 수 있는 이유....44
2.6.2 ID가 설정되는 방법....45
2.6.3 setuid 기능 제한 지침....45
2.7 스택 버퍼 오버플로 공격 방지....46
2.8 무인 터미널 및 워크스테이션 보호....46
2.8.1 /etc/inittab 및 실행 수준을 사용하여 액세스 제어....47
2.8.2 터미널 장치 파일 보호....47
2.8.3 화면 잠금 구성....48
2.8.3.1 TMOUT 변수 구성....48
2.8.3.2 CDE 잠금 관리자 구성....48
2.9 원격 장치의 시스템 액세스 방지....48
2.9.1 /etc/dialups 및 /etc/d_passwd를 사용하여 액세스 제어....49
2.10 로그인 배너 보안 유지....50
2.11 root 계정 보호....51
2.11.1 root 계정 액세스 모니터링....51
2.11.2 제한된 수퍼유저 액세스를 위해 Restricted SMH Builder 사용....51
2.11.3 수퍼유저 액세스 검토....52

3 HP-UX Bastille....53
3.1 특징 및 장점....53
3.2 HP-UX Bastille 설치....53
3.3 HP-UX Bastille 사용....54
3.3.1 대화형으로 HP-UX Bastille 사용....54
3.3.2 비대화형으로 HP-UX Bastille 사용....56
3.3.3 시스템 구성....56
3.4 HP-UX Bastille을 사용하여 변경 사항 되돌리기....58
3.5 파일 위치....58
3.6 팁 및 문제 해결....59
3.7 HP-UX Bastille 제거....60

4 HP-UX Standard Mode Security Extensions....63
4.1 개요....63
4.2 보안 속성 및 사용자 데이터베이스....64

4.2.1 시스템 보안 속성....64
4.2.2 시스템 범위 속성 구성....64
4.2.3 사용자 데이터베이스 구성 요소....65
4.2.3.1 구성 파일....65
4.2.3.2 명령....65
4.2.3.3 속성....65
4.2.3.4 맨페이지....66
4.2.4 사용자 데이터베이스에서 속성 구성....66
4.2.5 사용자 데이터베이스 문제 해결....67

5 원격 액세스 보안 관리....69
5.1 인터넷 서비스 및 원격 액세스 서비스 개요....69
5.1.1 ftp 보안 유지....70
5.1.2 익명 ftp 보안 유지....70
5.1.3 /etc/ftpd/ftpusers를 사용하여 액세스 거부....71
5.1.4 스푸핑에 대한 기타 보안 솔루션....71
5.2 inetd 데몬....72
5.2.1 inetd 보안 유지....73
5.2.1.1 /var/adm/inetd.sec를 사용하여 액세스 거부 또는 허용....73
5.3 TCP Wrappers를 사용하여 스푸핑 방지....73
5.3.1 TCP Wrappers의 추가 기능....74
5.3.2 TCP Wrappers는 RPC 서비스와 함께 작동하지 않음....74
5.4 Secure Internet Services....74
5.5 관리 도메인 제어....75
5.5.1 네트워크 제어 파일의 사용 권한 설정 확인....76
5.6 HP-UX SSH(Secure Shell)를 사용하여 원격 세션 보안 유지....76
5.6.1 HP-UX Secure Shell의 주요 보안 기능....77
5.6.2 HP-UX Secure Shell의 소프트웨어 구성 요소....77
5.6.3 HP-UX Secure Shell 실행....78
5.6.3.1 ssh 클라이언트 실행....78
5.6.3.2 sftp 클라이언트 실행....79
5.6.3.3 scp 클라이언트 실행....79
5.6.4 HP-UX Secure Shell 권한 분리....79
5.6.5 HP-UX Secure Shell 인증....80
5.6.5.1 GSS-API....81
5.6.5.2 공개 키 인증....81
5.6.5.3 호스트 기반 및 공개 키 인증....81
5.6.5.4 암호 인증....81
5.6.6 통신 프로토콜....82
5.6.7 HP-UX Secure Shell 및 HP-UX 시스템....82
5.6.8 연관된 기술....82
5.6.9 Strong Random Number Generator 요구 사항....83
5.6.10 TCP Wrappers 지원....83

5.6.11 chroot 디렉토리 jail....83
II 데이터 보호....85

6 파일 시스템 보안....87
6.1 파일 액세스 제어....87
6.1.1 파일 액세스 권한 설정....88
6.1.2 파일 소유권 설정....89
6.1.3 디렉토리 보호....89
6.1.4 사용자 계정과 관련된 파일 보호....90
6.1.5 fsck를 사용하여 파일 손상 검색 및 수정....90
6.2 액세스 제어 목록 설정....91
6.3 HFS ACL 사용....91
6.3.1 HFS ACL 및 HP-UX 명령과 호출....92
6.4 JFS ACL 사용....94
6.4.1 JFS ACL 정의....94
6.4.2 시스템에서 JFS ACL을 생성하는 방법....94
6.4.3 최소 JFS ACL....94
6.4.4 추가 JFS ACL user 및 group 항목....95
6.4.5 JFS ACL group 및 class 항목....95
6.4.6 setacl 및 getacl 명령 사용....96
6.4.7 class 항목에 대한 chmod의 영향....96
6.4.8 최소 JFS ACL 변경 예제....96
6.4.9 기본 JFS ACL....98
6.4.10 setacl 명령을 사용하여 JFS ACL 변경....99
6.4.10.1 수정 및 삭제 옵션 사용....99
6.4.10.2 -f 옵션 사용....99
6.4.10.3 유효 사용 권한 및 setacl -n....99
6.5 JFS ACL과 HFS ACL 비교....100
6.5.1 JFS와 HFS의 명령 및 함수 매핑....100
6.6 ACL 및 NFS....101
6.7 /dev 장치 특수 파일의 보안 고려 사항....101
6.8 디스크 파티션 및 논리 볼륨 보호....102
6.9 파일 시스템 마운트 및 마운트 해제에 대한 보안 지침....103
6.10 네트워크의 파일 보안 제어....104
6.10.1 네트워크 제어 파일의 사용 권한 설정 확인....104
6.10.2 NFS 환경에 마운트된 파일....104
6.10.2.1 서버 취약성....105
6.10.2.2 클라이언트 취약성....105
6.10.2.3 NFS 마운트된 파일을 보호하는 방법....105

7 구획....107
7.1 개요....107
7.1.1 구획 아키텍처....107

7.1.2 기본 구획 구성.....109
7.2 구획 구조 계획.....109
7.3 구획 활성화.....110
7.4 구획 구성 수정.....110
7.4.1 구획 규칙 변경.....111
7.4.2 구획 이름 변경.....111
7.5 구획 구성 요소.....111
7.5.1 구획 구성 파일.....111
7.5.2 구획 명령.....112
7.5.3 구획 맨페이지.....112
7.6 구획 규칙 및 구문.....112
7.6.1 구획 정의.....113
7.6.2 파일 시스템 규칙.....114
7.6.3 IPC 규칙.....115
7.6.4 네트워크 규칙.....116
7.6.5 기타 규칙.....118
7.6.6 예제 규칙 파일.....119
7.7 구획의 응용 프로그램 구성.....119
7.8 구획 문제 해결.....119
7.9 discover 모드를 사용하여 초기 구획 구성 생성.....120
7.10 HP Serviceguard 클러스터의 구획.....120

8 Fine-grained 권한.....123
8.1 개요.....123
8.2 Fine-grained 권한 구성 요소.....123
8.2.1 명령.....123
8.2.2 맨페이지.....124
8.3 사용 가능한 권한.....124
8.4 Fine-grained 권한을 사용하여 응용 프로그램 구성.....126
8.4.1 권한 모델.....127
8.4.2 복합 권한.....128
8.5 Fine-grained 권한의 보안 관련 사항.....128
8.5.1 권한 에스컬레이션.....128
8.6 HP Serviceguard 클러스터의 Fine-grained 권한.....128
8.7 Fine-grained 권한 문제 해결.....128
III ID 보호.....131

9 HP-UX Role-Based Access Control.....133
9.1 개요.....133
9.2 액세스 제어 기본 사항.....134
9.2.1 역할을 사용하여 액세스 제어 단순화.....134
9.3 HP-UX RBAC 구성 요소.....135
9.3.1 HP-UX RBAC 액세스 제어 정책 전환.....136

9.3.2 HP-UX RBAC 구성 파일....136
9.3.3 HP-UX RBAC 명령....137
9.3.4 HP-UX RBAC 맨페이지....137
9.3.5 HP-UX RBAC 아키텍처....138
9.3.6 HP-UX RBAC 예제 사용 및 작업....139
9.4 HP-UX RBAC 배포 실행....140
9.4.1 역할 계획....140
9.4.2 역할의 권한 부여 계획....141
9.4.3 명령 매핑 계획....141
9.4.4 HP-UX RBAC 제한 사항....142
9.5 HP-UX RBAC 구성....142
9.5.1 역할 구성....143
9.5.1.1 역할 만들기....143
9.5.1.2 사용자에게 역할 할당....144
9.5.1.3 그룹에 역할 할당....145
9.5.2 권한 부여 구성....145
9.5.3 추가 명령 권한 부여 및 권한 구성....146
9.5.4 Fine-grained 권한을 사용하여 HP-UX RBAC 구성....147
9.5.5 구획을 사용하여 HP-UX RBAC 구성....149
9.6 HP-UX RBAC 사용....149
9.6.1 privrun 명령을 통해 권한을 사용하여 응용 프로그램 실행....150
9.6.1.1 Serviceguard 클러스터의 HP-UX RBAC....151
9.6.2 privedit 명령을 사용하여 액세스가 제어되는 파일 편집....152
9.6.3 ACPS를 사용하여 privrun 및 privedit 사용자 정의....153
9.7 HP-UX RBAC 문제 해결....153
9.7.1 rbacdbchk 데이터베이스 구문 도구....153
9.7.2 privrun -v 정보....154

10 감사 관리....155
10.1 감사 구성 요소....155
10.1.1 명령....156
10.1.2 감사 구성 파일....156
10.1.3 감사 맨페이지....156
10.2 시스템 감사....157
10.2.1 감사 구현 계획....157
10.2.2 감사 활성화....157
10.2.3 감사 비활성화....158
10.2.4 감사 파일 모니터링....158
10.2.5 성능 고려 사항....158
10.2.6 감사 시스템 관리 지침....159
10.3 사용자 감사....159
10.4 감사 이벤트....160
10.5 감사 기록....161

10.5.1 감사 기록 구성.....162
10.5.2 감사 기록 모니터링 및 관리.....162
10.6 감사 로그 보기.....163
10.6.1 audisp 명령 사용 예제.....164
10.7 자체 감사.....164
10.8 HP-UX RBAC 감사.....165
10.8.1 HP-UX RBAC 기준 및 /etc/rbac/aud_filter 파일.....166
10.8.2 HP-UX RBAC 기준 감사 절차.....166

A 트러스트된 시스템.....169
A.1 트러스트된 시스템 설치.....169
A.2 트러스트된 시스템 감사.....170
A.3 트러스트된 암호 및 시스템 액세스 관리.....170
A.3.1 암호 파일.....171
A.3.1.1 /etc/passwd 파일.....171
A.3.1.2 /tcb/files/auth/ 데이터베이스.....172
A.3.2 암호 선택 사항 및 생성.....173
A.3.3 암호 변경기간 설정 기능.....173
A.3.4 암호 내역 및 암호 재사용.....173
A.3.5 시간 기반 액세스 제어.....174
A.3.6 장치 기반 액세스 제어.....174
A.3.7 트러스트된 시스템 데이터베이스 조작.....174
A.4 트러스트된 백업 및 복구 지침.....175

B 기타 보안 제품.....177
B.1 HP-UX HIDS.....177
B.2 보안 패치.....177
B.3 HP-UX IPFilter.....177
B.4 HP-UX Secure Shell 177

용어.....179

색인.....187

그림 목록

2-1 PAM 아래의 HP-UX 인증 모듈....31
3-1 HP-UX Bastille 사용자 인터페이스....55
6-1 파일 및 디렉토리 사용 권한 필드....88
7-1 구획 아키텍처....108
9-1 HP-UX RBAC 아키텍처....139
9-2 privrun을 호출한 후 예제 작업....140

표 목록

3-1 HP-UX Bastille 질문 모듈.....55
4-1 사용자 데이터베이스 구성 파일.....65
4-2 사용자 데이터베이스 명령.....65
4-3 사용자 속성.....65
4-4 사용자 데이터베이스 맨페이지.....66
5-1 인터넷 서비스 구성 요소와 액세스 확인, 권한 부여 및 인증.....69
5-2 HP-UX Secure Shell의 소프트웨어 구성 요소.....77
6-1 파일 및 디렉토리 권한의 차이점.....88
6-2 HFS ACL 명령.....93
6-3 HFS ACL 시스템 호출.....93
6-4 ACL 항목에 영향을 주는 명령 및 호출.....93
6-5 HFS 및 JFS ACL 매핑.....101
7-1 구획 구성 파일.....111
7-2 구획 명령.....112
7-3 구획 맨페이지.....112
8-1 Fine-grained 권한 명령.....124
8-2 Fine-grained 권한 맨페이지.....124
8-3 사용 가능한 권한.....124
9-1 사용자별 권한 부여 예제.....134
9-2 역할별 권한 부여 예제.....135
9-3 HP-UX RBAC 구성 파일.....136
9-4 HP-UX RBAC 명령.....137
9-5 HP-UX RBAC 맨페이지.....137
9-6 예제 계획 결과.....143
10-1 감사 명령.....156
10-2 감사 구성 파일.....156
10-3 감사 맨페이지.....156
10-4 audevent 명령 옵션.....160

보기 목록

2-1 의사 및 특별 시스템 계정.....40
6-1 HFS ACL 만들기.....92
6-2 여러 HFS ACL 일치 항목.....92

설명서 정보

발행 정보

발행 날짜와 제품 번호로 설명서의 버전을 확인할 수 있습니다. 새로운 버전의 설명서가 발행될 때 발행 날짜가 변경됩니다.

새 버전의 설명서를 받으려면 해당 제품 지원 서비스에 가입해야 합니다. 자세한 내용은 HP 영업 담당자에게 문의하십시오.

다음 웹 사이트에서 이 설명서의 여러 버전을 찾을 수 있습니다.

http://docs.hp.com/ko

2008년 3월 제품 번호 5992-3402

- **시스템 보호**, **데이터 보호** 및 **ID 보호**의 세 개 부분으로 문서가 구성됨
- HP-UX Standard Mode Security Extensions 문서에 대한 장이 추가됨(4장 참조)
- Security Patch Check가 Software Assistant로 교체됨
- HP-UX Bastille 사용자 인터페이스를 표시하는 그림이 추가됨(섹션 3.3 참조)
- HP-UX Bastille 구성 로그 파일 `assessment-log.config`가 추가됨(섹션 3.5 참조)
- 그 밖에 여러 가지가 편집됨

2007년 10월 제품 번호 5992-2916

- HP-UX Bastille에 대해 설명하는 장이 추가됨

2007년 8월 제품 번호 5992-1933

- 섀도 패스워드를 지원하지 않는 제품 목록에서 PRM(Process Resource Manager)이 제거됨(섹션 2.4.5 참조)
- `permission_list`에서 `search`가 `nsearch`로 수정됨(섹션 7.6.2 참조)

2007년 2월 제품 번호 5991-6494

제1판

참고: **HP-UX 시스템 관리 설명서**의 각 볼륨은 독립적으로 업데이트할 수 있습니다. 따라서 각 볼륨의 최신 버전이 시기별로 다를 수 있으며 서로 일치하지 않을 수도 있습니다. 각 볼륨의 최신 버전은 다음 웹 사이트에서 볼 수 있습니다.

http://docs.hp.com(영문) 및 http://docs.hp.com/ko(한글)

대상 독자

HP-UX 시스템 관리 설명서는 HP-UX 릴리즈 11i v3부터 HP-UX 시스템을 관리해야 하는 모든 기술 수준의 HP-UX 시스템 관리자를 위해 작성되었습니다.

이 설명서 세트의 많은 항목은 이전 릴리즈에 적용되지만 HP-UX 11i v3에서 변경된 사항도 많습니다. 따라서 이전 릴리즈에 대한 자세한 내용은 **시스템 및 작업 그룹 관리, 시스템 관리자를 위한 설명서**를 참조하십시오.

설명서 세트 정보

HP-UX 시스템 관리 설명서에서는 HP-UX 11i v3을 실행하는 시스템 관리에 필요한 핵심적인 작업 세트 및 관련 개념에 대해 설명합니다.

개요 HP-UX 11i 와 그 구성 요소, 이들 사이의 관계에 대해 자세히 설명합니다.

구성 관리 시스템 설정 및 하위 시스템의 동작을 구성 및 사용자 정의하기 위해 수행해야 할 작업 중 많은 부분에 대해 설명합니다.

논리 볼륨 관리 HP LVM(Logical Volume Manager)을 사용하여 물리 볼륨, 볼륨 그룹 및 논리 볼륨을 구성하는 방법에 대해 설명합니다.

보안 관리 HP-UX 11i의 데이터 및 시스템 보안 기능에 대해 설명합니다.

루틴 관리 작업 시스템이 원활하게 실행되도록 유지하기 위해 수행해야 할 지속적인 여러 작업에 대해 설명합니다.

HP-UX 시스템 관리 설명서: 보안 관리는 **시스템 보호**, **데이터 보호** 및 **ID 보호**의 세 개 부분으로 구성됩니다. 각 부분의 내용은 다음과 같습니다.

1장 부팅 및 설치 프로세스와 관련된 보안 고려 사항에 대해 설명합니다.

2장 운영 체제가 설치된 후 사용자 및 시스템 보안을 관리하는 방법에 대해 설명합니다.

3장 HP-UX 운영 체제의 보안을 강화시키기 위해 HP-UX Bastille을 사용하여 시스템을 강화하고 잠그는 방법에 대해 설명합니다.

4장 HP-UX Standard Mode Security Extentions의 기능 및 구성 요소에 대해 설명합니다.

5장 시스템에 대한 원격 액세스를 보호하는 방법에 대해 설명합니다.

6장 파일 시스템을 제어 및 보호하는 방법에 대해 설명합니다.

7장 시스템의 구성 요소를 서로 분리하는 방법 및 구획에 대해 설명합니다.

8장 수퍼유저의 기능을 권한 집합으로 나누는 방법 및 Fine-grained 권한에 대해 설명합니다.

9장 HP-UX Role-Based Access Control의 기능 및 구성 요소에 대해 설명합니다.

10장 감사 시스템의 관리에 대해 설명합니다.

부록 A 트러스트된 시스템에 대해 설명합니다.

부록 B 기타 보안 제품에 대해 설명합니다.

HP-UX 11i 릴리즈 이름 및 릴리즈 ID

HP-UX 11i를 사용하여 HP는 가용성이 높고 관리 가능한 보안 운영 체제를 제공합니다. HP-UX 11i는 업무상 중요한 엔터프라이즈 기술 컴퓨팅 환경을 지원하며 HP 9000 시스템과 HP Integrity 서버에서 모두 사용할 수 있습니다.

각 HP-UX 11i 릴리즈에는 관련된 릴리즈 이름과 릴리즈 ID가 있습니다. `uname` 명령에 `-r` 옵션을 사용하면 릴리즈 ID가 반환됩니다. HP-UX 11i에 사용 가능한 릴리즈 목록은 다음 표를 참조하십시오.

릴리즈 ID	릴리즈 이름	지원되는 프로세서 아키텍처
B.11.11	HP-UX 11i v1	HP 9000
B.11.23	HP-UX 11i v2	Intel™ Itanium™
B.11.23	HP-UX 11i v2, 2004년 9월	HP 9000 Itanium
B.11.31	HP-UX 11i v3	HP 9000 Itanium

HP-UX 11i의 각각의 버전에서 지원되는 시스템 및 프로세서 아키텍처에 대한 자세한 내용은 실행 중인 HP-UX 버전의 HP-UX 11i 시스템 릴리즈 노트를 참조하십시오. 예를 들어, **HP-UX 11i v3 릴리즈 노트**를 참조할 수 있습니다.

HP-UX 정보 찾기

다음 표에는 HP-UX에 대한 일반 시스템 관리 정보를 찾을 수 있는 위치가 나와 있습니다. 그러나 특정 제품에 대한 정보는 들어 있지 않습니다.

수행할 작업	참조 자료	위치
다음 사항 확인: • HP-UX 릴리즈에서 변경된 내용 • 운영 환경의 내용 • 특정 릴리즈의 펌웨어 요구 사항 및 지원되는 시스템	사용하는 HP-UX 버전과 관련된 HP-UX 11i 릴리즈 노트. 예를 들어, **HP-UX 11i v3 릴리즈 노트**를 참조할 수 있습니다.	• HP Instant Information 미디어 • **http://docs.hp.com**(영문) 및 **http://docs.hp.com/ko**(한글) • `/usr/share/doc/` 디렉토리 `/usr/share/doc` 디렉토리에는 사용자의 HP-UX 버전에 해당하는 원본 릴리즈 노트만이 들어 있습니다. 개정된 릴리즈 노트에 대해서는 최신 HP Instant Information 미디어 또는 **http://docs.hp.com**(영문) 및 **http://docs.hp.com/ko**(한글)을 참조하십시오.
HP-UX 설치 또는 업데이트	• **HP-UX를 설치 또는 업데이트하기 전에** • **HP-UX 11i 설치 및 업데이트 설명서** 참고: 해당 HP-UX 버전에 대한 설명서를 참조하십시오.	• 미디어 키트(운영 환경과 함께 제공됨) • HP Instant Information 미디어 • **http://docs.hp.com**(영문) 및 **http://docs.hp.com/ko**(한글)
HP-UX 시스템 관리	HP-UX 11i v3 이후 릴리즈: • **HP-UX 시스템 관리 설명서**(다중 볼륨 모음) 기타 시스템 관리 정보 소스: • **nPartition 관리 설명서** • Planning Superdome Configurations(백서)	• HP Instant Information CD-ROM • **http://docs.hp.com**(영문) 및 **http://docs.hp.com/ko**(한글) • Planning Superdome Configurations(백서)

관련 정보

보안 및 HP-UX에 대한 추가 정보는 http://docs.hp.com에서 볼 수 있습니다. 특히 다음 설명서를 참조할 수 있습니다.

- **HP-UX AAA Server Administrator's Guide**
- **HP-UX Host Intrusion Detection System Administrator's Guide**
- **HP-UX IPFilter Administrator's Guide**
- **HP-UX IPSec Administrator's Guide**
- **HP-UX Secure Shell Release Notes**

표기법

이 설명서에서 사용하는 표기법은 다음과 같습니다.

reboot(1M)	HP-UX 맨페이지입니다. **reboot**는 명령 이름이고 **1M**은 **HP-UX Reference**의 절입니다. 웹 및 Instant Information 미디어에서는 맨페이지 자체에 대한 링크일 수 있습니다. HP-UX 명령줄에서 "`man reboot`" 또는 "`man 1M reboot`"를 입력하면 맨페이지가 표시됩니다. 자세한 내용은 *man*(1)을 참조하십시오.
Book Title	책 제목입니다. 웹 및 Instant Information 미디어에서는 책 자체에 대한 링크일 수 있습니다.
KeyCap	키보드 키 이름입니다. **Return** 키와 **Enter** 키는 같은 키를 나타냅니다.
Emphasis	강조 텍스트입니다.
Emphasis	강한 강조 텍스트입니다.
Term	중요한 단어나 문구를 소개합니다.
`ComputerOut`	컴퓨터 화면에 표시되는 텍스트입니다.
`UserInput`	사용자가 입력하는 명령이나 텍스트입니다.
`Command`	명령 이름이거나 명령 구문입니다.
`Variable`	명령이나 함수에서 대체할 수 있는 변수 이름이거나 가능한 값이 여러 개인 정보를 표시할 때 사용합니다.
[ ]	형식 및 명령 설명에서 선택적인 내용입니다.
{ }	형식 및 명령 설명에서 필수적인 내용입니다. 내용이 |로 구분된 목록인 경우 그 중 하나를 선택해야 합니다.
. . .	앞에 있는 요소를 원하는 만큼 반복할 수 있습니다.
|	선택 목록에서 항목을 구분합니다.

I부 시스템 보호

엔터프라이즈 보안에서 가장 중요한 요소는 시스템 최소화 및 강화입니다. HP-UX 11i는 필요한 서비스만 실행하여 잠재적인 공격 지점을 최소화함으로써 알려진 취약성과 알려지지 않은 취약성을 해결하도록 설계된 보안 기능 집합을 제공합니다.

이 절에서는 시스템을 공격으로부터 보호하고 위협을 감지 및 대응하는 다음 HP-UX 도구에 대해 설명합니다.

- 안전하게 HP-UX 운영 환경 설치(1장)
- 사용자 및 시스템 보안 관리(2장)
- HP-UX Bastille(3장)
- Standard Mode Security Extensions(4장)
- 원격 액세스 보안 관리(5장)

1 안전하게 HP-UX 운영 환경 설치

이 장에서는 부팅 및 설치 프로세스와 관련된 보안 고려 사항에 대해 설명합니다. 이 장의 내용은 다음과 같습니다.

- 설치 보안 고려 사항(섹션 1.1)
- 부팅 프로세스 중 보안 문제 방지(섹션 1.2)
- root에 대한 로그인 보안 활성화(섹션 1.3)
- 부팅 인증을 사용하여 권한 없는 액세스 방지(섹션 1.4)
- 설치 시간 보안 옵션 설정(섹션 1.5)
- 보안 패치 설치(섹션 1.6)
- 백업 및 복구에 관한 설치 후 보안 팁(섹션 1.7)

1.1 설치 보안 고려 사항

새 운영 체제를 설치하거나 업데이트하기 전에 보안 고려 사항을 확인해야 합니다. 다음 보안 방법을 설치 준비 과정의 일부로 삼으십시오.

- 미디어 키트의 내용을 검토합니다. http://docs.hp.com(영문) 및 http://docs.hp.com(한글)에서 릴리즈 노트 및 기타 관련 정보를 참조하십시오.
- 필요한 소프트웨어와 필요 없는 소프트웨어를 결정합니다. 필요 없는 소프트웨어는 설치하지 마십시오. 보안 소프트웨어 제품 결정에 대한 도움말은 이 설명서의 다른 장을 참조하십시오.
- 보안 수정이 완료될 때까지 네트워크, 특히 공개 네트워크에서 시스템 연결을 끊거나 해제합니다. 배포하려는 보안의 수준(있을 경우)을 고려합니다. 자세한 내용은 섹션 1.5를 참조하십시오.
- 시스템 콘솔이 물리적으로 보호되어 있으며 LAN 콘솔의 연결이 끊어져 있거나 `telnet`과 같은 일반 텍스트 프로토콜이 허용/보호되는 네트워크를 통해서만 사용되는지 확인합니다. 이는 중요한 보안 고려 사항입니다. 시스템 콘솔에 대한 액세스를 제한하면 권한 없는 사람이 시스템의 보안 설정을 변경하는 것을 방지할 수 있습니다.
- 최신 패치, 특히 보안 패치를 설치합니다. 자세한 내용은 섹션 1.6을 참조하십시오.
- 백업 및 복구 시스템을 유지 관리합니다. 자세한 내용은 섹션 1.7을 참조하십시오.

1.2 부팅 프로세스 중 보안 문제 방지

부팅 시퀀스 중에 보안 문제가 발생할 수 있습니다. 부팅 프로세스가 인터럽트되어 권한 없는 사용자가 시스템에 액세스할 수 있습니다. 특정 시스템 파일이 다시 부팅되기 전에 잘못 또는 악의적으로 변경된 경우 다시 부팅하는 도중 및 다시 부팅한 후에 시스템에서 문제가 발생할 수 있습니다. 따라서 다음과 같은 예방 작업을 수행합니다.

- 시스템과 시스템 콘솔이 물리적으로 안전하며 권한이 부여된 사용자만 액세스할 수 있는지 확인합니다.
- 부팅 인증 기능을 통해 지정된 사용자만 시스템을 단일 사용자 모드로 부팅할 수 있도록 제한합니다. 자세한 내용은 섹션 1.4를 참조하십시오.

- 시스템 파일이 쓰기 금지되어 있는지 확인합니다. 일부 파일은 읽기 금지해야 할 수도 있습니다.

다음은 컴퓨터를 켜거나 다시 설정할 때 발생하는 부팅 시퀀스를 요약한 내용입니다. 부팅 시퀀스에 대한 자세한 내용은 **HP-UX 시스템 관리 설명서: 루틴 관리 작업**을 참조하십시오.

1. 부팅 중에 자동 부팅 시퀀스를 재정의할 수 있는 약 10초 간의 대기 시간이 있습니다. 이때 침입자가 부팅 시퀀스를 인터럽트하고 시스템에 침투할 수 있습니다.

   아무 키나 눌러 부팅 시퀀스를 인터럽트하면 root 액세스 권한을 얻을 수 있습니다. ISL에서 명령을 입력하라는 메시지를 표시합니다. 다음 명령을 입력하면 시스템이 단일 사용자 모드가 됩니다.

   ```
   ISL> hpux -is
   ```

   부팅 인증을 사용하지 않는 경우 암호 없이 root로 로그인할 수 있습니다.

   부팅 인증을 사용하면 지정된 사용자만 root로 로그인할 수 있습니다.

2. 부팅 시퀀스가 인터럽트되지 않으면 초기화 프로세스가 계속됩니다.
3. HP-UX는 초기화 과정을 거친 다음 일반 작업을 시작하고 로그인 준비를 마칩니다. 이때 침입자가 이미 root 액세스 권한을 얻은 경우 다른 보안 문제가 발생할 수 있습니다.

침입자가 부팅 프로세스를 인터럽트하면 시스템에 대한 root 액세스 권한을 얻어 이론상 해당 시스템을 소유하게 됩니다. 시스템을 소유하면 다양한 메커니즘을 통해 시스템을 변경할 수 있습니다.

1.3 root에 대한 로그인 보안 활성화

`rlogind` 및 `telnetd`와 같은 많은 네트워크 프로토콜은 네트워크 통신을 암호화하지 않으므로 침입자가 쉽게 네트워크에서 관리 암호를 스니핑할 수 있습니다. 이러한 비보안 프로토콜은 사용을 최소화하십시오.

이러한 프로토콜을 통한 관리 로그인을 차단하려면 `/etc/securetty` 파일을 사용하여 시스템 콘솔에서만 root 계정에 로그인하도록 제한할 수 있습니다. 예를 들어, root 로그인을 콘솔로만 제한하려면 콘솔로 구성된 한 줄이 포함된 `/etc/security` 파일을 만듭니다. 자세한 내용은 *login*(1)을 참조하십시오.

1.4 부팅 인증을 사용하여 권한 없는 액세스 방지

부팅 인증 기능은 암호 인증을 사용하여 단일 사용자 모드 부팅을 보호합니다. 권한이 부여된 사용자만 시스템을 단일 사용자 모드로 부팅할 수 있도록 시스템을 구성할 수 있습니다. 시스템을 다시 부팅하기 전에 부팅 인증 기능을 활성화해야 합니다.

부팅 인증은 `/etc/default/security` 파일의 다음 두 개 속성에 의해 구성됩니다.

- `BOOT_AUTH`는 부팅 인증을 활성화하거나 비활성화합니다. 부팅 인증을 활성화하려면 `BOOT_AUTH=1`을 지정합니다. 기본적으로 인증은 비활성화되어 있습니다(`BOOT_AUTH=0`).
- `BOOT_USERS`는 부팅 인증 기능이 활성화될 때 root로 로그인할 수 있는 사용자를 정의합니다. `BOOT_USERS`에 나열된 이름은 쉼표로 구분되어 있습니다. 예를 들면 다음과 같습니다.

  `BOOT_USERS=root,mary,jack,amy,jane`

  `BOOT_USERS=root`(기본값)

`/etc/default/security` 구성 파일에 대해서는 2장 및 *security*(4)에서 설명합니다.

1.5 설치 시간 보안 옵션 설정

설치 시간 보안 옵션을 사용하여 HP-UX Bastille 보안 잠금 엔진을 구성할 수 있습니다. 이 엔진에는 HP-UX IPFilter 방화벽이 포함될 수 있습니다. 시스템 설치가 완료된 후 이 엔진은 미리 구성된 보안 수준 중 하나를 갖게 됩니다.

설치 중에 미리 구성된 다음 네 가지 보안 수준 중 하나를 선택할 수 있습니다.

Sec00Tools	선택적 보안 기능을 활성화하지 않고 보안 인프라를 설치합니다. 이것이 기본값입니다.
Sec10Host	HP-UX IPFilter 방화벽 구성 없이 호스트 기반 잠금 시스템을 설치합니다. 이 보안 수준에서는 대부분의 네트워크 서비스가 비활성화됩니다. *bastille*(1M) 명령을 실행하여 이러한 서비스를 복구할 수 있습니다.
Sec20MngDMZ	HP-UX IPFilter 방화벽을 사용하여 들어오는 트래픽을 대부분 차단하는 관리되는 잠금 시스템을 설치합니다.
Sec30DMZ	호스트 기반의 IPFilter 네트워크 잠금인 DMZ 전체 잠금 시스템을 설치합니다. HP-UX IPFilter는 들어오는 연결을 거의 모두 차단합니다.

HP-UX Bastille에 대한 자세한 내용은 3장을 참조하십시오. HP-UX IPFilter에 대한 자세한 내용은 다음 웹 사이트에 있는 **HP-UX IPFilter Administrator's Guide**를 참조하십시오.

http://docs.hp.com/en/internet.html#IPFilter

1.6 보안 패치 설치

설치 후 즉시 HP-UX SWA(Software Assistant)를 사용하여 필수 패치와 권장 패치를 적용합니다.

SWA는 HP-UX 시스템에서 패치 관리 및 보안 정보 관리를 통합 및 간소화하는 명령줄 기반 도구입니다. SWA 도구는 SPC(Security Patch Check)를 대신하며, HP에서 게시한 HP-UX 소프트웨어 관련 보안 정보로 최신 상태를 유지하는 데 사용되는 HP 권장 유틸리티입니다.

참고: Software Assistant 소프트웨어 도구를 사용하면 시스템 보안을 향상시킬 수 있지만 시스템 보안이 보장되지는 않습니다.

SWA에 대한 자세한 내용은 다음을 참조하십시오.

- HP-UX Software Assistant 웹 사이트:
  https://www.hp.com/go/swa
- *swa*(1M)
- 다음 위치에 있는 **HP-UX Software Assistant System Administration Guide**
  http://docs.hp.com

1.7 백업 및 복구에 관한 설치 후 보안 팁

시스템이 실행된 후에도 보안을 유지 관리해야 합니다. 지속적으로 시스템 백업 및 복구 파일을 유지 관리하십시오. 몇 가지 고려할 지침은 다음과 같습니다.

- `fbackup` 및 `frecover` 명령만 사용하여 선택적으로 파일을 백업하고 복구합니다. `fbackup` 및 `frecover`를 사용하는 경우에만 ACL(액세스 제어 목록)이 유지됩니다. ACL을 구현하지 않는 시스템에서 사용할 파일을 백업하고 복구할 경우에는 이러한 명령에 `-A` 옵션을 사용합니다. *fbackup*(1M) 및 *frecover*(1M)를 참조하십시오.
- 다른 시스템에 파일을 복구할 경우 두 시스템에서 사용자의 사용자 이름과 그룹 이름이 일치해야 합니다.
- 백업 미디어는 민감한 요소이므로 필요한 경우에만 미디어에 대한 액세스를 허용합니다.
- 백업 테이프에 레이블을 붙인 다음 안전하게 보관합니다. 외부 저장소에 보관하는 것이 보안상 가장 좋습니다. 최소 6개월 동안 아카이브를 보관한 다음 미디어를 다시 사용합니다.
- 매일 증분 백업을 수행하고 매주 전체 백업을 수행합니다.

  조직 내 정보의 흐름에 백업 일정을 맞춥니다. 예를 들어, 주로 사용하는 데이터베이스가 금요일마다 업데이트되면 금요일 밤에 주별 백업을 예약할 수 있습니다.
- 일정에 따라 모든 파일을 백업해야 하는 경우에는 백업하기 전에 모든 사용자에게 로그오프하도록 요청합니다. `fbackup` 명령은 백업이 수행될 때 파일이 변경되고 있을 경우 경고 메시지를 표시합니다.
- 최근 백업의 로그 파일을 조사하여 백업하는 동안 발생한 문제를 식별합니다. 백업 로그 파일에 제한적인 사용 권한을 설정합니다.
- `frecover` 명령을 사용할 경우 파일을 덮어쓸 수 있습니다. 그러나 파일에는 파일이 백업될 때 설정된 사용 권한 및 ACL이 유지됩니다.
- 사전에 복구 프로세스를 테스트하여 응급 상황에 데이터 전체를 복구할 수 있는지 확인합니다.
- 다른 시스템으로부터 파일을 복구할 때 새 시스템에 사용자 및 그룹이 없으면 `chown` 명령을 실행하여 현재 상주하고 있는 시스템에 사용자 ID 및 그룹 ID를 설정할 수 있습니다. 지정된 그룹이 없는 새 시스템에 파일을 복구하면 파일은 `frecover` 명령을 실행하고 있는 사용자의 그룹 소유권을 갖게 됩니다. 서로 다른 시스템에서 소유자 및 그룹 이름의 의미가 다르면 복구 결과를 예측할 수 없으며 원하는 결과가 나타나지 않을 수 있습니다.
- 정전이 발생해도 파일이 손실되지 않아야 하지만 사용자가 정전 후에 파일이 손실되었다고 보고할 경우 백업 테이프에서 복원하기 전에 `/lost+found` 디렉토리에서 해당 파일을 찾아봅니다.
- 복구할 테이프의 내용을 확인하려면 `frecover` 명령에 `-I` 옵션을 사용하여 테이프에 있는 파일의 인덱스를 미리 봅니다. 파일 시스템의 기존 사용 권한은 백업에서 그대로 유지됩니다. `frecover` 명령은 파일의 사용 권한에서 금지할 경우 사용자가 파일을 읽을 수 없게 합니다.
- `/etc/passwd` 또는 `/tcb/files`에 있는 파일과 같이 중요한 파일은 제자리에 복구하지 마십시오. 대신 임시 디렉토리(`/tmp`를 사용하면 안 됨)에 파일을 복원하고 이 디렉토리 사용 권한을 `drwx------`로 지정하여 다른 사용자가 사용할 수 없게 합니다. 복원된 파일을 대체할 파일과 비교하여 필요한 만큼 변경합니다.
- 감사를 설정해야 합니다. 시스템을 복구할 때 감사는 자동으로 활성화되지 않습니다.

2 사용자 및 시스템 보안 관리

이 장에서는 운영 체제가 설치된 후의 기본 사용자 보안에 대해 다루며 로그인, 암호 및 시스템과의 기타 사용자 상호 작용을 중심으로 설명합니다. 이 장의 내용은 다음과 같습니다.

- 사용자 액세스 관리(섹션 2.1)
- 로그인 중 사용자 인증(섹션 2.2)
- PAM을 사용하여 사용자 인증(섹션 2.3)
- 암호 관리(섹션 2.4)
- 시스템 보안 속성 정의(섹션 2.5)
- setuid 및 setgid 프로그램 처리(섹션 2.6)
- 스택 버퍼 오버플로 공격 방지(섹션 2.7)
- 무인 터미널 및 워크스테이션 보호(섹션 2.8)
- 원격 장치의 시스템 액세스 방지(섹션 2.9)
- 로그인 배너 보안 유지(섹션 2.10)
- root 계정 보호(섹션 2.11)

2.1 사용자 액세스 관리

권한이 부여된 사용자는 유효한 사용자 이름(로그인 이름)과 암호를 제공하여 시스템에 액세스할 수 있습니다. 각 사용자는 `/etc/passwd` 파일에서 한 개 항목으로 정의됩니다. HP SMH(System Management Homepage)를 사용하여 사용자 계정을 추가, 제거, 비활성화, 다시 활성화 또는 수정할 수 있습니다.

암호에 대한 자세한 내용은 *passwd*(4), *passwd*(1) 및 이 설명서의 섹션 2.4를 참조하십시오.

2.1.1 사용자 계정 모니터링

다음은 사용자 계정 모니터링에 관한 지침입니다.

- `last`, `lastb` 및 `who` 명령의 출력을 정기적으로 검사하여 비정상적인 로그인을 확인합니다.
- 계정을 가진 모든 사용자에게 시스템에 액세스할 정당한 비즈니스 요구가 있는지 확인합니다.
- 여러 사용자가 같은 사용자 계정을 공유할 경우 주의해야 합니다. 두 사용자가 같은 사용자 계정을 공유할 수 없게 하십시오.
- 같은 UID(사용자 ID)를 공유하는 사용자 계정이 없는지 확인합니다.
- 모든 계정에 정기적으로 변경되는 보안 암호가 있는지 확인합니다.
- 모든 사용자 홈 디렉토리에 적절한 사용 권한이 있는지 확인합니다. 대부분의 홈 디렉토리에는 다른 사용자에 대한 읽기 권한이 있지만 쓰기 권한은 없습니다. 보호 기능을 향상시키려면 디렉토리 소유자에 대해서만 읽기, 쓰기 및 실행 권한을 설정합니다.

- 모든 사용자가 보안 정책을 이해하는지 확인합니다. 회사 보안 정책 파일을 각 홈 디렉토리에 저장합니다.
- /etc/passwd 파일 또는 기타 적합한 사용자 데이터베이스를 검사하여 사용되지 않는 계정 및 특히 회사를 퇴직한 사용자의 계정을 확인합니다.
- root 계정을 검사하여 root 액세스 권한을 가진 사용자를 확인합니다.
- 여러 사용자가 root 계정에 액세스할 수 있는 경우와 연관된 위험을 최소화하려면 HP-UX Role-based Access Control을 구현합니다. 자세한 내용은 9장을 참조하십시오.
- 게스트 계정을 검사하여 사용 빈도를 확인합니다.

2.1.2 게스트 계정 모니터링

최상위 수준의 보안을 위해 게스트 계정이나 열린 계정을 허용하지 마십시오. 게스트 계정이 있을 경우 다음을 수행합니다.

- 게스트 암호를 자주 변경합니다. 사용자가 암호를 지정할 수 있습니다.
- 제한된 셸(rsh)을 사용하여 시스템 액세스를 제한합니다. rsh 명령에 대한 자세한 내용은 *sh*(1) 및 *sh-posix*(1)를 참조하십시오.
- 게스트 계정을 잊어버리는 경우가 많습니다. 사용하고 있지 않은 게스트 계정을 비활성화하려면 다음 방법 중 하나를 사용합니다.
  - 사용자별 보안 속성을 사용하여 지정된 비활성 기간(일) 후에 자동으로 계정을 비활성화합니다. 자세한 내용은 *security*(4) 및 섹션 2.5.2.2를 참조하십시오.
  - 다음 명령을 사용하여 게스트 계정을 잠급니다.

    ```
    # passwd -l guest
    ```

  - 다음 명령을 사용하여 게스트 계정을 삭제합니다.

    ```
    # userdel guest
    ```

- at 작업을 예약하여 자동으로 임시 계정을 잠급니다.

  ```
  # at now +14 days passwd -l tempacct
  ```

- 정기적으로 /var/adm/wtmp 및 /var/adm/sulog 파일을 검색하여 사용되지 않은 계정을 확인합니다.

자세한 내용은 *sh*(1) 및 *su*(1)를 참조하십시오.

2.1.3 응용 프로그램 사용자 계정 만들기

사용자가 응용 프로그램을 시작하기 위한 용도로만 HP-UX를 사용하는 경우에는 셸에 액세스할 필요가 없습니다. 이러한 사용자는 데이터베이스 관리 시스템과 같은 응용 프로그램만 사용하면 되고 HP-UX 기능에 액세스할 필요가 없습니다.

HP-UX에 대한 액세스를 제한하려면 사용자가 로그인한 후 특정 명령만 실행되도록 /etc/passwd 파일을 수정합니다. /etc/passwd 파일에는 로그인 중에 필요한 중요한 정보가 들어 있습니다.

- 사용자 이름

- 암호화된 암호
- 사용자 ID
- 그룹 ID
- 주석 필드
- 홈 디렉토리
- 로그인 프로그램

일반적으로 로그인 프로그램은 /bin/sh와 같은 셸이지만 셸이 아니어도 됩니다. 응용 프로그램을 로그인 셸로 식별하여 캡티브 계정(사용자를 직접 응용 프로그램에 로그인시키는 계정)을 만들 수 있습니다.

다음은 사용자가 date 명령만 실행할 수 있도록 제한하는 예제입니다. /etc/passwd 항목은 다음과 같습니다.

```
username:rc70x.4,sx2:20:1:run only date command:/home/date:/usr/bin/date
```

로그인 프롬프트에서 사용자는 username과 해당 암호를 입력합니다. date 명령이 실행되면 즉시 사용자가 로그아웃됩니다.

```
login:username
Password:xxxxxx
Tue Nov 14 18:38:38 PDT 2006
```

2.1.4 그룹 계정 관리

그룹이 프로젝트 관련 파일을 공유하거나 액세스해야 하는 경우 다음 단계에 따라 보안을 유지합니다.

1. 각 구성원에 해당하는 항목이 /etc/passwd에 있는지 확인합니다.
2. /etc/group 파일에 그룹에 대한 항목을 만듭니다.
3. 그룹에 대한 공유 디렉토리를 만듭니다.

   ```
   drwxrwx-- root project /home/projects
   ```

4. 각 그룹 구성원의 ~/.profile에 umask를 설정합니다. 다음 예제에서 그룹의 사용자는 파일을 읽고 쓰고 실행할 수 있지만 다른 사용자는 이러한 작업을 수행할 수 없습니다.

   ```
   umask u=rwx,g=rwx, o=
   ```

2.2 로그인 중 사용자 인증

시스템 및 해당 리소스에 액세스하려면 사용자가 로그인해야 합니다. 시스템에 대한 액세스를 제어하여 권한이 없는 사용자가 시스템에 액세스하지 못하도록 차단할 수 있습니다. 그러나 권한이 없는 사용자가 액세스 권한을 얻은 경우에도 리소스를 사용하는 프로그램을 실행하고 시스템 데이터에 액세스하지 못하도록 차단할 수 있습니다. 이 절에서는 사용자 이름을 입력한 시간부터 셸 프롬프트가 표시되는 시간까지의 login 프로세스 중에 발생하는 동작에 대해 설명합니다.

2.2.1 login 프로세스 설명

다음 단계에서는 login 프로세스에 대해 설명합니다. 이 정보는 고유한 사용자 이름을 만들고 암호 보안 정책을 유지 관리하는 것이 얼마나 중요한지 보여 줍니다. 자세한 내용은 *login*(1)을 참조하십시오.

1. 시스템이 설치된 후 데스크탑 로그인 관리자에서 로그인 화면을 표시합니다. CDE(Common Desktop Environment)가 설치된 경우 CDE 로그인 화면이 표시됩니다.
2. `init` 프로그램이 사용자 이름을 묻는 `getty` 프로세스를 시작합니다. 사용자 이름을 입력합니다. `getty` 프로그램이 해당 사용자 이름을 `login` 프로그램에 전달합니다.
3. `login` 프로그램이 `/etc/passwd`에서 사용자 이름을 검색합니다.
   - 사용자 이름이 있으면 `login`이 4단계로 이동합니다.
   - 사용자 이름이 없으면 `login`이 다음 확인을 수행합니다.
     — 암호를 묻는 메시지를 표시합니다(`Password:`).
     — 잘못된 암호를 입력하면 시스템에서 `Invalid login` 오류 메시지를 표시합니다.
     — `/var/adm/btmp` 파일이 있으면 업데이트합니다. `/var/adm/btmp` 파일은 잘못된 로그인 시도를 추적합니다. 자세한 내용은 섹션 2.2.2를 참조하십시오.
     — 3회 연속해서 잘못된 로그인 시도가 있으면 종료됩니다.
4. `login` 프로세스에서 `/etc/passwd` 파일을 확인합니다.
   - 암호 필드가 설정되어 있으면 `login`이 암호를 묻는 메시지를 표시하고 5단계로 이동합니다.
   - 암호 필드가 설정되어 있지 않으면 사용자는 암호를 입력할 필요가 없으며 `login`이 6단계로 이동합니다.
5. `login` 프로세스에서 이 암호를 `/etc/passwd`에 있는 암호화된 암호와 비교합니다.
   - 암호가 일치하면 `login`이 6단계로 이동합니다.
   - 암호가 일치하지 않으면 `login`에서 `Invalid login`을 표시합니다. `login` 프로세스는 3회 연속 로그인 시도를 허용합니다. 사용자가 세 번 잘못된 로그인을 사용하면 `login`이 종료됩니다.
6. `login` 프로세스에서 잘못된 로그인을 추적하는 `/var/adm/wtmp` 파일을 업데이트합니다. 자세한 내용은 섹션 2.2.2를 참조하십시오.

   로그인에 성공하면 사용자 및 그룹 ID, 그룹 액세스 목록 및 작업 디렉토리가 초기화됩니다.
7. `login` 프로세스가 `/etc/passwd` 파일의 명령 필드에 있는 명령을 실행합니다. 일반적으로 명령 필드는 `/bin/ksh`, `/bin/csh` 또는 `/bin/sh`와 같은 셸의 경로 이름입니다. 명령 필드가 비어 있으면 기본값은 `/bin/sh`입니다.

   명령 필드는 셸이 아니어도 됩니다. 다른 명령을 실행하는 예제는 섹션 2.1.3을 참조하십시오.
8. 셸 초기화가 완료된 후 시스템에서 프롬프트를 표시하고 사용자 입력을 기다립니다.

`login` 프로세스에서 PAM(Pluggable Authentication Modules)을 사용하여 추가 사용자 인증을 수행하게 할 수 있습니다. 자세한 내용은 *pam.conf*(4) 및 섹션 2.3을 참조하십시오.

2.2.2 로그인 추적 파일(btmp 및 wtmp) 확인

다음 파일에서 로그인을 기록합니다.

- /var/adm/btmp 파일은 실패한 로그인을 추적합니다.
- /var/adm/wtmp 파일은 성공한 로그인을 추적합니다.

lastb 명령을 사용하여 /var/adm/btmp 파일을 읽고 권한 없는 사용자가 로그인하려고 시도했는지 확인합니다.

last 명령을 사용하여 /var/adm/wtmp 파일을 읽습니다.

last 및 lastb 명령은 가장 최근 사용자 정보를 내림차순으로 표시합니다.

wtmp 및 btmp 파일은 제한 없이 확장되므로 정기적으로 확인합니다. 더 이상 필요 없는 정보를 정기적으로 제거하여 파일이 너무 커지지 않도록 합니다. wtmp 및 btmp 파일은 해당 파일을 유지 관리하는 프로그램에서 만들어지지 않습니다. 이 파일을 제거하면 로그인 기록이 해제됩니다.

사용자가 로그인 중에 수행하는 일반적인 실수는 로그인 프롬프트에 암호나 암호의 일부를 입력하는 것입니다. 이 실패한 로그인은 btmps 파일에 기록되어 암호나 암호의 일부를 노출합니다. 이런 이유로 btmps에 파일 보호를 설정하여 관리자만 읽을 수 있도록 해야 합니다.

```
# chmod 400 /var/adm/btmps
```

보안 정책상 사용자의 이전 세션을 다른 사용자가 볼 수 없도록 해야 하는 경우 /var/adm/wtmp 파일의 파일 보호도 변경해야 할 수 있습니다.

자세한 내용은 *last*(1), *utmp*(4) 및 *wtmp*(4)를 참조하십시오.

utmp 데이터베이스는 /var/adm/utmp에 따라 utmpd 명령에서 관리하고 동기화하는 사용자 계정 데이터베이스입니다. 응용 프로그램은 utmps 데이터베이스에 액세스할 수 있습니다. *utmpd*(1M) 및 *utmps*(4)를 참조하십시오.

2.2.2.1 last 명령 예제

이 절에서는 last 명령을 사용하는 예제를 보여 줍니다. 다음 명령은 콘솔 터미널의 모든 세션과 모든 root 세션을 표시합니다.

```
# last root console | more
root pts/1 Mon Mar 12 16:22 - 18:04 (01:41)
abcdeux console Mon Mar 12 10:13 - 10:19 (00:06)
root pts/2 Fri Mar 9 13:51 - 15:12 (01:21)
abcdeux console Thu Mar 8 12:21 - 12:22 (00:00)
root pts/ta Wed Mar 7 15:38 - 18:13 (02:34)
```

다음 명령은 다시 부팅된 시간을 표시합니다.

```
# last reboot
reboot system boot Sun Mar 28 18:06 still logged in
reboot system boot Sun Mar 28 17:48 - 18:06 (00:17)
reboot system boot Sun Mar 28 17:40 - 17:48 (00:08)
reboot system boot Thu Feb 19 18:25 - 17:40 (37+23:15)
reboot system boot Mon Feb 16 13:56 - 18:25 (3+04:28)
```

2.2.3 로그인한 사용자 확인

who 명령은 /etc/utmp 파일을 검사하여 현재 사용자 로그인 정보를 얻습니다. 또한 who 명령은 로그인, 로그오프, 다시 부팅, 시스템 시계 변경 사항 및 init 프로세스에 의해 시작된 프로세스를 표시할 수 있습니다.

who -u 명령을 사용하여 현재 로그인한 사용자를 모니터링합니다. 예를 들면 다음과 같습니다.

```
# who -u
aperson console Aug 5 11:28 old 5796 system.home.company.com
aperson pts/0 Aug 17 18:11 0:03 24944 system
aperson pts/1 Aug 5 11:28 1:14 5840 system
```

자세한 내용은 *who*(1)를 참조하십시오.

2.3 PAM을 사용하여 사용자 인증

PAM(Pluggable Authentication Modules)은 인증, 계정 관리, 세션 관리 및 암호 서비스를 제공하는 산업 표준 프레임워크입니다. 이 절에서는 PAM의 개요를 제공하고 PAM 구성 파일인 /etc/pam.conf와 /etc/pam_user.conf에 대해 설명합니다.

자세한 내용은 *pam*(3), *pam_**(5), *pam.conf*(4), *pam_user.conf*(4) 및 *security*(4)를 참조하십시오.

2.3.1 개요

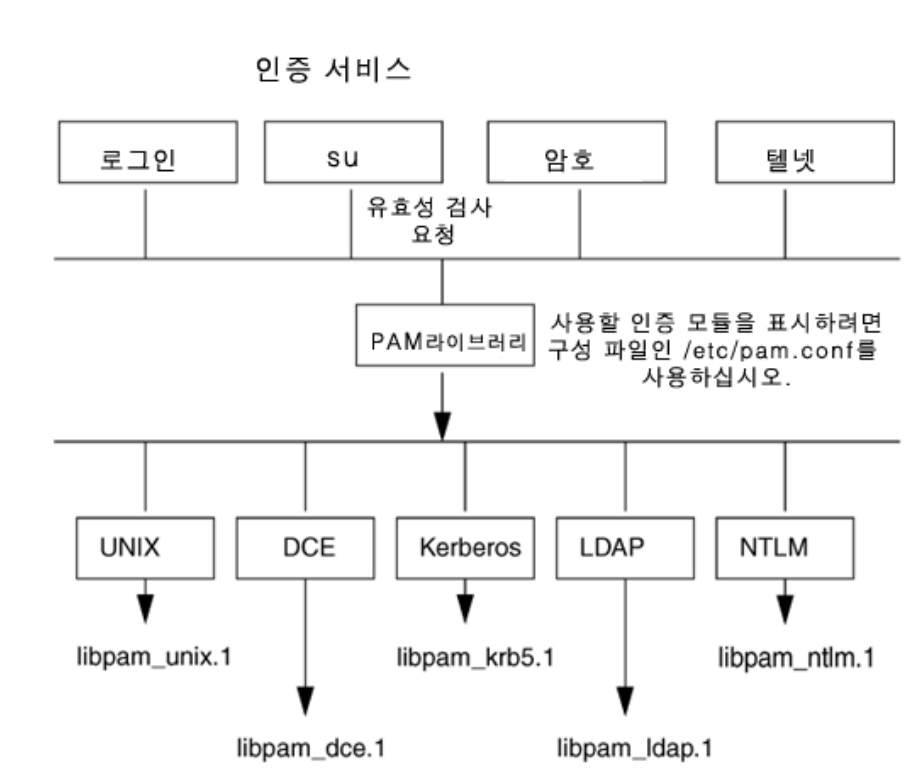
PAM을 사용하면 시스템에서 사용 가능한 인증 서비스를 융통성 있게 선택할 수 있습니다. PAM 프레임워크를 사용하여 응용 프로그램을 수정하지 않고도 새 인증 서비스 모듈을 연결하고 사용 가능하게 할 수도 있습니다.

사용자가 login 또는 rlogin을 사용하여 로컬 또는 원격으로 로그인할 때마다 사용자를 확인하거나 유효한 시스템 사용자로 인증해야 합니다. 시간이 지나면서 인증 방법이 향상되고 변경되면 로그인 서비스도 변경되어야 합니다. 단순히 인증 코드를 변경할 목적으로 로그인 서비스를 계속 변경하는 것을 방지하기 위해 로그인 코드를 수정하지 않고 다른 인증 방법을 사용할 수 있도록 PAM이 개발되었습니다.

따라서 로그인 인증, 계정 확인 및 암호 수정 시 PAM 인터페이스를 사용합니다.

사용자 인증이 필요한 프로그램은 PAM에 요청을 보냅니다. 그러면 올바른 검증 방법이 결정되고 적절한 응답이 반환됩니다. 프로그램에서는 사용되고 있는 인증 방법을 알 필요가 없습니다. 개요는 그림 2-1을 참조하십시오.

그림 2-1 PAM 아래의 HP-UX 인증 모듈



인증 방법은 다음 PAM 시스템 파일을 사용하여 시스템 범위 및 개별 사용자별로 지정됩니다.

`/etc/pam.conf` 시스템 범위 제어 파일입니다. 서비스와 쌍을 이루는 서비스 모듈을 정의합니다. 이러한 모듈은 시스템 기본값으로 간주됩니다.

`/etc/pam_user.conf` 개별 사용자 제어 파일입니다. 특정 사용자의 서비스 모듈에서 사용할 옵션을 정의합니다. 이 파일은 선택적 파일입니다.

자세한 내용은 *pam*(3), *pam.conf*(4), *pam_updbe*(5), *pam_user.conf*(4)를 참조하십시오.

2.3.2 PAM 라이브러리

PAM 서비스 모듈은 공유 라이브러리에 의해 구현됩니다. PAM을 사용하면 HP-UX에서 여러 인증 기술을 함께 사용할 수 있습니다. `/etc/pam.conf` 구성 파일은 사용할 인증 모듈을 결정합니다. PAM 라이브러리는 다음과 같습니다.

- `PAM_DCE`

  PAM_DCE 모듈을 사용하면 DCE를 시스템 항목 서비스(예: `login`, `telnet`, `rlogin`, `ftp`)로 통합할 수 있습니다. PAM_DCE 모듈은 인증, 계정 관리 및 암호 관리 모듈에 대한 기능을 제공합니다. 이러한 모듈은 PAM_DCE 라이브러리 `/usr/lib/security/pam_dce.sl`을 통해 지원됩니다. 자세한 내용은 *pam_dce*(5)를 참조하십시오.

- PAM_HPSEC

  PAM_HPSEC 모듈은 HP-UX와 관련해서 인증, 계정 관리, 암호 관리 및 세션 관리 확장을 관리합니다. login, dtlogin, ftp, su, remsh, rexec, ssh 등의 서비스에는 /usr/lib/security/$ISA/libpam_hpsec.so.1을 사용해야 합니다. 이러한 서비스는 하나 이상 비선택적 모듈 위의 스택 맨 위에 libpam_hpsec.so.1을 배치해야 합니다. 또한 pam_hpsec 모듈은 /etc/default/security에 정의된 여러 속성을 강제로 시행합니다. 자세한 내용은 *pam_hpsec*(5) 및 *security*(4)를 참조하십시오.

- PAM_KRB5

  Kerberos는 일반 텍스트로 암호를 전송하지 않고 네트워크에서 보안 통신을 활성화하는 네트워크 인증 프로토콜입니다. KDC(Key Distribution Center)는 암호를 인증한 다음 TGT(Ticket Granting Ticket)를 발급합니다. PAM Kerberos 공유 라이브러리는 /usr/lib/security/libpam_krb5.1입니다. 자세한 내용은 *pam_krb5*(5)를 참조하십시오.

- PAM_LDAP

  LDAP(Lightweight Directory Access Protocol)는 디렉토리 서비스를 통해 사용자, 그룹 및 네트워크 관리 정보를 중앙 집중화하는 표준입니다. 인증은 LDAP 디렉토리 서버에서 수행됩니다. 자세한 내용은 http://docs.hp.com/hpux/11iv2/index.html에 있는 LDAP-UX 설명서를 참조하십시오.

- PAM_NTLM

  PAM NT LAN Manager를 사용하면 시스템 로그인 도중 Windows 서버에 대해 HP-UX 사용자를 인증할 수 있습니다. PAM NTLM은 NT 서버를 사용하여 HP-UX 시스템에 로그인하는 사용자를 인증합니다. 자세한 내용은 http://docs.hp.com/hpux/11iv2/index.html에 있는 **HP CIFS Client 관리 설명서**를 참조하십시오.

- PAM_UNIX

  PAM_UNIX 모듈은 인증, 계정 관리, 세션 관리, 암호 관리 등 네 가지 PAM 모듈에 대한 기능을 모두 제공합니다. 모듈은 PAM UNIX 라이브러리 /usr/lib/security/libpam_unix.1을 통해 지원됩니다. 자세한 내용은 *pam_unix*(5)를 참조하십시오.

- PAM_UPDBE

  PAM에 대한 사용자 정책 정의 서비스 모듈 /usr/lib/security/libpam_updbe.1은 사용자 구성 파일 /etc/pam_user.conf에 정의된 옵션을 읽고 후속 서비스 모듈에서 사용할 수 있도록 이 정보를 PAM 핸들에 저장합니다. 자세한 내용은 *pam_updbe*(5)를 참조하십시오.

2.3.3 /etc/pam.conf를 사용한 시스템 범위 구성

PAM 구성 파일 /etc/pam.conf는 사용자를 인증하는 데 사용되는 보안 메커니즘을 정의합니다. 기본값을 사용하면 표준 HP-UX 및 트러스트된 시스템 모두에서 일반적인 시스템 작업을 수행할 수 있습니다. 또한 개별 사용자 제어 및 DCE 통합 로그인 기능이 지원됩니다.

참고: DCE의 경우 auth.adm 유틸리티를 사용하여 원하는 구성 파일을 만듭니다. 이 파일은 기능상 이전 HP 통합 로그인 auth.conf 파일과 같습니다. 자세한 내용은 *auth.adm*(1m)을 참조하십시오.

사용자가 로그인하거나 암호를 변경할 수 있으려면 libpam 및 libpam_unix PAM 라이브러리와 /etc/pam.conf 구성 파일이 시스템에 있어야 합니다.

HP-UX 인증은 /etc/pam.conf 파일에 따라 달라집니다. 이 파일의 소유자는 root여야 하고 파일 사용 권한은 다음과 같아야 합니다.

```
-r--r--r-- 1 root sys  1050 Nov  8 10:16 /etc/pam.conf
```

이 파일이 손상되거나 시스템에서 이 파일을 찾을 수 없는 경우 root가 단일 사용자 모드로 콘솔에 로그인하여 문제를 해결할 수 있습니다.

보호되는 서비스 이름은 시스템 제어 파일 /etc/pam.conf에서 네 가지 테스트 범주(module-type)인 인증, 계정, 세션 및 암호 아래에 나열됩니다.

자세한 내용은 *pam*(3), *pam.conf*(4) 및 *pam_user.conf*(4)를 참조하십시오.

2.3.4 샘플 /etc/pam.conf 파일

다음은 샘플 /etc/pam.conf 파일의 일부 목록입니다. 파운드(#)로 시작하는 줄은 주석 줄입니다. /etc/pam.conf에는 인증 관리, 계정 관리, 세션 관리 및 암호 관리 섹션이 있습니다.

```
#
# PAM configuration
#
# Notes:
#
# If the path to a library is not absolute, it is assumed to be
# relative to the directory /usr/lib/security/$ISA/
#
# For PA applications, /usr/lib/security/$ISA/libpam_unix.so.1 is a
# symbolic link that points to the corresponding PA (32 or 64-bit) PAM
# backend library.
#
# The $ISA (i.e. Instruction Set Architecture) token will be replaced
# by the PAM engine with an appropriate directory string.
# See pam.conf(4).
#
# Also note that the use of pam_hpsec(5) is mandatory for some of
# the services. See pam_hpsec(5).
#
# Authentication management
#
login    auth required  libpam_hpsec.so.1
login    auth required  libpam_hpsec.so.1
su       auth required  libpam.hpsec.so.1  bypass_setaud
su       auth required  libpam_unix.so.1
dtlogin  auth required  libpam_hpsec.so.1
dtlogin  auth required  libpam_unix.so.1
dtaction auth required  libpam_hpsec.so.1
dtaction auth required  libpam_unix.so.1
ftp      auth required  libpam_hpsec.so.1
ftp      auth required  libpam_unix.so.1
rcomds   auth required  libpam_hpsec.so.1
rcomds   auth required  libpam_unix.so.1
sshd     auth required  libpam_hpsec.so.1
sshd     auth required  libpam_unix.so.1
OTHER    auth required  libpam_unix.so.1
#
# Account management
```

```
#
login    account required       libpam_hpsec.so.1
login    account required       libpam_unix.so.1
su       account required       libpam_hpsec.so.1
su       account required       libpam_unix.so.1
```

2.3.5 /etc/pam_user.conf 사용자 구성 파일

PAM 구성 파일 /etc/pam_user.conf는 사용자별로 PAM을 구성합니다. 이 파일은 선택 사항이며, PAM 응용 프로그램이 사용자에 따라 다르게 동작해야 하는 경우에만 필요합니다.

/etc/pam_user.conf에 나열하여 개별 사용자에게 다른 옵션을 할당합니다. 여기에 나열된 login-name에 대해 여기에 나열된 options는 /etc/pam.conf의 module-type 및 module-path에 대해 지정된 모든 options를 대체합니다.

/etc/pam_user.conf의 항목은 다음 구문을 사용합니다.

```
login-name module-type module-path options
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

login-name	사용자의 로그인 이름입니다.
module-type	/etc/pam.conf에 지정된 module-type입니다.
module-path	/etc/pam.conf의 module-type과 연관된 module-path입니다.
options	모듈에 의해 인식되는 0개 이상의 옵션입니다.

/etc/pam_user.conf의 기본 내용은 주석입니다.

```
#
# This file defines PAM configuration for a user. The configuration
# here overrides pam.conf.
#
# The format for each entry is:
# user_name  module_type  module_path options
#
# For example:
#
# user_a        auth      /usr/lib/security/libpam_unix.1     debug
# user_a        auth      /usr/lib/security/libpam_dce.1      try_first_pass
# user_a        password  /usr/lib/security/libpam_unix.1     debug
#
# user_b        auth      /usr/lib/security/libpam_unix.1     debug use_psd
# user_b        password  /usr/lib/security/libpam_unix.1     debug use_psd
#
# See the pam_user.conf(4) manual page for more information
#
```

2.3.6 예제: PAM이 로그인에 대해 작동하는 방식

다음 예제에서는 /etc/pam.conf 파일이 구성된 방식에 따라 login의 auth 프로세스에 대해 설명합니다.

- /etc/pam.conf에 다음과 같은 단일 표준 login auth가 포함되어 있으면 login이 정상적으로 진행됩니다.

  ```
  login     auth  required  /usr/lib/security/libpam_unix.1
  ```

- 다음과 같은 시스템 범위 login auth 항목이 두 개 이상 있으면 순서대로 수행됩니다.

  ```
  login     auth  required  /usr/lib/security/libpam_unix.1
  login     auth  required  /usr/lib/security/libpam_dce.1
  ```

  이 경우 표준 HP-UX login 프로세스가 실행된 다음 DCE 인증 프로세스가 발생합니다. 둘 다 제대로 실행되면 로그인이 성공적으로 완료됩니다. 둘 중 하나가 실패해도 두 프로세스가 모두 수행됩니다.

- 서로 다른 사용자에 대해 서로 다른 인증 방법이 필요한 경우에는 /etc/pam.conf의 인증 모듈 앞에 특수한 항목인 libpam_udpbe를 지정합니다(쉽게 참조할 수 있도록 줄에 번호가 매겨져 있음).

  ```
  #/etc/pam.conf
  #1
  login     auth  required  /usr/lib/security/libpam_udpbe.1
  #2
  login     auth  required  /usr/lib/security/libpam_unix.1
  #3
  login     auth  required  /usr/lib/security/libpam_dce.1
  ```

  그런 다음 /etc/pam_user.conf에 영향을 받는 각 사용자에 대한 항목을 지정합니다.

  ```
  #/etc/pam_user.conf
  #4
  allan  auth  /usr/lib/security/libpam_unix.1  debug
  #5
  allan  auth  /usr/lib/security/libpam_dce.1   try_first_pass
  #6
  isabel auth  /usr/lib/security/libpam_unix.1  debug  use_psd
  ```

  allan이 로그인할 때 /etc/pam.conf의 첫 번째 줄로 인해 PAM이 /etc/pam_user.conf를 읽게 됩니다. /etc/pam_user.conf의 네 번째 및 다섯 번째 줄에 있는 모듈 경로가 /etc/pam.conf의 두 번째 및 세 번째 줄에 있는 모듈 경로와 일치하므로 PAM은 임시로 /etc/pam.conf의 두 번째 및 세 번째 줄에 있는 null options 필드를 각각 debug 및 try_first_pass로 대체합니다. 그런 다음 두 번째 및 세 번째 줄에 지정된 모듈이 변경된 옵션으로 실행됩니다.

  isabel이 로그인할 때 /etc/pam.conf의 첫 번째 줄로 인해 PAM이 /etc/pam_user.conf를 읽게 되고 /etc/pam.conf의 두 번째 줄에 있는 options 필드를 임시로 debug use_psd로 대체하며 세 번째 줄은 변경되지 않습니다. 그런 다음 두 번째 및 세 번째 줄에 지정된 모듈이 변경된 옵션으로 실행됩니다.

  george가 로그인할 때 /etc/pam.conf의 첫 번째 줄로 인해 PAM이 /etc/pam_user.conf를 읽게 됩니다. george에 대한 항목이 없으므로 /etc/pam_user.conf의 두 번째 및 세 번째 줄은 변경되지 않습니다. 두 번째 및 세 번째 줄에 지정된 모듈이 변경 없이 실행됩니다.

2.4 암호 관리

암호는 가장 중요한 개별 사용자 식별 기호입니다. 시스템에서는 암호로 사용자를 인증하여 시스템에 액세스하도록 합니다. 암호는 사용하거나 저장하거나 알려질 때 보안상 취약해지므로 항상 비밀로 유지해야 합니다. 다음 절에서는 암호에 대해 자세히 설명합니다.

2.4.1 시스템 관리자의 책임

암호 보안에 대한 책임은 시스템 관리자 및 시스템의 사용자 모두에게 있습니다. 시스템 관리자가 수행하는 보안 작업은 다음과 같습니다.

- 모든 사용자가 암호를 갖도록 합니다.
- 표준 암호 및 그룹 파일인 `/etc/passwd` 및 `/etc/group`을 포함한 모든 시스템 파일에서 적절한 사용 권한을 유지 관리합니다.
- 시스템에 더 이상 액세스할 수 없는 사용자의 사용자 ID와 암호를 삭제하거나 무효화합니다.
- 응용 프로그램 암호가 모두 암호화되었는지 확인합니다.
- `/var/adm/btmp` 및 `/var/adm/wtmp`의 사용 권한이 제대로 설정되었는지 확인합니다.
- 단일 게스트 액세스를 위한 일회성 암호를 구현합니다.
- 사용자에게 암호 보안에 관한 책임을 알립니다.
- 암호 변경기간 설정 기능을 사용하여 사용자가 정기적으로 암호를 변경하도록 강제합니다.
- 최근 암호를 다시 사용할 수 없게 합니다.
- `/etc/default/security` 파일에서 시스템 범위 보안 속성을 구성합니다. 자세한 내용은 섹션 2.5 및 *security*(4)를 참조하십시오.
- 섀도 패스워드를 사용하도록 시스템을 변환합니다. 자세한 내용은 섹션 2.4.5, *shadow*(4) 및 *pwconv*(1M)를 참조하십시오.

2.4.2 사용자 책임

모든 사용자는 다음 규칙을 준수해야 합니다.

- 암호를 기억하고 항상 비밀을 유지합니다.
- 즉시 초기 암호를 변경하고 계속해서 변경합니다.
- 상태 변경 및 의심스러운 보안 위반을 모두 보고합니다.
- 암호를 입력할 때 아무도 보지 않도록 합니다.

2.4.3 좋은 암호의 기준

암호를 선택할 때는 다음 지침을 따르고 이 지침을 사용자에게 알립니다.

- 6자에서 80자 사이의 암호를 선택합니다. 특수 문자에는 별표와 슬래시 같은 제어 문자 및 기호가 포함될 수 있습니다. 표준 모드에서는 처음 8자만 사용됩니다.
- 모든 언어에서 사용되는 단어는 선택하지 마십시오. 철자를 거꾸로 써도 안 됩니다. 이를 찾아서 일치시킬 수 있는 소프트웨어 프로그램이 있습니다.
- 가족이나 애완 동물 이름 또는 취미와 같이 사용자와 쉽게 연관되는 암호는 선택하지 마십시오.
- `asdfghjkl`과 같이 키보드 상에서 나란히 있는 문자 또는 로그인 이름을 반복하여 사용하지 마십시오(예를 들어, 로그인이 `ann`인 경우 `annann`과 같은 암호는 좋지 않음).

- 맞춤법이 틀린 단어나 관련 없는 두 단어의 결합을 사용하여 적절한 암호를 만듭니다. 또는 좋아하는 제목이나 문구의 첫 번째 문자를 연결하여 암호로 사용하는 것도 좋습니다.
- 음절을 결합하여 발음할 수 있는 어려운 단어를 만드는 암호 생성기를 사용하는 것도 좋습니다.
- 다른 사용자와 암호를 공유하지 마십시오. 암호를 공유하지 않도록 관리해야 합니다.
- 항상 암호를 설정합니다. /etc/passwd 파일의 암호 필드를 비워 두지 마십시오.

2.4.4 /etc/passwd 암호 파일 변경

표준 시스템에서는 하나의 암호 파일 /etc/passwd를 유지 관리합니다.

모든 암호는 입력한 즉시 암호화되어 암호 파일 /etc/passwd에 저장됩니다. 암호화된 암호만 비교하는 데 사용됩니다.

암호 파일을 변경해야 하는 경우 다음 지침을 따릅니다.

- 빈 암호 필드나 null 암호 필드를 허용하지 마십시오. 이는 보안 문제입니다. 암호 필드가 비어 있으면 모든 사용자가 해당 계정의 암호를 설정할 수 있습니다.
- 암호 파일을 직접 편집하지 마십시오. HP SMH 또는 useradd, userdel 또는 usermod 명령을 사용하여 암호 파일 항목을 수정합니다. 직접 암호 파일을 편집해야 하는 경우 vipw 명령을 사용하거나 pwck 명령으로 파일을 확인합니다. 자세한 내용은 *vipw*(1M) 및 *pwck*(1M)를 참조하십시오.

2.4.4.1 passwd 명령 예제

다음은 몇 가지 유용한 passwd 명령 예제입니다.

- 사용자 암호 다시 설정:

  # **passwd user1**

- 다음에 로그인할 때 암호 변경 강제:

  # **passwd -f user1**

- 계정 잠금 또는 비활성화:

  # **passwd -l user2**

- 암호 변경기간 설정 기능 활성화:

  # **passwd -n 7 -x 28 user1**

- 특정 사용자의 암호 변경기간 설정 상태 보기:

  # **passwd -s user**

- 모든 사용자의 암호 변경기간 설정 상태 보기:

  # **passwd -sa**

2.4.4.2 /etc/passwd 파일 형식

/etc/passwd 파일은 로그인할 때 사용자를 인증하는 데 사용됩니다. 파일에는 HP-UX 시스템의 모든 계정에 대한 항목이 들어 있습니다. 각 항목은 콜론으로 구분된 일곱 개의 필드로 이루어져 있습니다. 일반적인 /etc/passwd 항목은 다음과 같습니다.

```
robin:Z.yxGaSvxAXGg:102:99:Robin Hood,Rm 3,x9876,408-555-1234:/home/robin:/usr/bin/sh
```

필드에는 다음 정보(순서대로 나열)가 들어 있으며 콜론으로 구분되어 있습니다.

1. robin - 최대 8자로 구성된 사용자(로그인) 이름
2. Z.yxGaSvxAXGg - 암호화된 암호 필드
3. 102 - 사용자 ID. 0에서 MAXINT-1(2,147,483,646 또는 2^{31} -2와 같음) 사이의 정수
4. 99 - /etc/group의 그룹 ID. 0에서 MAXINT-1 사이의 정수
5. Robin Hood,Rm 3,x9876,408-555-1234 - 사용자의 전체 이름, 위치, 전화 번호 등의 정보를 식별하는 데 사용되는 주석 필드. 역사적인 이유로 gecos 필드라고도 합니다
6. /home/robin - 홈 디렉토리. 사용자의 초기 로그인 디렉토리
7. /usr/bin/sh - 사용자가 로그인할 때 실행되는 로그인 셸 경로 이름

사용자는 passwd를 호출하여 암호를 변경할 수 있고 chfn으로 주석 필드(다섯 번째 필드)를 변경할 수 있으며 chsh로 로그인 프로그램 경로 이름(일곱 번째 필드)을 변경할 수 있습니다. 나머지 필드는 시스템 관리자가 설정합니다. 사용자 ID는 고유해야 합니다. 자세한 내용은 *chfn*(1), *chsh*(1), *passwd*(1) 및 *passwd*(4)를 참조하십시오.

2.4.5 /etc/shadow 섀도 패스워드 파일

악의적인 암호 해독자가 점점 더 강력한 컴퓨팅 능력을 갖게 되면서 /etc/passwd 파일에 있는 숨겨지지 않은 암호가 공격 받기 쉬워지고 있습니다.

섀도 패스워드는 섀도 패스워드 파일에 있는 암호화된 패스워드를 숨겨 시스템 보안을 향상시킵니다. 공개적으로 읽을 수 있는 /etc/passwd 파일에 저장되어 있는 암호화된 암호를 해당 권한이 있는 사용자만 액세스할 수 있는 /etc/shadow 파일로 옮길 수 있습니다.

다음 명령을 사용하여 섀도 패스워드를 활성화, 확인 및 비활성화합니다.

- pwconv 명령은 섀도 패스워드 파일을 만들고 /etc/passwd 파일에 있는 암호화된 패스워드를 /etc/shadow 파일에 복사합니다.
- pwck 명령은 /etc/passwd 및 /etc/shadow 파일을 검사하여 불일치를 확인합니다.
- pwunconv 명령은 /etc/shadow 파일에 있는 암호화된 암호와 변경기간 설정 정보를 /etc/passwd 파일에 복사하고 /etc/shadow 파일을 삭제합니다.

자세한 내용은 *pwconv*(1M), *pwck*(1M), *pwunconv*(1M) 및 *shadow*(4)를 참조하십시오.

섀도 패스워드 기능과 관련해서 다음 사항에 주의합니다.

- 섀도 패스워드 기능이 활성화되면 응용 프로그램이 암호와 변경기간 설정 정보를 얻기 위해 직접 /etc/passwd 파일의 패스워드 필드에 액세스할 경우 응용 프로그램이 영향을 받을 수 있습니다. 이 필드에는 이제 x가 포함되어 있으며, 이것은 해당 정보가 /etc/shadow에 있다는 것을 나타냅니다.

  PAM 인터페이스를 사용하여 인증하는 응용 프로그램은 영향을 받지 않습니다.

프로그램 방식으로 /etc/shadow 파일에 액세스하려면 **getspent()** 호출을 사용합니다. 이러한 호출은 /etc/passwd에 대한 **getpwent()** 호출과 비슷합니다. 자세한 내용은 *getspent*(3C) 및 *getpwent*(3C)를 참조하십시오.

- /etc/nsswitch.conf 파일의 섀도 패스워드는 files, NIS 및 LDAP 이름 서비스에서 지원되지만 다른 이름 서버 전환 백엔드에서는 지원되지 않을 수 있습니다. files, NIS 및/또는 LDAP만 사용하도록 시스템을 구성하려면 /etc/nsswitch.conf의 passwd 줄에 files, NIS 및/또는 LDAP만 포함되게 합니다. /etc/nsswitch.conf가 없거나 passwd 줄이 없는 경우 기본값은 files만 사용합니다. 자세한 내용은 *nsswitch.conf*(4)를 참조하십시오.
- 섀도 패스워드는 다른 UNIX 시스템에서 제공되는 사실상의 표준을 기반으로 합니다.

/etc/default/security에 정의된 다음 속성은 섀도 패스워드에 적용됩니다. 자세한 내용은 섹션 2.5 및 *security*(4) 맨페이지를 참조하십시오.

- INACTIVITY_MAXDAYS - 사용되지 않아 계정을 만료하기까지의 기간(일)
- PASSWORD_MINDAYS - 암호를 변경할 수 있기까지의 최소 기간(일)
- PASSWORD_MAXDAYS - 암호가 유효한 최대 기간(일)
- PASSWORD_WARNDAYS - 사용자에게 암호 만료를 경고하기까지의 기간(일)

섀도 패스워드는 다음 제품에서 지원됩니다.

- LDAP(Lightweight Directory Access Protocol)
- LDAP(Ignite-UX 디렉토리 액세스 제어)
- Serviceguard

패스워드가 /etc/passwd에 있다고 가정하는 응용 프로그램에서는 섀도 패스워드가 지원되지 않습니다.

자세한 내용은 다음 맨페이지를 참조하십시오.

passwd(1), *pwck*(1M), *pwconv*(1M), *pwunconv*(1M), *getspent*(3C), *putspent*(3C), *nsswitch.conf*(4), *passwd*(4), *security*(4), *shadow*(4)

2.4.6 /etc/passwd에서 의사 계정 제거 및 주요 하위 시스템 보호

일반적으로 /etc/passwd 파일에는 개별 사용자와 연관된 항목이 아니며 실제 대화형 로그인 셸이 없는 많은 "의사 계정"이 들어 있습니다.

date, who, sync, tty 등 몇 가지 항목은 사용자 편의를 위해 개선되어 로그인하지 않고 실행할 수 있는 명령을 제공합니다. 보안을 강화하기 위해 배포된 /etc/passwd에서는 이러한 항목이 제거되어, 로그인한 사용자만 이러한 프로그램을 실행할 수 있습니다.

다른 항목은 파일의 소유자이므로 /etc/passwd에 남아 있습니다. adm, bin, daemon, hpdb, lp 및 uucp와 같은 소유자가 있는 프로그램은 전체 하위 시스템을 포함하며 특별한 경우를 나타냅니다. 이러한 프로그램은 보호하거나 사용하는 파일에 대한 액세스 권한을 부여하므로 /etc/passwd에 나열된 항목을 사용하여 의사 계정으로 작동할 수 있어야 합니다. 일반적인 의사 계정 및 특별 계정은 보기 2-1에 표시되어 있습니다.

보기 2-1 의사 및 특별 시스템 계정

```
root::0:3::/:/sbin/sh
daemon:*:1:5::/:/sbin/sh
bin:*:2:2::/usr/bin:/sbin/sh
sys:*:3:3::/:
adm:*:4:4::/var/adm:/sbin/sh
uucp:*:5:3::/var/spool/uucppublic:/usr/lbin/uucp/uucico
lp:*:9:7::/var/spool/lp:/sbin/sh
nuucp:*:11:11::/var/spool/uucppublic:/usr/lbin/uucp/uucico
hpdb:*:27:1:ALLBASE:/:/sbin/sh
nobody:*:-2:-2::/:
```

이러한 하위 시스템은 root 액세스 권한(`uid 0`)을 부여하지 않고 제어 하에 있는 프로그램에 대한 액세스 권한을 부여할 수 있습니다. 대신 실행 파일에 대한 `setuid` 비트가 설정되며 프로세스의 유효 사용자가 실행 파일의 소유자에 해당합니다. 예를 들어, `cancel` 명령은 `lp` 하위 시스템의 일부이며 유효 사용자 `lp`로 실행합니다.

`setuid`를 설정하면 하위 시스템의 보안 조정 기능은 전체 시스템이 아닌 하위 시스템에 포함된 모든 프로그램의 보안을 강제로 시행합니다. 따라서 보안 문제에 대한 하위 시스템의 취약성도 해당 하위 시스템 파일에만 제한됩니다. 다른 하위 시스템의 프로그램에는 문제의 영향이 미치지 않습니다. 예를 들어, `lp`의 프로그램은 `daemon`의 프로그램에 영향을 미치지 않습니다.

2.4.7 HP-UX Secure Shell을 사용한 로그인 보안 유지

HP-UX Secure Shell은 안전한 원격 로그인, 파일 전송 및 원격 명령 실행 기능을 제공합니다. 모든 클라이언트-서버 통신이 암호화됩니다. 네트워크를 통과하는 암호는 일반 텍스트로 전송되지 않습니다. 자세한 내용은 *ssh*(1) 및 섹션 5.6을 참조하십시오.

2.4.8 NIS에 저장된 암호 보안 유지

NIS(Network Information Service)는 NFS(Network File System)의 일부입니다. NIS를 사용하면 중앙의 한 곳, 즉 마스터 서버에서 여러 호스트의 구성을 관리할 수 있습니다. 각 호스트에 개별적으로 호스트 구성을 저장하는 대신 이 정보는 중앙의 한 곳에 통합됩니다. `/etc/password` 파일은 NIS 서버에 저장된 여러 개의 구성 파일 중 하나입니다.

`/etc/shadow` 섀도 패스워드 파일은 NIS에서 지원되지 않습니다.

NIS에 대한 자세한 내용은 **NFS Services Administrator's Guide**를 참조하십시오.

2.4.9 LDAP 디렉토리 서버에 저장된 암호 보안 유지

LDAP-UX Client Services는 PAM과 상호 운용되어 LDAP 디렉토리 서버에 저장된 암호를 인증합니다. `PAM_LDAP` 라이브러리에서 인증 서비스를 제공합니다.

2.5 시스템 보안 속성 정의

보안 속성은 시스템 구성의 제어를 강화하여 암호, 로그인 및 감사에 향상된 보안 기능을 추가합니다.

20개 이상의 속성이 있으며, 이러한 속성에 대해서는 *security*(4)에서 설명합니다. 속성 범주는 다음과 같이 요약됩니다.

로그인 속성	이 속성은 로그인 시간, 허용되는 로그인 횟수 및 계정을 잠그기까지 허용되는 로그인 실패 횟수 등의 로그인 작업을 제어합니다.
암호 속성	이 속성은 암호 길이, 문자 수 및 문자 유형, 내역 깊이, 암호를 변경할 기간(일), 암호 만료 등의 암호 작업을 제어합니다.
부팅 속성	이 속성은 부팅 인증을 정의하여 시스템을 단일 사용자 모드로 부팅할 권한이 있는 사용자를 정의합니다. 1장의 부팅 인증 정보를 참조하십시오.
사용자 전환(su) 속성	이 속성은 PATH 환경 값, su 명령에 대한 root 그룹 이름 및 su에서 특정 환경 변수를 전파할지 여부를 정의합니다. 자세한 내용은 *su*(1)를 참조하십시오.
감사 속성	이 속성은 사용자를 감사할지 여부를 제어합니다. 감사 속성은 login 프로세스 중에 확인됩니다. HP-UX 감사에 대한 자세한 내용은 *audit*(5)를 참조하십시오.
umask 속성	이 속성은 pam_unix 또는 pam_hpsec에 의해 시작된 모든 세션의 umask()를 제어합니다. 자세한 내용은 *pam_unix*(5) 및 *pam_hpsec*(5)를 참조하십시오. umask 속성은 login 프로세스 중에 확인됩니다.

시스템은 다음 파일을 사용하여 속성을 처리합니다.

- /etc/default/security
- /var/adm/userdb
- /etc/security.dsc
- /etc/passwd
- /etc/shadow

각 속성의 사용자별 값은 /etc/password, /etc/shadow 또는 /var/adm/userdb의 사용자 데이터베이스 중 한 위치에만 있습니다. 각 속성과 해당 사용자별 위치에 대해서는 *security*(4) 맨페이지에서 설명합니다.

시스템은 다음과 같은 방식으로 적용할 속성을 확인합니다.

- 시스템은 /var/adm/userdb 사용자 데이터베이스, /etc/passwd 파일 또는 /etc/shadow 파일의 사용자별 속성 값을 검사합니다.
- 사용자별 값이 없으면 시스템은 /etc/default/security에 있는 구성 가능한 시스템 범위 기본 속성을 검사합니다.
- 구성 가능한 시스템 범위 기본 속성이 없으면 시스템은 /etc/security.dsc의 기본 속성을 사용합니다.

보안 속성 설명 파일 /etc/security.dsc는 /etc/default/security 및 /var/adm/userdb의 사용자 데이터베이스에서 정의할 수 있는 속성을 표시합니다. 일부 속성은 구성 가능하고 일부는 내부 속성입니다. 어떤 방식으로든 /etc/security.dsc 파일을 수정하지 마십시오.

2.5.1 시스템 범위 속성 구성

다음 단계에서는 시스템 범위 보안 속성을 정의하는 방법에 대해 설명합니다.

1. 구성 가능한 시스템 범위 속성 기본값을 설명하는 *security*(4) 맨페이지를 검토합니다. 이러한 속성은 `/etc/default/security` 파일에서 구성되며, 이 파일에 대해서는 *security*(4) 맨페이지에도 설명되어 있습니다.

   `/etc/default/security` 파일에 속성이 정의되어 있지 않으면 시스템은 `/etc/security.dsc` 파일에 정의된 기본값을 사용합니다. `/etc/security.dsc` 파일에 대한 설명은 *userdb*(4) 맨페이지를 참조하십시오.

2. 구성 가능한 시스템 범위 기본값을 변경하려면 `vi`와 같은 텍스트 편집기를 사용하여 보안 구성 파일 `/etc/default/security`를 편집합니다. 이 파일은 외부 사용자가 읽을 수 있고 root만 쓸 수 있습니다.

   `/etc/default/security` 파일의 각 줄은 주석이나 속성 구성 정보입니다. 주석 줄은 파운드(#) 기호로 시작합니다. 주석이 아닌 줄은 `attribute=value` 쌍의 형식을 사용합니다(예: `PASSWORD_MAXDAYS=30`).

2.5.2 사용자별 속성 구성

다음 명령을 사용하여 개별 사용자의 특정 속성을 구성할 수 있습니다. 사용자별 속성을 구성하면 시스템 범위 기본값을 재정의합니다.

`userdbset`	`/etc/default/security` 파일에 정의된 시스템 범위 기본값을 재정의하기 위해 지정된 사용자의 속성을 변경합니다. 예제는 섹션 2.5.2.1을 참조하고, 자세한 내용은 *userdbset*(1M)를 참조하십시오.
`userdbget`	특정 사용자나 모든 사용자의 사용자 정의 값을 표시합니다. 자세한 내용은 *userdbget*(1M)를 참조하십시오.
`userdbck`	사용자 정의 값을 확인하거나 수정합니다. 자세한 내용은 *userdbck*(1M)를 참조하십시오.

예를 들어, `amy` 사용자에 대해서만 `PASSWORD_MAXDAYS`를 60일에서 30일로 변경할 수 있습니다. `amy`의 암호는 60일 동안이 아니라 30일 동안 유효합니다. 다른 모든 사용자에 대해서는 시스템 범위 값 60일이 적용됩니다.

사용자의 속성 값을 변경하려면 다음 절차를 사용합니다.

1. 사용자별 값을 설정하는 방법과 시스템 범위 속성 및 값에 대해 설명하는 *security*(4) 맨페이지를 검토합니다. 모든 속성에 사용자별 값이 있는 것은 아닙니다.
2. `userdbset`, `userdbget` 및 `userdbck` 명령에 대한 맨페이지를 검토합니다.
3. 수정할 사용자와 해당 사용자에 적용할 속성을 결정합니다. 예를 들어, 회계 부서의 사용자는 30일마다 암호를 변경하고 학생은 매 분기마다 암호를 변경하도록 설정할 수 있습니다.
4. `userdbset` 명령을 사용하여 사용자의 속성을 변경합니다.

   사용자별 정보는 `/var/adm/userdb` 디렉토리의 사용자 데이터베이스에 저장됩니다. 사용자 데이터베이스는 *userdb*(4) 맨페이지에서 설명합니다.

   `userdbset` 명령을 사용하여 모든 속성을 구성할 수는 없습니다. 일부 사용자별 값은 `/etc/passwd` 및 `/etc/shadow` 파일에서 정의됩니다. 자세한 내용은 *security*(4)를 참조하십시오.

5. `userdbget` 명령을 사용하여 사용자 정보를 가져옵니다.

2.5.2.1 userdbset를 사용한 사용자별 속성 정의 예제

다음 예제에서 userdbset 명령은 사용자 joe의 사용자 정의 속성을 모두 삭제합니다. joe가 로그인하면 /etc/default/security의 시스템 범위 기본값이 joe에 적용됩니다.

```
# /usr/sbin/userdbset -d -u joe
```

다음으로 userdbset는 최소 암호 길이를 7로 설정하고 UMASK를 0022(8진수 022)로 설정합니다. 이러한 변경 사항은 joe에 대해서만 적용됩니다.

```
# /usr/sbin/userdbset -u joe MIN_PASSWORD_LENGTH=7 UMASK=0022
```

다음 예제에서 userdbset는 사용자 amy의 모든 속성을 표시합니다.

```
# /usr/sbin/userdbget -u amy
amy AUDIT_FLAG=1
amy DISPLAY_LAST_LOGIN=0
```

표시에서 amy에 대한 감사 플래그는 활성화되고 마지막 로그인 기능은 비활성화됩니다.

2.5.2.2 INACTIVITY_MAXDAYS 및 섀도 패스워드 파일

/etc/default/security 파일에 정의된 INACTIVITY_MAXDAYS 속성은 시스템 범위로 비활성 계정을 만료할지 여부를 제어합니다. 시스템 범위 기본값을 재정의하고 사용자별로 INACTIVITY_MAXDAYS를 구성하려면 useradd -f 명령이나 usermod -f 명령을 사용합니다. 사용자별 구성을 삭제하려면 userdel 명령을 사용합니다. 자세한 내용은 *useradd*(1M), *usermod*(1M) 및 *userdel*(1M) 맨페이지를 참조하십시오.

userdbset 명령을 사용하여 사용자별로 INACTIVITY_MAXDAYS를 구성할 수는 없습니다. INACTIVITY_MAXDAYS 속성은 섀도 패스워드 파일의 비활성 필드와 관련이 있습니다. useradd 및 usermod 명령은 지정된 사용자에 대해 섀도 패스워드 파일의 비활성 필드를 수정합니다. 자세한 내용은 *security*(4) 맨페이지에 있는 INACTIVITY_MAXDAYS 설명을 참조하십시오.

2.5.3 사용자 데이터베이스 문제 해결

사용자 데이터베이스 문제를 해결하려면 다음 절차를 사용합니다.

문제 1: 사용자 보안 속성이 잘못 구성된 것 같습니다. 사용자 데이터베이스에 있는 사용자 정보가 잘못 구성된 것 같으면 다음 명령을 실행합니다.

```
# userdbget -u username
```

사용자 username에 대해 구성된 속성이 표시됩니다. 속성이 잘못 구성된 경우 해당 속성을 다시 구성합니다.

문제 2: 사용자 데이터베이스가 제대로 작동하지 않습니다. 사용자 데이터베이스를 확인해야 하는 경우 다음 명령을 입력합니다.

```
# userdbck
```

userdbck 명령은 사용자 데이터베이스의 문제를 식별하고 복구합니다.

2.6 setuid 및 setgid 프로그램 처리

보안상 시스템에 위험할 수 있으므로 setuid(사용자 ID 설정) 프로그램과 setgid(그룹 ID 설정) 프로그램을 확인합니다. 시스템 공격자는 주로 다음 두 가지 방법 중 하나로 setuid 및 setgid 프로그램을 이용할 수 있습니다.

- 대화형으로 또는 스크립트를 통해 setuid 또는 setgid 프로그램에서 공격자가 정의한 명령을 실행하도록 합니다.
- 프로그램에서 만든 데이터를 의미 없는 데이터로 대체합니다.

setuid 및 setgid 프로그램의 보안을 유지하려면 다음 지침을 따릅니다.

- setuid 및 setgid 프로그램에 대한 변경 사항을 확인합니다.
- 불필요하게 setuid 프로그램으로 표시되는 모든 프로그램에 대해 자세히 조사합니다.
- 불필요하게 setuid 프로그램으로 표시되는 프로그램의 사용 권한을 setgid 프로그램으로 변경합니다. 자세한 내용은 *chmod*(1) 및 *chmod*(2)를 참조하십시오.

  ls 명령의 긴 형식(ll 또는 ls -l)은 소유자 실행 권한에 대해 - 또는 x 대신 S 또는 s를 표시하여 setuid 프로그램을 보여 줍니다. 또한 그룹 실행 권한에 대해 - 또는 x 대신 S 또는 s를 표시하여 setgid 프로그램을 보여 줍니다.

  setuid 및 setgid 시스템 파일을 찾을 수 있지만 사용자 정의하지 않은 경우 이러한 파일은 출시될 때와 동일한 사용 권한을 가집니다.

- 사용자가 특히 자신보다 다른 사용자에게 setuid를 사용하는 경우 setuid 프로그램을 사용할 수 없게 합니다.
- 외부 소스에서 가져온 모든 프로그램의 코드를 트로이 목마로 알려진 유해 프로그램에 대해 검사합니다. 검사할 소스가 없는 setuid 프로그램은 복원하거나 설치하지 마십시오.
- 특정 수퍼유저 프로그램에 사용자가 액세스할 수 있도록 허용하려면 Restricted SMH를 사용하는 것이 좋습니다. Restricted SMH을 사용하면 수퍼유저가 아닌 사용자가 SMH의 특정 영역에 액세스할 수 있습니다. 자세한 내용은 *smh*(1M)를 참조하십시오.

2.6.1 setuid 및 setgid 프로그램이 위험할 수 있는 이유

모든 프로그램은 실행될 때마다 네 개의 ID 번호(실제 및 유효 사용자 ID(ruid 및 euid)와 실제 및 유효 그룹 ID(rgid 및 egid))를 가진 프로세스를 만듭니다. 일반적으로 이러한 ID 쌍은 동일합니다.

그러나 setuid 또는 setgid 프로그램을 실행하면 소유자와 연관된 것에서 객체에 대한 것으로 프로세스의 euid 또는 egid가 변경됩니다. 시작된 프로세스는 객체로부터 속성을 받고 사용자에게 프로그램의 소유자 및 그룹과 같은 액세스 권한을 부여합니다.

- setuid 비트가 설정된 경우 프로세스의 권한은 파일 소유자의 권한으로 설정됩니다.
- setgid 비트가 설정된 경우 프로세스의 권한은 파일 그룹의 권한으로 설정됩니다.
- setuid 및 setgid 비트가 모두 설정되어 있지 않으면 프로세스의 권한은 변경되지 않습니다.
- 특히 위험한 경우로 프로그램이 root에 대한 setuid이면 사용자는 root에서 사용할 수 있는 모든 권한을 얻습니다. 이러한 경우는 프로그램이 시스템 보안을 위반하는 방식으로 사용될 수 있으므로 위험합니다. 위험성은 조금 떨어지지만 다른 setuid 및 setgid의 경우에도 마찬가지로 이러한 문제가 발생할 수 있습니다.

보안상, 스크립트의 setuid 및 setgid 비트는 일반적으로 HP-UX 커널에 의해 무시됩니다. 이 규칙은 secure_sid_scripts 튜너블을 변경하여 완화시킬 수 있지만 이 튜너블을 기본값에서 변경하지 않는 것이 좋습니다. 이 튜너블에 대한 자세한 내용은 *secure_sid_scripts*(5)를 참조하십시오.

2.6.2 ID가 설정되는 방법

ID는 다음과 같은 방법으로 설정됩니다.

- `ruid` 및 `rgid`는 `ruid` 및 `rgid`를 설정하는 `login` 프로세스에서 상속됩니다. `uid` 및 `gid` 값은 `/etc/passwd`에서 지정됩니다.
- 또한 `login` 명령은 `ruid`, `euid`, `rgid` 및 `egid`를 변경합니다.
- `su` 명령은 `euid` 및 `ruid`를 변경합니다.
- `newgrp` 명령은 gid를 변경할 수 있습니다.
- **`chmod()`** 시스템 호출이나 `chmod` 명령을 사용하여 `setuid` 및 `setgid` 비트를 설정합니다. 자세한 내용은 *chmod*(1) 및 *chmod*(2)를 참조하십시오.

2.6.3 setuid 기능 제한 지침

기존 시스템에 `root`에 대한 `setuid` 프로그램을 추가하는 경우에는 주의해야 합니다. `root`에 대한 `setuid` 프로그램을 추가하면 시스템 구성이 변경되어 보안 문제가 발생할 수 있습니다.

관리 및 프로그래밍과 관련된 다음 권장 사항에 따라 권한이 부여된 프로그램의 사용을 제한하십시오.

- 반드시 필요한 경우에만 `setuid` 및 `setgid`를 사용합니다.
- 다른 사용자가 `setuid` 프로그램에 쓸 수 없도록 합니다.
- 가능한 한 `setuid` 대신 `setgid`를 사용하여 코딩 결함 또는 보안 문제로 인한 손상 범위를 줄입니다.
- 정기적으로 새로운 또는 수정된 `setuid` 및 `setgid` 프로그램이 있는지 파일 시스템을 검색합니다. `ncheck -s` 명령을 사용할 수 있습니다.
- `setuid` 및 `setgid` 프로그램이 수행하는 작업을 정확히 알고, 해당 프로그램이 의도한 작업만 수행하는지 확인합니다. 의도하지 않은 작업을 수행할 경우에는 프로그램 또는 해당 `setuid` 속성을 제거합니다.
- `setuid` 프로그램을 복사해야 하는 경우 대상 파일의 모드가 올바른지 확인합니다.
- 별로 중요하지 않은 데이터에 대해 테스트할 수 있도록 `setuid` 또는 `setgid` 속성 없이 `setuid` 프로그램을 작성합니다. 코드를 검토하고 영향을 받는 모든 부서가 새 프로그램이 보안을 유지한다는 사실에 만족한 후에만 이러한 속성을 적용합니다.
- `setuid` 프로그램에서 의도한 사용자 이외의 다른 사용자가 쓸 수 있는 파일을 만들지 않도록 합니다.
- **`exec*()`** 시스템 호출보다 먼저 `euid`를 다시 설정합니다. **`exec*()`**는 다른 라이브러리 루틴에서 호출될 수 있으며 셸을 분기(fork)하여 프로그램을 실행하는 루틴(**`popen()`**, **`system()`**, **`execlp()`** 및 **`execvp()`** 포함)을 사용하는 경우 주의해야 합니다. 자세한 내용은 *exec*(2), *popen*(3S) 및 *system*(3S)를 참조하십시오.
- `setuid` 프로그램을 작성할 때 권한이 필요한 코드 부분 주위에 **`setresuid()`**를 사용하여 취약성의 여지를 줄입니다. 자세한 내용은 *setresuid*(2)를 참조하십시오.
- **`exec*()`**를 호출하기 전에 불필요한 파일 설정자를 모두 닫습니다.
- 프로그램 환경에 있는 모든 변수(`PATH`, `IFS`) 및 `umask` 값이 충분히 제한적인지 확인합니다.

- **creat()** 시스템 호출을 사용하여 잠금 파일을 만들지 마십시오. 대신 **lockf()** 또는 **fcntl()**을 사용합니다. 자세한 내용은 *lockf*(2) 및 *fcntl*(2)을 참조하십시오.
- 적절한 매개 변수 길이 검증 없이 **sprintf()**, **strcpy()**, **strcat()** 등을 사용하여 버퍼 오버런을 방지하도록 특히 주의합니다. 자세한 내용은 *printf*(3S) 및 *string*(3C)을 참조하십시오.

2.7 스택 버퍼 오버플로 공격 방지

대량의 데이터를 프로그램으로 전달하는 것을 **스택 버퍼 오버플로 공격**이라고 합니다. 일반적으로 이 데이터에는 프로그램이 속아서 실행하는 명령이 들어 있습니다. 이러한 공격은 시스템에 권한 없이 액세스하거나 데이터를 삭제 또는 변경하거나 정당한 사용자에 대해 서비스 거부를 발생시키는 데 사용됩니다.

스택 버퍼 오버플로 공격을 모니터링하려면 다음 변경 사항을 확인합니다.

- 다른 프로그램을 실행하는 `setuid` 프로그램
- 예기치 않게 사용자 ID 0이 할당되는 프로그램. 사용자 ID 0은 수퍼유저나 root에만 사용됩니다.

스택 버퍼 오버플로 공격을 방지하려면 다음을 수행합니다.

- `executable_stack` 커널 튜너블 매개 변수를 활성화합니다.
- `chatr +es` 명령을 사용합니다.

`executable_stack` 커널 튜너블 매개 변수를 사용하여 프로그램이 해당 스택의 코드를 실행하지 않도록 할 수 있습니다. 이렇게 하면 침입자가 잘못된 데이터를 프로그램에 전달하여 프로그램이 해당 프로그램 스택에 있는 임의의 코드를 실행하게 하는 것을 방지할 수 있습니다.

`executable_stack` 커널 튜너블 매개 변수는 전체적으로 스택 버퍼 오버플로 방지 기능을 활성화하거나 비활성화합니다. `0`을 설정하면 스택을 실행할 수 없게 되며 보안상 더 안전합니다. 기본적으로 `executable_stack`은 이전 버전과의 호환성을 위해 `1`로 설정되어 있으며, 이 경우 스택 실행이 허용되므로 보호 기능이 없습니다. HP SMH 또는 `kmtune` 명령을 사용하여 `executable_stack`의 값을 변경합니다.

스택 버퍼 오버플로 방지 기능을 관리하는 다른 방법은 `chatr` 명령의 `+es` 옵션을 사용하는 것입니다. 예를 들어, `executable_stack`이 0으로 설정되어 있지만 프로그램이 해당 스택을 실행할 필요가 없는 경우 다음 `chatr` 명령을 사용하여 해당 프로그램의 스택 실행을 허용합니다.

chatr -es enable program

자세한 내용은 *chatr*(1), *kmtune*(1M) 및 *executable_stack*(5)을 참조하십시오.

2.8 무인 터미널 및 워크스테이션 보호

무인 워크스테이션과 터미널은 권한 없는 사용자의 공격에 특히 취약합니다. 현관문을 잠그지 않은 것처럼 누구에게나 열려 있습니다. 이 절에서는 다음과 같이 이 위험을 줄이는 방법에 대해 설명합니다.

- `/etc/inittab` 및 실행 수준을 사용하여 액세스를 제어합니다. `/etc/inittab`를 편집하여 다른 실행 수준에서 실행되어야 하는 장치를 식별합니다.
- 사용자 터미널 세션에 대한 외부 사용자의 액세스를 거부하여 터미널 장치를 보호합니다.

- 화면 잠금을 구성합니다.

2.8.1 /etc/inittab 및 실행 수준을 사용하여 액세스 제어

실행 수준은 특정 프로세스 집합이 실행되도록 허용된 시스템 상태입니다. 프로세스와 기본 실행 수준은 /etc/inittab에서 정의됩니다. 실행 수준은 0-6, s 또는 S입니다. 프로세스가 시스템과 동일한 실행 수준에 있지 않으면 해당 프로세스는 종료됩니다. 프로세스가 동일한 실행 수준에 있으면 시작되거나 계속 실행됩니다.

다음은 터미널과 모뎀이 선택한 실행 수준에서 실행될 수 있게 하는 예제입니다. ttp1과 ttp2는 모두 실행 수준 2와 3에 있습니다.

```
ttp1:23:respawn:/usr/sbin/getty -h tty0p1 9600
ttp2:23:respawn:/usr/sbin/uugetty -h ttypd0p2 9600
```

다음은 cron 작업을 사용하는 터미널과 모뎀을 비활성화하기 위해 정상 근무 시간 후에 실행 수준을 변경하는 예제입니다. 근무 시간 중에는 실행 수준이 3이고 ttp1 및 ttp2 터미널이 실행 수준 2와 3에 있기 때문에 두 터미널을 사용할 수 있습니다. 월-금, 오전 8:00시에 시스템 실행 수준이 3으로 설정됩니다.

```
# crontab -e
0 8 * * 1-5 /sbin/init 3
0 17 * * * /sbin/init 4
```

매일 오후 5:00시(앞의 예제에서 17은 17:00시 또는 오후 5:00시를 의미함)에 시스템 실행 수준이 4로 변경됩니다. ttp1 및 ttp2 터미널은 실행 수준 2와 3에 있으므로 오후 5:00시 이후에 작동할 수 없습니다.

2.8.2 터미널 장치 파일 보호

침입자는 열린 터미널에 액세스할 수 있는 경우 명령을 다른 터미널 창으로 리디렉션할 수 있습니다. 다음 예제에서 제거(rm) 명령은 /dev/tty0p0으로 리디렉션됩니다.

```
# echo "\r rm -r / \r\033d" > /dev/tty0p0
```

메시지가 터미널에 기록되지 않게 하려면 mesg -n(또는 mesg n) 명령을 사용할 수 있습니다. 이 명령은 해당 권한이 없는 사용자에 대해 쓰기 권한을 취소합니다. 자세한 내용은 *mesg*(1) 및 *write*(1)를 참조하십시오.

```
# vi ~/.shrc
mesg n
```

워크스테이션이나 터미널을 보호하는 다른 방법은 xhost 명령을 사용하는 것입니다. 자세한 내용은 *xhost*(1)를 참조하십시오. xhost 명령은 워크스테이션에 연결할 수 있는 호스트 및 사용자의 이름을 정의합니다.

```
# xhost +Another.system
```

모든 시스템과 사용자가 워크스테이션에 액세스할 수 있게 하여 액세스 제어를 해제하려면 다음 명령을 사용합니다.

```
# xhost +
```

2.8.3 화면 잠금 구성

이 절에서는 TMOUT 변수와 CDE 잠금 관리자를 사용하여 화면 잠금을 구성하는 방법에 대해 설명합니다.

2.8.3.1 TMOUT 변수 구성

비활성 터미널을 자동으로 잠그도록 TMOUT 변수를 구성할 수 있습니다.

다른 시스템을 자주 사용하며 .profile 파일을 다른 시스템으로 복사하는 경우 TMOUT 변수를 .profile에 추가하는 것이 더 편리합니다. 일반적으로 한 시스템만 사용하는 경우에는 터미널 잠금 방법 중 하나를 사용할 수 있습니다.

TMOUT 변수를 구성하려면 다음과 같이 .profile 파일을 편집합니다.

```
# vi ~/.profile
export TMOUT=600 #(600초 동안 사용하지 않으면 잠금)
```

600을 다른 원하는 값으로 변경할 수 있습니다.

2.8.3.2 CDE 잠금 관리자 구성

특정 비활성 시간 후에 화면을 잠그도록 CDE 잠금 관리자를 구성할 수 있습니다. 10분 동안 사용하지 않을 경우 화면을 잠그도록 CDE 잠금 관리자를 구성하려면 다음 명령을 입력합니다.

```
# cp /usr/dt/config/C/sys.resources /etc/dt/config/C/sys.resources
# vi /etc/dt/config/C/sys.resources
dtsession*lockTimeout: 10
```

스타일 관리자 작업 패널을 사용하여 CDE 잠금 관리자를 조정할 수도 있습니다. 이렇게 하려면 screen 아이콘을 클릭합니다.

2.9 원격 장치의 시스템 액세스 방지

원격 액세스에 의한 시스템 침투를 방지하려면 다음 주의 사항을 준수합니다.

- 모든 대화형 모뎀에 대해 하드웨어 다이얼 백 시스템의 사용을 요청합니다.
- /etc/dialups 및 선택적으로 /etc/d_passwd에 모뎀 장치에 대한 항목을 추가하여 모뎀 사용자의 암호를 추가적으로 요청합니다. 자세한 내용은 섹션 2.9.1을 참조하십시오.
- 사용자에게 다이얼 인 계정을 자주 바꾸도록 합니다.
- 사용자가 직원이 아니게 되면 즉시 시스템 액세스를 취소합니다.
- 원격 사용을 검토하기 위한 정기 감사 일정을 수립합니다.
- 모뎀 및 다이얼 백 장비를 단일 HP-UX 시스템에 연결하여 네트워크 서비스가 해당 지점에서 대상 시스템에 도달하도록 합니다.
- UUCP 액세스에 대해 다이얼 백의 예외를 만듭니다. 적절하게 UUCP를 구성하면 추가적인 제한이 가능합니다. 자세한 내용은 *uucp*(1)를 참조하십시오.

  다른 예외는 kermit을 통한 파일 전송이 될 수 있습니다. 자세한 내용은 *kermit*(1)를 참조하십시오.
- 알려지지 않은 요인으로 인해 보안 문제가 발생하면 네트워크와 전화 액세스를 모두 종료하고 네트워크 관리자에게 알립니다.

- 다이얼 백 모뎀 시스템을 구성할 때 보안을 최대화하려면 다이얼 아웃 기능에만 다이얼 아웃 메커니즘을 사용합니다. 다이얼 인을 받아들이도록 구성하지 마십시오. 다이얼 인 서비스에 대해서는 다른 전화선의 다른 모뎀을 사용합니다.
- 모뎀의 전화 번호를 등록되지 않은 상태로 다른 비즈니스 전화와 별도의 시스템에 보관합니다. 다이얼 인 전화 번호를 공개하지 마십시오.
- 물리적으로 모뎀의 보안을 유지합니다.
- 호출자 ID를 사용하여 모뎀에 들어오는 모든 호출을 식별합니다.
- 모뎀 회선에서 호출 전달이나 다른 추가 전화 서비스를 허용하지 마십시오. 또한 휴대폰 모뎀을 사용하지 마십시오.
- 원격 및 로컬 액세스를 위해 HP-UX AAA 서버 제품 설치를 고려합니다. HP-UX AAA 서버는 산업 표준 RADIUS(Remote Authentication Dial-In User Service) 프로토콜을 사용하여 네트워크 진입점에서 사용자 네트워크 액세스의 인증, 권한 부여 및 계정을 제공합니다. 자세한 내용은 **HP-UX AAA Server Administrator's Guide**를 참조하십시오.
- Mobile IPv6을 사용하는 모바일 연결의 경우 HP-UX IPSec를 사용하여 Mobile IPv6 클라이언트와 홈 에이전트 간의 Mobile IPv6 프로토콜 메시지를 암호화하고 인증합니다. 자세한 내용은 **HP-UX IPSec Administrator's Guide**를 참조하십시오.

2.9.1 /etc/dialups 및 /etc/d_passwd를 사용하여 액세스 제어

원격 사용자 식별의 보안을 강화하려면 `/etc/dialups` 및 `/etc/d_passwd` 파일에 항목을 추가합니다. 이러한 파일은 로그인의 다이얼 업 보안 기능을 제어하는 데 사용됩니다. 자세한 내용은 *dialups*(4) 및 *login*(1)을 참조하십시오.

`/etc/dialups` 파일이 있으면 login 프로세스는 터미널을 `/etc/dialups`에 나열된 터미널과 비교합니다. 터미널이 `/etc/dialups`에 있을 경우 `login`에서 암호를 요청합니다. 이 암호는 `/etc/d_passwd`에 있는 암호와 비교됩니다.

또한 `/etc/passwd` 파일이 암호를 확인하는 데 사용됩니다.

다음은 `/etc/dialups` 파일을 구성하는 예제입니다.

```
# vi /etc/dialups(허용되는 터미널 나열)
/dev/ttyd0p1
/dev/ttyd0p2
# vi /etc/d_passwd
/usr/bin/sh:xxxencrypted-passwordxxxxxxxxx:comments
/usr/bin/ksh:xxxencrypted-passwordxxxxxxxx:comments
/sbin/sh:xxxencrypted-passwordxxxxxxxxx:comments
```

사용자에게 다음 메시지가 표시됩니다.

```
Login:
Password:
Dialup password:
```

`/etc/d_passwd`에 있는 암호를 변경하려면 다음과 같이 `passwd` 명령을 사용합니다.

```
# passwd -F /etc/d_passwd shell_path
```

shell_path는 /etc/d_passwd에 나열된 셸 경로입니다.

2.10 로그인 배너 보안 유지

로그인 배너는 시스템 이름, 릴리즈 버전, 시스템 용도 등의 시스템 정보를 표시하는 데 사용되는 경우가 많습니다. 이 정보는 권한 없는 사용자가 시스템을 파악하는 데 도움이 될 수 있습니다. 다음은 보다 안전한 로그인 배너를 만들기 위한 몇 가지 지침입니다.

- 법무 부서와 논의하여 적절한 메시지를 결정합니다.
- 권한 없는 사용을 금지하는 경고를 배너 메시지에 추가합니다.
- 로그인 방법에 관계없이 모든 배너에 같은 내용이 표시되게 합니다.

다음과 같은 방법으로 배너를 수정할 수 있습니다.

- /etc/copyright 및 /etc/motd에 정의된 login 배너를 수정합니다.
- /etc/issue에 정의된 telnet 배너를 수정합니다. telnetd -b 배너 파일 명령은 사용자 정의 배너를 정의합니다. /etc/issue를 로그인 배너로 사용하려면 /etc/inetd.conf 파일에 다음 줄을 추가합니다.

  **telnet stream tcp nowait root /usr/lbin/telnetd **
  telnetd -b /etc/issue

  inetd가 telnetd를 시작하면 /etc/issue에 있는 배너가 사용됩니다. 자세한 내용은 *inetd*(1M), *telnetd*(IM) 및 *inetd.conf*(4)를 참조하십시오.

- ftpd 구성 파일인 /etc/ftpd/ftpaccess에 정의된 ftp 배너를 수정합니다. 인사말, 배너, 호스트 이름, 메시지 등의 다른 표시되는 메시지는 /etc/ftpd/ftpaccess에서 정의됩니다. 자세한 내용은 *ftpdaccess*(4) 및 *ftpd*(1M)를 참조하십시오.

다음은 로그인 배너를 표시하는 안전하지 않은 telnet 예제입니다.

```
# telnet computerAmy
```

telnet 로그인 배너는 릴리즈 버전과 컴퓨터 종류를 표시합니다. 권한 없는 사용자가 telnet을 사용하여 computerAmy에 액세스하려는 경우 이 정보가 큰 도움이 될 수 있습니다.

다음은 보다 안전한 로그인 배너를 표시하는 telnet 예제입니다.

```
$ telnet computerMom
Trying...
Connected to computerMom.city.company.com.
Escape character is '^]'.
Local flow control on
Telnet TERMINAL-SPEED option ON
************************************************************
This is a private system operated for Hewlett-Packard company business. Authorization from HP
management is required to use this system. Use by unauthorized persons is prohibited.
************************************************************
login: Connection closed by foreign host.
```

2.11 root 계정 보호

다음은 root 계정 보호를 위한 제안입니다.

- root 암호를 공유하지 마십시오.
- /를 root 홈 디렉토리로 사용하지 마십시오.
- `last -R` 및 `lastb -R`의 출력을 검사하여 비정상적이거나 실패한 root 로그인과 root로 로그인한 사람을 확인합니다.
- `/var/adm/sulog`를 검사하여 `su root` 명령을 사용하려는 시도를 확인합니다.
- `logins -d` 명령을 사용하여 UID가 0인 권한 없는 계정을 찾습니다.

다음 절에서는 root 계정을 보호하는 방법에 대해 자세히 설명합니다.

2.11.1 root 계정 액세스 모니터링

root 액세스 권한이 필요한 시스템 관리자가 두 명 이상 있는 경우 이러한 관리자를 추적하는 방법에 대한 제안은 다음과 같습니다.

- 시스템 콘솔에서 직접 root 로그인만 허용합니다. 다음과 같이 `console` 항목만 있는 `/etc/securetty` 파일을 만듭니다.

  ```
  #echo console > /etc/securetty
  ```

  이 제한 사항은 UID가 0인 모든 로그인 이름에 적용됩니다. 자세한 내용은 *login*(1)을 참조하십시오.

- 관리자가 개인 계정에서 `su root` 명령을 사용하여 root에 액세스하도록 합니다. 예를 들면 다음과 같습니다.

  ```
  login:me
  $ su root
  password:xxxx
  ```

- `/var/adm/sulog`를 모니터링하여 `su`를 사용해서 root에 액세스한 사람을 확인합니다.
- 각 시스템 관리자에 대해 별도의 root 계정을 구성합니다.

  ```
  # vipw
  root:xxx:0:3::/home/root:/sbin/sh
  root1:xxx:0:3::/home/root1:/sbin/sh
  root2:xxx:0:3::/home/root2:/sbin/sh
  ```

- 다음과 같이 각 시스템 관리자의 내역 파일을 모니터링합니다.

  ```
  #more ~root1/.sh_history
  #more ~root2/.sh_history
  ```

- `/var/adm/syslog`에서 성공 및 실패한 `su` 시도를 모니터링합니다.

2.11.2 제한된 수퍼유저 액세스를 위해 Restricted SMH Builder 사용

수퍼유저가 아닌 사용자에게 제한된 수퍼유저 액세스 권한을 부여해야 하는 경우 Restricted SMH Builder를 활성화할 수 있습니다. Restricted SMH Builder를 사용하여 사용자에 대해 선택된 SMH 영역을 활성화하거나 비활성화할 수 있습니다. Restricted SMH Builder를 활성화하려면 다음을 입력합니다.

smh -r

제한된 액세스 권한을 가진 사용자가 SMH를 실행하면 정의된 영역에서 수퍼유저 상태를 갖게 되며 메뉴에서 해당 SMH 영역만 볼 수 있습니다. SMH의 다른 모든 영역은 사용자에게 표시되지 않습니다. 액세스 권한이 없는 사용자가 SMH를 실행하면 수퍼유저여야 한다는 오류 메시지가 표시됩니다.

SMH에 다른 응용 프로그램을 추가하고 제한된 액세스를 설정할 수도 있습니다.

2.11.3 수퍼유저 액세스 검토

/var/adm/sulog 파일은 실패를 포함하여 su root 명령의 모든 시도를 기록합니다. 성공한 시도에는 더하기(+) 플래그가 지정되고 실패한 시도에는 빼기(-) 플래그가 지정됩니다. root만 /var/adm/sulog 파일을 볼 수 있습니다. 예를 들면 다음과 같습니다.

su root

```
Password:
```

ll /var/adm/sulog

```
-rw------- 1 root root 690 Aug 17 19:37 /var/adm/sulog
```

다음 예제에서 userone은 성공적으로 su 명령을 사용하여 root에 액세스했습니다. 두 번째 사용자 usertwo는 실패했습니다. 또한 usertwo는 su를 사용하여 gooduser1에 액세스하는 데에도 실패했습니다.

more /var/adm/sulog

```
SU 08/17 19:10 + 0 userone-root
SU 08/17 19:36 - 0 usertwo-root
SU 08/17 19:36 - 0 usertwo-root
SU 08/17 19:36 + 0 userone-root
SU 08/17 19:37 - 0 usertwo-gooduser1
```

3 HP-UX Bastille

HP-UX Bastille은 보안 강화 및 잠금 도구로, HP-UX 운영 체제의 보안을 향상하는 데 사용할 수 있습니다. Bastion Host와 비슷한 기능과 기타 강화 및 잠금 확인 목록을 인코딩하여 시스템별로 사용자 정의 잠금을 제공합니다.

Bastille은 원래 Linux 시스템에서 사용하기 위해 개방형 소스 커뮤니티에서 개발했습니다. HP는 Bastille의 내용에 상당한 기여를 했으며 이를 HP-UX Bastille을 개발하는 기초로 활용했습니다.

이 장의 내용은 다음과 같습니다.

- 기능 및 이점(섹션 3.1)
- HP-UX Bastille 설치(섹션 3.2)
- HP-UX Bastille 사용(섹션 3.3)
- HP-UX Bastille을 사용하여 변경 사항 되돌리기(섹션 3.4)
- 파일 위치(섹션 3.5)
- 팁 및 문제 해결(섹션 3.6)
- HP-UX Bastille 제거(섹션 3.7)

3.1 특징 및 장점

HP-UX Bastille에서는 다음과 같은 기능과 이점을 제공합니다.

- 시스템 잠금
  - 데몬 및 시스템 설정을 더욱 안전하게 구성합니다.
  - `pwgrd`와 같은 필요하지 않은 서비스를 해제합니다.
  - 사용자 인터페이스를 통해 사용자를 교육합니다.
  - 자동으로 실행되도록 Software Assistant 및 Security Patch Check를 구성합니다.
  - IPFilter 기반 방화벽을 구성합니다.
- 보안 구성 상태 보고
  - 보안 구성 상태 보고서를 생성합니다.
  - HP-UX Bastille 구성 기준을 만들고 시스템의 현재 상태를 저장된 기준(편차)과 비교합니다.
- SIM(System Insight Manager)과 통합
  - 시스템을 잠그고 SIM을 통해 보고서를 생성합니다.
  - SIM 서버 잠금에 대한 미리 테스트된 구성 `SIM.config`를 제공합니다.

3.2 HP-UX Bastille 설치

HP-UX Bastille은 운영 환경 미디어에 권장 소프트웨어로 포함되며 Ignite-UX 또는 Update-UX에서 설치 및 실행할 수 있습니다. HP-UX Bastille 설치에 대한 자세한 내용은 **HP-UX 11i v3 설치 및 업데이트 설명서**를 참조하십시오.

HP-UX Bastille의 최신 버전을 다운로드하려면 다음 웹 사이트를 참조하십시오.

http://www.hp.com/go/bastille

3.3 HP-UX Bastille 사용

HP-UX Bastille을 대화형 또는 비대화형으로 사용하여 다음을 수행할 수 있습니다.

- 시스템을 잠그려면(보안 구성 파일 만들기 또는 기존 보안 구성 파일 적용) 다음을 입력합니다.

  ```
  # bastille -x
  ```

- 여러 시스템에서 보안 구성 파일을 복제하려면 다음을 입력합니다.

  ```
  # bastille -b -f file
  ```

- 시스템의 구성 상태에 대한 보고서를 생성하려면 다음을 입력합니다.

  ```
  # bastille --assess
  ```

- HP-UX Bastille 구성 기준을 만들고 시스템의 현재 상태를 저장된 기준과 비교합니다. 기준을 저장하려면 다음을 입력합니다.

  ```
  # bastille_drift --save_baseline baseline
  ```

  시스템 상태를 지정된 기준과 비교하려면 다음을 입력합니다.

  ```
  # bastille_drift --from_baseline baseline
  ```

자세한 내용은 *bastille*(1M) 및 *bastille_drift*(1M)를 참조하십시오.

참고: 새 소프트웨어나 패치를 설치할 때마다 `bastille_drift` 유틸리티를 다시 실행하여 이러한 패치나 소프트웨어가 시스템 상태를 변경했는지 확인합니다. 또한 `bastille_drift` 유틸리티를 사용하면 `swverify`가 `-x fix=true` 옵션 또는 `-F` 옵션을 사용하여 실행되어 공급업체별 수정 스크립트를 실행할 때 시스템 상태 변경을 식별할 수 있습니다.

3.3.1 대화형으로 HP-UX Bastille 사용

HP-UX Bastille은 Perl/Tk를 통해 구현되는 X 인터페이스를 사용하여 대화형으로 실행됩니다. 이 인터페이스에는 X 서버가 필요하며 다음과 같은 기능을 제공합니다.

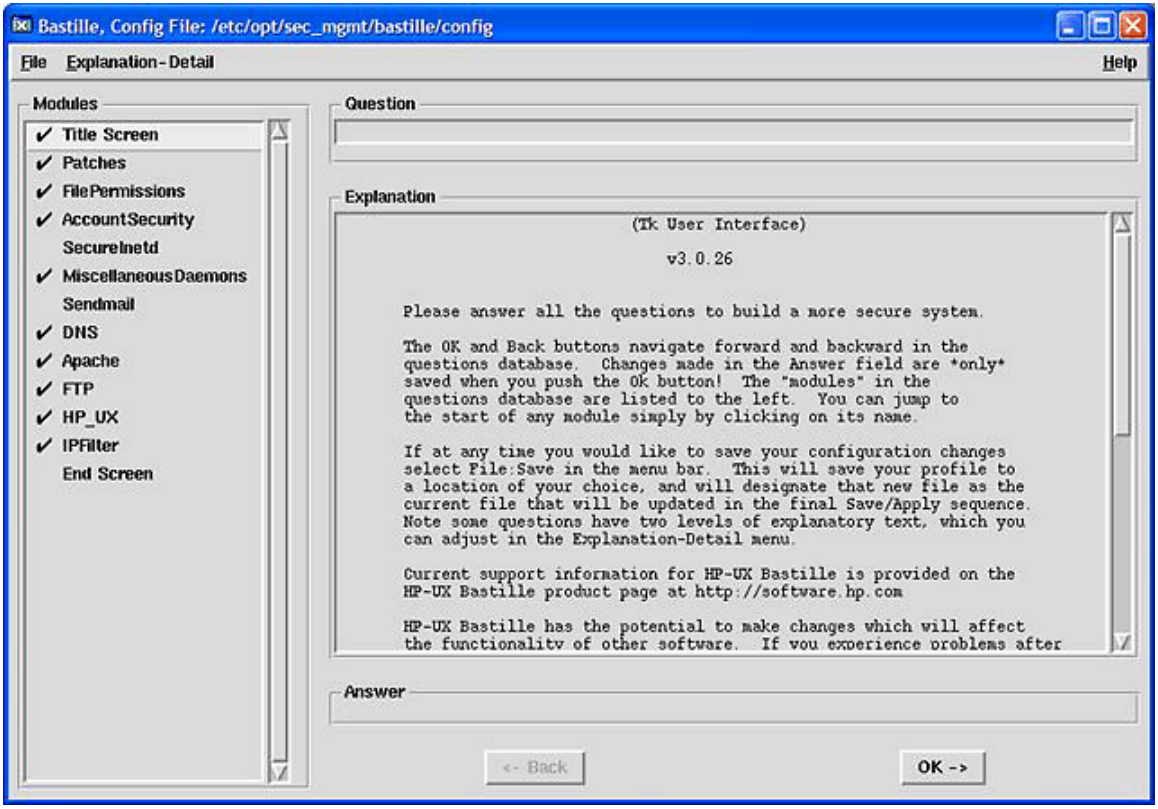
- 질문 모듈 간 임의 액세스
- 사용자의 진행 상황을 표시하는 완료 표시기
- 다음 옵션을 사용하여 암호화된 채널을 통한 X11 트래픽의 터널링

  ```
  # ssh -X
  ```

  자세한 내용은 *ssh*(1)를 참조하십시오.

그림 3-1은 HP-UX Bastille 사용자 인터페이스의 주 화면을 보여 줍니다.

그림 3-1 HP-UX Bastille 사용자 인터페이스



사용자 인터페이스에서는 모듈별로 그룹화된 일련의 질문을 통해 사용자를 안내하여 교육합니다(표 3-1 참조). 각 질문에서는 보안 문제에 대해 설명하고 HP-UX 시스템을 잠그는 데 필요한 결과 작업을 설명합니다. 또한 각 결정의 비용과 이점에 대해 높은 수준으로 자세히 설명하며 사용자는 도구를 사용하여 문제를 처리하는 방법을 결정합니다.

모든 질문에 대답하면 HP-UX Bastille에서는 각 잠금 단계를 수행할 수 있도록 자동 지원을 제공합니다. 자동으로 수행할 수 있는 작업을 수행한 다음 사용자가 수행해야 하는 나머지 수동 작업에 대해 수행할 작업 목록을 만듭니다. HP-UX Bastille 잠금 프로세스를 완료하려면 해당 수행할 작업을 수행해야 합니다.

표 3-1 HP-UX Bastille 질문 모듈

모듈 이름	설명
Patches Applications	보안 정보 준수 확인에 도움이 되도록 설치 및 구성합니다.
File Permissions	`SUID` 및 다른 권한 조정을 수행합니다.
Account Security	로그인 설정 및 `cron`에 대한 액세스를 구성합니다.
Secure `inetd`	필요하지 않은 `inetd` 서비스를 해제합니다.
Miscellaneous Daemons	필요하지 않은 경우가 많고 보안 위험이 있는 서비스를 해제합니다.

표 3-1 HP-UX Bastille 질문 모듈 (계속)

모듈 이름	설명
sendmail	메일을 더욱 안전하게 구성하거나 사용자가 메일을 해제할 수 있게 합니다.
DNS	DNS를 해제하거나 더욱 안전하게 구성합니다.
Apache	Apache 웹 서버를 더욱 안전하게 구성합니다.
FTP	FTP 서버를 더욱 안전하게 구성합니다.
HP-UX	HP-UX 플랫폼에 고유한 보안 구성 작업을 수행합니다.
IPFilter	IPFilter 기반 방화벽을 만듭니다.

3.3.2 비대화형으로 HP-UX Bastille 사용

구성 엔진을 통해 직접 보안 강화를 수행할 수 있습니다. 이 방법은 동일한 운영 체제와 응용 프로그램을 설치한 여러 시스템으로 보안 구성을 복제하는 데 유용합니다. 구성 엔진에서는 미리 정의된 구성 파일을 사용합니다. 이 옵션에서는 대화형 세션에서 기본 위치에 만든 파일을 사용하거나 다음과 같이 `-f` 옵션에서 지정한 대체 파일을 사용할 수 있습니다.

```
# bastille -b -f file
```

3.3.3 시스템 구성

시스템을 구성하거나 나중에 다른 시스템에서 다시 사용할 수 있는 구성 파일을 만들려면 다음 단계를 수행합니다.

1. HP-UX Bastille에서 시스템 구성과 설정을 변경해야 하므로 `root` 사용자로 변경합니다. HP-UX Bastille이 로컬에서 실행되고 있지 않은 경우 `ssh`(Secure Shell) 또는 IPSec를 통해 X11 트래픽을 터널링하여 네트워크 노출을 제한하거나 로컬 사용자 및 원격 사용자의 공격을 해결하는 더 완벽한 데스크탑 공유 솔루션을 사용해야 합니다.
2. 시스템을 변경할 경우 HP-UX Bastille을 대화형을 사용할지 비대화형으로 사용할지 결정합니다. 처음 사용자는 배포 시 `DMZ.config`와 같이 미리 작성된 구성 파일이 함께 제공되지 않은 경우 HP-UX Bastille을 대화형으로 실행하여 구성 프로파일을 만듭니다. 한 시간 정도를 들여 모든 질문을 읽고 대답합니다.

   HP-UX Bastille의 대화형 및 비대화형 사용에 대한 자세한 내용은 섹션 3.3을 참조하십시오.
3. 2단계의 결정에 따라 해당하는 절차(대화형 또는 비대화형)를 수행합니다.

   대화형 절차

   a. HP-UX Bastille을 시작합니다.

      처음 사용자는 HP-UX Bastille을 대화형으로 실행하여 구성 프로파일을 만들어야 합니다. 이 도구는 설치 시 `PATH` 환경 변수가 업데이트되므로 HP-UX Bastille을 설치한 후 로그아웃했다가 다시 로그인한 경우 도구를 시작하려면 다음을 입력합니다.

      ```
      # bastille
      ```

      `PATH`가 업데이트되지 않은 경우 다음을 입력하여 HP-UX Bastille을 시작합니다.

      ```
      # /opt/sec_mgmt/bastille/bin/bastille
      ```

      현재 구성과 관련된 질문 범주만 표시됩니다.

b. 질문에 대답합니다.

질문은 기능별로 분류되어 있으며 범주가 완료되었는지 여부를 나타내기 위해 확인 표시를 완료 표시기로 사용합니다. 이를 통해 진행 상황을 추적할 수 있습니다.

질문에 대답할 때 **Explanation-Detail** 메뉴를 사용하여 더 자세하거나 덜 자세한 설명 간을 전환합니다. 일부 질문에는 자세한 대답과 간단한 대답 중 하나가 없습니다.

c. 구성을 저장하고 변경 사항을 적용합니다.

언제든지 메뉴 표시줄을 사용하여 구성 파일을 저장하거나 로드할 수 있습니다. **Save As** 옵션을 사용하면 도구에서는 구성이 완료되지 않았거나 나중에 변경될 수 있다고 간주하므로 더 큰 구성 파일이 생성됩니다. **Save/Apply** 옵션을 사용하면 도구에서는 구성이 완료되었다고 간주하며 추가 필터링을 적용할 수 있습니다. 생성되는 구성 파일의 크기는 다를 수 있지만 HP-UX Bastille의 기능은 두 경우 모두 동일합니다. **Save/Apply** 옵션에서는 항상 구성 파일을 HP-UX Bastille 제목 표시줄에 나열된 현재 위치에 저장합니다.

비대화형 절차

a. 구성 파일이 아직 없는 경우 HP-UX Bastille을 대화형으로 실행하여 구성 파일을 만듭니다.

일부 배포에서는 기본 구성 파일이 제공되지 않습니다. 이 경우 처음에 HP-UX Bastille을 대화형으로 실행하여 구성 파일을 만들어야 합니다. 자세한 내용은 "대화형 절차"를 참조하십시오.

b. 구성 파일을 복제하려는 각 시스템에 복사합니다.

구성 파일 `/etc/opt/sec_mgmt/bastille/config`를 첫 번째 시스템의 위치에서 다른 시스템의 동일한 위치로 복사합니다.

```
# bastille -b -f file
```

참고: 일부 질문은 운영 체제나 설치된 보안 소프트웨어와 관련이 있으므로 복제할 시스템에는 구성 파일을 만든 시스템과 동일한 운영 체제 및 소프트웨어가 설치되어 있어야 합니다.

c. 복제할 각 시스템에 HP-UX Bastille을 설치합니다.

이 작업은 한꺼번에 수행할 수 있으며 나중에 검토하기 위해 작업 및 오류 로그를 수집할 수 있습니다. 다음을 입력합니다.

```
# bastille -b
```

4. 로그 파일을 검토합니다.

로그를 실시간으로 보려면 다음을 입력합니다.

```
# tail -f log_file
```

작업 로그 파일 `/var/opt/sec_mgmt/bastille/log/action-log`에는 시스템을 변경할 때 HP-UX Bastille에서 수행한 특정 단계가 포함됩니다. 이 로그 파일은 시스템에 변경 사항을 적용하는 경우에만 만들어집니다.

오류 로그 파일 /var/opt/sec_mgmt/bastille/log/error-log에는 HP-UX Bastille에서 시스템을 변경할 때 발생한 오류가 포함됩니다. 이 로그 파일은 실행 중 오류가 발생하는 경우에만 만들어집니다.

5. 수행할 작업 목록에 나열된 항목을 수행합니다.

   이 도구에서는 자동으로 실행할 수 있는 작업을 수행한 후 사용자가 수동으로 수행해야 하는 나머지 작업에 대해 설명하는 수행할 작업 목록 /var/opt/sec_mgmt/bastille/TODO.txt를 만듭니다. 다시 부팅이 필요한 변경 사항이 있는 경우 이 목록에 다시 부팅도 포함됩니다.

   안전한 구성을 위해서는 수행할 작업 목록의 작업을 완료해야 합니다.

참고: 변경 사항을 시스템에 적용한 경우에만 수행할 작업 목록이 만들어집니다.

3.4 HP-UX Bastille을 사용하여 변경 사항 되돌리기

보안 구성을 HP-UX Bastille을 실행하기 전의 상태로 되돌리려면 다음을 입력합니다.

```
# bastille -r
```

이전 HP-UX Bastille 상태로 복원하기 위해 수동 작업을 수행해야 하는 경우 되돌리기 프로세스에서는 var/opt/sec_mgmt/bastille/TOREVERT.txt 파일을 만듭니다 되돌리기 프로세스를 완료하려면 이 파일에 나열된 작업을 수행해야 합니다.

되돌리기 옵션을 실행한 후 TOREVERT.txt 파일에서 되돌리기 프로세스를 완료하기 위해 수행해야 하는 수동 작업이 있는지 확인합니다. 이 파일의 위치는 /var/opt/sec_mgmt/bastille/TOREVERT.txt입니다.

참고: 일부 방화벽 옵션 외에 시스템 되돌리기는 HP-UX Bastille을 사용하여 시스템을 덜 안전하게 만드는 유일한 방법입니다.

3.5 파일 위치

다음 목록에서는 주요 HP-UX Bastille 파일에 대해 설명하고 해당 위치를 보여 줍니다.

- **기본 구성 파일** - 다른 파일 이름이 지정되지 않은 경우 가장 최근에 저장한 세션에 대한 대답이 포함되어 있습니다.

  /etc/opt/sec_mgmt/bastille/config

- **구성 로그 파일** - HP-UX Bastille을 통해 적용된 파일은 두 시스템에 동일한 HP-UX 버전, 비슷하게 설치된 응용 프로그램 세트 및 구성이 있는 경우 HP-UX Bastille이 완전히 잠근 Bastille 관련 항목을 잠급니다. 시스템이 다른 경우 구성 파일이 대상 시스템과 관련이 없는 추가 질문을 포함할 수 있거나 원격 시스템에 필요한 질문을 누락할 수 있습니다. HP-UX Bastille은 전자의 경우에는 사용자에게 알려 주고, 후자의 경우에는 오류를 생성합니다. 그런 다음 누락된 질문에 응답할 기회를 주거나 그래픽 인터페이스에서 기타 누락 질문을 제거합니다.

  /var/opt/sec_mgmt/bastille/log/Assessment/assessment-log.config

- **오류 로그 파일** - HP-UX Bastille에서 시스템을 변경할 때 발생한 오류가 포함되어 있습니다.

/var/opt/sec_mgmt/bastille/log/error-log

- **작업 로그 파일** - HP-UX Bastille에서 시스템을 변경할 때 수행한 특정 단계가 포함되어 있습니다.

  /var/opt/sec_mgmt/bastille/log/action-log

- **수행할 작업 목록 파일** - 시스템이 안전할 수 있도록 사용자가 수행해야 하는 나머지 수동 작업이 포함되어 있습니다.

  /var/opt/sec_mgmt/bastille/TODO.txt

- **되돌리기 작업 스크립트** - HP-UX Bastille에서 변경한 파일을 HP-UX Bastille을 실행하기 전의 상태로 되돌립니다.

  /var/opt/sec_mgmt/bastille/revert/revert-actions

  이 스크립트는 되돌리기 기능의 일부입니다.

- **되돌리기 작업 파일** - 시스템을 HP-UX Bastille을 실행하기 전의 상태로 되돌리는 작업을 완료하기 위해 수행해야 하는 수동 작업이 포함되어 있습니다.

  /var/opt/sec_mgmt/bastille/TOREVERT.txt

- **평가 보고서 파일** - 이 파일은 HTML, 텍스트 및 HP-UX Bastille 로그 파일과 같은 형식으로 지정됩니다.

  /var/opt/sec_mgmt/bastille/log/Assessment/assessment-report.HTML

  /var/opt/sec_mgmt/bastille/log/Assessment/assessment-report.txt

  /var/opt/sec_mgmt/bastille/log/Assessment/assessment-report-log.txt

- **구성 편차 파일** - 마지막으로 HP-UX Bastille을 실행한 이후 시스템에서 발생한 구성 편차에 대한 정보가 포함되어 있습니다. 이 파일은 이전의 HP-UX Bastille 구성을 시스템에 적용한 경우에만 만들어집니다.

  /var/opt/sec_mgmt/bastille/log/Assessment/Drift.txt

3.6 팁 및 문제 해결

이 절에서는 알려진 문제 해결 문제에 대한 기본 정보를 제공합니다.

- HP-UX Bastille에서 변경한 내용으로 인해 다른 소프트웨어가 작동을 중지할 수 있습니다.

  문제를 해결하려면 다음을 입력합니다.

  # **bastille -r**

  이 명령은 시스템을 HP-UX Bastille을 실행하기 전의 상태로 되돌린 다음 문제가 제거되었는지 확인합니다.

- $DISPLAY not set, cannot use X.

  사용자가 X 인터페이스를 요청했지만 $DISPLAY 환경 변수가 설정되어 있지 않습니다. 문제를 해결하려면 이 환경 변수를 원하는 디스플레이로 설정합니다.

- System is in original state...

사용자가 HP-UX Bastille에서 변경한 내용을 `-r` 옵션을 사용하여 되돌리려고 했지만 되돌릴 변경 사항이 없습니다.

- `Must run HP-UX Bastille as root`

  HP-UX Bastille에서 변경하는 내용은 시스템 파일에 영향을 주므로 `root` 사용자로 HP-UX Bastille을 실행해야 합니다.

- 파일을 열거나 복사하거나 읽을 수 없습니다.

  이러한 작업을 수행하는 데 문제가 있다는 오류 메시지는 일반적으로 로컬 시스템에서 root 사용자를 신뢰하지 않는 NFS 파일 시스템과 관련이 있습니다. 자세한 내용은 `fstab` 맨페이지의 `options` 절을 참조하십시오.

- 개별 구성 파일과 관련된 오류

  시스템이 너무 심하게 수정되어 HP-UX Bastille에서 효과적으로 변경할 수 없다거나 HP-UX Bastille 설치 디렉토리의 파일, 위치 또는 권한이 변경되었음을 나타내는 개별 구성 파일과 관련된 문제를 알리는 오류입니다.

- HP Secure Shell에서 암호가 만료될 때 시스템을 즉시 잠그는 경우 다음의 HP IT 리소스 센터에서 다운로드할 수 있는 PAM 패치: PHCO_24839(HP-UX 11.11)가 필요할 수 있습니다.

  https://www2.itrc.hp.com/service/patch/mainPage.do

- HP-UX Bastille에서는 IPFilter를 사용하여 방화벽을 구성합니다.

  가장 일반적인 충돌은 방화벽과 관련되어 발생합니다. HP-UX Bastille에서 명시적으로 해제하지 않은 네트워크 서비스가 작동하지 않는 경우 방화벽 규칙이 필요한 포트를 통과하는지 확인해야 합니다.

  자세한 내용은 *ipfstat*(8) 및 *ipmon*(8)을 참조하십시오.

3.7 HP-UX Bastille 제거

HP-UX Bastille을 시스템에서 제거해도 시스템을 HP-UX Bastille을 실행하기 전의 상태로 되돌리지는 않습니다. 대신 소프트웨어를 제거하면 `revert-actions` 스크립트가 남습니다. 이 스크립트를 사용하면 관리자가 HP-UX Bastille에서 수행한 구성 파일을 HP-UX Bastille을 설치하지 않고도 원래 상태로 되돌릴 수 있습니다. 대부분의 경우 HP-UX Bastille 변경 사항은 파일 수준에서 기록되므로 `revert-actions` 스크립트는 수정된 파일만 되돌릴 수 있습니다.

경우에 따라 동일한 파일에서 사용자가 중간에 변경한 내용이 있더라도 HP-UX Bastille에서 더 세부적으로 변경하여 프로그래밍 방식으로 되돌릴 수 있습니다. 예를 들어 사용 권한이 변경된 파일을 수정한 경우에도 사용 권한을 원래 상태로 되돌릴 수 있습니다.

1. `swremove`를 사용하여 HP-UX Bastille을 HP-UX 시스템에서 제거할 수 있습니다.
2. (선택 사항) HP-UX Bastille을 제거한 시스템의 변경 사항을 되돌리려면 다음 명령을 입력합니다.

```
# cd /var/opt/sec_mgmt/bastille/revert/
# chmod 0500 revert-actions
# ./revert-actions
# mv revert-actions revert-actions.last
```

3. 되돌리기 작업 목록 /var/opt/sec_mgmt/bastille/TOREVERT.txt이 만들어졌는지 확인합니다. 이 파일이 있으면 목록의 작업을 수행하여 되돌리기 프로세스를 완료합니다.

4 HP-UX Standard Mode Security Extensions

이 장에서는 HP-UX SMSE(HP-UX Standard Mode Security Extensions)에 대해 설명합니다. 이 장의 내용은 다음과 같습니다.

- 개요(섹션 4.1)
- 보안 속성 및 사용자 데이터베이스(섹션 4.2)

4.1 개요

HP-UX SMSE(HP-UX Standard Mode Security Extensions)는 사용자와 운영 체제 보안을 향상시키는 기능 그룹입니다. HP-UX SMSE에는 표준 모드의 시스템에 대한 HP-UX 감사 시스템, 암호 및 로그인 개선 사항 또는 변경 사항이 포함되어 있습니다. 이전에는 이러한 기능이 트러스트된 모드로 변환된 시스템에서만 지원되었습니다. HP-UX SMSE에서는 표준 모드 시스템에서 이러한 기능을 사용할 수 있습니다.

참고: 트러스트된 모드로 실행되는 시스템에서는 HP-UX SMSE를 사용하지 **않는 것이 좋습니다**. HP-UX SMSE는 HP-UX 시스템을 트러스트된 모드로 변환해서만 여러 계정과 암호 정책을 사용할 수 있는 표준 모드에서 사용할 수 있습니다. HP-UX SMSE로 구성된 정책은 트러스트된 모드로 실행되는 시스템에서는 강제 적용되지 않습니다.

시스템이 트러스트된 모드로 변환되었는지 여부를 확인하려면 다음 파일을 확인합니다.

`/tcb/files/auth/system/default`

이 파일이 있을 경우 시스템이 트러스트된 모드로 실행 중입니다. 시스템을 표준 모드로 다시 변환하려면 `sam(1M)` 명령을 사용합니다.

각 HP-UX SMSE 보안 기능으로 지원하는 구성에 대한 자세한 내용은 *security*(4)를 참조하십시오.

HP-UX SMSE는 새로운 기능인 **사용자 데이터베이스**를 제공합니다. 이전에는 모든 HP-UX 보안 속성과 암호 정책 제한이 시스템 범위를 기준으로 설정되었습니다. 사용자 데이터베이스가 도입되면서 이제 사용자별로 시스템 범위 기본값을 재정의하는 보안 속성을 설정할 수 있습니다.

아래의 트러스트된 모드 기능은 HP-UX SMSE 표준 모드에서 사용할 수 있습니다.

- 시스템의 모든 사용자 및 이벤트 감사
- 마지막으로 성공한 사용자 로그인과 실패한 사용자 로그인 표시
- 지나치게 많은 인증이 실패하면 사용자 계정 잠금
- 암호 내역 표시
- 비활성 계정 만료
- null 암호를 사용하는 사용자 로그인 차단
- 특정 기간에 사용자 로그인 제한
- `userdbset` 명령 사용이 사용자의 권한 부여를 기준으로 제한될 수 있습니다. 자세한 내용은 *userdbset*(1M)를 참조하십시오.

- `userstat` 명령은 로컬 사용자의 계정 상태를 표시합니다. 로컬 사용자 계정의 상태를 확인하고 계정 잠금과 같은 비정상 상태를 보고합니다. 자세한 내용은 *userstat*(1M)를 참조하십시오.

4.2 보안 속성 및 사용자 데이터베이스

이전에는 표준 모드에서 HP-UX 보안 속성과 암호 정책 제한이 시스템 범위를 기준으로 설정되었습니다. 사용자 데이터베이스가 도입되면서 이제 사용자별로 시스템 범위 기본값을 재정의하는 보안 속성을 설정할 수 있습니다.

4.2.1 시스템 보안 속성

보안 속성은 암호, 로그인 및 감사와 같은 보안 구성을 제어하는 방법을 정의합니다. 보안 속성 설명 파일 `/etc/security.dsc`는 `/etc/default/security`, `/var/adm/userdb`의 사용자 데이터베이스 또는 이 두 파일에서 정의할 수 있는 속성을 표시합니다. 일부 속성은 구성 가능하고 일부는 내부 속성입니다.

주의: 어떤 방식으로든 `/etc/security.dsc` 파일을 수정하지 마십시오.

사용자가 로그인하면 시스템에서 다음 순서로 적용 가능한 보안 속성을 확인합니다.

1. 시스템은 다음 위치에서 사용자별 속성을 검사합니다.
   - `/var/adm/userdb`
   - `/etc/passwd`
   - `/etc/shadow`

   참고: 각 사용자별 속성에 대해 값은 위의 세 파일 중 하나에 저장됩니다. 각 파일에 저장되는 속성을 확인하려면 *security*(4)를 참조하십시오.
2. 사용자별 값이 없으면 시스템은 `/etc/default/security`에 있는 구성된 시스템 범위 속성을 검사합니다.
3. 구성된 시스템 범위 속성이 없으면 시스템은 `/etc/security.dsc`의 기본 속성을 사용합니다.

4.2.2 시스템 범위 속성 구성

시스템 범위 속성을 구성하려면 다음 단계를 수행합니다.

1. 사용 가능한 리소스를 사용하여 구성을 계획합니다. 시스템 범위 속성 구성에 대한 자세한 내용은 *security*(4)를 참조하십시오.
2. 시스템 범위 기본값을 변경하려면 `vi`와 같은 텍스트 편집기를 사용하여 `/etc/default/security` 파일을 편집합니다. 주석은 파운드 기호(#)로 시작합니다. 속성은 `attribute=value` 형식으로 작성됩니다.

   예를 들어, 암호에서 시스템 범위 최소 대문자 개수를 2로 설정하려면 `/etc/default/security`에 다음 값을 입력합니다.

   `PASSWORD_MIN_UPPER_CASE_CHARS=2`

참고: 시스템 범위 보안 속성에 대한 변경 사항은 바로 적용되지 않습니다. 암호 속성은 사용자가 다음에 자신의 암호를 변경할 때 적용됩니다. 로그인 속성은 사용자가 다음에 로그인할 때 적용됩니다.

4.2.3 사용자 데이터베이스 구성 요소

HP-UX SMSE의 사용자 데이터베이스 기능에는 HP-UX 시스템의 특정 사용자에게 적용할 수 있는 파일, 명령, 맨페이지 및 사용자별 속성이 포함되어 있습니다. 사용자 데이터베이스의 이러한 요소는 모두 다음 절에서 설명합니다.

4.2.3.1 구성 파일

표 4-1에서는 사용자 데이터베이스에 사용되는 파일에 대해 간략히 설명합니다.

표 4-1 사용자 데이터베이스 구성 파일

파일	설명
/var/adm/userdb	대부분의 사용자별 정보를 저장합니다.

4.2.3.2 명령

표 4-2에서는 사용자 데이터베이스의 항목을 수정하고 관리하는 데 사용할 수 있는 명령을 설명합니다.

표 4-2 사용자 데이터베이스 명령

명령	설명
userdbset	사용자 데이터베이스에 구성된 속성 값을 변경합니다.
userdbget	사용자 데이터베이스에 구성된 속성 값을 표시합니다.
userdbck	사용자 데이터베이스에서 정보 무결성을 확인합니다.
userstat	로컬 사용자 계정의 상태를 보고합니다.

4.2.3.3 속성

개별 사용자에 대해 다음 보안 속성을 사용할 수 있습니다.

표 4-3 사용자 속성

속성	설명
ALLOW_NULL_PASSWORD	빈 암호를 사용한 로그인을 허용하거나 거부합니다.
AUDIT_FLAG	사용자를 감사하거나 감사를 중지합니다.
AUTH_MAXTRIES	사용자가 시스템에서 잠길 때까지 허용된 로그인 실패 횟수를 정의합니다.
DISPLAY_LAST_LOGIN	사용자의 마지막 로그인에 대한 정보를 표시합니다.
LOGIN_TIMES	로그인 기간을 제한합니다.
MIN_PASSWORD_LENGTH	최소 암호 길이를 정의합니다.

표 4-3 사용자 속성 (계속)

속성	설명
NUMBER_OF_LOGINS_ALLOWED	사용자마다 허용된 동시 로그인 수를 정의합니다.
PASSWORD_HISTORY_DEPTH	암호 내역 깊이를 정의합니다.
PASSWORD_MIN_LOWER_CASE_CHARS	암호에 필요한 최소 소문자 수를 정의합니다.
PASSWORD_MIN_UPPER_CASE_CHARS	암호에 필요한 최소 대문자 수를 정의합니다.
PASSWORD_MIN_DIGIT_CHARS	암호에 필요한 최소 숫자 수를 정의합니다.
PASSWORD_MIN_SPECIAL_CHARS	암호에 필요한 최소 특수 문자 수를 정의합니다.
UMASK	파일 생성에 대한 umask를 정의합니다.

참고: 이전 목록에는 사용자 데이터베이스에서 구성할 수 있는 보안 속성만 들어 있습니다. HP-UX 시스템 보안 속성에 대한 전체 목록에 대해서는 *security*(4)를 참조하십시오.

4.2.3.4 맨페이지

표 4-4에서는 사용자 데이터베이스에 사용되는 맨페이지에 대해 간략히 설명합니다.

표 4-4 사용자 데이터베이스 맨페이지

맨페이지	설명
userdb(4)	사용자 데이터베이스 사용 개요를 제공합니다.
userdbset(1M)	`userdbset` 기능과 구문에 대해 설명합니다.
userdbget(1M)	`userdbget` 기능과 구문에 대해 설명합니다.
userdbck(1M)	`userdbck` 기능과 구문에 대해 설명합니다.
userstat(1M)	`userstat` 기능과 구문에 대해 설명합니다.

4.2.4 사용자 데이터베이스에서 속성 구성

이전 HP-UX 시스템에서 보안 속성과 암호 정책 제한이 시스템 범위를 기준으로 설정되었습니다. HP-UX SMSE를 사용하여 사용자 기준으로 몇 가지 보안 속성을 구성할 수 있습니다. 사용자별로 구성된 속성은 시스템 범위 구성 속성을 재정의합니다.

사용자의 속성 값을 수정하려면 다음 단계를 수행합니다.

1. 수정할 사용자와 해당 사용자에 적용할 속성을 결정합니다.

   예를 들어, 사용자 joe가 월요일 오전 8시부터 오후 5시까지만 시스템에 로그인하도록 할 수 있습니다.

2. `userdbset` 명령을 사용하여 다음과 같이 속성을 변경합니다.

   # **`userdbset -u user-name attribute-name=attribute-value`**

   예를 들어, 사용자 joe가 오전 8시부터 오후 5시까지만 시스템에 로그인할 수 있도록 지정하려면 다음을 입력합니다.

```
# userdbset -u joe LOGIN_TIMES=Mo0800-1700
```

4.2.5 사용자 데이터베이스 문제 해결

사용자 데이터베이스 문제를 해결하려면 다음 절차를 사용합니다.

문제 1: 사용자 보안 속성이 잘못 구성된 것 같습니다. 사용자 데이터베이스에 있는 사용자 정보가 잘못 구성된 것 같으면 다음 명령을 실행합니다.

```
# userdbget -u username
```

사용자 username에 대해 구성된 속성이 표시됩니다. 속성이 잘못 구성된 경우 해당 속성을 다시 구성합니다. 자세한 내용은 "사용자 데이터베이스에서 속성 구성"을 참조하십시오.

문제 2: 사용자 데이터베이스가 제대로 작동하지 않습니다. 사용자 데이터베이스를 확인해야 하는 경우 다음 명령을 실행합니다.

```
# userdbck
```

`userdbck` 명령은 사용자 데이터베이스의 문제를 식별하고 복구합니다.

5 원격 액세스 보안 관리

HP-UX는 파일 전송, 원격 로그인, 원격 명령 실행, IP 주소 및 네트워크 클라이언트 관리, 라우팅 프로토콜, 메일 교환, 네트워크 서비스, `inetd`에 의해 시작된 보안 메커니즘, 인터넷 수퍼 데몬 등 여러 가지 원격 액세스 서비스를 제공합니다.

이 장의 내용은 다음과 같습니다.

- 인터넷 서비스 및 원격 액세스 서비스 개요(섹션 5.1)
- inetd 데몬(섹션 5.2)
- TCP Wrappers를 사용하여 스푸핑 방지(섹션 5.3)
- Secure Internet Services(섹션 5.4)
- 관리 도메인 제어(섹션 5.5)
- HP-UX SSH(Secure Shell)를 사용하여 원격 세션 보안 유지(섹션 5.6)

5.1 인터넷 서비스 및 원격 액세스 서비스 개요

이 절에서는 다양한 인터넷 서비스에 사용되는 인증 및 권한 부여 메커니즘과 보안 위험에 대해 간략하게 설명합니다.

HP-UX Internet Services에 대해서는 http://docs.hp.com/en/netcom.html#Internet%20Services에 있는 **HP-UX Internet Services Administrator's Guide** 및 **Using HP-UX Internet Services**에서 설명합니다.

또한 다음 웹 사이트에 있는 **HP-UX Remote Access Services Administrator's Guide**를 참조하십시오.

http://docs.hp.com/en/netcom.html#Internet%20Services

HP-UX Internet Services는 구성 파일에 설정된 권한 부여나 암호 확인을 통해 인증을 제공합니다. 인터넷 서비스 구성 요소 목록과 해당 액세스 확인 또는 권한 부여 메커니즘은 표 5-1을 참조하십시오.

표 5-1 인터넷 서비스 구성 요소와 액세스 확인, 권한 부여 및 인증

인터넷 서비스 구성 요소	액세스 확인, 권한 부여 또는 인증 메커니즘
`ftp`(파일 전송)	암호 확인. 또한 `/etc/inetsvcs.conf`에 정의된 Kerberos 인증 메커니즘을 사용할 수 있습니다. *ftp*(1)를 참조하십시오.
`rcp`(원격 복사)	`$HOME/.rhosts` 또는 `/etc/hosts.equiv` 파일의 항목. 또한 `/etc/inetsvcs.conf`에 정의된 Kerberos 인증 메커니즘을 사용할 수 있습니다. *rcp*(1)를 참조하십시오.
`rdist`(원격 파일 배포)	`$HOME/.rhosts` 또는 `/etc/hosts.equiv` 파일의 항목. *rdist*(1)를 참조하십시오.
`remsh`, `rexec`(원격 셸에서 실행)	`$HOME/.rhosts` 또는 `/etc/hosts.equiv` 파일의 항목. 또한 `/etc/inetsvcs.conf`에 정의된 Kerberos 인증 메커니즘을 사용할 수 있습니다. *remsh*(1)를 참조하십시오.

표 5-1 인터넷 서비스 구성 요소와 액세스 확인, 권한 부여 및 인증 (계속)

인터넷 서비스 구성 요소	액세스 확인, 권한 부여 또는 인증 메커니즘
`rlogin`(원격 로그인)	암호 확인 또는 `$HOME/.rhosts` 또는 `/etc/hosts.equiv` 파일의 항목. 또한 `/etc/inetsvcs.conf`에 정의된 Kerberos 인증 메커니즘을 사용할 수 있습니다. *rlogin*(1)을 참조하십시오.
`telnet`(TELNET 프로토콜을 사용한 원격 로그인)	암호 확인. `telnetd` 데몬에 의해 TAC 사용자 ID 옵션이 활성화되면 `telnet`에서 `$HOME/.rhosts` 또는 `/etc/hosts.equiv` 파일을 사용합니다. *telnet*(1) 및 *telnetd*(1M)를 참조하십시오.

참고: 암호를 비롯한 정보는 두 시스템 간에 일반 텍스트로 전달되고 암호화되지 않습니다. 잘 알려져 있고 서로 정의된 호스트 간에만, 방화벽을 사용하는 전용 내부 네트워크에서 인터넷 서비스를 사용하십시오. 트러스트되지 않은 네트워크에서 통신하는 경우 IPSec 또는 Kerberos를 사용하여 통신 보안을 유지합니다.

원격 액세스 서비스는 네트워크의 원격 시스템을 연결합니다. 기본적으로 원격 액세스 서비스는 비보안 환경에서 작동합니다. 보안 환경에서 작동하려면 Kerberos V5 네트워크 인증을 활성화합니다. 비보안 환경에서 원격 시스템에 액세스하려면 로그인 이름과 암호가 있어야 하고 인증 및 권한 부여를 위해 로그인 이름이 확인되지 않습니다. 보안 환경에서는 로그인 이름과 암호가 없어도 됩니다. 원격 시스템에 연결하려고 하면 Kerberos 프로토콜이 사용자가 원격 시스템에 액세스할 수 있는지 확인합니다.

5.1.1 ftp 보안 유지

권한이 없는 사용자가 `ftp` 명령을 사용하여 시스템에 액세스하려고 할 수 있습니다. 이 문제를 방지하기 위한 몇 가지 제안 사항은 다음과 같습니다.

- `ftpd -l` 명령을 사용하여 `/etc/inetd.conf`에서 `ftp` 로깅을 활성화합니다.
- `/var/adm/syslog/syslog.log` 및 `/var/adm/syslog/xferlog`의 `ftp` 로그에서 비정상적인 원격 액세스 시도를 확인합니다.

  syslogd(1M) 및 *xferlog*(5)를 참조하십시오.

- `/etc/ftpd/ftpusers`에 나열하여 `guest`, `root` 및 기타 계정의 `ftp` 액세스를 거부합니다.

  ftpusers(4)를 참조하십시오.

- 사용자의 `~/.netrc` 파일을 정기적으로 검색하고 제거합니다. `.netrc` 파일에는 `ftp` 자동 로그인 프로세스, `rexec()` 라이브러리 루틴 및 `rexec` 명령에 사용되는 로그인, 암호 및 계정 정보가 들어 있습니다.

  netrc(4)를 참조하십시오.

5.1.2 익명 ftp 보안 유지

`$HOME/.rhosts` 파일이 `/home/ftp`에 배치되어 있으면 권한이 없는 사용자가 `rlogin`을 사용하여 `ftp` 사용자로 로그인할 수 있습니다. `.rhosts` 파일은 암호 없이 `rcp`, `remsh` 또는 `rlogin`을 사용하여 로컬 계정에 액세스할 수 있는 호스트와 사용자를 지정합니다. 자세한 내용은 *hosts.equiv*(4)를 참조하십시오.

익명 ftp를 보다 안전하게 설정하기 위한 몇 가지 제안 사항은 다음과 같습니다.

- /home/ftp와 해당 자식을 모두 쓰기 불가능으로 설정합니다.

  $**chmod -R a -w /home/ftp**

- /etc/passwd의 ftp 항목이 올바르게 구성되었는지 확인합니다.

  ftp:*:500:100:Anonymous FTP user:/var/ftp:/usr/bin/false

- ~ftp/etc/passwd의 모든 암호가 별표(*)인지 확인합니다.

  $**more ~ftp/etc/passwd**
  root:*:0:3::/:/usr/bin/false daemon:*:1:5::/:/usr/bin/false

- 쓰기 가능한 pub 디렉토리가 있어야 하는 경우 1733 사용 권한을 사용합니다.

  $**chmod 1733 /home/ftp/pub**

- 디스크 할당량이나 cron 작업을 사용하여 /home/ftp/pub의 크기를 제어합니다.

  0 1 * * * find /home/ftp/pub/* -atime +1 exec rm -rf {} \;

- /var/adm/syslog/syslog.log에서 익명 ftp 작업을 확인합니다.

  $**tail /var/adm/syslog/syslog.log**

5.1.3 /etc/ftpd/ftpusers를 사용하여 액세스 거부

inetd 데몬은 /etc/services에 표시된 포트에서 서비스 요청을 받을 때 파일 전송 프로토콜 서버인 ftpd를 실행합니다. ftpd는 /etc/ftpd/ftpusers에 나열된 로컬 사용자 계정에 대한 원격 로그인을 거부합니다. 이러한 사용자 계정을 제한된 계정이라고 합니다. *ftpd*(1M), *privatepw*(1) 및 *services*(4)를 참조하십시오.

/etc/ftpd/ftpusers 파일에서 제한된 각 계정 이름은 한 줄로 표시되어야 합니다. 또한 ftpd는 해당 로그인 셸을 사용하지 않고 로컬 계정에 액세스하므로 /etc/passwd에 정의된 제한된 로그인 셸을 사용하여 사용자 계정을 추가합니다.

/etc/ftpd/ftpusers가 없으면 ftpd에서 보안 검사를 수행하지 않습니다. 자세한 내용은 *ftpusers*(4)를 참조하십시오.

HP-UX 11i의 ftpd 데몬은 WU-FTPD를 기반으로 합니다. WU-FTPD는 Washington University에서 개발된 ftpd 데몬의 HP 구현입니다. WU-FTPD에서는 액세스 제어가 향상되고 로깅 기능이 개선되었으며 가상 호스트가 지원되고 RFC1413(Identification Protocol)이 지원됩니다. 자세한 내용은 다음 웹 사이트에 있는 **HP-UX Remote Access Services Administrator's Guide**를 참조하십시오.

http://docs.hp.com/en/netcom.html#Internet%20Services

5.1.4 스푸핑에 대한 기타 보안 솔루션

스푸핑은 유효한 사용자나 호스트처럼 가장하여 시스템에 권한 없이 액세스하는 방법입니다. IP 주소와 호스트 이름을 스푸핑할 수 있으므로 inetd(인터넷 데몬)에 대해 /var/adm/inetd.sec 보안 파일을 사용하는 것은 안전한 보안 솔루션이 되지 않습니다. inetd에 대한 자세한 내용은 섹션 5.2를 참조하십시오.

다음 보안 기능 또는 제품은 대체 보안 솔루션입니다.

- IPFilter는 응용 프로그램 서버를 보호하는 시스템 방화벽으로 사용하기에 적합한 TCP/IP 패킷 필터입니다. 자세한 내용은 다음 위치에 있는 **HP-UX IPFilter Administrator's Guide**를 참조하십시오.

  http://docs.hp.com/en/internet.html#IPFilter

- TCP Wrappers는 추가 보안을 위해 `inetd`에서 호출하는 TCP 래퍼 데몬 `tcpd`를 제공합니다. 자세한 내용은 섹션 5.3 및 다음 위치에 있는 **HP-UX Internet Services Administrator's Guide**를 참조하십시오.

  http://docs.hp.com/en/netcom.html#Internet%20Services

- Secure Internet Services를 사용하면 `ftp`, `rcp`, `remsh`, `rlogin` 및 `telnet`에 대해 Kerberos 인증과 권한 부여를 사용할 수 있습니다. 사용자 암호 대신 암호화된 Kerberos 인증이 네트워크를 통한 전송을 기록합니다. http://docs.hp.com/en/netcom.html#Internet%20Services 에 있는 섹션 5.4, **Installing and Administering Internet Services** 및 다음 위치에 있는 **Configuration Guide for Kerberos Client Products on HP-UX**를 참조하십시오.

  http://docs.hp.com/en/internet.html#Kerberos

- IP 보안 프로토콜 집합인 IPSec는 데이터 무결성, 인증, 데이터 개인 정보 보호, 응용 프로그램 투명 보안, 암호화 등의 IP 네트워크 보안을 제공합니다.

  다음 위치에 있는 **HP-UX IPSec Administrator's Guide**를 참조하십시오.

  http://docs.hp.com/en/internet.html#IPSec

5.2 inetd 데몬

인터넷 데몬 `/usr/sbin/inetd`는 많은 인터넷 서비스의 마스터 서버입니다.

`inetd` 데몬은 일반적으로 `/sbin/init.d/inetd` 스크립트에 의해 부팅 과정에서 자동으로 시작됩니다.

`inetd` 데몬은 `/etc/inetd.conf` 구성 파일에 나열된 서비스에 대한 연결 요청을 모니터링하고 요청을 받으면 해당 서버를 시작합니다. 즉, 사용자가 `telnet`과 같은 인터넷 서비스를 사용하여 원격 시스템에 연결합니다. `inetd` 데몬은 연결을 완료하기 전에 호스트의 `telnet` 연결이 허용되는지 확인합니다. 액세스 허용 또는 거부에 대한 호스트 정보는 `/var/adm/inetd.sec` 파일에 들어 있습니다.

`inetd` 데몬은 다음과 같이 작동합니다.

1. 시스템을 부팅하는 동안 실행 수준 2로 시작됩니다(시스템 시동 스크립트에 `/sbin/init.d/inetd start` 명령이 있는 경우).
2. `/etc/inetd.conf`를 검사하여 제공할 서비스를 확인합니다. 자세한 내용은 *ftp*(1) 및 *inetd.conf*(4)를 참조하십시오.
3. `/etc/services`를 검사하여 `/etc/inetd.conf`에 나열된 서비스를 모니터링할 포트를 확인합니다. `/etc/services` 파일은 서비스 이름을 포트 번호에 매핑합니다. 자세한 내용은 *services*(4)를 참조하십시오.
4. 클라이언트로부터 인터넷 서비스 연결 요청을 받습니다. 예를 들어, 사용자가 `telnet`을 실행합니다.
5. `/var/adm/inetd.sec`를 검사하여 클라이언트의 액세스가 허용되는지 확인합니다. 자세한 내용은 *inetd.sec*(4)를 참조하십시오.

6. 로깅이 활성화되어 있으면 /var/adm/syslog/syslog.log에 요청을 기록합니다. 자세한 내용은 *syslogd*(1M)를 참조하십시오.
7. inetd에서 보안상 연결을 거부하는 경우 연결이 종료됩니다.
8. 연결 요청이 유효하면 inetd에서 유효한 연결 요청을 처리할 서버 프로세스를 시작합니다. 서버 프로세스에는 inetd 외에도 기타 보안 기능이 있을 수 있습니다.

5.2.1 inetd 보안 유지

/etc/inetd.conf 파일은 inetd 데몬이 시작할 수 있는 서비스를 나열하는 inetd 구성 파일입니다. /etc/inetd.conf에 나열된 각 서비스는 /etc/services 파일에도 표시되어야 합니다. /etc/services 파일은 서비스 이름을 포트 번호에 매핑합니다. 각 포트 번호에는 tcp 또는 udp와 같은 연관된 프로토콜 이름이 있습니다. 프로토콜의 각 항목에 일치하는 항목이 /etc/protocols 파일에 있어야 합니다.

inetd를 보다 안전하게 설정하기 위한 제안 사항은 다음과 같습니다.

- /etc/rc.config.d/netdaemons에서 inetd 로깅을 활성화합니다. 자세한 내용은 *rc.config.d*(4)를 참조하십시오.
- /etc/inetd.conf 및 /etc/services에서 변경 사항을 확인합니다. 권한이 없는 사용자가 root 액세스 권한을 얻어 /etc/services 및 /etc/inetd.conf 파일을 수정했을 수 있습니다. /etc/inetd.conf에서 사용하고 있지 않은 서비스의 이름을 찾습니다. /etc/services에서 http://www.iana.org의 IANA(Internet Assigned Numbers Authority)에 등록되어 있지 않은 포트 번호를 찾습니다. 인터넷 서비스에 대해 나열된 포트 번호가 IANA에 등록된 포트 번호와 일치하는지 확인합니다.
- /etc/inetd.conf에서 finger와 같은 불필요한 서비스를 주석 처리합니다. finger 명령은 암호를 요구하지 않고 사용자 정보를 표시합니다.
- /etc/inetd.conf에서 RPC(Remote Procedure Calls) 서비스를 주석 처리합니다.
- 서비스 거부 공격을 방지하기 위해 /etc/inetd.conf에서 inetd의 "일반적인 내부" 서비스를 주석 처리합니다. 악의적인 사용자가 inetd를 chargen(문자 생성기) 요청으로 오버로드할 수 있습니다. 자세한 내용은 *inetd*(1M) 및 *inetd.conf*(4)를 참조하십시오.

5.2.1.1 /var/adm/inetd.sec를 사용하여 액세스 거부 또는 허용

/etc/inetd.conf 파일 구성 외에도 /var/adm/inetd.sec라는 선택적 보안 파일을 구성하여 inetd에 의해 시작된 서비스에 대한 액세스를 제한할 수 있습니다. /var/adm/inetd.sec 파일은 각 서비스에 대한 액세스가 허용 또는 거부된 호스트를 표시합니다. 자세한 내용은 *inetd.conf*(4)를 참조하십시오.

예를 들면 다음과 같습니다.

```
login allow 10.3-5 192.34.56.5 ahost anetwork
login deny 192.54.24.5 cory.example.edu.testlan
```

5.3 TCP Wrappers를 사용하여 스푸핑 방지

TCP(전송 제어 프로토콜) Wrappers는 inetd에 의해 시작된 서비스의 보안을 향상시킵니다. /etc/inetd.sec를 사용하는 대신 TCP Wrappers를 사용할 수 있습니다. TCP Wrappers는 호스트 이름 및 호스트 주소 스푸핑을 방지합니다. 스푸핑은 유효한 사용자나 호스트처럼 가장하여 시스템에 권한 없이 액세스하는 방법입니다.

스푸핑을 방지하기 위해 TCP Wrappers는 ACL(액세스 제어 목록)을 사용합니다. ACL은 `/etc/hosts.allow` 및 `/etc/hosts.deny` 파일에 있는 시스템 목록입니다. TCP Wrappers는 호스트 이름-IP 주소 매핑을 확인하고 IP 원본 라우팅을 사용하여 패킷을 거부하도록 구성된 경우 IP 스푸핑을 방지합니다.

그러나 TCP Wrappers는 암호화 인증이나 데이터 암호화를 제공하지 않습니다. `inetd`와 마찬가지로 정보는 일반 텍스트로 전달됩니다.

TCP Wrappers는 HP-UX Internet Services 소프트웨어의 일부입니다. 자세한 내용은 http://docs.hp.com/en/netcom.html#Internet%20Services에 있는 **HP-UX Internet Services Administrator's Guide**와 다음 맨페이지를 참조하십시오.

tcpd(1M), *tcpdmatch*(1), *tcpdchk*(1), *tcpd.conf*(4), *hosts_access*(3), *hosts_access*(5) 및 *hosts_options*(5).

TCP Wrappers를 활성화하면 `inetd`는 요청된 서비스를 직접 실행하는 대신 TCP 래퍼 데몬 `tcpd`를 실행합니다. TCP Wrappers는 다음과 같이 작동합니다.

1. 클라이언트가 일반적인 방식(예: `telnet`)으로 `inetd`에 연결 요청을 보냅니다.
2. 서버 프로세스를 호출하는 대신 `inetd`는 TCP 래퍼 데몬(`tcpd`)을 호출합니다.
3. TCP 래퍼 데몬은 클라이언트의 연결 요청이 유효한지 확인합니다. `tcpd` 데몬은 요청을 기록하고 액세스 제어 파일(`/etc/hosts.allow` 및 `/etc/hosts.deny`)을 확인합니다.
4. 클라이언트가 유효하면 `tcpd`는 해당 서버 프로세스를 호출합니다.
5. 서버 프로세스에서 클라이언트 요청을 처리합니다. 예를 들어, `telnet` 연결이 완료됩니다.

5.3.1 TCP Wrappers의 추가 기능

로깅 동작, 사용자 이름 조회, 오류 동작 역조회 등의 구성 매개 변수를 `/etc/tcpd.conf` 구성 파일에 정의할 수도 있습니다. `tcpd` 데몬은 런타임 중에 이 구성 파일을 읽고 구성 매개 변수를 찾습니다.

트랩 설정 및 배너 메시지와 같은 다른 보안 기능에 대해 `/etc/hosts.allow` 및 `/etc/hosts.deny` 파일을 구성할 수 있습니다.

TCP Wrappers의 트랩 설정 기능을 사용하면 원격 호스트의 거부된 연결 시도 횟수에 따라 호스트에서 적절한 작업을 트리거할 수 있습니다.

배너 메시지 기능은 ACL 규칙이 액세스 제어 파일에 포함되어 있을 때 클라이언트에 메시지를 보냅니다.

5.3.2 TCP Wrappers는 RPC 서비스와 함께 작동하지 않음

TCP Wrappers는 TCP를 통해 RPC(Remote Procedure Call) 서비스와 함께 작동하지 않습니다. 이러한 서비스는 `/etc/inetd.conf` 파일에 `rpc` 또는 `tcp`로 등록되어 있습니다. 이 제한의 영향을 받는 중요한 서비스는 `on` 명령에 사용되는 `rexd`뿐입니다.

5.4 Secure Internet Services

SIS(Secure Internet Services)는 `ftp`, `rcp`, `remsh`, `rlogin` 및 `telnet`과 같은 원격 액세스 서비스에 대한 Kerberos V5 인증 및 권한 부여를 통합하는, 선택적으로 활성화되는 메커니즘입니다.

Secure Internet Services는 HP-UX Internet Services 제품의 일부이며, 이에 대해서는 http://docs.hp.com/en/netcom.html#Internet%20Services에 있는 **Using HP-UX Internet Services** 및 다음 맨페이지에서 설명합니다.

sis(5), *kinit*(1), *klist*(1), *kdestroy*(1M), *krbval*(1M), *k5dcelogin*(1M), *inetsvcs_sec*(1M) 및 *inetsvcs*(4).

SIS 명령을 실행하면 읽을 수 있는 형식으로 암호를 네트워크에서 전송할 필요가 없으므로 보안이 향상됩니다.

참고: SIS 라이브러리는 사용자에게 권한을 부여하거나 서비스를 인증하는 데 필요한 이상으로는 세션을 암호화하지 않습니다. 따라서 이러한 서비스는 데이터 또는 원격 서비스에 대해 무결성 확인이나 암호화 서비스를 제공하지 않습니다. 데이터를 암호화하려면 OpenSSL을 사용합니다. 자세한 내용은 http://docs.hp.com/en/internet.html#OpenSSL에 있는 **OpenSSL Release Notes**를 참조하십시오.

두 시스템이 모두 Kerberos V5 기반 보안 환경에서 작동될 때 Secure Internet Services를 사용하면 로컬 및 원격 호스트가 안전하고 신뢰할 수 있는 방식으로 서로를 식별할 수 있으며 원격 계정에 액세스할 수 있는 권한이 사용자에게 부여됩니다.

`ftp/ftpd`, `rlogin/rlogind` 및 `telnet/telnetd`의 경우 Kerberos V5 인증 메커니즘은 네트워크를 통해 암호 대신 암호화된 티켓을 보내 사용자를 확인하고 식별합니다. `rcp/remshd` 및 `remsh/remshd`의 경우에는 이러한 서비스의 보안 버전을 통해 원격 계정에 액세스할 수 있는 권한이 사용자에게 부여됩니다.

5.5 관리 도메인 제어

모든 네트워크 관리 프로그램은 root가 아닌 `uucp`, `nso` 또는 daemon과 같은 보호된 네트워크별 계정의 소유여야 합니다.

관리 도메인은 암호 확인 없이 사용자가 서로 액세스할 수 있도록 허용하는 네트워크 서비스를 통해 연결된 시스템 그룹입니다. 관리 도메인은 호스트 시스템에서 시스템 사용자가 이미 확인되었다고 가정합니다. 관리 도메인을 식별하고 제어하려면 다음 단계를 사용합니다.

1. 파일 시스템을 내보낼 노드를 `/etc/exports`에 표시합니다. `/etc/exports` 파일에는 파일 시스템 경로 이름과 파일 시스템에 액세스할 수 있는 시스템 또는 시스템 그룹 목록으로 구성된 항목이 들어 있습니다. `/etc/exports` 항목에는 시스템 그룹의 이름이 포함될 수 있습니다. `/etc/netgroup`을 검사하여 그룹에 포함된 개별 시스템을 확인할 수 있습니다.
2. 동일한 암호 데이터베이스가 있는 노드를 `/etc/hosts.equiv`에 표시합니다.
3. 관리 도메인에 있는 각 노드의 권한이 포함되지 않은 노드로 확장되지 않는지 확인합니다. 도메인의 각 노드에 대해 2단계와 3단계를 반복합니다.
4. 관리 도메인의 각 노드에서 root 및 로컬 보안을 제어합니다. 도메인의 한 시스템에서 수퍼유저 권한이 있는 사용자는 도메인의 모든 시스템에 대해 이러한 권한을 얻을 수 있습니다.
5. 관리 도메인의 암호 파일 간에 사용자 이름, `uid` 및 `gid`의 일관성을 유지합니다.
6. 관리 도메인의 모든 노드에서 그룹 파일 간의 일관성을 유지합니다. 예를 들어, `hq` 및 `mfg` 시스템을 사용하여 일관성을 확인하려면 `mfg` 시스템의 root 파일 시스템이 원격으로 `hq`에 `/nfs/mfg/`로 마운트된 경우 다음 `diff` 명령을 입력합니다.

   $**diff /etc/group /nfs/mfg/etc/group**

   차이가 표시되면 두 `/etc/group` 파일이 일치하지 않는 것이며 이는 허용되지 않습니다.

5.5.1 네트워크 제어 파일의 사용 권한 설정 확인

`/etc` 디렉토리에 있는 네트워크 제어 파일은 네트워크 자체에 대한 액세스를 제공하므로 보안 대상이 됩니다. 네트워크 제어 파일은 일반 사용자가 쓸 수 없도록 해야 합니다.

모든 시스템 파일의 모드, 소유자 및 그룹을 신중하게 설정합니다. 이러한 파일을 정기적으로 검사하여 변경 사항을 확인하고 수정합니다.

자주 사용하는 네트워크 제어 파일은 다음과 같습니다.

- `/etc/exports`

  NFS 클라이언트로 내보낼 수 있는 파일 디렉토리 목록. 자세한 내용은 *exports*(4)를 참조하십시오.

- `/etc/hosts`

  네트워크 호스트 및 해당 IP 주소 목록. 자세한 내용은 *hosts*(4)를 참조하십시오.

- `/etc/hosts.equiv`

  액세스가 허용되고 로컬 호스트와 동등한 원격 호스트 목록. 자세한 내용은 *hosts.equiv*(4)를 참조하십시오.

- `/etc/inetd.conf`

  인터넷 서비스 구성 파일. 자세한 내용은 *inetd.conf*(4)를 참조하십시오.

- `/etc/netgroup`

  네트워크 범위의 그룹 목록. 자세한 내용은 *netgroup*(4)을 참조하십시오.

- `/etc/networks`

  네트워크 이름 및 해당 네트워크 번호 목록. 자세한 내용은 *networks*(4)를 참조하십시오.

- `/etc/protocols`

  프로토콜 이름 및 번호 목록. 자세한 내용은 *protocols*(4)를 참조하십시오.

- `/etc/services`

  공식적인 서비스 이름 및 별칭과 서비스에서 사용하는 포트 번호 및 프로토콜 목록. 자세한 내용은 *services*(4)를 참조하십시오.

5.6 HP-UX SSH(Secure Shell)를 사용하여 원격 세션 보안 유지

HP-UX Secure Shell은 개방형 소스 SSH 제품인 OpenSSH 제품을 기반으로 합니다 (http://www.openssh.org). HP-UX Secure Shell은 비보안 네트워크에서 클라이언트와 원격 호스트 간의 보안 연결을 가능하게 합니다. 이 보안 연결의 주요 속성은 다음과 같습니다.

- 클라이언트와 원격 호스트 둘 다에 대한 강력한 인증
- 클라이언트와 원격 호스트 간의 통신에 대한 강력한 암호화 및 공개 키 암호화
- 클라이언트가 원격 호스트에서 명령을 실행하는 데 사용할 보안 연결

HP-UX Secure Shell은 `telnet`, `remsh`, `rlogin`, `ftp` 및 `rcp`와 같은 자주 사용하는 함수와 명령의 보안 대체 항목을 제공합니다.

HP-UX Secure Shell 설명서는 `ssh` 클라이언트에 대한 *ssh*(1) 맨페이지와 `sshd` 서버에 대한 *sshd*(8) 맨페이지를 참조하십시오. 두 맨페이지에는 제품과 함께 제공되는 다른 HP-UX Secure Shell 맨페이지에 대한 참조가 들어 있습니다.

또한 http://docs.hp.com/en/internet.html#Secure%20Shell에 있는 **HP-UX Secure Shell Release Notes**를 참조하십시오.

5.6.1 HP-UX Secure Shell의 주요 보안 기능

HP-UX Secure Shell의 주요 보안 기능은 다음과 같습니다.

- 강력한 암호화

  클라이언트와 원격 호스트 간의 모든 통신은 Blowfish, 3DES, AES 및 arcfour와 같은 특허 없는 암호화 알고리즘을 사용하여 암호화됩니다. 암호 등의 인증 정보는 네트워크에서 일반 텍스트로 전송되지 않습니다. 또한 암호화는 강력한 공개 키 기반 암호화와 더불어 잠재적 보안 공격을 차단합니다.

- 강력한 인증

  HP-UX Secure Shell은 클라이언트와 서버 간의 강력한 인증 방법 집합을 지원합니다. 인증은 양방향일 수 있습니다. 서버는 클라이언트를 인증하고 클라이언트도 서버를 인증합니다. 이렇게 하면 다양한 보안 문제로부터 세션을 보호할 수 있습니다. 지원되는 인증 방법에 대해서는 섹션 5.6.5에서 설명합니다.

- 포트 전달

  클라이언트와 원격 호스트 간의 CP/IP 연결 리디렉션(및 그 반대)을 **포트 전달** 또는 **SSH 터널링**이라고 합니다. HP-UX Secure Shell은 포트 전달을 지원합니다. 예를 들어, 포트 전달을 사용하여 클라이언트와 서버 간의 `ftp` 트래픽(또는 전자 메일 클라이언트와 POP/IMAP 서버 간의 전자 메일 트래픽)을 리디렉션할 수 있습니다. 클라이언트가 직접 서버와 통신하는 대신 보안 채널을 통해 트래픽을 `sshd` 서버로 리디렉션할 수 있으며, 그런 다음 `sshd` 서버에서 트래픽을 실제 서버 시스템의 지정된 포트로 전달할 수 있습니다.

- 기본 HP-UX 보안 기능과의 통합

  HP-UX Secure Shell 제품은 중요한 HP-UX 보안 기능과 통합되어 있습니다. 자세한 내용은 섹션 5.6.7을 참조하십시오.

5.6.2 HP-UX Secure Shell의 소프트웨어 구성 요소

HP-UX Secure Shell 소프트웨어는 클라이언트 및 서버 구성 요소 집합으로 이루어져 있습니다. 자세한 내용은 표 5-2를 참조하십시오.

표 5-2 HP-UX Secure Shell의 소프트웨어 구성 요소

구성 요소	설명	위치	대응하는 비보안 구성 요소
`ssh`	Secure Shell 클라이언트는 `telnet` 및 `remsh`의 보안 대체 항목이며 보안 기능이 있는 `remsh`와 가장 유사합니다.	클라이언트	`remsh`, `telnet`, `rlogin`
`slogin`	`ssh`에 대한 심볼릭 링크입니다.	클라이언트	`remsh`, `telnet`, `rlogin`

표 5-2 HP-UX Secure Shell의 소프트웨어 구성 요소 (계속)

구성 요소	설명	위치	대응하는 비보안 구성 요소
scp	클라이언트 보안 복사 및 서버 보안 복사를 수행합니다.	클라이언트 및 서버	rcp
sftp	보안 ftp 클라이언트입니다.	클라이언트	ftp
sshd	보안 셸 데몬입니다.	서버	remshd, telnetd, rlogind
sftp-server	보안 ftp 데몬입니다.	서버	ftpd
ssh-rand-helper	sshd가 서버에서 /dev/random 또는 /dev/urandom을 찾을 수 없을 때 사용되는 Random Number Generator. HP-UX에는 커널에 상주하는 Random Number Generator인 rng가 포함되어 있습니다. rng가 구성 해제되어 있으면 sshd는 prngd를 사용합니다.	서버	해당 사항 없음
ssh-agent	클라이언트에서 서버로의 "자동" 키 기반 로그인 도구입니다.	클라이언트 및 서버	rhosts 파일 메커니즘
ssh-add	클라이언트의 키 쌍을 ssh-agent에 알리는 도구입니다.	클라이언트	해당 사항 없음
ssh-keygen	공개 키 인증을 위해 키 쌍을 생성하는 도구입니다.	클라이언트	해당 사항 없음
ssh-keyscan	Secure Shell 데몬(sshd)을 실행하는 호스트 집합에 대한 공개 키를 수집하는 클라이언트 도구입니다.	클라이언트	해당 사항 없음
ssh-keysign	호스트 기반 인증 중에 필요한 디지털 서명을 생성하는 도구입니다. ssh()에서 로컬 호스트 키 호스트 기반 인증에 액세스하는 데 사용됩니다.	클라이언트	해당 사항 없음

5.6.3 HP-UX Secure Shell 실행

표 5-2에 나열된 Secure Shell 클라이언트를 실행하기 전에 먼저 Secure Shell 서버 데몬 sshd를 시작합니다. sshd 데몬은 서버 시스템의 /opt/ssh/etc 디렉토리에 있는 sshd_config 파일에서 초기 구성 값을 가져옵니다. sshd_config에 있는 가장 중요한 구성 지시어 중 하나는 sshd 데몬에서 지원하는 인증 방법 집합입니다. 자세한 내용은 섹션 5.6.5를 참조하십시오.

5.6.3.1 ssh 클라이언트 실행

ssh 클라이언트 응용 프로그램은 sshd 서버와의 소켓 연결을 설정합니다. sshd 서버는 자식 sshd 프로세스를 시작합니다. 이 자식은 연결 소켓을 상속 받고 선택된 인증 방법을 기반으로 클라이언트를 인증합니다. 인증에 성공한 경우에만 성공적으로 보안 클라이언트 세션이 설정됩니다.

세션이 만들어진 후 모든 후속 통신은 클라이언트와 이 자식 sshd 프로세스 간에 직접 이루어집니다. 이제 클라이언트는 서버에서 원격 명령을 실행할 수 있습니다. ssh 클라이언트의 명령 요청이 있을 때마다 자식 sshd 프로세스에서 해당 명령을 실행할 셸 프로세스를 시작합니다.

간단히 말해서 실행 중인 ssh 클라이언트-서버 세션은 다음 프로세스로 구성됩니다.

- sshd 서버에 연결된 모든 클라이언트 시스템에는 현재 이 클라이언트 시스템에서 설정된 각 ssh 연결에 대한 하나의 ssh 클라이언트 프로세스가 있습니다.
- 서버 시스템에는 하나의 부모 sshd 프로세스와 서버에 연결된 동시 ssh 클라이언트 개수만큼의 자식 sshd 프로세스가 있습니다. 서버에 권한 분리가 활성화되어 있으면 서버에서 실행되는 자식 sshd 프로세스의 수가 두 배로 증가합니다. 자세한 내용은 섹션 5.6.4를 참조하십시오.
- ssh 클라이언트에서 명령 실행 요청이 있을 때마다 서버 시스템의 해당 자식 sshd 프로세스는 셸 프로세스를 시작하고 UNIX 파이프를 사용하여 명령 요청을 이 셸 프로세스로 전달합니다. 이 셸 프로세스는 UNIX 파이프를 사용하여 명령 실행 결과를 자식 sshd 프로세스로 반환하고 명령 실행이 완료되면 종료됩니다.

5.6.3.2 sftp 클라이언트 실행

sftp 클라이언트 응용 프로그램은 sftp 클라이언트 응용 프로그램이 ssh 클라이언트를 시작하도록 하고 UNIX 파이프를 사용하여 이 클라이언트와 통신합니다. 그런 다음 ssh 클라이언트는 sshd 서버와의 소켓 연결을 설정합니다.

서버 상호 작용의 나머지 부분은 섹션 5.6.3.1에서 설명한 ssh 클라이언트의 경우와 유사합니다. 차이점은 원격 명령을 실행할 셸을 시작하는 대신 자식 sshd 프로세스가 sftp-server 프로세스를 시작한다는 것입니다. 이 sftp 세션 중의 모든 후속 통신은 다음 프로세스 간에 발생합니다.

- UNIX 파이프를 사용하여 클라이언트 시스템에서 sftp 클라이언트와 ssh 클라이언트 간에
- 설정된 연결 소켓을 통해 ssh 클라이언트와 자식 sshd 프로세스 간에
- UNIX 파이프를 사용하여 자식 sshd 프로세스와 sftp 서버 프로세스 간에

5.6.3.3 scp 클라이언트 실행

scp 클라이언트의 경우는 sftp 클라이언트 실행과 거의 동일합니다. 차이점은 sftp-server 프로세스를 시작하는 대신 자식 sshd 프로세스가 scp 프로세스를 시작한다는 것입니다. scp 세션 중의 모든 후속 통신은 다음 프로세스 간에 발생합니다.

- 클라이언트 시스템에서 scp 클라이언트와 ssh 클라이언트 간에, UNIX 파이프 사용
- 설정된 연결 소켓을 통해 ssh 클라이언트와 자식 sshd 프로세스 간에
- UNIX 파이프를 사용하여 자식 sshd 프로세스와 scp 서버 프로세스 간에

5.6.4 HP-UX Secure Shell 권한 분리

HP-UX Secure Shell은 **권한 분리** 기능을 통해 보다 향상된 수준의 보안을 제공합니다. 섹션 5.6.3에서 설명한 것처럼 부모 sshd 및 자식 sshd 프로세스는 권한이 부여된 사용자로 실행됩니다. 권한 분리를 활성화하면 사용자 연결당 하나의 추가 프로세스가 시작됩니다.

ssh 클라이언트가 권한 분리에 대해 구성된 sshd 서버에 연결하면 부모 sshd 프로세스가 권한이 부여된 자식 sshd 프로세스를 시작합니다. 권한 분리를 활성화하면 자식 sshd 프로세스는 권한이 없는 자식 sshd 프로세스를 추가로 시작합니다. 권한이 없는 자식 sshd 프로세스는

연결 소켓을 상속 받습니다. 클라이언트와 서버 간의 모든 후속 통신은 권한이 없는 이 자식 sshd 프로세스를 사용하여 이루어집니다.

클라이언트의 원격 명령 실행 요청은 대부분 비권한이며 권한이 없는 이 자식 sshd 프로세스에서 시작된 셸에 의해 처리됩니다. 권한이 없는 자식 sshd 프로세스에서 권한이 부여된 기능을 실행해야 하는 경우 UNIX 파이프를 사용하여 권한이 부여된 부모 sshd 프로세스와 통신합니다.

권한 분리는 침입자에 의한 잠재적 손상을 제한하는 데 도움이 됩니다. 예를 들어, 셸 명령을 실행하는 동안 버퍼 오버플로 공격이 발생할 경우 권한이 없는 프로세스 내에서 제어되므로 잠재적 보안 위험이 제한됩니다.

참고: 권한 분리는 HP-UX Secure Shell의 기본 구성입니다. sshd_config 파일에서 UsePrivilegeSeparation NO를 설정하여 권한 분리를 해제할 수 있습니다. 잠재적 보안 위험이 있으므로 권한 분리를 해제할 경우 신중하게 고려하십시오.

5.6.5 HP-UX Secure Shell 인증

HP-UX Secure Shell은 다음 인증 방법을 지원합니다.

- GSS-API(Kerberos 기반 클라이언트 인증)
- 공개 키 인증
- 호스트 기반 인증
- 암호 인증

클라이언트는 원격 sshd 데몬과 연결될 때 원하는 인증 방법(앞에 나열된 방법 중 하나)을 선택하고 연결 요청의 일부로 적절한 자격 증명을 제공하거나 서버에서 보낸 프롬프트에 응답합니다. 모든 인증 방법이 이런 방식으로 작동합니다.

서버가 성공적으로 연결을 설정하려면 클라이언트로부터 적절한 키, 암호구, 암호 또는 자격 증명을 받아야 합니다.

sshd 인스턴스에서 보안 요구 사항을 기반으로 지원되는 인증 방법 중 일부만 지원하도록 선택할 수 있습니다.

HP-UX Secure Shell은 앞에 나열된 인증 방법을 지원하지만 시스템 관리자가 환경의 특정 보안 요구 사항을 기반으로 sshd 인스턴스에서 제공되는 인증 방법을 제한할 수 있습니다. 예를 들어, HP-UX Secure Shell 환경에서 모든 클라이언트가 공개 키 또는 Kerberos 방법을 사용하여 인증해야 한다고 지정할 수 있으며, 이로 인해 나머지 방법은 비활성화될 수 있습니다. 지원되는 인증 방법 활성화 및 비활성화는 sshd_config 파일에 지정된 구성 지시어를 통해 이루어집니다.

ssh 클라이언트 연결 요청이 있으면 서버는 먼저 지원되는 인증 방법 목록으로 응답합니다. 이 목록은 sshd 서버에서 지원하는 인증 방법과 이러한 방법이 시도되는 시퀀스를 나타냅니다. 클라이언트는 이러한 인증 방법을 하나 이상 생략할 수 있습니다. 클라이언트에서 방법이 시도되는 시퀀스를 변경할 수도 있습니다. 이렇게 하려면 클라이언트 구성 파일 /opt/ssh/etc/ssh_config의 구성 지시어를 사용합니다.

HP-UX Secure Shell에서 지원하는 인증 방법은 다음 절에 요약되어 있습니다.

5.6.5.1 GSS-API

Kerberos 기반 클라이언트 인증인 GSS-API(Generic Security Service application Programming Interface)를 사용하는 경우 클라이언트가 미리 Kerberos 자격 증명을 받아야 하며, 해당 클라이언트 디렉토리에 Kerberos 구성 파일이 있어야 합니다. 클라이언트는 `sshd` 데몬과 연결될 때 연결 시 자격 증명을 제공합니다. 서버는 이러한 자격 증명을 이 특정 사용자의 자격 증명 복사본과 일치시킵니다. 또한 선택적으로 클라이언트 호스트 환경의 타당성을 설정할 수 있습니다.

자세한 내용은 *gssapi*(5), *kerberos*(9) 및 http://docs.hp.com/en/internet.html#Kerberos에 있는 Kerberos 관리자 설명서를 참조하십시오.

5.6.5.2 공개 키 인증

공개 키 인증을 사용하려면 Secure Shell 환경에 다음 설정이 있어야 합니다.

- 클라이언트와 서버 둘 다에 키 쌍이 있어야 합니다. 모든 `ssh` 클라이언트와 모든 `sshd` 서버가 `ssh-keygen` 유틸리티를 사용하여 자체적으로 키 쌍을 생성해야 합니다.
- 클라이언트는 통신해야 하는 모든 `sshd` 서버에 해당 공개 키를 알려야 합니다. 이렇게 하려면 각 클라이언트의 공개 키를 모든 관련 서버의 미리 지정된 디렉토리에 복사합니다.
- 클라이언트는 통신해야 하는 모든 서버의 공개 키를 구해야 합니다. 클라이언트는 `ssh-keyscan` 유틸리티를 사용하여 공개 키를 받습니다.

이 설정이 완료되면 `sshd` 서버에 연결하는 `ssh` 클라이언트가 공개 키와 개인 키를 사용하여 인증됩니다. 공개 키 암호화에 대한 자세한 내용은 **공개 키 암호화**를 참조하십시오.

HP-UX Secure Shell은 공개 키 인증을 단순화하는 추가 기능을 제공합니다. 일부 환경에서는 항상 암호 프롬프트에 응답할 필요가 없도록 설정할 수 있습니다. 둘 다 클라이언트 시스템에서 실행되는 `ssh-agent` 및 `ssh-add` 프로세스를 조합해서 사용하면 암호에 응답하지 않아도 됩니다. 클라이언트는 `ssh-add` 유틸리티를 통해 `ssh-agent` 프로세스에 모든 키 정보를 등록합니다. 그런 다음 클라이언트와 서버 간의 공개 키 인증은 `sshd` 데몬이 클라이언트와 상호 작용할 필요 없이 `ssh-agent`에 의해 수행됩니다.

5.6.5.3 호스트 기반 및 공개 키 인증

호스트 기반 및 공개 키 인증은 보다 안전한 공개 키 인증 방법의 확장입니다. 클라이언트와 서버 둘 다의 키 쌍이 있는 것 외에도 이 방법을 사용하면 클라이언트 환경에서 통신할 서버를 제한할 수 있습니다. 클라이언트의 홈 디렉토리에 `.rhosts` 파일을 만들어 이 제한을 구현합니다.

5.6.5.4 암호 인증

암호 인증 방법은 단일 사용자 ID 및 암호 기반 로그인을 사용합니다. 이 로그인은 `/etc/passwd`에 지정된 사용자 로그인을 기반으로 하거나 PAM 기반일 수 있습니다.

HP-UX Secure Shell은 서버 시스템에서 사용할 수 있는 PAM 모듈과 완전히 통합됩니다. 이 목적을 위해 `/opt/ssh/etc/sshd_config` 파일에는 UsePAM 구성 지시어가 있습니다. `YES`로 설정하면 클라이언트에서 암호 인증 요청이 있을 때마다 `sshd`는 PAM 구성 파일(`/etc/pam.conf`)을 확인합니다. 그런 다음 구성된 PAM 모듈을 통해 암호 인증이 성공할 때까지 순서대로 수행됩니다. PAM 인증에 대한 자세한 내용은 *pam.conf*(4)를 참조하십시오.

PAM 인증을 무시하려면 UsePAM 지시어를 `NO`로 설정합니다. 그러면 클라이언트에서 암호 인증 요청이 있을 때마다 `sshd`가 서버의 PAM 구성 설정을 무시합니다. 대신 `sshd`는 **`getpwnam()`** 라이브러리 호출을 직접 호출하여 사용자 암호 정보를 가져옵니다.

HP-UX Secure Shell은 PAM_UNIX, PAM_LDAP 및 PAM_KERBEROS를 사용하여 테스트되었습니다. 또한 PAM_DCE 및 PAM_NTLM과 같은 다른 PAM 모듈에서 작동합니다.

5.6.6 통신 프로토콜

HP-UX Secure Shell 사용자는 SSH-1 또는 SSH-2 프로토콜을 사용하여 원격 `sshd` 데몬과 연결할 수 있습니다. SSH-2가 더 안전하므로 SSH-1 대신 권장됩니다.

5.6.7 HP-UX Secure Shell 및 HP-UX 시스템

HP-UX Secure Shell은 실제로 셸이 아니라 호스트에서 안전하게 원격 셸 세션을 실행하기 위해 클라이언트와 원격 호스트 간에 보안 연결을 만드는 메커니즘입니다. 보안 연결을 설정하기 위해 HP-UX Secure Shell에서 인증과 세션 생성을 대부분 수행합니다. HP-UX Secure Shell에서 사용하는 기능의 일부 목록은 다음과 같습니다.

- `login` 시도 기록

  `telnet` 또는 `remsh`와 마찬가지로 HP-UX Secure Shell은 성공한 세션과 실패한 세션을 각각 `/var/adm/wtmp` 및 `/var/adm/btmp` 파일에 기록합니다. 자세한 내용은 *utmp*(4)를 참조하십시오.

- PAM 모듈

  섹션 5.6.5에서 설명한 것처럼 HP-UX Secure Shell은 클라이언트 세션에 대해 PAM 인증을 사용할 수 있습니다. PAM 인증을 선택하면 HP-UX Secure Shell은 `/etc/pam.conf` 파일을 사용하고 인증을 위해 해당 PAM 모듈을 호출합니다. `/etc/pam.conf` 파일에 대한 자세한 내용은 *pam.conf*(4)를 참조하십시오.

- `/etc/default/security` 파일 사용

  로그인 동작, 암호 및 기타 보안 구성을 정의하는 속성이 들어 있는 시스템 범위 구성 파일입니다. 몇 가지 제한은 있지만 HP-UX Secure Shell에서 이러한 속성을 사용할 수 있습니다. 이에 대해서는 HP-UX Secure Shell의 `/opt/ssh/README.hp` 파일에서 설명합니다.

  `/etc/default/security` 파일에 대한 자세한 내용은 *security*(4)에 있습니다.

- 섀도 패스워드

  HP-UX Secure Shell은 HP-UX 섀도 패스워드 기능과 통합됩니다. 자세한 내용은 *shadow*(4)를 참조하십시오.

- 컨트롤 시스템 로그(`syslog`)

  HP-UX Secure Shell은 `syslog`를 사용하여 중요한 메시지를 씁니다. 자세한 내용은 *syslog*(3C) 및 *syslogd*(1M)를 참조하십시오.

- 감사 로깅

  HP-UX Secure Shell은 해당 코드로 감사 로깅(트러스트된 모드)을 구현했습니다. 자세한 내용은 *audit*(5)를 참조하십시오.

5.6.8 연관된 기술

HP-UX Secure Shell은 다음 기술을 사용하여 테스트되었습니다.

- Kerberos 5 및 GSS-API

- OpenSSL
- IPv6
- TCP Wrappers
- PAM(PAM_UNIX, PAM_Kerberos, PAM_LDAP)
- HP-UX Strong Random Number Generator

5.6.9 Strong Random Number Generator 요구 사항

모든 암호화 키 기반 제품과 마찬가지로 HP-UX Secure Shell에는 Random Number Generator가 필요합니다. HP-UX Strong Random Number Generator 장치 특수 파일 `/dev/urandom`과 `/dev/random`을 찾고 처음 발견한 장치 특수 파일을 사용합니다. 이러한 두 파일이 시스템에 없으면 HP-UX Secure Shell은 내부 Random Number Generator인 `ssh-rand-helper`를 사용합니다.

HP-UX Strong Random Number Generator는 HP-UX Secure Shell의 성능과 엔트로피(임의성 및 생성된 키 보안 측정값)를 향상시키며 재현할 수 없는 난수를 생성합니다. HP-UX Secure Shell에서는 HP-UX Strong Random Number Generator를 사용하는 것이 좋습니다.

HP-UX Strong Random Number Generator는 기본적으로 사용할 수 있습니다. 자세한 내용은 *random*(7)을 참조하십시오.

5.6.10 TCP Wrappers 지원

HP-UX Secure Shell 데몬 `sshd`는 아카이브 라이브러리 `libwrap.a`와 연결되어 TCP Wrappers를 지원합니다. 섹션 5.3을 참조하십시오.

5.6.11 chroot 디렉토리 jail

`chroot`는 디렉토리 jail입니다. 지정된 디렉토리에 있는 응용 프로그램을 시작하고 사용자가 해당 디렉토리와 하위 디렉토리에만 액세스할 수 있도록 제한합니다. 사용자가 지정된 디렉토리 위의 디렉토리를 변경하는 것을 방지합니다. 응용 프로그램을 사용하는 동안 파일 및 디렉토리 액세스를 해당 응용 프로그램 사용자로 제한하는 데 사용됩니다.

응용 프로그램에 대해 `chroot`를 활성화해야 합니다. 새 디렉토리를 만들고 관련된 파일 집합을 새로 만든 디렉토리로 복사해야 합니다.

선택적으로 `chroot` 디렉토리를 사용하여 `ssh`, `scp` 및 `sftp`를 설정할 수 있습니다.

`/opt/ssh/README.hp`에 있는 HP-UX Secure Shell `README` 파일에서는 `chroot` 기능, `chroot` 설정 스크립트 및 이 스크립트가 `chroot` 환경에 대해 `ssh`, `sftp` 및 `scp`를 활성화하기 위해 복사하는 특정 파일에 대해 설명합니다. *chroot*(1M)를 참조하십시오.

`chroot` 설정 스크립트는 HP-UX Secure Shell 소프트웨어 제품(Secure Shell 4.30.004/005)의 일부인 `/opt/ssh/utils/ssh_chroot_setup.sh` 파일에 있습니다.

II부 데이터 보호

HP-UX 11i에서는 전송 중, 사용 중 및 휴지 상태의 형태로 데이터를 보호합니다. 이러한 세 가지 형식으로 데이터를 보호하도록 설계된 보안 기능을 사용하여 HP-UX 11i 고객은 데이터 손실에서 뿐만 아니라 고객 신뢰 부문에서 발생할 수 있는 문제를 최소화할 수 있습니다. 이 절의 내용은 다음과 같습니다.

- 파일 시스템 보안(6장)
- 구획(7장)
- Fine-grained 권한(8장)

6 파일 시스템 보안

이 장에서는 파일 시스템 보안에 대해 설명합니다. 이 장의 내용을 읽기 전에 파일과 파일 시스템에 대한 기초 지식이 있어야 합니다.

데이터가 파일에 저장되므로 파일 보호 방법을 이해하는 것은 매우 중요합니다. 이 장의 내용은 다음과 같습니다.

- 파일 액세스 제어(섹션 6.1)
- 액세스 제어 목록 설정(섹션 6.2)
- HFS ACL 사용(섹션 6.3)
- JFS ACL 사용(섹션 6.4)
- JFS ACL과 HFS ACL 비교(섹션 6.5)
- ACL 및 NFS(섹션 6.6)
- /dev 장치 특수 파일의 보안 고려 사항(섹션 6.7)
- 디스크 파티션 및 논리 볼륨 보호(섹션 6.8)
- 파일 시스템 마운트 및 마운트 해제에 대한 보안 지침(섹션 6.9)
- 네트워크의 파일 보안 제어(섹션 6.10)

6.1 파일 액세스 제어

작업 그룹, 파일 사용 권한, 파일 소유권 및 구획 규칙에 따라 지정된 파일에 액세스할 수 있는 사용자가 결정됩니다. 가장 간단한 파일 액세스 규칙은 표준 UNIX 파일 사용 권한입니다.

그룹이 소유한 파일을 해당 그룹 내에서 공유하고 외부 사용자로부터 보호할 수 있도록 사용자를 그룹으로 나눌 수 있습니다.

일반적인 UNIX 파일 사용 권한은 `-l` 플래그가 있는 `ls` 명령을 사용하여 표시됩니다. 사용 권한은 시스템의 소유자와 그룹에 부여되는 액세스 종류(읽기, 쓰기 및 실행 권한)를 나타냅니다. 일반적인 UNIX 파일 보호 기능을 사용하면 파일과 디렉토리에 액세스할 수 있는 사용자를 어느 정도 제어할 수 있지만 소유 사용자와 소유 그룹의 범위를 넘어서 개별 사용자와 그룹의 액세스 권한을 정의할 수는 없습니다. UNIX 파일 사용 권한에 대한 간략한 설명은 다음과 같습니다.

각 파일과 각 디렉토리에는 9개 사용 권한이 연관되어 있습니다. 파일과 디렉토리에 있는 세 가지 유형의 사용 권한은 다음과 같습니다.

- `r`(읽기)
- `w`(쓰기)
- `x`(실행)

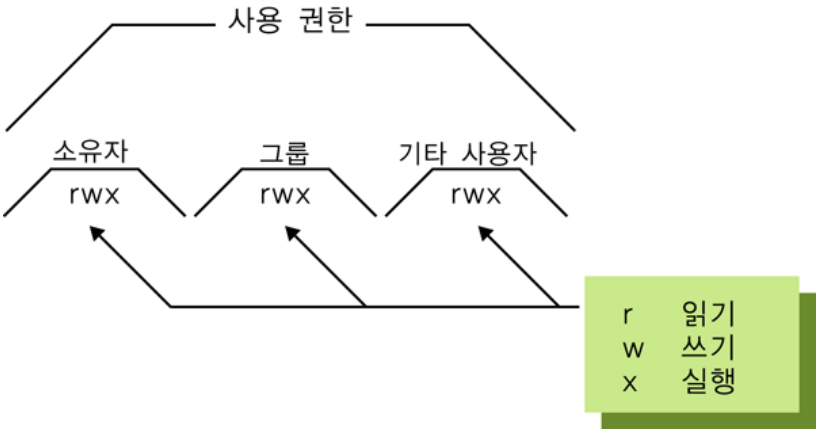
이러한 세 가지 사용 권한이 다음 세 클래스의 사용자에 대해 각각 발생합니다.

- `u`(사용자/소유자)
- `g`(그룹)
- `o`(기타, 외부 사용자라고도 함)

r 권한을 선택하면 사용자가 파일을 보거나 인쇄할 수 있습니다. w 권한을 선택하면 사용자가 파일을 쓰거나 수정할 수 있습니다. x 권한을 선택하면 사용자가 파일을 실행하거나 디렉토리를 검색할 수 있습니다.

그림 6-1에서는 일반적인 사용 권한 필드를 보여 줍니다.

그림 6-1 파일 및 디렉토리 사용 권한 필드



파일이나 디렉토리의 사용자/소유자는 일반적으로 해당 파일이나 디렉토리를 만든 사람입니다. 파일 소유자는 chmod 명령을 사용하여 파일 사용 권한을 변경할 수 있습니다.

그룹은 파일이 속하는 그룹을 지정합니다. 파일 소유자는 chgrp 명령을 사용하여 파일의 그룹 ID를 변경할 수 있습니다.

세 가지 사용 권한 유형의 의미는 일반 파일과 디렉토리에서 약간 다릅니다. 자세한 내용은 표 6-1을 참조하십시오.

표 6-1 파일 및 디렉토리 권한의 차이점

사용 권한	파일	디렉토리
r(읽기)	내용을 보거나 인쇄할 수 있습니다.	내용을 읽을 수 있지만 검색할 수는 없습니다. 일반적으로 r과 x가 함께 사용됩니다.
w(쓰기)	내용을 변경하거나 삭제할 수 있습니다.	항목을 추가하거나 제거할 수 있습니다.
x(실행)	파일을 프로그램으로 사용할 수 있습니다.	디렉토리를 검색할 수 있습니다.

6.1.1 파일 액세스 권한 설정

chmod 명령은 파일의 소유자, 그룹 구성원 또는 다른 모든 사용자의 액세스 유형(읽기, 쓰기 및 실행 권한)을 변경합니다. 파일 소유자나 해당 권한을 가진 사용자만 파일 액세스를 변경할 수 있습니다. *chmod*(1)를 참조하십시오.

기본적으로 파일과 디렉토리의 초기 읽기 및 쓰기 권한 집합은 만든 사람의 umask 값에 의해 결정됩니다. 기본 파일 사용 권한을 변경하려면 umask 명령을 사용합니다. *umask*(1)를 참조하십시오.

파일 모드 생성 마스크에서 비트가 설정되면 파일 모드의 해당 사용 권한 비트가 지워집니다(비활성화됨). 반대로 마스크에서 비트가 지워지면 새로 만든 파일의 해당 파일 모드 비트가 활성화될 수 있습니다.

예를 들어, `umask`가 8진수 022이면 그룹 및 다른 쓰기 권한을 비활성화하는 `u=rwx,g=rx,o=rx` 마스크가 만들어집니다.

6.1.2 파일 소유권 설정

`chown` 명령은 파일 소유권을 변경합니다. 소유자를 변경하려면 해당 파일을 소유하고 있거나 적절한 권한이 있어야 합니다.

`chgrp` 명령은 파일 그룹 소유권을 변경합니다. 그룹을 변경하려면 해당 파일을 소유하고 있거나 적절한 권한이 있어야 합니다.

자세한 내용은 *chown*(1) 및 *chgrp*(1)를 참조하십시오.

6.1.3 디렉토리 보호

일반적으로 표준 사용 권한이나 ACL을 통해 디렉토리에 쓸 수 있는 경우 파일 자체의 사용 권한에 관계없이 누구든지 해당 디렉토리에 있는 파일을 제거할 수 있습니다. 디렉토리의 파일이 삭제되지 않도록 보호하려면 다음을 수행합니다.

- 삭제되지 않도록 보호할 디렉토리에 대한 쓰기 권한을 제거합니다.

  이 방법은 사용자의 개인 디렉토리인 경우에 특히 유용합니다. 다음 명령을 실행하면 다른 사용자가 `mydir` 디렉토리를 읽고 검색할 수 있지만 소유자만 디렉토리에서 파일을 삭제할 수 있습니다.

  ```
  # chmod 755 mydir
  ```

  chmod(1) 및 *chmod*(2)를 참조하십시오.

- 디렉토리에 대해 **고정 비트**를 설정합니다.
- 고정 비트는 모든 파일 모드의 특수 비트입니다. 고정 비트를 설정하면 사용자가 해당 디렉토리에서 다른 사용자의 파일을 제거할 수 없습니다. 디렉토리에 대해 고정 비트를 설정하면 파일 소유자, 디렉토리 소유자 또는 해당 권한을 가진 사용자만 파일을 삭제하거나 파일 이름을 바꿀 수 있습니다.

  이 방법은 권한이 부여된 많은 사용자가 액세스하는 임시 또는 프로젝트 디렉토리(예: `/tmp` 및 `/var/tmp`)에 효율적입니다. 다음 명령을 사용하면 모든 사용자가 `/mfgproj`에서 파일을 만들고 읽고 쓸 수 있지만 파일 소유자, 디렉토리 소유자 또는 해당 권한을 가진 사용자만 파일을 삭제할 수 있습니다.

  ```
  # chmod a+rwxt /mfgproj
  ```

  고정 비트 설정은 임시 파일에 사용되는 디렉토리에 중요합니다. 임시디렉토리가 고정으로 설정되어 있지 않으면 공격자가 임시 파일이 만들어질 때까지 기다린 다음 파일을 삭제하고 수정된 내용을 사용하여 같은 이름으로 새 파일을 다시 만들어 사용자 프로그램의 예상 동작을 변경할 수 있습니다. 대부분의 경우 응용 프로그램은 이러한 변경을 인식하지 못하므로 공격자 대신 악의적인 동작을 수행할 수 있습니다.

6.1.4 사용자 계정과 관련된 파일 보호

사용자 계정과 관련된 파일을 보호하려면 다음 지침을 수행하십시오.

- 홈 디렉토리는 소유자를 제외하고 아무도 쓸 수 없어야 합니다. 그렇지 않으면 모든 사용자가 디렉토리에 파일을 추가하고 제거할 수 있습니다.
- `.profile`, `.kshrc`, `.login` 및 `.cshrc` 파일은 계정 소유자를 제외한 다른 사용자가 쓸 수 없도록 해야 합니다.
- 사용자의 `.rhosts` 파일은 소유자 이외의 다른 사용자가 읽거나 쓸 수 없도록 해야 합니다. 이렇게 하면 사용자의 다른 계정을 다른 사용자가 추측할 수 없으며 `.rhosts` 파일을 다른 사용자가 편집하여 해당 시스템에 액세스하는 것을 방지할 수 있습니다. 자세한 내용은 *hosts.equiv*(4)를 참조하십시오.
- `.netrc` 파일은 원격 로그인에 대한 `login` 인증을 건너뛰고 사용자의 암호화되지 않은 암호를 포함하므로 사용하지 마십시오. 사용할 경우 소유자 이외의 다른 사용자가 `.netrc` 파일을 읽거나 쓸 수 없도록 해야 합니다. 자세한 내용은 *netrc*(4)를 참조하십시오.

6.1.5 fsck를 사용하여 파일 손상 검색 및 수정

다음 문제는 손상된 파일 시스템을 나타낼 수 있습니다.

- 파일에 잘못된 데이터(불필요 정보)가 들어 있습니다.
- 파일이 잘리거나 데이터가 손실되었습니다.
- 예기치 않게 파일이 사라지거나 위치가 바뀝니다.
- 사용자의 터미널, 시스템 콘솔 또는 시스템 로그에 오류 메시지가 나타납니다.
- 디렉토리 또는 목록 파일을 변경할 수 없습니다.
- 시스템이 다시 부팅되지 않습니다.

사용자나 다른 사용자가 파일 시스템의 문제를 쉽게 식별할 수 없으면 `fsck` 명령을 사용하여 파일 시스템을 검사합니다. `fsck` 명령은 파일 시스템 불일치를 찾고 수정하는 주요 도구입니다. `fsck` 명령은 `/etc/fstab`에 나열된 파일 시스템을 검사합니다.

`fsck` 유틸리티는 파일 손상을 감지할 수 없습니다. `fsck`에서 오류를 찾지 못하면 파일 시스템 손상 문제가 아닐 가능성이 큽니다. 즉, 기본 데이터가 손실되거나 손상된 경우에도 파일 시스템을 사용할 수 있습니다. 다음과 같은 다른 파일 문제를 하나 이상 찾아 보십시오.

- 사용자, 프로그램 또는 응용 프로그램에서 파일을 삭제하거나 덮어쓰거나 이동하거나 잘랐습니다.
- 파일이 만들어질 때 특정 디렉토리와 연관된 파일 시스템이 해당 디렉토리에 마운트되지 않았을 수 있습니다.
- 현재 파일 시스템이 마운트된 디렉토리에 파일이 저장되어 있습니다. 파일은 있지만 액세스할 수 없습니다. 파일에 액세스하려면 파일 시스템을 마운트 해제합니다.
- 파일 보호나 소유권으로 인해 액세스할 수 없습니다. `chmod` 또는 `chown` 명령을 사용하여 파일 사용 권한을 변경합니다.

6.2 액세스 제어 목록 설정

ACL(액세스 제어 목록)을 통해 일반적인 파일 액세스 권한보다 세부적으로 파일을 보호할 수 있습니다. ACL을 사용하여 사용자가 속한 그룹과 관련이 없는 개별 사용자에게 파일 액세스를 허용하거나 제한할 수 있습니다. 파일 소유자나 해당 권한을 가진 사용자만 ACL을 만들 수 있습니다.

JFS(Journaled File System) 및 HFS(High-Performance File System)는 ACL을 지원하지만 다른 메커니즘과 구문을 사용합니다.

JFS는 VxFS(Veritas Journaled File System)의 HP-UX 구현입니다. HFS는 UFS(UNIX 파일 시스템)의 HP-UX 버전이고 이전 버전의 HP-UX와 호환됩니다.

ACL(액세스 제어 목록)은 파일과 연관된 사용자, 그룹 및 모드 항목 집합입니다. 이 목록은 가능한 모든 사용자 ID 및 그룹 ID 조합의 사용 권한을 지정합니다. 액세스 제어 목록을 사용하면 일반적인 UNIX 파일 사용 권한을 사용할 때보다 더 세부적으로 파일 액세스를 제어할 수 있습니다. ACL을 통해 일반적인 제어는 물론 개별 사용자 및 특정 그룹에 대해 파일 액세스를 부여하거나 제한할 수 있습니다.

HFS 및 JFS 파일 시스템은 모두 ACL을 지원하지만 서로 다른 메커니즘과 구문을 사용합니다.

참고: HFS는 더 이상 사용되지 않으며 향후 릴리즈의 운영 체제에서 제거될 것입니다.

HP-UX는 두 가지 별도 JFS 제품을 지원합니다. 기본 JFS 제품은 운영 체제에 포함되어 있고 선택적 고급 제품인 OnLineJFS는 별도로 설치됩니다. 두 JFS 제품은 모두 ACL을 지원합니다.

자세한 내용은 *setacl*(1), *getacl*(1), *aclv*(5), *chacl*(1), *lsacl*(1) 및 *acl*(5)을 참조하십시오.

6.3 HFS ACL 사용

`chacl` 명령을 사용하여 HFS ACL 사용 권한을 설정하고 `lsacl` 명령을 사용하여 표시합니다. 자세한 내용은 보기 6-1을 참조하십시오.

중요: HFS ACL 사용 권한이 할당된 파일에 대한 작업을 수행할 경우 `chmod`에 `-A` 옵션을 사용해야 합니다. `-A` 옵션을 사용하지 않으면 `chmod`는 파일에서 ACL 사용 권한을 삭제합니다. 구문은 다음과 같습니다.

```
# chmod -A mode file
```

`chacl` 명령은 `chmod` 명령의 수퍼셋입니다. `chmod` 명령을 사용하여 지정한 보다 일반적인 사용 권한에 `chacl` 명령을 사용하여 지정한 특수한 사용 권한이 추가됩니다.

파일에 ACL이 있으면 `ll` 명령은 사용 권한 문자열 뒤에 더하기 기호(+)를 표시합니다.

`user.group`이 둘 이상의 HFS ACL 항목과 일치하면 보다 명확한 항목이 높은 우선 순위를 갖습니다. 자세한 내용은 보기 6-2를 참조하십시오.

보기 6-1 HFS ACL 만들기

이 예제에서 chmod 명령은 myfile에 대한 쓰기 권한을 사용자 allan으로만 제한합니다. 또한 chmod 명령은 이전 HFS ACL을 삭제합니다.

```
$ chmod 644 myfile
$ ll myfile
-rw-r--r--   1 allan      users          0 Sep 21 16:56 myfile
$ lsacl myfile
(allan.%,rw-)(%.users,r--)(%.%,r--) myfile
```

lsacl 명령은 기본 소유자, 그룹 및 다른 사용 권한에 해당하는 기본값만(ACL 제외) 표시합니다.

chacl 명령은 myfile에 대한 읽기 및 쓰기 권한을 다른 사용자에게 부여합니다.

```
$ chacl 'naomi.users=rw' myfile
$ ll myfile
-rw-r--r--+  1 allan      users          0 Sep 21 16:56 myfile
$ lsacl myfile
(naomi.users,rw-)(allan.%,rw-)(%.users,r--)(%.%,r--) myfile
```

여기서 ll 사용 권한 표시에는 +가 붙어 있습니다. 이는 ACL이 있으며 ll 사용 권한 문자열이 변경되지 않았다는 것을 나타냅니다. lsacl에서 추가적으로 표시된 항목은 users 그룹에 있는 사용자 naomi가 myfile에 대한 읽기 및 쓰기 권한이 있음을 나타냅니다.

보기 6-2 여러 HFS ACL 일치 항목

사용자의 user.group 조합이 둘 이상의 ACL 항목과 일치하면 가장 명확한 항목이 가장 높은 우선 순위를 갖습니다. 이 예제에서는 첫 번째 항목이 파일 사용 권한을 설정합니다.

```
$ chmod 644 myfile
```

myfile에 대해 chacl 명령을 사용하여 사용자 naomi에 대한 쓰기 전용 항목을 추가합니다.

```
$ chacl naomi.%=w myfile
$ lsacl myfile
(naomi.%,-w-)(allan.%,rw-)(%.users,r--)(%.%,r--) myfile
```

이제 사용자 naomi는 naomi.%에 정의된 ACL을 사용하여 myfile 파일에 대한 쓰기 권한을 갖게 되지만 파일에 대한 읽기 권한은 없게 됩니다. 그 이유는 naomi.%가 %.users 및 %.%에 정의된 ACL 중 가장 높은 우선 순위를 갖기 때문입니다.

lsacl 명령은 HFS ACL을 가장 특수한 것부터 순서대로 표시합니다. 즉, 사용 권한 일치는 왼쪽에서 오른쪽으로 시도됩니다.

6.3.1 HFS ACL 및 HP-UX 명령과 호출

다음 명령 및 시스템 호출은 HFS 파일 시스템의 ACL에서 작동합니다.

표 6-2 HFS ACL 명령

명령	설명
chacl	파일의 HFS ACL을 변경합니다.
getaccess	파일에 대한 사용자의 액세스 권한을 표시합니다.
lsacl	파일의 HFS ACL을 표시합니다.

표 6-3 HFS ACL 시스템 호출

시스템 호출	설명
getaccess()	파일에 대한 사용자의 유효한 액세스 권한을 가져옵니다.
getacl(), **fgetacl()**	HFS ACL 정보를 가져옵니다.
setacl(), **fsetacl()**	HFS ACL 정보를 설정합니다.
acltostr()	HFS ACL 구조를 문자열 형식으로 변환합니다.
chownacl()	HFS 파일의 ACL에 표시된 소유자 또는 그룹을 변경합니다.
cpacl(), **fcpacl()**	HFS ACL 및 모드 비트를 한 파일에서 다른 파일로 복사합니다.
setaclentry(), **fsetaclentry()**	HFS 파일의 ACL 항목을 추가, 수정 또는 삭제합니다.
strtoacl()	HFS ACL 구조를 구문 분석하고 문자열 형식으로 변환합니다.
strtoaclpatt()	HFS ACL 패턴 문자열을 구문 분석하고 배열로 변환합니다.

ACL 항목은 다음 명령, 시스템 호출 및 서브루틴 라이브러리에 의해 영향을 받으며 때때로 예기치 않은 방식으로 영향을 받습니다.

표 6-4 ACL 항목에 영향을 주는 명령 및 호출

명령 또는 호출	설명
chmod	기본적으로 HFS ACL을 삭제합니다. HFS ACL을 유지하려면 `-A` 옵션을 사용합니다.
chmod()	HFS ACL 항목을 삭제합니다. HFS ACL 항목을 저장하고 복원하려면 **getacl()** 및 **setacl()**을 사용합니다.
cpset	파일의 선택적인 ACL 항목을 설정하지 않습니다.
find	HFS 또는 JFS 파일 시스템에서 특정 ACL 패턴과 일치하거나 이를 포함하는 ACL 항목을 갖는 파일을 식별합니다.
ls -l	긴 형식은 파일의 사용 권한 비트 뒤에 더하기 기호(+)를 표시하여 ACL이 있음을 나타냅니다.
mailx	/var/mail/* 파일의 선택적인 ACL 항목을 지원하지 않습니다.
compact, compress, cp, ed, pack, unpack	ACL 항목을 새로 만든 파일로 복사합니다.

표 6-4 ACL 항목에 영향을 주는 명령 및 호출 (계속)

명령 또는 호출	설명
`frecover`, `fbackup`	파일을 선택적으로 복구하고 백업하려면 이러한 명령만 사용하십시오. ACL을 지원하지 않는 시스템에서 복구하려면 ACL 시스템에서 백업할 때 `-A` 옵션을 사용합니다.
`ar`, `cpio`, `ftio`, `shar`, `tar`, `dump`, `restore`	이러한 명령은 아카이브 및 복원 시 ACL을 유지하지 않습니다. **`stat()`**에서 반환된 `st_mode` 값을 사용합니다.
`rcs`, `sccs`	이러한 명령은 ACL을 지원하지 않습니다.

HFS 액세스 제어 목록은 새 파일 시스템을 만들 때 추가 "연속 inode"를 사용합니다. 다음 명령을 사용할 때 이를 고려하십시오.

- `fsck`: ACL 항목이 있는 파일의 개수를 `icont`의 값으로 반환합니다. 참조되지 않은 연속 inode를 지우려면 `-p` 옵션을 사용합니다. *fsck*(1M)를 참조하십시오.
- `diskusg`, `ncheck`: 연속 inode를 무시합니다. *diskusg*(1M) 및 *ncheck*(1M)를 참조하십시오.
- `mkfs`: 새 디스크에 연속 inode를 허용합니다. *mkfs*(1M)를 참조하십시오.

6.4 JFS ACL 사용

이 절에서는 JFS ACL 및 이를 사용하는 방법에 대해 설명합니다.

참고: JFS ACL을 사용하려면 디스크 레이아웃 버전 4를 사용하는 VxFS 파일 시스템이 있어야 합니다. 파일 시스템을 버전 4로 업그레이드하는 방법에 대한 자세한 내용은 *vxupgrade*(1M)를 참조하십시오.

6.4.1 JFS ACL 정의

JFS ACL에는 특정 사용자 및 그룹의 이름을 지정하고 각각에 부여된 권한을 나타내는 한 줄짜리 항목이 들어 있습니다. 또한 JFS ACL이 있으므로 `ls -l` 명령을 사용하여 표시된 `group` 사용 권한 비트의 의미가 변경됩니다.

JFS ACL에는 `user` 항목, `group` 항목, `class` 항목 및 `other` 항목을 포함하여 항상 네 개 이상의 항목이 있습니다. JFS ACL에 이 네 항목만 들어 있는 경우 부여되는 사용 권한은 표준 UNIX 시스템 사용 권한 비트로 표시되는 사용 권한과 정확히 같습니다.

6.4.2 시스템에서 JFS ACL을 생성하는 방법

JFS 파일 시스템에서 파일이 만들어질 때마다 시스템은 소유자 사용 권한에 대한 `user` 항목, 소유 그룹 사용 권한에 대한 `group` 항목, 소유 그룹 사용 권한에 대한 `class` 항목 및 기타 그룹 사용 권한에 대한 `other` 항목이 들어 있는 최소 JFS ACL을 파일에 대해 초기화합니다. 사용자에 의해 또는 부모 디렉토리에 지정된 기본 항목의 결과로 항목이 추가될 수 있습니다.

6.4.3 최소 JFS ACL

앞에서 정의한 네 가지 기본 항목이 포함된 ACL을 최소 JFS ACL이라고 합니다. 최소 ACL 예제는 다음과 같습니다.

```
user::rw-
group::r--
class:r--
other:---
```

- `user` 항목은 파일 소유자의 사용 권한을 나타내고 소유자 사용 권한 비트에 직접 매핑됩니다. 첫 번째 항목은 파일 소유자에 적용되기 때문에 사용자 이름을 나타낼 필요가 없습니다. 이 예제 ACL 항목은 파일 소유자에게 읽기 및 쓰기 권한을 부여합니다.
- `group` 및 `class` 항목은 파일 소유 그룹의 구성원에게 할당되는 사용 권한을 지정합니다. 예제 ACL 항목은 파일 소유 그룹에 읽기 전용 권한을 부여합니다. `group` 및 `class` 항목에 대해서는 섹션 6.4.5에서 자세히 설명합니다.
- `other` 항목은 다른 항목에서 사용 권한을 부여하거나 거부하지 않은 사용자의 사용 권한을 지정하는 기타 항목입니다. 예제 `other` 항목은 파일의 소유자가 아니고 파일 소유 그룹에 없는 모든 사용자의 액세스를 거부합니다.

이 파일에 대해 `ls -l`을 수행하여 표시된 사용 권한 비트는 다음과 같습니다.

```
rw-r-----
```

다음 절에서는 추가 JFS ACL 항목이 파일 액세스에 미치는 영향 및 사용 권한 비트의 해석에 대해 설명합니다.

6.4.4 추가 JFS ACL user 및 group 항목

시스템의 특정 사용자 및 그룹에 사용 권한을 부여하거나 거부하려는 경우 이전 절에서 설명한 최소 항목 네 개에 `user` 및 `group` 항목을 13개까지 더 추가할 수 있습니다.

예를 들어, 파일 ACL의 다음 항목은 `boss`로 로그인한 사용자에게 읽기, 쓰기 및 실행 권한을 부여합니다.

```
user:boss:rwx
```

예를 들어, 다음 항목이 포함된 ACL은 `spies` 그룹에 있는 사용자의 액세스를 거부합니다.

```
group:spies:---
```

6.4.5 JFS ACL group 및 class 항목

최소 ACL이 있는 파일에서는 소유 `group`과 `class`의 ACL 항목이 동일합니다. 그러나 추가 항목이 있는 파일에서는 소유 `group`과 `class`의 ACL 항목이 다릅니다. 소유 `group` 항목은 사용 권한을 특정 그룹인 소유 `group`에 부여합니다.

`class` 항목은 보다 일반적입니다. 이 항목은 추가 `user` 및 `group` 항목에서 부여할 수 있는 최대 사용 권한을 지정합니다.

특정 사용 권한이 `class` 항목에 부여되지 않은 경우 모든 ACL 항목(첫 번째 `user`(소유자) 항목 및 `other` 항목 제외)에서 이 사용 권한을 부여할 수 없습니다. 모든 사용 권한은 특정 사용자 또는 그룹에 대해 거부될 수 있습니다. `class` 항목은 파일 사용 권한에 대한 상한값의 기능을 합니다.

ACL에 둘 이상의 `group` 또는 `user` 항목이 있는 경우 이러한 추가 항목에 의해 부여된 유효 사용 권한이 `class` 항목에 의해 제한되므로 추가 `user` 및 `group` 항목은 `group class` 항목으로 참조됩니다.

6.4.6 setacl 및 getacl 명령 사용

`setacl` 및 `getacl` 명령을 사용하여 ACL을 변경하고 볼 수 있습니다.

`setacl` 명령을 사용하여 다음 방법 중 하나로 ACL을 변경합니다.

- 디렉토리의 기본 ACL을 포함하여 파일의 전체 ACL을 바꿉니다.
- 디렉토리의 기본 항목을 포함하여 하나 이상의 항목을 추가, 수정 또는 삭제합니다.

`getacl` 명령은 ACL 항목을 표시합니다. `user` 및 `group`의 파일 사용 권한 비트는 다음과 같이 이러한 항목의 특별한 경우로 해석됩니다.

- 소유자 사용 권한을 나타내는 비트는 지정된 사용자 ID 없이 `user` 항목으로 표시됩니다.
- 그룹 사용 권한을 나타내는 비트는 지정된 그룹 ID 없이 `group` 항목으로 표시됩니다.

  ACL에는 이러한 특별한 `user` 및 `group` 항목이 하나씩 있어야 합니다. 개수에 관계없이 ACL에 추가 `user` 항목과 `group` 항목이 있을 수 있지만 각각 사용자 ID 또는 그룹 ID가 있어야 합니다. 기타 사용자에게 부여될 사용 권한의 사용 권한 비트를 나타내는 `other` 항목은 ACL에 하나만 포함될 수 있습니다.

명령 설명은 *setacl*(1) 및 *getacl*(1)을 참조하십시오.

6.4.7 class 항목에 대한 chmod의 영향

파일에 최소 ACL이 있는 경우 소유 `group` 및 `class` ACL 항목은 동일하며 `chmod`가 두 항목에 모두 영향을 미칩니다. 그러나 파일의 ACL에 선택적인 추가 항목이 있는 경우 다음과 같은 결과가 발생합니다.

- `class` ACL 항목이 더 이상 소유 `group` ACL 항목과 같을 필요가 없습니다.
- `chmod`가 `class` ACL 항목에는 영향을 미치지만 소유 `group` 항목에는 영향을 미치지 않습니다.
- 소유 `group` 항목을 변경하려면 `setacl` 명령을 사용해야 합니다.

6.4.8 최소 JFS ACL 변경 예제

JFS ACL `class` 항목의 기능을 설명하기 위해 이 절에서는 `chmod` 및 `setacl`이 최소 JFS ACL이 있는 파일과 `group class` 항목이 있는 파일에 미치는 영향에 대해 설명합니다.

참고: `getacl` 및 `setacl` 명령의 사용에 대한 자세한 내용은 섹션 6.4.10에 설명되어 있습니다. 또한 *getacl*(1) 및 *setacl*(1)을 참조하십시오.

읽기 전용(`444`) 사용 권한과 최소 JFS ACL이 있는 `exfile` 파일이 있다고 가정합니다. **`ls -l`** 명령은 `exfile`에 대한 사용 권한을 보여 줍니다.

```
$ ls -l exfile
-r--r--r-- 1 jsmith users 12 Sep 20 15:02 exfile
```

`getacl` 명령은 최소 JFS ACL인 `exfile`에 대해 다음 출력을 표시합니다.

```
$ getacl exfile
# file: exfile
# owner: jsmith
# group: users
user::r--
```

```
group::r--
class:r--
other:r--
```

chmod 명령을 사용하여 쓰기 권한을 exfile에 추가하면 소유 group 및 class ACL 항목이 모두 변경됩니다. 예를 들어, 다음과 같은 getacl 명령 출력을 찾습니다.

```
$ chmod 666 exfile
$ getacl exfile
# file: exfile
# owner: jsmith
# group: users
user::rw-
group::rw-
class:rw-
other:rw-
```

이제 class ACL 항목에는 영향을 미치고 소유 group 항목에는 영향을 미치지 않는 다른 사용자 및 그룹 항목을 추가합니다. 아래의 첫 번째 setacl 명령은 guest 사용자에게 읽기 전용 권한을 부여합니다. 다른 ACL 항목은 영향을 받지 않습니다. 그러나 두 번째 setacl 명령은 dev 그룹에 읽기-실행 권한을 부여하고 사용 권한의 상한값(class 항목)이 실행 권한을 포함하도록 확장됩니다.

```
$ setacl -m u:guest:r-- exfile
$ setacl -m g:dev:r-x exfile
$ getacl exfile# file: exfile
# owner: jsmith
# group: users
user::rw-
user:guest:r--
group::rw-
group:dev:r-x
class:rwx
other:rw-
```

그런 다음 chmod 명령이 group에서 쓰기 및 실행 권한을 제거하고 실제로 class 권한을 읽기 전용으로 축소합니다. 변경하지 않는 한, 소유 group 유효 사용 권한도 읽기 전용으로 축소됩니다.

```
$ chmod g-wx exfile
$ getacl exfile
# file: exfile
# owner: jsmith
# group: users
user::rw-
user:guest:r--
group::rw-      # effective:r--
group:dev:r-x  # effective:r--
class:r--
other:rw-
```

other 사용 권한은 변경되지 않습니다. class 항목은 첫 번째 user(소유자) 항목 또는 other 항목에서 부여할 수 있는 액세스 권한을 제한하지 않습니다.

그런 다음 `ls -l` 명령이 `exfile`의 사용 권한을 표시합니다. 사용 권한 문자열의 끝에 있는 더하기 기호(+)는 파일의 ACL이 있음을 나타냅니다.

```
$ ls -l exfile
-rw-r--rw-+ 1 jsmith users 12 Sep 20 15:02 exfile
```

6.4.9 기본 JFS ACL

한 디렉토리에 만들어진 모든 파일에 대해 특정 ACL 항목을 지정할 수 있습니다. 예를 들어, 두 사람이 함께 작업하는 경우 자신의 디렉토리에 있는 파일에 다른 사람이 쓸 수 있도록 허용할 수 있습니다.

디렉토리의 모든 파일에 대해 원하는 액세스 권한을 부여하는 ACL 항목을 지정할 수 있지만 새 파일을 만들 때마다 해당 항목을 다시 추가해야 합니다. 기본 ACL 항목을 사용하여 파일을 만들 때마다 시스템에서 자동으로 이 작업을 수행하도록 할 수 있습니다.

기본 ACL 항목은 다음과 같습니다.

```
default:user:boss:rw-
```

기본 ACL은 디렉토리에만 지정할 수 있으며 사용자에게 부여되는 디렉토리 액세스 권한에는 영향을 주지 않습니다. 기본 ACL은 디렉토리에 만들어진 파일에 적용됩니다.

새로 만든 파일이 디렉토리인 경우 기본 ACL 항목으로 인해 다음 두 가지 영향이 나타납니다.

- 디렉토리에 만들어진 파일과 같은 방식으로 디렉토리에 원하는 사용 권한이 부여되고 거부될 수 있도록 기본이 아닌 해당 ACL 항목이 만들어집니다.
- 새 하위 디렉토리가 부모 디렉토리와 동일한 기본 ACL을 갖도록 기본 항목이 자체적으로 복사됩니다.

예를 들어, `projectdir` 디렉토리에 만들어진 모든 파일을 특정 사용자가 읽을 수 있도록 하려면 다음과 같이 해당 기본 항목을 만들 수 있습니다.

```
$ setacl -m d:u:boss:r,d:u:jjones:r,d:u:dev:r projectdir
```

```
$ getacl projectdir
# file: projectdir
# owner: jsmith
# group: users
user::rw-
user:boss:rw-
user:jjones:rw-
user:jdoe:---
group::rw-
group:dev:rw-
class:rw-
other:---
default:user:boss:r---
default:user:jjones:r--
default:group:dev:r--
```

새로 만든 파일이 디렉토리인 경우 동일한 ACL 항목이 생성됩니다. 또한 기본 항목 자체도 ACL에 추가됩니다.

이러한 항목이 추가되면 `projectdir` 디렉토리에 만들어진 모든 새 파일이 앞에서 설명한 것처럼 기본 항목이 없는 ACL을 갖게 됩니다.

6.4.10 setacl 명령을 사용하여 JFS ACL 변경

이 절에서는 setacl 명령을 사용하는 추가 예제를 제공합니다.

6.4.10.1 수정 및 삭제 옵션 사용

다음 setacl 명령은 -m(수정) 옵션을 사용하여 boss 사용자에게 junk 파일에 대한 읽기 전용 권한을 부여합니다.

```
$ setacl -m u:boss:r-- junk
```

dev 그룹의 모든 사용자에게 읽기 및 쓰기 권한을 부여하려면 setacl -m 명령에 그룹(g:) 매개 변수를 사용합니다.

```
$ setacl -m g:dev:rw- junk
```

-d 옵션은 항목을 삭제하므로 -d를 사용하여 ACL 항목에 사용 권한을 지정하지 마십시오. 예를 들어, 다음 명령은 그룹 dev에 대한 항목을 삭제합니다.

```
$ setacl -d g:dev junk
```

6.4.10.2 -f 옵션 사용

여러 개의 항목을 추가하거나 변경하는 경우 서로 다른 절차를 사용할 수 있습니다. ACL을 파일에 저장하고 파일을 편집한 다음 새 ACL을 파일에 적용할 수 있습니다. 예를 들어, 다음 명령을 사용하여 ACL을 파일에 저장할 수 있습니다.

```
$ getacl junk > junk.acl
```

파일을 다음과 같이 편집합니다.

```
$ cat junk.acl
# file: junk
# owner: user1
# group: group1
user::rw-
user:user2:rw-
user:user3:rw-
user:user4:---
user:user5:r--
group::rw-
group:group2:rw-
group:group3:r--
group:group4:---
group:group5:rw-
class:rw-
other:r--
```

setacl -f 명령을 사용하여 ACL을 파일에 적용합니다.

```
$ setacl -f junk.acl junk
```

6.4.10.3 유효 사용 권한 및 setacl -n

일반적으로 setacl은 class 항목을 다시 계산하여 추가 ACL 항목에서 부여된 사용 권한이 실제로 부여되도록 합니다. -n 옵션을 지정하면 class 항목이 다시 계산되지 않고 기존 값이 사용됩니다. 그러면 ACL 항목을 통해 부여한 일부 사용 권한이 실제로 부여되지 않을 수 있습니다.

예를 들어, dev 그룹에 읽기 및 실행 권한을 추가하려면 다음과 같이 setacl -n 명령을 사용하여 이 ACL을 수정합니다.

```
$ getacl exfile
# file: exfile
# owner: jsmith
# group: users
user::rw-
group::rw-
class:rw-
other:rw-
$ setacl -n -m group:dev:r-x exfile
$ getacl exfile
# file: exfile
# owner: jsmith
# group: users
user::rw-
group::rw-
group:dev:r-x     #effective r--
class:rw-
other:rw-
```

dev 그룹 ACL 항목은 지정된 대로 추가되지만 실행 권한은 실제로 부여되지 않습니다. 실행 권한은 class 항목에 의해 거부되며 -n이 지정되었으므로 class 항목이 다시 계산되지 않습니다. -n을 사용하지 않으면 class가 class:rwx로 다시 설정되며 effective 주석이 사용되지 않습니다.

6.5 JFS ACL과 HFS ACL 비교

JFS ACL은 POSIX ACL 표준을 준수합니다.

JFS ACL은 형식(내부 및 외부)과 기능 면에서 모두 HFS ACL과 다릅니다.

JFS ACL과 HFS ACL의 기능적 차이는 다음과 같습니다.

- JFS 디렉토리의 ACL은 기본 항목을 가질 수 있으며 이 기본 항목은 그 후 해당 디렉토리에 만들어진 파일에 적용됩니다. HFS ACL에는 이러한 기능이 없습니다.
- HFS ACL의 소유자는 ACL에서 제어하는 파일의 소유자와 다를 수 있습니다. JFS ACL의 소유자는 해당 파일의 소유자입니다.
- HFS ACL에는 특정 그룹의 특정 사용자에 대해 서로 다른 항목이 있을 수 있습니다. 예를 들어, userx는 users 그룹의 구성원으로 읽기 및 쓰기 권한을 가질 수 있지만 other 그룹의 구성원으로는 읽기 권한만 가질 수 있습니다.

6.5.1 JFS와 HFS의 명령 및 함수 매핑

표 6-5에서는 JFS ACL 및 HFS ACL에 대해 동일한 기능을 하는 명령 및 함수의 맨페이지를 보여줍니다.

표 6-5 HFS 및 JFS ACL 매핑

HFS 이름	대응하는 JFS 이름
chacl(1)	*setacl*(1)
lsacl(1)	*getacl*(1)
getacl(2)	*acl*(2)
fgetacl(2)	-없음-
setacl(2)	*acl*(2)
fsetacl(2)	-없음-
acltostr(3C)	-없음-
chownacl(3C)	-없음-
cpacl(3C)	-없음-
setaclentry(3C)	-없음-
strtoacl(3C)	-없음-
-없음-	*aclsort*(3C)
acl(5)	*aclv*(5)

6.6 ACL 및 NFS

NFS(Network File System)에는 원격 파일에 대한 ACL 정보를 전달하는 기능이 없습니다. 따라서 NFS를 사용할 경우 원격 파일의 ACL을 볼 수 없습니다. `ls -l` 명령에서 원격 파일에 ACL이 있음을 표시하지는 않지만 액세스 권한에 대한 ACL 제어는 계속 적용됩니다.

각각의 맨페이지 항목은 이러한 상황에서의 다양한 시스템 호출, 라이브러리 호출 및 명령의 동작을 지정합니다.

중요: NFS에서 알리지 않고 선택적인 항목을 삭제할 수 있으므로 네트워크를 통해 선택적인 항목이 있는 파일을 전송하거나 원격 파일을 조작할 때는 주의하십시오.

6.7 /dev 장치 특수 파일의 보안 고려 사항

시스템의 모든 장치에 대한 액세스는 장치 특수 파일에 의해 제어되므로 프로그램이 장치에 관계없이 실행될 수 있습니다. 이러한 파일은 적절한 사용 및 최대 보안이 가능하도록 사용 권한이 설정되어 제공됩니다.

다른 장치 특수 파일을 설치하는 경우 올바른 사용 권한 설정에 대한 자세한 내용은 *insf*(1M)를 참조하십시오.

장치 특수 파일도 다른 파일처럼 취약할 수 있으므로 다음 주의 사항을 준수하십시오.

- 모든 장치 특수 파일을 `/dev` 디렉토리에 보관합니다.
- 메모리 파일 `/dev/mem` 및 `/dev/kmem`에는 중요한 사용자 정보가 들어 있으므로 이러한 파일에 쉽게 액세스할 수 없도록 보호합니다. 예를 들어, 메모리에서 `login` 프로그램 호

출을 감시하는 프로그램이 사용자가 암호를 입력할 때 login 프로그램 버퍼에서 암호를 복사할 수 있습니다. 다음과 같이 파일 보호를 설정해야 합니다.

```
crw-r-----   1 bin        sys          3 0x000001 Jun  9  2006 /dev/kmem
crw-r-----   1 bin        sys          3 0x000000 Jun  9  2006 /dev/mem
```

- 모든 디스크 특수 파일을 보호합니다.
  - 일반 사용자가 모든 디스크 특수 파일에 쓸 수 없도록 하여 의도하지 않은 데이터 손상을 방지합니다. group 및 other의 쓰기 권한을 해제합니다.
  - 디스크 특수 파일을 읽을 수 없도록 하여 정보의 유출을 방지합니다. other의 읽기 권한을 해제합니다.

  다음과 같이 파일 보호를 설정해야 합니다.

```
brw-r-----   1 bin        sys         31 0x002000 Feb 18  2004 /dev/dsk/c0t2d0
crw-r-----   1 bin        sys        188 0x002000 Aug  3  2004 /dev/rdsk/c0t2d0
brw-r-----   1 root       sys         64 0x000002 Jun 11  2006 /dev/vg00/lvol2
crw-r-----   1 root       sys         64 0x000002 Jun 11  2006 /dev/vg00/rlvol2
```

- write 또는 talk 프로그램을 사용하여 사용자가 통신할 수 있도록 허용한 경우 모든 사용자가 HP-UX 시스템의 터미널 포트에 쓸 수 있습니다. 소유자에게만 읽기 권한을 허용합니다.
- 개별 사용자가 터미널 장치 또는 개인 프린터 이외의 장치 특수 파일을 소유할 수 없게 합니다.
- 출처를 모르는 디스크나 다른 마운트 가능한 장치를 서비스 상태로 전환하기 전에 해당 파일에서 장치 특수 파일과 setuid 프로그램을 확인합니다. 자세한 내용은 섹션 6.9를 참조하십시오.

6.8 디스크 파티션 및 논리 볼륨 보호

LVM(Logical Volume Manager)은 일반적인 디스크 관리 도구입니다. LVM은 디스크 파티션보다 쉽게 디스크를 나누며 볼륨이 여러 개의 디스크에 걸쳐 있을 수 있습니다. 볼륨은 하나의 물리 디스크 파티션으로 나타나는 논리 장치입니다. 파일 시스템이나 데이터베이스를 만드는 응용 프로그램 등에 볼륨을 가상 디스크 파티션으로 사용할 수 있습니다.

디스크 파티션 및 논리 볼륨에 관한 몇 가지 보안 고려 사항은 다음과 같습니다.

- 디스크 파티션 및 논리 볼륨에 대한 장치 특수 파일은 root 및 디스크 백업에 사용된 계정만 읽을 수 있도록 해야 합니다. 자세한 내용은 섹션 6.7을 참조하십시오.
- 소유권 및 사용 권한이 inode에 저장되므로 마운트된 파티션에 대한 쓰기 권한이 있는 모든 사용자는 해당 파티션의 모든 파일에 대해 사용자 ID를 설정할 수 있습니다. 소유자에 관계없이 **chmod()** 시스템 호출 및 기타 보안 검사를 건너뛰고 파일이 변경됩니다.

  장치 특수 파일이 쓰기 가능한 경우 사용자가 해당 파일을 열고 원시 디스크에 액세스할 수 있습니다. 그런 다음 직접 파일 시스템을 편집하거나 파일을 읽거나 파일 사용 권한과 소유자를 변경할 수 있습니다.

  파일 사용 권한으로 장치 특수 파일에 대한 액세스를 금지하고 root만 읽을 수 있게 합니다.

- 데이터베이스 응용 프로그램과 같은 프로그램에서 파티션에 직접 액세스해야 하는 경우 해당 파티션을 프로그램에 대해 배타적으로 예약합니다. 사용자가 직접 파티션에 액세스

할 수 있는 경우 파티션을 파일 시스템으로 마운트하지 마십시오. 파티션을 파일 시스템으로 마운트하면 사용자가 기본 파일 시스템을 편집할 수 있습니다.

파일 보안이 HP-UX 파일 시스템이 아닌 파일 사용 권한 설정에 의해 적용된다고 프로그램 사용자에게 알립니다.

6.9 파일 시스템 마운트 및 마운트 해제에 대한 보안 지침

`mount` 명령을 사용하면 제거 가능한 파일 시스템 및 디스크 또는 파티션을 기존 파일 트리에 연결할 수 있습니다. `mount` 명령은 `/etc/fstab`이라는 파일을 사용합니다. 이 파일에는 사용 가능한 파일 시스템 및 해당 마운트 지점 목록이 들어 있습니다. `/etc/fstab` 파일을 root만 쓸 수 있지만 다른 사용자가 읽을 수 있게 설정합니다. 파일 시스템 마운트에 대한 자세한 내용은 *fstab*(4)를 참조하십시오.

파일 시스템 또는 디스크를 마운트할 때 다음 주의 사항을 준수하십시오.

- 새 파일 시스템을 마운트할 마운트 지점 디렉토리(`/mnt` 등)를 만듭니다. 파일이 들어 있는 디렉토리에 파일 시스템을 마운트하면 이러한 파일에 액세스할 수 없게 되므로 마운트하지 마십시오.

  마운트된 파일 시스템의 마운트 지점은 파일 시스템의 root 디렉토리에 대한 사용 권한 및 소유권을 얻게 됩니다.

- 디스크 경로 이름에 사용 권한 및 액세스 제어 목록 항목을 설정하여 디스크에 대한 액세스를 제어합니다.
- `mount` 명령에 `-r` 옵션을 사용하여 파일 시스템을 읽기 전용으로 마운트합니다. 물리적으로 쓰기 금지된 파일 시스템은 이런 방식으로 마운트해야 합니다.
- 새로운 또는 외부 파일 시스템을 마운트할 때 미디어가 안전하지 않다고 가정합니다.
  - `PATH` 환경 변수에 "`.`"(현재 디렉토리)이 포함되지 않도록 합니다. 그렇지 않으면 새 파일 시스템을 조사하는 동안 `ls` 또는 비슷한 명령의 트로이 목마 버전이 실행될 수 있습니다.
  - `fsck` 명령을 실행하여 파일 시스템이 기술적으로 손상되지 않았는지 확인합니다. *fsck*(1M)를 참조하십시오.
  - `ncheck_hfs -s` 또는 `ncheck_vxfs -s` 명령을 실행하여 `setuid` 및 `setgid` 프로그램과 장치 파일을 검색하고 이상한 점이 있는지 조사합니다. `-s` 옵션은 보안 정책의 숨겨진 위반을 찾는 데 사용됩니다. 자세한 내용은 *ncheck_hfs*(1M) 및 *ncheck_vxfs*(1M)를 참조하십시오.
  - 사용 권한을 `700`(`drwx------`)으로 설정하여 root로 제한되는 디렉토리를 만듭니다.

    ```
    # mkdir /securefile
    # chmod 700 /securefile
    ```

  - 해당 위치에 외부 파일 시스템을 읽기 전용으로 마운트합니다.

    ```
    #  mount -r /dev/disk1 /securefile
    ```

  - 권한이 부여된 프로그램에 대해 모든 디렉토리를 검사하고 각 프로그램의 ID를 확인합니다.

— 시스템을 읽기 및 쓰기 권한으로 다시 마운트하고 이전 단계에서 발견한 파일에서 불필요한 setuid 및 setgid 사용 권한을 모두 제거합니다. 이러한 주의 사항은 사용자가 개인 파일 시스템 마운트를 요청하는 경우 특히 중요합니다.

이러한 테스트를 수행한 이후에만 파일 시스템을 마운트 해제하고 원하는 위치에 다시 마운트해야 합니다.

- 사용할 수 없도록 하거나 제거할 계정을 가진 사용자의 마운트된 파일 시스템은 모두 마운트 해제하십시오.

NFS 환경에 마운트된 파일에 대한 자세한 내용은 섹션 6.10.2를 참조하십시오.

6.10 네트워크의 파일 보안 제어

보안의 관점에서 보면 네트워크로 연결된 시스템이 독립형 시스템보다 훨씬 취약합니다. 네트워크를 사용하면 시스템 액세스 가능성이 높아지지만 그만큼 보안 위반의 위험도 증가합니다.

네트워크에서 보안을 완전히 제어할 수는 없지만 네트워크에 있는 각 노드의 보안을 제어하여 시스템 또는 사용자 생산성의 유용성을 줄이지 않고 침투 위험을 제한할 수 있습니다.

모든 네트워크 관리 프로그램은 root가 아닌 uucp, nso 또는 daemon과 같은 보호된 네트워크 별 계정의 소유여야 합니다.

6.10.1 네트워크 제어 파일의 사용 권한 설정 확인

모드, 소유자 및 그룹은 모든 시스템 파일에서 신중하게 설정됩니다. 이러한 파일을 정기적으로 검사하여 변경 사항을 확인합니다. 원래 값에서 변경된 사항을 확인하고 수정합니다.

/etc 디렉토리에 있는 네트워크 제어 파일에 특히 주의합니다. 이러한 파일은 네트워크 자체에 대한 액세스를 제공하므로 권한이 없는 액세스를 얻으려는 사용자에게 중요합니다. 네트워크 제어 파일은 일반 사용자가 쓸 수 없도록 해야 합니다. 이러한 파일은 다음과 같습니다.

exports	NFS 클라이언트로 내보낼 파일 시스템 목록
hosts	네트워크 호스트 및 해당 주소
hosts.equiv	로컬 호스트와 동일하게 액세스가 허용되는 원격 호스트
inetd.conf	인터넷 구성 파일
netgroup	네트워크 범위의 그룹 목록
networks	네트워크 이름 및 해당 주소
protocols	프로토콜 이름 데이터베이스
services	서비스 이름 데이터베이스

6.10.2 NFS 환경에 마운트된 파일

NFS(Network File System)는 다음과 같은 편리한 기능을 제공합니다.

- 파일 공간 절약
- 일관된 파일 사용 유지 관리
- 협력적인 사용자 환경 제공

NFS는 /etc/exports 파일을 통해 액세스를 제어하여 서버와 클라이언트 시스템 사이의 파일 공유를 능률적으로 처리합니다. /etc/exports의 항목은 서버에 있는 파일 시스템을 클라이언트 시스템 또는 지정된 목록의 시스템에 마운트할 수 있는 사용 권한을 제공합니다. 파일

시스템이 /etc/exports에 배치된 후에는 NFS 마운트를 수행할 수 있는 모든 사용자가 정보를 사용할 수 있습니다. 따라서 NFS 클라이언트 사용자는 서버 시스템에 로그인하지 않고도 서버 파일 시스템에 액세스할 수 있습니다. 내보낸 파일 시스템에 대한 액세스를 제어하는 방법은 *exports*(4)를 참조하고 보안 지침은 섹션 6.10.2.3을 참조하십시오.

6.10.2.1 서버 취약성

/etc/exports 파일에 제한적인 사용 권한을 설정하여 서버 보안을 유지합니다. root 권한은 NFS 전체에 유지되지 않습니다. 따라서 클라이언트 시스템에 root 권한이 있어도 서버에 특별한 액세스 권한이 있는 것은 아닙니다.

서버에서는 로컬로 해당 사용자의 사용 권한을 검사하는 것과 동일하게 클라이언트에 대해 원격으로 사용 권한을 검사합니다. 서버측에서는 네트워크를 통해 받은 클라이언트의 사용자 ID 및 그룹 ID를 서버 파일의 사용자 ID 및 그룹 ID와 비교하여 서버 파일에 대한 클라이언트의 액세스를 제어합니다. 검사는 커널에서 이루어집니다.

NFS 클라이언트에서 권한이 있는 사용자는 해당 권한을 이용하여 NFS 서버에 제한 없이 액세스할 수 있습니다.

참고: 사용자 자신의 노드 정책보다 관대한 권한이 부여된 노드에는 파일 시스템을 내보내지 마십시오.

6.10.2.2 클라이언트 취약성

워크스테이션용 NFS의 이전 릴리즈에서는 /dev inode가 클라이언트의 디스크에 상주해야 했습니다. 이제는 NFS에서 클라이언트 마운트된 장치 특수 파일의 주 번호 및 보조 번호가 들어 있는 /dev inode가 서버측에 있을 수 있습니다. 이렇게 되면 서버측에서 발견된 파일과 inode 번호를 통해 장치 특수 파일에 액세스하여 클라이언트에 마운트된 장치 특수 파일에 있는 사용 권한을 재정의하는 트로이 목마를 만들 수 있는 가능성이 생깁니다.

시스템 위반자는 클라이언트측에 장치 특수 파일을 만들 수 있는 사용 권한이 없더라도 서버측의 root 권한을 사용하여 /dev/kmem과 같은 장치 특수 파일을 만들 수 있습니다. 새 /dev 파일은 클라이언트측에 있는 대상 장치와 같은 주 번호 및 보조 번호를 갖도록 만들어지지만 다음의 권한을 갖게 됩니다.

```
crw-rw-rw-
```

그런 다음 위반자는 클라이언트로 이동하여 일반 사용자로 로그인한 다음 NFS를 사용하여 새로 만들어진 서버측 장치 특수 파일을 열고 그릇된 방향으로 사용할 수 있습니다.

6.10.2.3 NFS 마운트된 파일을 보호하는 방법

NFS 마운트된 파일을 보호하기 위한 제안 사항은 다음과 같습니다.

- 가능하면 한 사람이 클라이언트와 서버 시스템을 모두 관리하도록 합니다.
- 서버 및 클라이언트 시스템에 대한 사용자 ID 및 그룹 ID를 동일하게 유지합니다.
- 서버에서 내보낸 파일 시스템의 /dev 파일을 정기적으로 확인합니다.
- /etc/passwd 클라이언트 파일에 대한 쓰기 권한을 가진 사용자를 제한합니다.
- 가장 엄격하게 제어하자면 네트워크를 통해 액세스할 수 있는 모든 호스트를 감사합니다.
- fstab nosuid 명령을 사용하여 root로 실행되어 시스템을 손상시킬 수 있는 setuid 프로그램으로부터 시스템을 보호합니다. 기본 마운트 옵션은 suid로, 이 경우 setuid 권한

이 있는 마운트된 프로그램이 누가 시작하든 관계없이 소유자 사용 권한으로 실행될 수 있습니다. 따라서 root가 `setuid` 권한이 있는 프로그램을 소유하는 경우 누가 시작하든 관계없이 root 권한으로 프로그램이 실행됩니다.

7 구획

이 장에서는 HP-UX 11i v3의 구획 기능에 대해 설명합니다. 이 장의 내용은 다음과 같습니다.

- 개요(섹션 7.1)
- 구획 구조 계획(섹션 7.2)
- 구획 활성화(섹션 7.3)
- 구획 구성 수정(섹션 7.4)
- 구획 구성 요소(섹션 7.5)
- 구획 규칙 및 구문(섹션 7.6)
- 구획의 응용 프로그램 구성(섹션 7.7)
- 구획 문제 해결(섹션 7.8)
- discover 모드를 사용하여 초기 구획 구성 생성(섹션 7.9)
- HP Servicegard 클러스터의 구획(섹션 7.10)

7.1 개요

구획은 시스템의 구성 요소를 서로 분리하는 방법입니다. 제대로 구성된 경우 구획은 HP-UX 시스템과 이 시스템에 있는 데이터를 효과적으로 보호할 수 있습니다.

구획을 사용하여 프로세스 또는 주체를 서로 분리하는 것은 물론 리소스나 객체와 분리할 수 있습니다.

개념상, 각 프로세스는 하나의 구획에 속하고 리소스는 두 가지 방법 중 하나로 처리됩니다.

1. 만들기 프로세스의 구획을 사용하여 리소스에 레이블이 지정됩니다. 이는 통신 끝점 및 공유 메모리 같은 임시 리소스에 구획이 할당되는 방법입니다.
2. 파일 및 디렉토리 같은 영구 리소스에 대해 다른 구획에 있는 프로세스가 리소스에 액세스할 수 있는 방법을 지정하는 액세스 목록과 리소스가 연관될 수 있습니다. 즉, 프로세스는 해당 구획에 대한 규칙이 있는 경우에만 다른 구획에 속하는 프로세스와 통신하거나 리소스에 액세스할 수 있습니다. 같은 구획에 속하는 프로세스는 규칙이 없어도 해당 구획의 리소스에 액세스하고 서로 통신할 수 있습니다.

구획은 주체를 객체와 구분합니다. 이 경우 관련된 주체와 객체의 가상 그룹을 만들 수 있습니다. 한 구획에서 실행 중인 서비스가 손상되어도 다른 구획에서 실행 중인 서비스에는 영향을 주지 않도록 시스템을 구성할 수 있습니다. 이렇게 하면 손상 범위가 영향을 받는 구획으로만 제한됩니다.

7.1.1 구획 아키텍처

구획은 한 시스템 내의 프로세스와 자식 프로세스를 분리합니다. 그림 7-1에서는 시스템의 다양한 부분에 액세스해야 하는 많은 핸들러 프로세스를 시작하는 부모 프로세스를 보여 줍니다. 시스템의 구획은 프로세스가 필요한 리소스에 액세스할 수 있도록 구성되어 있습니다.

그림 7-1 구획 아키텍처

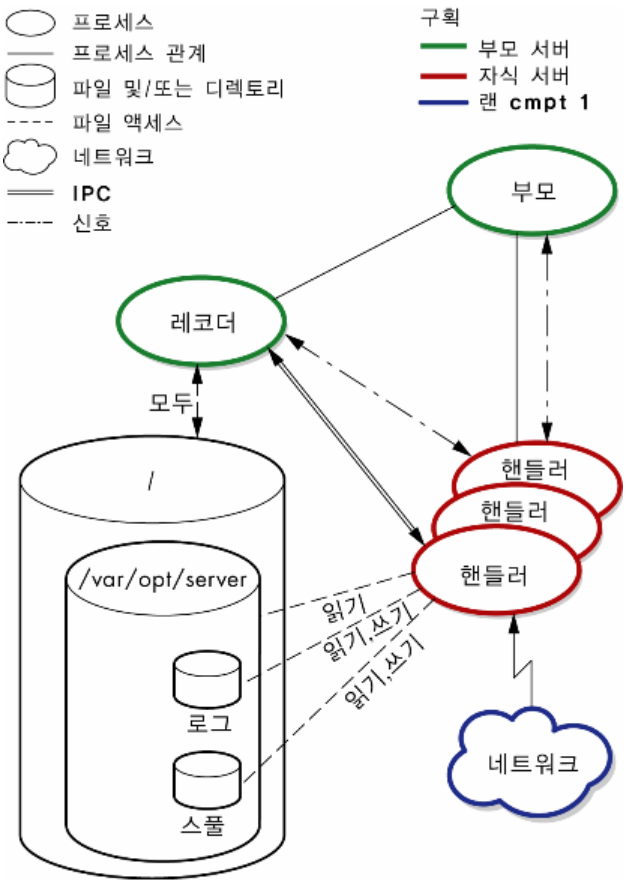


그림 7-1에서 부모 프로세스는 하나의 구획(구획 A)에 구성되어 있습니다. 작동하는 동안 부모 프로세스는 다른 구획(구획 B)에서 많은 핸들러 프로세스를 시작합니다. 핸들러 프로세스는 부모 프로세스의 구획 구성을 상속합니다. 이 시스템을 LAN에 연결하는 네트워크 카드는 다른 구획(구획 C)에 구성되어 있습니다. 파일 시스템은 구획 A에 대해 모든 액세스를 허용하지만 구획 B에 대해서는 부분 액세스만 허용하도록 구성되어 있습니다. 별도 구획에 있는 시스템 구성 요소 간의 통신은 다음과 같이 구성되어 있습니다.

- 모든 핸들러 프로세스가 네트워크와 통신하도록 구성되어 있습니다.
- 레코더가 파일 시스템에 액세스할 수 있습니다.
- 핸들러 프로세스에 파일 시스템 부분에 대한 읽기 및 읽기/쓰기 권한이 있습니다.
- 핸들러 프로세스는 IPC와 신호를 사용하여 부모 프로세스 및 레코더와 통신할 수 있습니다.
- 네트워크는 레코더 및 부모 프로세스와 분리되어 있습니다.

이 구획 구성은 파일 시스템과 레코더에 대한 보안을 제공합니다. 파일 시스템과 레코더는 둘 다 해당 구획에 의해 분리되어 있습니다. 핸들러 프로세스가 네트워크와 통신할 수 있지만 레코더나 부모 프로세스에서 네트워크에 액세스할 수는 없습니다.

7.1.2 기본 구획 구성

구획을 활성화하면 `INIT`라는 기본 구획이 만들어집니다. 시스템을 부팅할 때 `init` 프로세스는 이 구획에 속합니다. `INIT` 구획은 다른 모든 구획에 액세스할 수 있도록 정의되며 구획 규칙 파일에 정의되어 있지 않습니다.

중요: 규칙 파일에서 명시적 규칙을 만들어 `INIT` 구획을 다시 정의할 경우 해당 구획의 모든 특징이 사라지고 시스템을 다시 부팅하지 않으면 복원할 수 없습니다.

7.2 구획 구조 계획

구획 규칙을 만들기 전에 구획 구조를 계획해야 합니다.

구획 구조를 계획하려면 다음 질문에 대답하십시오.

- 이 시스템에 액세스하는 여러 사용자 그룹을 분리하시겠습니까? 예를 들어, 회계 부서와 인사 부서에서 모두 이 시스템을 사용하며 두 사용자 그룹을 분리해야 합니까?
- 방화벽 외부로 통신하는 이 시스템의 네트워크 인터페이스를 방화벽 내부에서만 통신하는 시스템의 나머지 부분과 분리하시겠습니까?
- 보안 정책에 구획을 사용하여 해결할 수 있는 요구 사항이나 문제가 있습니까?
- 보안 정책에서 특정 구획 규칙 구성을 지정하거나 제안합니까?

이러한 질문에 대답한 경우 이 대답을 사용하여 시스템의 각 부분을 특정 구획에 할당하는 방법을 결정합니다.

구획 구성을 계획할 때는 다음 권장 사항을 고려합니다.

- 모든 구획 구성 파일을 `/etc/cmpt` 디렉토리에 보관합니다.

  `#include` 지시어를 사용하여 시스템의 다른 위치에 구획 구성 파일을 만들 수 있습니다. 그러나 이 옵션은 사용하지 않는 것이 좋습니다. 대신 구획 구성 파일을 한 곳에 보관하여 쉽게 찾을 수 있게 합니다.

- 각 시스템 구성 요소에 대해 별도의 구획 구성을 만듭니다.

  두 구성 요소 간에 정의된 특정 소프트웨어 종속성이 없는 경우 다른 구성 요소에 대해 규칙을 혼합하지 마십시오. 구성 요소 구획에는 다른 구성 요소의 구획을 참조하는 규칙이 포함되지 않습니다. 구성 요소를 제거해야 하는 경우 구획 구성이 별도로 유지되면 보다 쉽게 구획 구성을 수정할 수 있습니다.

- 각 소프트웨어 구성 요소에 대해 하나의 구획 구성 파일을 만듭니다.

  이렇게 하면 시스템에서 소프트웨어를 제거할 경우 쉽게 구획 구성을 제거할 수 있습니다. 또한 소프트웨어 구성 요소와 관련된 모든 규칙을 쉽게 찾을 수 있습니다.

- 일부 소프트웨어 제품은 구획 규칙이 미리 구성된 상태로 제공됩니다. 이러한 규칙은 수정하지 마십시오.

제공된 구획 구성을 수정하기 전에 기존 구성을 이해해야 합니다. 소프트웨어 제품에 대한 설명서를 읽고 기존 구성을 꼼꼼하게 확인합니다.

주의: 기존 INIT 구획은 다시 정의하지 마십시오. INIT 구획을 변경하거나 다시 정의하면 자동으로 생성된 모든 정의가 삭제되고 구획이 제대로 작동하지 않습니다.

7.3 구획 활성화

시스템에서 구획 규칙을 활성화하려면 다음 단계를 수행합니다.

1. 구획 규칙을 계획합니다. 자세한 내용은 섹션 7.2를 참조하십시오.

팁: 신중하게 구획 규칙 구성을 계획하는 것이 좋습니다. 구성을 편집하고 프로덕션 시스템에서 구현한 후에는 변경하기가 어렵습니다. 구획 구성을 변경하는 경우 사용자 프로시저, 스크립트 및 도구를 변경해야 합니다.

2. 구획 규칙을 만듭니다. 이 단계를 완료하는 지침과 구획 규칙 구문에 대한 자세한 내용은 섹션 7.6을 참조하십시오.
3. (선택 사항) 다음 명령을 입력하여 구획 규칙을 미리 봅니다.

   ```
   # setrules -p
   ```

   -p 옵션은 구성된 규칙 목록을 구문 분석하고 구문과 의미의 불일치를 보고합니다. 시스템에서 구획 규칙을 활성화하기 전에 이 단계를 수행하는 것이 좋습니다.
4. (선택 사항) 구획 구성 파일의 백업 복사본을 만듭니다. 이러한 파일을 /etc/cmpt 디렉토리 외부에 보관하거나 .rules 접미사를 생략합니다. 이렇게 하면 편집 문제가 발생할 경우 쉽게 시작점으로 되돌릴 수 있습니다.
5. 다음 명령을 입력하여 구획 기능을 활성화합니다.

   ```
   # cmpt_tune -e
   ```
6. 시스템을 다시 부팅합니다. 이 단계는 필수입니다.

팁: 백업 파일을 보관합니다. 이렇게 하면 보다 쉽게 이전 구성으로 되돌릴 수 있습니다.

7.4 구획 구성 수정

시스템을 다시 부팅하지 않고 새 구획을 만들고 기존 구획을 수정할 수 있습니다. 구획 기능을 활성화 또는 비활성화하거나 구획을 완전히 제거하는 경우 시스템을 다시 부팅해야 합니다. 그러나 구획과 연관된 모든 규칙 및 해당 구획에 대한 모든 참조를 제거하면 다음에 다시 부팅할 때까지 구획을 시스템에 그대로 둘 수 있습니다.

구획 이름 변경의 의미에 대한 자세한 내용은 섹션 7.4.2를 참조하십시오.

새 구획 규칙을 추가하거나 필요 없는 규칙을 삭제하거나 기존 규칙을 수정할 수 있습니다. 기존 구획의 이름을 변경할 수도 있습니다.

다음 절에서는 구획 구성을 수정하는 방법에 대해 설명합니다.

7.4.1 구획 규칙 변경

1. (선택 사항) 수정할 구성 파일의 임시 백업 복사본을 만듭니다. 이러한 파일을 `/etc/cmpt` 디렉토리 외부에 보관하거나 `.rules` 접미사를 생략합니다. 이렇게 하면 편집 문제가 발생할 경우 쉽게 시작점으로 되돌릴 수 있습니다.
2. 다음 명령을 사용하여 현재 구획 규칙을 검사합니다.

   ```
   # getrules
   ```

3. 구획 규칙을 만들거나 수정합니다. 이 단계를 완료하는 지침과 구획 규칙 구문에 대한 자세한 내용은 섹션 7.6을 참조하십시오.
4. (선택 사항) 다음 명령을 입력하여 구획 규칙을 미리 봅니다.

   ```
   # setrules -p
   ```

   `-p` 옵션은 구성된 규칙 목록을 구문 분석하고 구문과 의미의 불일치를 보고합니다. 시스템에서 구획 규칙을 활성화하기 전에 이 단계를 수행하는 것이 좋습니다.
5. (선택 사항) 구획 구성 파일의 백업 복사본을 만듭니다.
6. `setrules` 명령을 실행하여 구성된 규칙을 로드합니다.

   ```
   # setrules
   ```

7.4.2 구획 이름 변경

구획 이름을 변경할 수 있습니다. 그러나 구획 이름을 변경하면 기존 구획 이름을 사용하여 구성된 응용 프로그램에 영향을 줄 수 있습니다. 구획 이름을 변경하는 경우 해당 구획에 구성된 응용 프로그램도 다시 구성해야 합니다.

참고: 구획 이름을 변경하는 경우 실제로 새 구획을 만들고 이전 이름을 가진 구획을 제거한 것입니다. 새 구획을 보려면 모든 참조를 변경해야 합니다. 다시 부팅할 때까지 이전 구획이 시스템에 남아 있습니다.

7.5 구획 구성 요소

구성 요소 기능은 구획을 구성하고 관리하는 데 사용되는 구성 파일 및 명령 집합으로 이루어져 있습니다. 구획 기능 사용을 지원하는 맨페이지가 있습니다. 이러한 구성 요소는 다음 절에 나열되어 있습니다.

7.5.1 구획 구성 파일

표 7-1에서는 구획 구성 요소에 사용되는 파일에 대해 설명합니다.

표 7-1 구획 구성 파일

구성 파일	설명
`/etc/cmpt`	구획 규칙 파일이 있는 디렉토리입니다.
`/etc/cmpt/*.rules`	시스템에 대해 구성된 구획 규칙이 들어 있는 파일입니다.
`/etc/cmpt/hardlinks/hardlinks.config`	여러 개의 하드링크가 가리키는 파일에 대한 구획 규칙의 일관성을 확인하기 위해 검색할 유효한 마운트 지점이 들어 있는 파일입니다.

7.5.2 구획 명령

표 7-2에는 구획을 관리하는 데 사용하는 명령이 포함되어 있습니다.

표 7-2 구획 명령

명령	설명
`cmpt_tune`	구획 기능을 쿼리, 활성화 및 비활성화합니다.
`setfilexsec`	구획 속성을 포함하여 바이너리 파일의 보안 속성을 설정합니다.
`getfilexsec`	구획 속성을 포함하여 바이너리 실행 파일과 연관된 보안 속성을 표시합니다.
`getprocxsec`	구획 속성을 포함하여 프로세스의 보안 속성을 표시합니다.
`getrules`	커널에서 현재 활성화된 구획 규칙을 표시합니다.
`setrules`	커널의 새 규칙이나 수정된 규칙을 활성화합니다. `-p` 옵션을 사용하면 수정된 규칙을 커널로 전달하지 않고 검토하기 위해 표시합니다.
`vhardlinks`	여러 개의 하드 링크가 있는 파일에 대해 구획 규칙의 일관성을 검사하여 충돌하는 액세스 규칙이 없도록 합니다.

7.5.3 구획 맨페이지

표 7-3에는 구획과 연관된 맨페이지가 포함되어 있습니다.

표 7-3 구획 맨페이지

맨페이지	설명
compartments(4)	구획 규칙 구문에 대해 설명합니다.
compartments(5)	구획 기능의 개요를 제공하고 구획 규칙 사용에 대해 설명합니다.
cmpt_tune(1M)	`cmpt_tune` 기능과 구문에 대해 설명합니다.
setfilexsec(1M)	`setfilexsec` 기능과 구문에 대해 설명합니다.
getfilexsec(1M)	`getfilexsec` 기능과 구문에 대해 설명합니다.
getprocxsec(1M)	`getprocxsec` 기능과 구문에 대해 설명합니다.
getrules(1M)	`getrules` 기능과 구문에 대해 설명합니다.
setrules(1M)	`setrules` 기능과 구문에 대해 설명합니다.
vhardlinks(1M)	`vhardlinks` 기능과 구문에 대해 설명합니다.

7.6 구획 규칙 및 구문

구획은 이름과 규칙 집합으로 구성됩니다. 이 절에서는 네 가지 유형의 구획 규칙에 대해 설명합니다.

- 파일 시스템 규칙
- IPC 규칙

- 네트워크 규칙은
- 기타 규칙

/etc/cmpt 디렉토리에 만든 규칙 파일에 규칙을 추가합니다. vi 또는 유사한 텍스트 편집기를 사용하여 이 파일을 편집할 수 있습니다. 규칙 파일에는 .rules 확장자가 있어야 합니다.

자세한 내용은 *compartments*(5)를 참조하십시오.

7.6.1 구획 정의

각 구획의 이름을 구성하고 하나 이상의 구획 규칙을 구획 이름과 연결하여 구획을 정의합니다. 순서에 관계없이 규칙을 지정할 수 있습니다.

구획 정의의 구문은 다음과 같습니다.

```
[sealed] [discover] compartment new_compartment_name { rules }
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

sealed	(선택 사항) 이 구획에 있는 프로세스가 **execve()**를 호출하여 권한을 얻거나 구획을 변경할 수 없습니다.
discover	(선택 사항) 구획 위반이 재정의되도록 규칙을 찾아서 자동으로 추가합니다. 이는 필요한 규칙을 확인하는 개발 기능이므로 실제 작업 시스템에서는 사용하면 안 됩니다. 이 키워드에 대한 자세한 내용은 섹션 7.9를 참조하십시오.
compartment	규칙을 구획 정의로 지정합니다.
new_compartment_name	새 구획과 연관된 레이블입니다. 이 레이블은 대/소문자를 구분합니다. 예를 들어, compartmenta와 CompartmentA는 서로 다른 구획입니다.
{}	이 구획에 대한 규칙을 묶습니다.

예를 들면 다음과 같습니다.

```
sealed compartment server_children {

/* Deny all access to any file system objects ... */
permission none /
}
```

참고: INIT 구획 이름은 대/소문자를 구분하지 않습니다. INIT, init 및 Init는 시스템에서 모두 같은 구획으로 간주됩니다.

구획 지정은 구문 분석이 시작되기 전에 cpp()를 사용하여 사전 처리됩니다. #include, #define, #ifdef 및 C 스타일 주석과 같은 cpp() 지시어를 사용하여 규칙 파일을 구성하고 문서화하는 것은 이러한 이유입니다.

7.6.2 파일 시스템 규칙

파일 시스템 규칙은 시스템에 있는 파일 및 디렉토리에 대한 프로세스의 액세스를 제어합니다. 명시적 규칙으로 상속을 재정의하지 않으면 파일 시스템 규칙은 부모 디렉토리에서 부모의 모든 하위 디렉토리 및 파일로 상속됩니다.

기본적으로 사용 권한을 지정하지 않으면 파일 시스템 객체에 대한 모든 사용 권한이 부여됩니다.

파일 시스템 규칙 구문은 다음과 같습니다.

```
(permission|perm) permission_list file_object
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

`permission` 또는 `perm`
: 파일이나 디렉토리에 대한 사용 권한을 설정합니다.

`permission_list`
: 파일이나 디렉토리에 적용할 수 있는 사용 권한 유형은 다음과 같습니다.

- `none`: 파일이나 디렉토리에 대한 모든 사용 권한을 거부합니다.
- `read`: 객체에 대한 읽기 권한을 제어합니다. 객체가 파일이면 파일 읽기 및 실행이 제어됩니다. 객체가 디렉토리이면 디렉토리 검색 및 나열이 제어됩니다. 또한 상속 때문에 디렉토리에 있는 모든 파일의 읽기가 제어됩니다. 파일을 실행하기 위해 열려면 읽기 권한이 있어야 합니다.
- `write`: 객체에 대한 쓰기 권한을 제어합니다. 객체가 파일이면 파일에 쓰기가 제어됩니다. 객체가 디렉토리이면 상속 때문에 디렉토리에 있는 모든 파일에 대한 쓰기가 제어됩니다.
- `create`: 객체를 만드는 기능을 제어합니다. 이 권한은 디렉토리 객체에만 적용됩니다. 또한 지정된 디렉토리에 있는 모든 디렉토리에 상속됩니다.
- `unlink`: 객체 삭제 기능을 제어합니다. 이 권한은 디렉토리 객체에만 적용됩니다. 또한 지정된 디렉토리에 있는 모든 디렉토리에 상속됩니다.
- `nsearch`: `file_object`가 디렉토리이면 요소 검색 기능을 제어합니다. 이 속성은 하위 디렉토리에 상속되지 않습니다.

`file_object`
: 파일이나 디렉토리의 전체 경로 이름입니다.

예를 들면 다음과 같습니다.

```
/* deny all permissions except read to entire system */
perm read  /

/* except for this directory   */
```

```
perm read,write,create,unlink /var/opt/server

/* just read and write log files, not create them */
perm  read,write /var/opt/server/logs
```

참고: 파일 시스템 객체에 대한 사용 권한을 부여하려면 구획 규칙에 적어도 해당 객체 위의 모든 디렉토리에 대한 읽기 권한이 있어야 합니다. 예를 들어, `/var/opt/tmp/file1`에 대한 읽기 및 쓰기 권한을 부여하려면 `/var/opt/tmp`, `/var/opt`, `/var` 및 `/`에 대한 읽기 권한을 부여해야 합니다.

7.6.3 IPC 규칙

IPC(Interprocess Communication) 규칙은 프로세스가 여러 구획에서 프로세스 간 통신 방법을 사용하는 방식을 제어합니다. IPC 통신 방법에는 프로세스 간 직접 통신이나 IPC 객체에 대한 공유 액세스가 포함됩니다. 객체가 프로세스와 연관되어 있으면 해당 객체를 만든 프로세스와 동일한 구획에 있습니다. 구획 규칙을 정의하여 객체에 액세스하는 프로세스와 액세스되는 객체 간의 관계를 설명합니다. 규칙에서 두 프로세스가 서로 통신한다고 설명하는 경우 두 번째 프로세스를 객체로 처리합니다. IPC 객체의 기본 동작은 규칙에서 명시적으로 허용하지 않는 한 서로 다른 구획 간의 모든 작업이 금지되는 것입니다.

두 가지 유형의 IPC 규칙이 있습니다. 첫 번째 규칙 유형의 구문은 다음과 같습니다.

```
(grant|access) (pty|fifo|uxsock|ipc) compartment_name
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

Access — 규칙이 객체 중심인지 주체 중심인지 지정합니다. 옵션은 다음과 같습니다.

- `grant`: 객체 중심 규칙을 지정합니다. 이 규칙을 사용하면 `compartment_name` 구획의 프로세스가 현재 구획에 있는 지정된 IPC 메커니즘에 액세스할 수 있습니다.
- `access`: 주체 중심 규칙을 지정합니다. 이 규칙을 사용하면 현재 구획의 프로세스가 `compartment_name` 구획에 있는 지정된 IPC 메커니즘에 액세스할 수 있습니다.

Method — 이 규칙이 적용되는 통신 방법을 지정합니다. 옵션은 다음과 같습니다.

- `pty`: 규칙이 프로세스 간 통신에 사용되는 `pty`에 적용되도록 지정합니다.
- `fifo`: 규칙이 FIFO에 적용되도록 지정합니다.
- `uxsock`: 규칙이 UNIX 도메인 소켓에 적용되도록 지정합니다.
- `ipc`: 규칙이 공유 메모리, 세마포어, 메시지 대기열 등의 SYSV 및 POSIX IPC 객체에 적용되도록 지정합니다.

compartment_name 이 구획의 프로세스가 통신할 수 있는 다른 구획의 이름입니다.

예를 들면 다음과 같습니다.

```
/* allow the children to access UNIX domain */
/* sockets created by the parent compartment */
grant  uxsock  server_children
```

두 번째 유형의 IPC 규칙은 프로세스 액세스를 제어합니다. 이 규칙 유형의 구문은 다음과 같습니다.

```
(send|receive) signal compartment_name
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

Direction 현재 구획의 프로세스에 다른 지정된 구획의 프로세스 동작을 보고 변경할 권한이 있는지 여부를 지정합니다. 옵션은 다음과 같습니다.

- `send`: 주체 중심 규칙을 지정합니다. 현재 구획의 프로세스가 신호를 보내고 `compartment_name` 구획에 있는 프로세스 데이터를 볼 수 있습니다.
- `receive`: 객체 중심 규칙을 지정합니다. `compartment_name` 구획의 프로세스가 신호를 보내고 현재 구획에 있는 프로세스 데이터를 볼 수 있습니다.

`signal` 이 규칙이 신호와 프로세스 표시에 적용되도록 지정합니다.

compartment_name 현재 구획의 프로세스가 액세스하여 프로세스 정보를 보거나 해당 정보가 표시될 수 있는 다른 구획의 이름입니다.

예를 들면 다음과 같습니다.

```
/* allow the parent to send signals to children */
   send signal server_children
```

7.6.4 네트워크 규칙

네트워크 규칙은 네트워크 인터페이스에 대한 액세스를 제어합니다. 또한 네트워크 규칙은 `INET` 도메인 통신(TCP/IP 소켓 및 스트림)을 사용하는 프로세스 간 통신을 제어합니다. 기본 동작은 네트워크에 대한 액세스 거부입니다.

네트워크 끝점은 해당 끝점을 만드는 프로세스의 구획이 레이블로 지정된 객체로 간주됩니다. 그러나 한 프로세스에서 네트워크 끝점을 만든 다음 다른 구획에서 실행되는 다른 프로세스로 전달할 수 있습니다. 액세스 확인은 현재 구획이 아니라 끝점이 만들어질 때 포함된 구획에서 수행됩니다. 또한 끝점은 새 연결을 받을 때 해당 구획 구성을 허용하는 끝점으로 전달합니다.

`INET` 도메인 끝점은 프로세스 간 통신에 자주 사용됩니다. 적절하게 구획을 구성하십시오.

네트워크 규칙의 구문은 다음과 같습니다.

```
(grant|deny) (server|client|bidir) (tcp|udp|raw [protonum] )
[port port_num] [peer[portport]] compartment_name
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

Access
: 지정된 구획의 네트워크 트래픽에 대한 구획 액세스를 허용하거나 거부합니다. 옵션은 다음과 같습니다.

- `grant`
- `deny`

Direction
: 규칙이 적용되는 방향을 지정합니다. 옵션은 다음과 같습니다.

- `server`: 이 규칙이 인바운드 요청에만 적용됩니다. TCP의 경우 들어오는 연결만 이 규칙에 의해 제어됩니다. UDP 및 RAW의 경우에는 모든 인바운드 패킷에 이 규칙이 적용됩니다.
- `client`: 이 규칙이 아웃바운드 요청에만 적용됩니다. TCP의 경우 연결 시작만 이 규칙에 의해 제어됩니다. UDP 및 RAW의 경우에는 모든 아웃바운드 패킷에 이 규칙이 적용됩니다.
- `bidir`: 이 규칙이 인바운드 요청과 아웃바운드 요청에 모두 적용됩니다. TCP의 경우 끝점에서 시작하고 받는 연결이 이 규칙에 의해 제어됩니다. UDP 및 RAW의 경우에는 끝점을 통과하는 모든 패킷에 이 규칙이 적용됩니다.

Protocol
: 이 규칙에 적용되는 네트워킹 프로토콜을 지정합니다. 옵션은 다음과 같습니다.

- `tcp`: 이 규칙이 TCP 프로토콜에 적용됩니다.
- `udp`: 이 규칙이 UDP 프로토콜에 적용됩니다.
- `raw`: 이 규칙이 INET 도메인의 다른 모든 프로토콜에 적용됩니다.

protonum
: 이 규칙에 대해 지정된 프로토콜 번호입니다. `protonum` 옵션은 `raw` 지정에만 관련이 있습니다.

`port`
: (선택 사항) 이 규칙이 특정 포트에 적용되도록 지정합니다.

port
: 이 규칙에 지정된 포트를 식별합니다.

`peer`
: (선택 사항) 포트 정보는 이 규칙에 대한 통신에 관련된 동등한 계층의 끝점에 적용됩니다.

compartment_name
: 피어 끝점과 연관된 구획 이름이나 이 규칙이 적용되는 인터페이스입니다.

예를 들면 다음과 같습니다.

```
/* allow all inbound TCP connections(any port)from interfaces labeled lancmpt1  */
grant server tcp lancmpt1
/* allow DNS client lookups (both TCP and UDP) through interface labeled lancmpt1 */
grant client tcp port 53 lancmpt1
grant bidir udp port 53 lancmpt1
/* allow only outbound telnet connections through interface labeled ifacelan0 */
```

```
grant client tcp peer port 23 ifacelan0
/* allow all TCP traffic except inbound telnet through interface labeled ifacelan0 */
/* the following two lines can be specified in either order */
grant bidir tcp ifacelan0
deny server tcp port 23 ifacelan0
/* allow inbound web server traffic through interface lan1cmpt */
grant server tcp port 80 lan1cmpt
```

네트워크 규칙에 대한 자세한 내용은 *compartments*(4)를 참조하십시오.

7.6.5 기타 규칙

다른 규칙 범주에 맞지 않는 규칙입니다.

네트워크 인터페이스 규칙 네트워크 인터페이스 규칙은 인터페이스가 속하는 구획을 지정합니다. 구획에 속해 있지 않은 네트워크 인터페이스는 온라인으로 가져올 수 없습니다.

참고: 보다 엄격한 보안 정책을 위해 프로세스에 할당된 구획과 별도의 구획에 네트워크 인터페이스를 구성하십시오. 각 구획에 대해 적절하게 네트워크 액세스 규칙을 정의합니다. 동등한 구획에는 항상 서로에 대한 모든 권한이 부여됩니다.

네트워크 인터페이스 규칙의 구문은 다음과 같습니다.

```
compartment compartment_name {
interface interface_or_ip[,interface_or_ip][...]
}
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

`interface`	인터페이스 정의로 지정합니다.
`interface_or_ip[,interface_or_ip][...]`	인터페이스 이름, IP 주소 또는 IP 주소 범위의 쉼표로 구분된 목록입니다. IP 주소 또는 범위는 선택적 마스크를 사용하여 IPv4 주소나 IPv6 주소로 지정될 수 있습니다.

예를 들면 다음과 같습니다.

```
compartment iface0 {
/* Define the compartment for the network interface lan0  */
 interface  lan0
/* All addresses in the range 192.168.0.0-192.168.0.255 */
  interface  192.160.0.0/24
}
compartment other_ifaces {
/* Define the compartment for two of the other network interfaces */
interface lan1,lan5
```

권한 제한 규칙 권한 제한 규칙은 권한 상속을 제어합니다. 권한 제한 규칙에 명명된 권한은 `execve(2)`를 호출하여 얻을 수 없습니다.

권한 제한 규칙의 구문은 다음과 같습니다.

```
disallowed privileges  privilege[,privilege[...]]
```

여기서 각 항목에 대한 설명은 다음과 같습니다.

`disallowed privileges`
`privilege[,privilege[...]]`

권한 제한 규칙으로 지정합니다.
쉼표로 구분된 권한 목록입니다. 다음 키워드를 추가로 사용할 수 있습니다.

- `all`: 모든 권한을 거부합니다.
- `none`: 모든 권한을 허용합니다.
- `!`: 제외를 나타냅니다.

예를 들면 다음과 같습니다.

```
/* Disallow all privileges except mount. */
disallowed privileges all,!mount

/* Disallow mount only. */
disallowed privileges none,mount
```

구획에 대해 권한 제한 규칙이 지정되어 있지 않으면 기본 권한 제한은 `INIT` 구획을 제외하고 모든 구획에 대해 `basicpolicy,mknod`입니다. `INIT` 구획 기본 권한 제한은 `none`입니다.

7.6.6 예제 규칙 파일

예제 규칙 파일은 `/etc/cmpt/examples/sendmail.example`에 있습니다.

7.7 구획의 응용 프로그램 구성

특정 구획에서 실행되도록 응용 프로그램을 구성할 수 있습니다. setfilexsec 명령을 사용하여 바이너리 파일의 구획 속성을 구성합니다. 예를 들어, `apple` 응용 프로그램을 `fruit` 구획에 구성하려면 다음 명령을 입력합니다.

```
# setfilexsec -c fruit apple
```

HP-UX RBAC를 사용하여 특정 구획에서 실행되도록 응용 프로그램을 구성할 수도 있습니다. 자세한 내용은 섹션 9.5.5를 참조하십시오.

7.8 구획 문제 해결

시스템 일부가 작동하지 않고 이 문제가 구획 구조로 인해 발생하는 것 같으면 다음과 같이 구획 규칙을 확인할 수 있습니다.

문제 1: 구성한 구획 규칙에 따라 액세스가 제어되고 있지 않습니다. 해결 방법: 규칙이 커널에 설정되어 있지 않을 수 있습니다. 규칙이 커널에 설정되어 있는지 확인하려면 다음 단계를 수행하십시오.

1. 다음 명령을 사용하여 커널에 있는 유효한 구획 규칙을 표시합니다.

   ```
   # getrules
   ```

2. 다음 명령을 사용하여 커널에 로드되지 않은 규칙을 포함하여 시스템에 구성된 모든 규칙을 표시합니다.

   ```
   # setrules -p
   ```

3. 두 명령의 출력을 비교합니다. 출력이 같으면 모든 규칙이 커널에 로드되었습니다. 출력이 다르면 규칙을 커널에 로드해야 합니다.
4. 다음 명령을 사용하여 규칙을 커널에 로드합니다:

```
# setrules
```

문제 3: 파일에 대한 액세스가 제대로 작동하지 않습니다. 해결 방법: 여러 개의 하드 링크가 이 파일을 가리키는 경우 구획 규칙 구성에 포함된 파일 액세스 규칙이 일치하지 않을 수 있습니다. 불일치를 확인하려면 다음 단계를 수행하십시오.

1. 다음 명령을 실행합니다.

   ```
   # vhardlinks
   ```

   출력에 불일치가 표시되면 2단계로 이동합니다.

2. 규칙을 수정하여 불일치를 제거합니다. 섹션 7.4에 설명된 절차를 수행하십시오.

문제 4: 네트워크 서버 규칙이 getrules 출력에 나타나지 않습니다. 해결 방법: 규칙이 내부적으로 관리되는 방식 때문에 지정된 구획에 대한 네트워크 서버 규칙이 `getrules` 명령의 대상 구획 출력에 나열될 수 있습니다.

예를 들면 다음과 같습니다.

```
/* telnet compartment rule to allow incoming telnet requests through compartment labeled ifacelan0
*/
grant server tcp port 23 ifacelan0
```

이 규칙을 지정하면 `getrules`의 `ifacelan0` 구획 출력 아래에 나열됩니다.

```
ACCESS              PROTOCOL     SRCPORT     DESPORT      DESCMPT
Grant client        tcp          0           23           telnet
```

7.9 discover 모드를 사용하여 초기 구획 구성 생성

discover 키워드를 사용하여 구획 정의에 태그를 지정할 수 있습니다. 자세한 내용은 섹션 7.6.1을 참조하십시오. `discover` 키워드를 사용하면 시스템에서 응용 프로그램이 제대로 작동하는 데 필요한 모든 규칙을 찾습니다. 이 기능은 테스트 환경에서만 사용해야 합니다.

`discover` 모드를 사용하려면 구획을 `discover`로 표시하고 일반적인 방식으로 응용 프로그램을 실행합니다. 시스템에서 모든 리소스 액세스를 식별하고 필요한 규칙을 만듭니다.

응용 프로그램의 초기 실행 후에 `getrules -m compartment_name` 명령을 사용하여 시스템에서 읽을 수 있는 규칙 버전을 생성합니다.

응용 프로그램이 테스트 환경에서 성공적으로 실행되려면 시스템에서 생성된 규칙이 필요하지만 이러한 규칙을 일반화해야 할 수도 있습니다. 예를 들어, 시스템에서 응용 프로그램이 아니라 커널이 포트 번호를 선택하는 익명 포트 범위의 포트 번호와 관련된 규칙을 생성할 수 있습니다. 응용 프로그램을 다시 실행할 때 다른 포트 번호가 사용되면 다른 규칙이 필요합니다. 이 경우 모든 포트나 적어도 익명 포트 범위의 포트 번호가 지정되도록 규칙을 일반화해야 합니다.

7.10 HP Serviceguard 클러스터의 구획

HP Serviceguard에서 구획을 사용하는 경우 모든 Serviceguard 데몬을 기본 `INIT` 구획에 구성해야 합니다. 그러나 Serviceguard 패키지는 다른 구획에 구성할 수 있습니다. Serviceguard에 필요한 데몬 및 ServiceGuard Extension for RAC에 대한 자세한 내용은 **Serviceguard 관리** 및 **Serviceguard Extension for RAC 사용 설명서**의 최신 버전을 참조하십시오.

Serviceguard 패키지는 특정 구획에 속할 수 있습니다. Serviceguard 패키지의 일부로 모니터링된 응용 프로그램도 특정 구획에 구성할 수 있습니다. 패키지에 대해 구획을 설정하는 경우 해당 구획이 패키지에 필요한 리소스(예: 볼륨 그룹, 파일 시스템, 네트워크 주소 등)에 액세스할 수 있는지 확인합니다. 구획 규칙은 노드에 특정하며 Serviceguard 장애 조치 작업 중에 적용되

지 않습니다. 장애 조치 후에 제대로 작동하려면 클러스터에 있는 모든 노드의 구획 구성이 같아야 합니다.

기본 LAN 인터페이스가 대기 LAN 인터페이스로 장애 조치될 때 대기 인터페이스가 온라인 상태가 아니면 기본 인터페이스의 구획 레이블이 자동으로 대기 인터페이스로 복사됩니다. 대기 인터페이스가 이미 온라인에 구성되어 있는 경우 장애 조치가 성공적으로 수행되려면 대기 인터페이스와 기본 인터페이스가 같은 구획에 구성되어 있어야 합니다. 대기 인터페이스가 기본 인터페이스와 다른 구획에 구성되어 있지만 장애 조치 시 오프라인 상태이면 인터페이스가 장애 조치될 때 대기 인터페이스가 기본 인터페이스 구획 구성으로 업데이트됩니다.

HP Serviceguard 노드나 패키지에 구획을 배포할 때 올바른 Serviceguard 작업을 유지하려면 다음을 수행합니다.

- 어떤 방식으로든 `INIT` 구획 지정을 수정하지 마십시오.
- `inetd`가 `INIT` 구획에서 실행되는지 확인합니다.
- 클러스터에 있는 모든 Serviceguard 데몬이 INIT 구획에서 실행되는지 확인합니다. 예를 들어, Serviceguard 버전 A.11.16의 데몬에는 `cmclconfd`, `cmcld`, `cmlogd`, `cmlvmd`, `cmomd` 및 `cmsnmpd`가 있습니다. 모든 Serviceguard 데몬 목록은 **Serviceguard 관리**를 참조하십시오.
- Serviceguard Extensions for RAC 클러스터에 대해 모든 Serviceguard 클러스터 요구 사항을 충족하는지 확인합니다. 또한 Serviceguard Extension for RAC 버전 A.11.16을 포함하는 클러스터가 `INIT` 구획에서 실행되려면 `cmsmgd` 데몬이 필요합니다. RAC 프로세스가 `libnmapi2` 라이브러리에 액세스할 수 있어야 하고 `cmsmgd`와 통신해야 합니다. 필수 데몬과 라이브러리에 대한 자세한 내용은 **ServiceGuard Extension for RAC 사용 설명서**를 참조하십시오.
- 대기 LAN 인터페이스를 구획에 구성하지 마십시오.
- 클러스터의 모든 노드에서 구획과 규칙을 동일하게 설정합니다. 구획 및 규칙은 시스템과 관련이 있고 시스템이 장애 조치될 때 전달되지 않습니다.

참고: 대기 인터페이스가 구획에 구성되어 있으면 기본 인터페이스에서 성공적으로 전환된 경우에도 `setrules` 명령을 실행할 때 이 구획이 대기 인터페이스에 적용됩니다. 구성된 대기 인터페이스 구획이 기본 인터페이스 구획과 일치하지 않으면 `setrules`를 실행할 때 기본 인터페이스 구획을 덮어씁니다. 이 경우 보안 위반이 발생할 수 있습니다.

구획, Fine-grained 권한 또는 RBAC 사용을 지원하기 위해 Serviceguard 스크립트에서 변경된 사항은 없습니다.

8 Fine-grained 권한

이 장에서는 HP-UX 11i의 Fine-grained 권한에 대해 설명합니다. 이 장의 내용은 다음과 같습니다.

- 개요(섹션 8.1)
- Fine-grained 권한 구성 요소(섹션 8.2)
- 사용 가능한 권한(섹션 8.3)
- Fine-grained 권한을 사용하여 응용 프로그램 구성(섹션 8.4)
- Fine-grained 권한의 보안 관련 사항(섹션 8.5)
- HP Serviceguard 클러스터의 Fine-grained 권한(섹션 8.6)
- Fine-grained 권한 문제 해결(섹션 8.7)

8.1 개요

UNIX 운영 체제는 일반적으로 "모두 또는 없음" 권한 모델을 사용합니다. 이 경우 수퍼유저(root 사용자와 같이 유효 `UID=0`인 사용자)에게는 거의 무제한 권한이 있고 다른 사용자에게는 특별한 권한이 거의 또는 전혀 없습니다.

HP-UX는 제한된 *smh*(1M), *privgrp*(4)에서 설명하는 권한 그룹, *shutdown*(1M)에서 설명하는 `shutdown.allow` 파일 및 *crontab*(1)에서 설명하는 `cron.allow` 파일을 포함하여 제한된 기능을 위임하는 기존의 여러 가지 방법을 제공합니다.

이러한 기존 방법은 Fine-grained 권한과 HP-UX RBAC 액세스 제어 프레임워크를 사용하여 대체할 수 있습니다.

HP-UX Fine-grained 권한 모델은 수퍼유저의 기능을 권한 집합으로 분할합니다. Fine-grained 권한은 프로세스에 부여됩니다. 각 권한은 해당 권한을 소유한 프로세스에 커널에서 제공하는 제한된 특정 서비스 집합에 대한 권한을 부여합니다.

자세한 내용은 *privileges*(5)를 참조하십시오.

8.2 Fine-grained 권한 구성 요소

HP-UX 11i의 Fine-grained 권한 기능에는 구성 파일, 명령 및 맨페이지가 포함되어 있습니다. 이러한 구성 요소를 사용하여 Fine-grained 권한을 구성하고 관리할 수 있습니다.

8.2.1 명령

표 8-1에서는 Fine-grained 권한 명령에 대해 간략하게 설명합니다.

표 8-1 Fine-grained 권한 명령

명령	설명
`setfilexsec`	바이너리 파일의 보안 속성을 설정합니다. 속성에는 유지되는 권한, 허용되는 권한, 구획, 권한 시작 플래그 등이 있습니다.
`getfilexsec`	바이너리 실행 파일과 연관된 보안 속성을 표시합니다. 속성에는 유지되는 권한, 허용되는 권한, 구획, 보안 속성 플래그 등이 있습니다.
`getprocxsec`	실행 중인 프로세스와 연관된 보안 속성을 표시합니다. 속성에는 유효 권한 집합, 유지되는 권한 집합, 허용되는 권한 집합, `euid`, 구획 이름 등이 있습니다.

8.2.2 맨페이지

표 8-2에서는 Fine-grained 권한 맨페이지에 대해 간략하게 설명합니다.

표 8-2 Fine-grained 권한 맨페이지

맨페이지	설명
privileges(5)	HP-UX 권한의 개요입니다.
privileges(3)	Fine-grained 권한 인터페이스에 대해 설명합니다.
setfilexsec(1M)	`setfilexsec` 기능과 구문에 대해 설명합니다.
getfilexsec(1M)	`getfilexsec` 기능과 구문에 대해 설명합니다.
getprocxsec(1M)	`getprocxsec` 기능과 구문에 대해 설명합니다.

8.3 사용 가능한 권한

Fine-grained 권한은 주로 개발자를 대상으로 합니다. 그러나 관리자도 이러한 응용 프로그램이 작동하는 방식을 이해하고 권한이 없는 응용 프로그램이 권한을 얻었는지 확인하기 위해 권한에 대해 알아둘 필요가 있습니다.

표 8-3에서는 권한과 그 용도를 보여 줍니다.

표 8-3 사용 가능한 권한

권한	설명
`PRIV_ACCOUNTING`	프로세스가 프로세스 계정 시스템을 제어할 수 있게 합니다.
`PRIV_AUDCONTROL`	프로세스가 감사 시스템을 시작, 수정 및 중지할 수 있게 합니다.
`PRIV_CHANGECMPT`	프로세스에 구획 변경 권한을 부여합니다.
`PRIV_CHANGEFILEXSEC`	프로세스가 바이너리에 권한을 부여할 수 있게 합니다.
`PRIV_CHOWN`	프로세스가 `chown()` 시스템 호출에 액세스할 수 있게 합니다.
`PRIV_CHROOT`	프로세스가 root 디렉토리를 변경할 수 있게 합니다.
`PRIV_CHSUBJIDENT`	프로세스가 UID, GID 및 그룹 목록을 변경할 수 있게 합니다. 또한 `chown()` 시스템 호출을 사용할 때 프로세스가 파일에 설정된 `suid` 또는 `sgid` 비트를 그대로 둘 수 있게 합니다.

표 8-3 사용 가능한 권한 (계속)

권한	설명
PRIV_CMPTREAD	프로세스가 달리 이러한 작업을 허용하지 않는 구획 규칙을 건너뛰고 읽거나 실행 또는 검색할 파일이나 디렉토리를 열 수 있게 합니다.
PRIV_CMPTWRITE	프로세스가 달리 이러한 작업을 허용하지 않는 구획 규칙을 건너뛰고 파일이나 디렉토리에 쓸 수 있게 합니다.
PRIV_COMMALLOWED	프로세스가 IPC 및 네트워킹 하위 시스템의 구획 규칙을 재정의할 수 있게 합니다.
PRIV_DACREAD	프로세스가 모든 임의 읽기, 실행 및 검색 액세스 제한을 재정의할 수 있게 합니다.
PRIV_DACWRITE	프로세스가 모든 임의 쓰기 권한 제한을 재정의할 수 있게 합니다.
PRIV_DEVOPS	프로세스가 테이프 또는 디스크 포맷과 같은 장치별 관리 작업을 수행할 수 있게 합니다.
PRIV_DLKM	프로세스가 커널 모듈을 로드하고, 로드된 커널 모듈에 대한 정보를 얻고, 동적으로 로드할 수 있는 커널 모듈의 전역 검색 경로를 변경할 수 있게 합니다.
PRIV_FSINTEGRITY	프로세스가 디스크 파티션의 크기나 경계 제거 또는 수정과 같은 디스크 작업을 수행하거나 전체 시스템에서 LVM 볼륨 그룹을 내보내고 가져올 수 있게 합니다.
PRIV_LIMIT	프로세스가 최대 제한 값을 초과하여 리소스 및 우선 순위 제한을 설정할 수 있게 합니다.
PRIV_LOCKRDONLY	프로세스가 lockf() 시스템 호출을 사용하여 읽기 전용 권한으로 열린 파일을 잠글 수 있게 합니다.
PRIV_MKNOD	프로세스가 mknod() 시스템 호출을 사용하여 문자 또는 블록 특수 파일을 만들 수 있게 합니다.
PRIV_MLOCK	프로세스가 plock 시스템 호출에 액세스할 수 있게 합니다.
PRIV_MOUNT	프로세스가 mount() 및 umount() 시스템 호출을 사용하여 파일 시스템을 마운트 및 마운트 해제할 수 있게 합니다.
PRIV_MPCTL	프로세스가 프로세서 바인딩, 로컬 도메인 바인딩 또는 시작 정책을 변경할 수 있게 합니다.
PRIV_NETADMIN	프로세스가 네트워크 라우팅 테이블 구성 및 인터페이스 정보 쿼리를 비롯한 네트워크 관리 작업을 수행할 수 있게 합니다.
PRIV_NETPRIVPORT	프로세스가 권한이 부여된 포트에 바인딩할 수 있게 합니다. 기본적으로 포트 번호 0-1023은 권한이 부여된 포트입니다.
PRIV_NETPROMISCUOUS	프로세스가 불규칙한 모드로 수신하도록 인터페이스를 구성할 수 있게 합니다.
PRIV_NETRAWACCESS	프로세스가 원시 인터넷 네트워크 프로토콜에 액세스할 수 있게 합니다.
PRIV_OBJSUID	프로세스에 OWNER 권한이 있는 경우 프로세스가 임의 파일에 suid 또는 sgid 비트를 설정할 수 있게 합니다. 또한 프로세스가 파일의 소유권을 변경할 수 있는 경우 suid 또는 sgid 비트를 지우지 않고 파일의 소유권을 변경할 수 있게 합니다.
PRIV_OWNER	프로세스가 파일이나 리소스의 소유자와 일치하는 UID와 관련해서 모든 제한을 재정의할 수 있게 합니다.

표 8-3 사용 가능한 권한 (계속)

권한	설명
PRIV_PSET	프로세스가 시스템 pset 구성을 변경할 수 있게 합니다.
PRIV_REBOOT	프로세스가 다시 부팅하는 작업을 수행할 수 있게 합니다.
PRIV_RTPRIO	프로세스가 rtprio() 시스템 호출에 액세스할 수 있게 합니다.
PRIV_RTPSET	프로세스가 RTE psets를 제어할 수 있게 합니다.
PRIV_RTSCHED	프로세스가 POSIX.4 실시간 우선 순위를 설정할 수 있게 합니다.
PRIV_RULESCONFIG	프로세스가 시스템에 구획 규칙을 추가하고 수정할 수 있게 합니다.
PRIV_SELFAUDIT	프로세스가 audwrite() 시스템 호출을 사용하여 자체적으로 감사 기록을 생성할 수 있게 합니다.
PRIV_SERIALIZE	프로세스가 serialize() 시스템 호출을 사용하여 직렬화 표시된 다른 프로세스와 함께 대상 프로세스의 순차적 실행을 강제할 수 있게 합니다.
PRIV_SPUCTL	프로세스가 Instant Capacity 제품의 특정 관리 작업을 수행할 수 있게 합니다.
PRIV_SYSATTR	프로세스가 튜너블 설정과 호스트 이름, 도메인 이름 및 사용자 할당량 수정을 포함하여 시스템 속성을 관리할 수 있게 합니다.
PRIV_SYSNFS	프로세스가 파일 시스템 내보내기, getfh() 시스템 호출, NFS 파일 잠금, NFS 인증 해지, NFS 커널 데몬 스레드 만들기 등의 NFS 작업을 수행할 수 있게 합니다.
PRIV_TRIALMODE	프로세스가 평가 모드 정보를 syslog 파일에 기록할 수 있게 합니다.

8.4 Fine-grained 권한을 사용하여 응용 프로그램 구성

Fine-grained 권한을 지원하도록 작성 또는 수정된 응용 프로그램을 권한 인식 응용 프로그램이라고 합니다. setfilexsec 명령을 사용하여 권한 인식 응용 프로그램을 등록해야 합니다. 등록하면 바이너리 파일과 연관된 보안 속성이 구성 파일에 저장되고 다시 부팅해도 지속성이 유지됩니다. 이 작업은 일반적으로 SD-UX 유틸리티를 사용하여 권한 인식 응용 프로그램을 설치하고 구성할 때 자동으로 수행됩니다.

이전 HP-UX 응용 프로그램이나 기존 응용 프로그램은 권한을 인식하지 않습니다. UID=0으로 실행되는 기존 응용 프로그램이 Fine-grained 권한을 사용하여 실행되도록 구성할 수 있습니다. HP-UX RBAC를 사용하여 기존 응용 프로그램을 구성하려면 섹션 9.5.4를 참조하십시오.

팁: HP-UX RBAC를 사용하여 실행 시 다양한 권한이 필요한 응용 프로그램을 구성하는 것이 좋습니다.

권한 인식 응용 프로그램의 보안 속성을 구성하려면 다음과 같이 setfilexsec 명령을 사용합니다.

```
# setfilexsec [options] filename
```

setfilexsec 명령은 로컬 파일 시스템의 바이너리에 권한을 할당합니다. NFS 서버에서 직접 다른 시스템이 파일을 수정하는 경우 setfilexsec에 의해 설정된 확장 속성이 제거되지 않으므로 NFS(네트워크 파일 시스템)에서 얻은 바이너리에는 권한을 할당하면 안 됩니다.

setfilexsec의 옵션은 다음과 같습니다.

-d 구성 파일과 커널에서 이 파일의 보안 정보를 삭제합니다.

-D 구성 파일에서만 이 파일의 보안 정보를 삭제합니다. 삭제한 파일의 보안 정보를 지우는 데 사용됩니다.

-r 유지되는 최소 권한을 추가하거나 변경합니다.

-R 유지되는 최대 권한을 추가하거나 변경합니다.

-p 허용되는 최소 권한을 추가하거나 변경합니다.

-P 허용되는 최대 권한을 추가하거나 변경합니다.

-f 보안 속성 플래그를 설정합니다.

getfilexsec 명령은 setfilexsec 명령을 사용하여 설정된 바이너리 파일의 확장 속성을 표시합니다.

```
# getfilexsec filename
```

8.4.1 권한 모델

각 프로세스에는 다음 세 가지 권한 집합이 연관되어 있습니다.

- 허용되는 권한 집합

  프로세스가 발생시킬 수 있는 최대 권한 집합입니다. 프로세스는 이 집합에서 모든 권한을 삭제할 수 있지만 이 집합에 권한을 추가할 수는 없습니다. 이 집합의 권한을 프로세스의 유효 권한 집합에 추가할 수 있습니다.

- 유효 권한 집합

  프로세스에 대해 현재 활성화된 권한 집합입니다. 권한 인식 프로세스는 필요한 권한만 이 집합에 유지되도록 언제든지 유효 권한 집합을 수정할 수 있습니다. 프로세스는 유효 권한 집합에서 모든 권한을 제거할 수 있지만 허용되는 권한 집합의 권한만 추가할 수 있습니다.

  유효 권한 집합은 항상 허용되는 권한 집합의 하위 집합입니다.

- 유지되는 권한 집합

  프로세스가 **execve()** 시스템 호출을 수행할 때 유지되는 권한 집합입니다. 프로세스는 이 집합에서 모든 권한을 제거할 수 있지만 이 집합에 권한을 추가할 수는 없습니다.

  유지되는 권한 집합은 항상 허용되는 권한 집합의 하위 집합입니다.

첫 번째 프로세스 init는 작은 권한 집합으로 시작됩니다. 그런 다음 exec 계열 호출(execv, execve 등)을 사용하여 다른 바이너리를 실행하는 다른 프로세스를 만듭니다. 이 exec 호출 중에 바이너리의 확장 속성인, setfilexsec 명령을 사용하여 설정된 속성으로 인해 이러한 프로세스는 부모 프로세스에 없는 권한을 얻거나 부모 프로세스에 있는 권한을 잃을 수 있습니다. 예를 들어, 바이너리에 DACREAD의 허용되는 최소값이 있는 경우(바이너리에 setfilexsec -p DACREAD가 수행된 경우) 새 프로세스는 부모 프로세스에 해당 권한이 있는지 여부에 관계없이 DACREAD 권한을 갖게 됩니다. 반면 프로세스에 이미 DACREAD 권한이 있지만 프로세스가 실행하는 바이너리의 허용되는 최대값에 이 권한이 없는 경우(예: 파일에서 이미 setfilexsec -P none ….가 수행된 경우) 바이너리 실행 결과로 해당 권한을 잃게 됩니다.

8.4.2 복합 권한

복합 권한은 미리 정의된 단순 권한 집합을 지정하는 간단한 방법입니다.

복합 권한은 다음과 같습니다.

- `BASIC`

  기본적으로 모든 프로세스가 사용할 수 있는 기본 권한입니다. 프로세스는 이 집합에서 권한을 하나 이상 삭제할 수 있습니다.

- `BASICROOT`

  기본 권한 및 일반적으로 UID=0과 연관된 기능을 제공하는 권한입니다.

- `POLICY`

  정책 재정의 권한과 정책 구성 권한입니다. 정책 재정의 권한은 구획 규칙을 재정의합니다. 정책 구성 권한은 권한 구성 방법을 제어합니다.

각 복합 권한에 있는 모든 권한 목록은 *privileges*(5)를 참조하십시오.

8.5 Fine-grained 권한의 보안 관련 사항

Fine-grained 권한은 분산 시스템에서 전파되지 않고 로컬 시스템에만 적용됩니다. 예를 들어, `PRIV_DACREAD` 및 `PRIV_DACWRITE` 권한을 가진 시스템 프로세스는 파일 읽기나 쓰기에 대한 다른 시스템의 임의 제한을 재정의할 수 없습니다.

8.5.1 권한 에스컬레이션

경우에 따라 프로세스에 특정 권한이나 권한 집합을 부여할 때 이 프로세스가 명시적으로 부여되지 않은 추가 권한을 얻을 수 있습니다. 이를 권한 에스컬레이션이라고 합니다. 예를 들어, `PRIV_DACWRITE` 권한을 가진 프로세스는 중요한 운영 체제 파일을 덮어쓸 수 있으며, 해당 프로세스에 Fine-grained 권한을 추가로 부여할 수 있습니다.

8.6 HP Serviceguard 클러스터의 Fine-grained 권한

HP Serviceguard에서 권한 인식 응용 프로그램을 모니터링할 수 있습니다. Fine-grained 권한을 지원하기 위해 Serviceguard 패키지 구성 파일이나 Serviceguard 패키지 관리에서 변경된 사항은 없습니다. Fine-grained 권한 사용을 지원하기 위해 Serviceguard 스크립트에서 변경된 사항은 없습니다.

Serviceguard 노드나 패키지에 HP-UX 11i Fine-grained 권한을 배포할 때 올바른 Serviceguard 작업을 유지하려면 다음을 수행합니다.

- `INIT` 구획에서 root(`UID=0`)에 모든 권한이 있는지 확인합니다.
- Fine-grained 권한 구현으로 인해 Serviceguard 클러스터에서 보안 위험이 발생하지 않는지 확인합니다.

8.7 Fine-grained 권한 문제 해결

시스템 일부가 작동하지 않고 이 문제가 Fine-grained 권한으로 인해 발생하는 것 같으면 다음과 같이 Fine-grained 권한 구성을 확인할 수 있습니다.

문제 1: 바이너리 파일에 Fine-grained 권한이 할당된 경우에도 exec()를 사용하여 바이너리에 액세스하는 프로세스는 할당된 Fine-grained 권한을 받을 수 없습니다. 해결 방법: 다음 경우 중 하나를 확인합니다.

- 해당 파일이 스크립트입니까?

  셸 스크립트에 할당된 Fine-grained 권한은 모두 무시됩니다.

- Fine-grained 권한이 할당된 이후 파일이 변경되었습니까?

  파일을 수정하면 해당 Fine-grained 권한 속성이 손실됩니다. 파일을 수정하기 전이나 후에 다음 명령을 실행합니다.

  ```
  # setfilexsec -d filename
  ```

  그런 다음 파일에 할당할 권한 속성을 추가합니다.

Fine-grained 권한 문제 해결에 대한 자세한 내용은 *setfilexsec*(1M)를 참조하십시오.

문제 2: 프로세스가 있으면 안 되는 권한이 있거나 있어야 하는 권한이 없습니다. 해결 방법: 다음 `getprocxsec` 명령을 사용하여 프로세스에 있는 권한을 확인합니다.

```
# getprocxsec -per pid
```

이 명령은 프로세스에 대해 허용되는 권한 집합, 유효 권한 집합 및 유지되는 권한 집합을 표시합니다. 자세한 내용은 *getprocxsec*(1M)를 참조하십시오.

프로세스에 올바른 권한이 없는 경우 이 프로세스를 만든 바이너리 파일을 올바른 권한으로 구성합니다. 자세한 내용은 "Fine-grained 권한을 사용하여 응용 프로그램 구성"을 참조하십시오.

III부 ID 보호

오늘날 글로벌 엔터프라이즈 기업에서는 ID를 관리하는 작업이 쉽지 않습니다. 특히 다양한 개인 정보 보호 법규 및 규정에 여러 국가의 직원, 계약자, 파트너 및 공급자가 포함되도록 ID 관리 요구 사항이 증가합니다. HP-UX 11i는 권한이 부여된 모든 수행 작업을 감사하는 동안 사용자 인증 및 액세스 관리를 단순화합니다.

이 절의 내용은 다음과 같습니다.

- HP-UX Role-Based Access Control(9장)
- 감사 관리(10장)

9 HP-UX Role-Based Access Control

이 장에서는 HP-UX RBAC(HP-UX Role-Based Access Control)에 대해 설명합니다. 이 장의 내용은 다음과 같습니다.

- 개요(섹션 9.1)
- 액세스 제어 기본 사항(섹션 9.2)
- HP-UX RBAC 구성 요소(섹션 9.3)
- HP-UX RBAC 배포 계획(섹션 9.4)
- HP-UX RBAC 구성(섹션 9.5)
- HP-UX RBAC 사용(섹션 9.6)
- HP-UX RBAC 문제 해결(섹션 9.7)

9.1 개요

보안, 특히 플랫폼 보안은 항상 엔터프라이즈 인프라에 중요한 문제였습니다. 그렇지만 이전에는 많은 조직이 이러한 보안 개념을 개인의 책임과 최소 권한으로 무시하거나 간과한 경우가 많았습니다. 그러나 HIPAA(Health Insurance Portability and Accountability Act) 및 Sarbanes-Oxley Act를 포함하여 최근에 미국에서 제정된 법률은 이러한 보안 개념의 중요성을 강조하는 데 도움이 되었습니다.

대부분의 엔터프라이즈 환경에서는 여러 사용자가 시스템을 관리합니다. 이를 위해 대체로 관리자에게 root라고 알려진 일반적인 공유 계정에 대한 암호를 제공합니다. root 계정은 root 암호를 가진 관리자가 모든 작업을 수행할 수 있게 하여 액세스 제어 관리를 단순화하는 동시에 액세스 제어 관리에 대한 몇 가지 내재된 문제를 제공합니다. 예를 들면 다음과 같습니다.

- 관리자 사용자에게 root 암호를 제공한 후에는 이러한 사용자를 쉽게 제한할 수 없습니다.
- 최상의 경우에서도 관리자 한 명의 액세스를 해지하려면 공통 암호를 변경하고 다른 관리자에게 알려야 합니다. 보다 현실적으로, 대체 액세스 메커니즘이 이미 구현되었을 수 있으므로 단순히 암호를 변경하는 것만으로는 효과적으로 액세스 권한을 해지할 수 없습니다.
- 공유 root 계정을 사용할 경우 개인의 책임을 실현하기가 거의 불가능합니다. 따라서 보안 이벤트가 발생한 후 제대로 분석하기 어려울 뿐만 아니라 불가능한 경우도 있습니다.

HP-UX RBAC(Role-Based Access Control) 기능은 일반적이지만 적절하게 구성된 사용자 계정에 일련의 작업을 할당하는 기능을 제공하여 이러한 문제를 해결합니다. 또한 HP-UX RBAC는 사용자 단위로 개별 권한 부여를 할당하고 해지하는 관리 오버헤드를 줄입니다.

HP-UX RBAC는 다음 기능을 제공합니다.

- HP-UX와 관련해서 미리 정의된 구성 파일을 통해 빠르고 쉬운 배포 가능
- PAM(Plugable Authentication Module)을 통해 융통성 있게 다시 인증하여 명령별 제한 허용
- HP-UX 감사 시스템과 통합되어 하나의 통합된 감사 추적 생성
- 플러그형 아키텍처를 통해 액세스 제어 결정의 사용자 정의 가능

9.2 액세스 제어 기본 사항

액세스 제어 시스템은 일련의 제약 조건을 기반으로 리소스에 대한 액세스를 제한하는 것을 목적으로 합니다. 일반적으로 이러한 제약 조건 및 연관된 속성은 다음 범주로 나뉩니다.

- 주체: 리소스에 액세스하려는 엔터티. 운영 체제 컨텍스트에서 주체는 일반적으로 사용자나 사용자와 연관된 프로세스입니다.
- 작업: 리소스에 대해 수행되는 작업. 작업은 직접 응용 프로그램이나 명령에 해당할 수 있습니다. HP-UX RBAC의 경우 작업은 점으로 구분된 계층 구조 문자열(예: `hpux.user.add`)입니다.
- 객체: 작업의 대상. 최종 리소스와 같은 경우가 많지만 다를 수도 있습니다.

액세스 제어 요청은 이러한 요소를 결합한 질문으로 간주될 수 있습니다. 여기서 질문에 대한 응답(일반적으로 허용 또는 거부)에 따라 리소스에 대한 액세스를 허용할지 여부가 결정됩니다. 예를 들면 다음과 같습니다.

사용자 `ron`은 객체 `/dev/dsk/c0t1d0`에 대해 작업 `hpux.fs.mount`를 수행할 권한이 있습니까?

권한 부여란 용어는 액세스 제어의 동의어로 사용되는 경우가 많습니다. HP-UX RBAC에서 권한 부여는 객체에 대해 작업을 수행할 수 있는 권한을 나타냅니다. 표 9-1과 같이 사용자는 각각 하나의 리소스에 대한 액세스를 허용하는 일련의 권한 부여를 가질 수 있습니다.

표 9-1 사용자별 권한 부여 예제

권한 부여의 작업 구성 요소	사용자			
	ron	lisa	jim	liz
`hpux.user.add`				
`hpux.user.delete`				
`hpux.user.modify`				
`hpux.user.password.modify`	•	•	•	•
`hpux.network.nfs.start`	•			
`hpux.network.nfs.stop`	•			
`hpux.network.nfs.config`	•			
`hpux.fs.backup`	•	•		
`hpux.fs.restore`	•	•		

참고: 표 9-1에서는 권한 부여의 작업 요소만 보여 주고 권한 부여의 객체 요소는 표시하지 않습니다.

9.2.1 역할을 사용하여 액세스 제어 단순화

앞의 개요에서 설명한 액세스 제어의 기본 개념 외에, 이 절에서는 액세스 제어 정책을 나타내는 방법과 결정 방법에 대해 다룹니다.

앞의 액세스 제어 개요에서는 액세스 제어 정책을 나타내는 방법과 결정 방법에 대해 다루지 않았습니다. 한 가지 방법은 단순히 사용자 목록과 각 사용자에게 할당된 권한 부여(작업, 객체 쌍)를 유지 관리하는 것입니다. 이 방법은 각 사용자의 권한 부여 집합이 다른 사용자와 완전히 다를 수 있으므로 융통성이 크다는 장점이 있습니다.

그러나 사용자를 추가할 때 각 사용자에게 필요한 권한 부여를 정확히 결정해야 하므로 이 방법은 관리가 어렵습니다. 또한 감사를 수행할 때 각 사용자를 개별적으로 검사하여 연관된 권한 부여를 확인해야 합니다.

HP-UX RBAC는 공통 권한 부여 요구를 가진 사용자를 역할로 그룹화하여 이러한 문제를 해결합니다. 역할은 권한 부여 할당과 감사를 단순화하는 그룹화 메커니즘으로 사용됩니다. 직접 사용자에게 권한 부여를 할당하는 대신 역할에 권한 부여를 할당합니다. 사용자를 시스템에 추가할 때 일련의 역할을 할당하면 이러한 역할에 따라 수행할 수 있는 작업과 액세스할 수 있는 리소스가 결정됩니다.

역할에 할당된 권한 부여를 나열하는 표 9-2를 각 사용자에게 할당된 권한 부여를 나열하는 표 9-1과 비교해 보십시오. 두 표를 비교하면 역할이 어떻게 권한 부여 할당을 단순화하는지 확인할 수 있습니다.

표 9-2 역할별 권한 부여 예제

권한 부여의 작업 구성 요소	역할			
	UserAdmin	NetworkAdmin	BackupOper	Admin
hpux.user.add	•			•
hpux.user.delete	•			•
hpux.user.modify	•			•
hpux.user.password.modify				•
hpux.network.nfs.start		•		•
hpux.network.nfs.stop		•		•
hpux.network.nfs.config		•		•
hpux.fs.backup			•	•
hpux.fs.restore			•	•

참고: 표 9-2에서는 권한 부여의 작업 요소만 보여 주고 권한 부여의 객체 요소는 표시하지 않습니다.

9.3 HP-UX RBAC 구성 요소

다음은 주요 HP-UX RBAC 구성 요소 목록입니다.

privrun 래퍼 명령

privrun은 사용자와 연관된 권한 부여를 기반으로 권한 부여를 확인하고 선택적으로 사용자를 다시 인증한 후, 응용 프로그램을 수정하지 않고 권한

	을 사용하여 기존 응용 프로그램을 호출합니다.
`privedit` 명령	`privedit`는 사용자와 연관된 권한 부여를 기반으로 파일 사용 권한이나 ACL(액세스 제어 목록)로 인해 사용자가 일반적으로 편집할 수 없는 파일을 편집할 수 있게 합니다.
ACPS(Access Control Policy Switch)	주체가 객체에 대해 작업을 수행할 권한이 있는지 확인합니다.
액세스 제어 정책 모듈	HP-UX RBAC 데이터베이스 파일을 평가하고 매핑 정책을 서비스 액세스 제어 요청에 적용합니다.
관리 명령	HP-UX RBAC 데이터베이스 파일을 편집하고 유효성을 검사합니다.

다음 절에서는 HP-UX RBAC 구성 요소에 대해 자세히 설명합니다.

9.3.1 HP-UX RBAC 액세스 제어 정책 전환

HP-UX RBAC 액세스 제어 정책 전환은 액세스 제어 결정을 내려야 하는 응용 프로그램과 RBAC 데이터베이스에 있는 정책 정보를 해석한 후 결정 응답을 제공하는 액세스 제어 정책 모듈 간의 사용자 정의 가능한 인터페이스입니다. 그림 9-1과 같이 HP-UX RBAC 아키텍처의 해당 위치에서 ACPS는 액세스 제어 정책 모듈과 액세스 제어 결정을 내리는 응용 프로그램 간의 인터페이스를 제공합니다.

ACPS에는 다음 인터페이스가 있으며, 이에 대해서는 해당 맨페이지에서 자세히 설명합니다.

- ACPS API(Application Programming Interface)
- ACPS SPI(Service Provider Interface)
- `/etc/acps.conf`

ACPS의 관리 인터페이스는 `/etc/acps.conf` 구성 파일입니다. `/etc/acps.conf` 구성 파일은 ACPS가 참조하는 정책 모듈, 모듈이 참조되는 시퀀스 및 액세스 제어 결정이 필요한 응용 프로그램에 결과를 제공하기 위해 모듈의 응답을 결합하는 규칙을 결정합니다. 이 ACPS 구현을 사용하여 기존의 역할 기반 액세스 제어 응용 프로그램을 수정하지 않고 사용자 정의 정책을 강제로 시행하는 모듈을 만들 수 있습니다.

참고: ACPS 및 해당 인터페이스에 대한 자세한 내용은 *acps*(4), *acps.conf*(4), *acps_api*(3) 및 *acps_spi*(3)를 참조하십시오.

9.3.2 HP-UX RBAC 구성 파일

표 9-3에서는 HP-UX RBAC 파일을 나열하고 간략하게 설명합니다.

표 9-3 HP-UX RBAC 구성 파일

구성 파일	설명
`/etc/rbac/auths`	유효한 모든 권한 부여가 들어 있는 데이터베이스 파일입니다.
`/etc/rbac/cmd_priv`	명령과 파일 권한 부여 및 권한이 들어 있는 `privrun` 데이터베이스 파일입니다.

표 9-3 HP-UX RBAC 구성 파일 (계속)

구성 파일	설명
/etc/rbac/role_auth	각 역할에 대한 권한 부여를 정의하는 데이터베이스 파일입니다.
/etc/rbac/roles	구성된 모든 역할을 정의하는 데이터베이스 파일입니다.
/etc/rbac/user_role	각 사용자에 대한 역할을 정의하는 데이터베이스 파일입니다.
/etc/acps.conf	ACPS에 대한 구성 파일입니다.
/etc/rbac/aud_filter	감사할 특정 HP-UX RBAC 역할, 작업 및 객체를 식별하는 감사 필터 파일입니다.

9.3.3 HP-UX RBAC 명령

표 9-4에서는 HP-UX RBAC 명령을 나열하고 간략하게 설명합니다.

표 9-4 HP-UX RBAC 명령

명령	설명
privrun	권한 부여를 확인하고 선택적으로 사용자를 다시 인증한 후 권한을 사용하여 기존 응용 프로그램을 호출합니다.
privedit	권한이 부여된 사용자가 액세스 제어를 받는 파일을 편집할 수 있게 합니다.
roleadm	/etc/rbac/user_role, /etc/rbac/role_auth 및 /etc/rbac/roles 파일에 있는 역할 정보를 편집합니다.
authadm	/etc/rbac/role_auth 및 /etc/rbac/roles 파일에 있는 권한 부여 정보를 편집합니다.
cmdprivadm	/etc/rbac/cmd_priv 데이터베이스에 있는 명령 권한 부여와 권한을 편집합니다.
rbacdbchk	HP-UX RBAC 및 privrun 데이터베이스 파일에 있는 권한 부여와 구문을 확인합니다.

9.3.4 HP-UX RBAC 맨페이지

표 9-5에서는 HP-UX RBAC 맨페이지를 나열하고 간략하게 설명합니다.

표 9-5 HP-UX RBAC 맨페이지

맨페이지	설명
rbac(5)	HP-UX RBAC 기능에 대해 설명합니다.
acps(3)	ACPS 및 해당 인터페이스에 대해 설명합니다.
acps.conf(4)	ACPS 구성 파일과 해당 구문에 대해 설명합니다.
acps_api(3)	ACPS 응용 프로그래밍 인터페이스에 대해 설명합니다.
acps_spi(3)	ACPS 서비스 제공자 인터페이스에 대해 설명합니다.
privrun(1m)	privrun 기능과 구문에 대해 설명합니다.
privedit(1m)	privedit 기능과 구문에 대해 설명합니다.
roleadm(1m)	roleadm 기능과 구문에 대해 설명합니다.
authadm(1m)	authadm 기능과 구문에 대해 설명합니다.

표 9-5 HP-UX RBAC 맨페이지 (계속)

맨페이지	설명
cmdprivadm(1m)	`cmdprivadm` 기능과 구문에 대해 설명합니다.
rbacdbchk(1m)	`rbacdbchk` 기능과 구문에 대해 설명합니다.

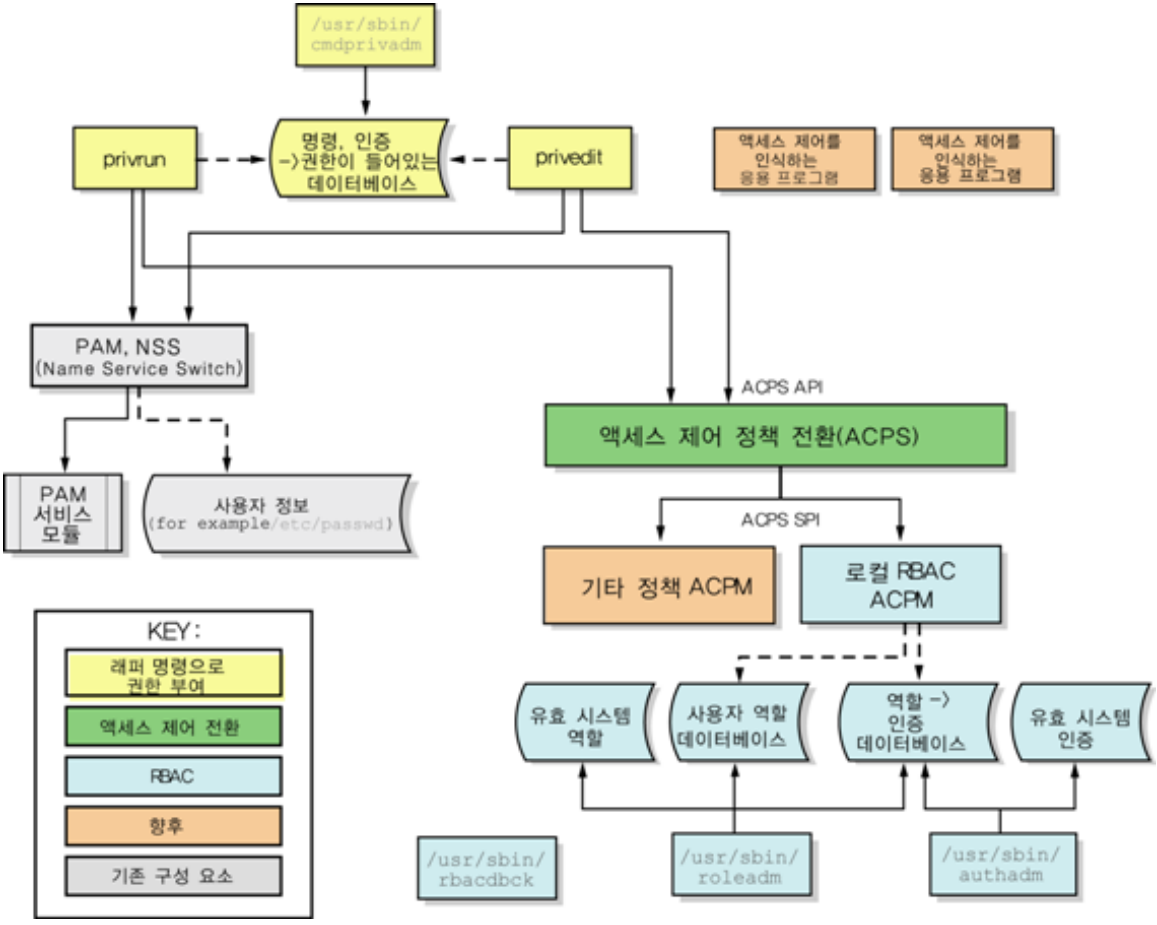
9.3.5 HP-UX RBAC 아키텍처

HP-UX RBAC의 주요 구성 요소는 기존 명령, 응용 프로그램 및 스크립트를 호출하는 `privrun` 명령입니다. `privrun` 명령은 ACPS 하위 시스템을 사용하여 액세스 제어 요청을 합니다. 액세스 요청은 사용자-역할 매핑과 역할-권한 부여 매핑을 정의하는 일련의 구성 파일을 기반으로 승인되거나 거부됩니다.

액세스 요청이 승인되면 `privrun`은 UID, GID, Fine-grained 권한 및 구획이 하나 이상 포함될 수 있는 추가 권한을 사용하여 대상 명령을 호출합니다. 권한은 대상 명령이 성공적으로 실행될 수 있도록 구성되어 있습니다.

그림 9-1에서는 HP-UX RBAC 아키텍처를 보여 줍니다.

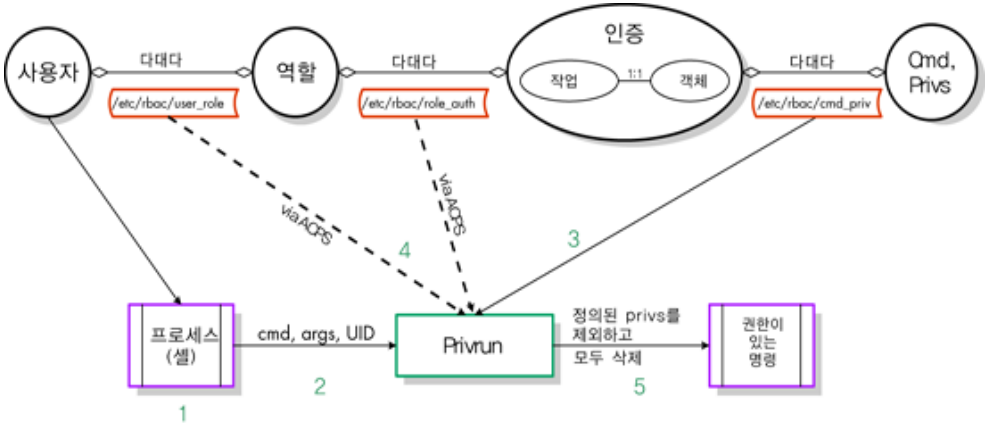
그림 9-1 HP-UX RBAC 아키텍처



9.3.6 HP-UX RBAC 예제 사용 및 작업

그림 9-2 및 이후 주석에서는 `privrun`의 샘플 호출과 `privrun`에서 사용자가 명령을 호출할 수 있는지 확인하는 데 사용하는 구성 파일을 보여 줍니다.

그림 9-2 privrun을 호출한 후 예제 작업



1. 사용자와 연관된 프로세스, 특히 셸은 높은 액세스 권한을 사용하여 대상 명령을 실행할 목적으로 `privrun`을 실행합니다.
2. 대상 명령줄(명령 및 인수)은 명시적으로 `privrun`에 전달되고, 호출하는 사용자의 UID는 프로세스 컨텍스트에 의해 암시적으로 전달됩니다.
3. `privrun`은 `/etc/rbac/cmd_priv` 데이터베이스에서 지정된 명령줄과 일치하는 항목(또는 항목 집합)을 찾습니다. 또한 일치하는 각 항목은 필요한 권한 부여(작업, 객체 쌍) 및 사용자에게 지정된 권한 부여가 있는 경우 결과 권한을 지정합니다.
4. `privrun`은 일치하는 각 `/etc/rbac/cmd_priv` 항목에 대해 ACPS를 호출합니다. ACPS의 HP-UX RBAC 백엔드는 `/etc/rbac/user_role` 및 `/etc/rbac/role_auth` 데이터베이스를 참조하여 사용자에게 지정된 권한 부여가 있는지 확인하고 이 결과를 다시 `privrun`에 전달합니다.
5. 프로세스와 연관된 사용자에게 요청된 명령과 관련해서 `/etc/rbac/cmd_priv` 데이터베이스에 지정된 필수 권한 부여가 있는 경우 `privrun`은 `/etc/rbac/cmd_priv` 항목에 지정된 권한을 제외하고 모든 권한을 삭제하고 요청된 명령을 실행합니다. `privrun` 명령이 `UID=0`으로 설정되고 필요한 모든 권한을 사용하여 시작됩니다.

9.4 HP-UX RBAC 배포 실행

HP-UX RBAC를 배포하기 전에 다음 계획 단계를 따릅니다.

1. 사용자의 역할을 계획합니다.
2. 역할의 권한 부여를 계획합니다.
3. 권한 부여-명령 매핑을 계획합니다.

다음 절에서는 이러한 단계에 대해 자세히 설명합니다.

9.4.1 역할 계획

시스템 사용자에 대해 적절한 역할 집합을 계획하는 것은 HP-UX RBAC 배포의 중요한 첫 단계입니다. 일부 엔터프라이즈에는 이러한 역할 집합이 이미 있으며 HP-UX RBAC를 구성할 때 재사용할 수 있습니다. 시스템의 관리자 사용자와 연관된 기존 작업을 기반으로 역할을 디자인해야 하는 경우가 보다 일반적입니다.

역할을 디자인할 때는 다음 지침을 고려합니다.

- 역할 수는 시스템의 총 사용자 수보다 훨씬 적어야 합니다. 각 사용자에게 특수 역할이 필요한 경우 역할 사용과 연관된 관리 편의성이 모두 사라집니다.
- 역할은 사용자의 실제 비즈니스 역할과 관련이 있어야 합니다.
- 사용자는 여러 역할을 가질 수 있으므로 단순히 여러 비즈니스 역할에 공통된 권한 부여를 그룹화하기 위해 일부 역할을 디자인할 수 있습니다. 이 방법을 사용하면 해당 권한 부여를 포함하여 계층 구조로 다른 역할이 포함될 역할을 역할을 디자인할 수 있습니다.

9.4.2 역할의 권한 부여 계획

역할을 정의한 후 각 역할과 연관된 권한 부여를 계획할 수 있습니다. 역할이 기존 작업 계층 구조와 일치하면 간단하게 권한 부여를 할당할 수 있습니다. 시스템 정의 권한 부여를 모두 나열하려면 다음 명령을 입력합니다.

```
# authadm list sys
```

기존 권한 부여 계층 구조가 역할과 일치하지 않으면 각 역할과 연관된 권한 부여를 정의하는 것이 더 복잡해집니다. 이러한 정의에 도움이 되도록 다음 단계를 사용할 수 있습니다.

1. 각 역할에서 자주 사용하는 시스템 명령을 표시합니다.
2. 이러한 명령을 /etc/rbac/cmd_priv 데이터베이스에 있는 명령과 비교합니다.
3. 이전 단계를 수행한 후 일치하는 항목을 찾은 경우 이러한 항목을 권한 부여 할당의 지침으로 사용합니다.

예를 들어, 원하는 역할 중 하나가 useradd, usermod, userdel 등의 명령을 자주 실행하는 UserOperator라고 가정합니다. 이 역할에 적합한 권한 부여를 확인하려면 다음 명령을 입력합니다.

```
 # grep useradd /etc/rbac/cmd_priv
/usr/sbin/useradd:dflt:(hpux.user.add,*):0/0//:dflt:dflt:dflt:
```

이 예제에서 /usr/sbin/useradd 명령에는 hpux.user.add 권한 부여가 필요합니다. 직접 이 권한 부여를 할당하거나 hpux.user.*를 권한 부여로 할당할 수 있습니다.

와일드카드를 사용하여 권한 부여를 할당할 때는 주의하십시오. 이 권한 부여를 할당하면 실제로 다음과 같은 여러 권한 부여가 할당됩니다.

```
# grep hpux.user. /etc/rbac/cmd_priv
/usr/sbin/pwgrd:dflt:(hpux.user.cache.admin,*):0/0// :dflt :dflt :dflt :
/usr/sbin/userdel:dflt:(hpux.user.delete,*):0/0// :dflt :dflt :dflt :
/usr/sbin/groupdel:dflt:(hpux.user.group.delete,*):0/0// :dflt :dflt :dflt :
/usr/sbin/useradd:dfl:(hpux.user.add,*):0/0//:dflt:dflt:dflt:
/usr/sbin/usermod:dflt:(hpux.user.modify,*):0/0// :dflt :dflt :dflt :
/usr/sbin/groupadd:dflt:(hpux.user.group.add,*):0/0// :dflt :dflt :dflt :
/usr/sbin/groupmod:dflt:(hpux.user.group.modify,*):0/0// :dflt :dflt :dflt :
/usr/sbin/vipw:dflt:(hpux.user.modify,*):0/0// :dflt :dflt :dflt :
```

9.4.3 명령 매핑 계획

정의된 임의의 역할에서 자주 사용하지만 제공된 미리 정의된 /etc/rbac/cmd_priv 파일에 없는 명령을 정의합니다. /etc/rbac/cmd_priv 파일은 권한 부여와 명령 간의 매핑을 정의합니다. 각 명령에 대해 다음을 결정합니다.

- 명령의 전체 경로

- 명령을 실행하기 전에 확인할 필수 권한 부여
- 명령에 필요한 특수 권한(예: `euid=0`)

`/etc/rbac/cmd_priv` 파일의 작업 및 객체 항목을 구성하는 텍스트 문자열은 임의로 지정할 수 있지만 논리상 하나의 명령이나 명령 집합에 해당해야 합니다. `/etc/rbac/cmd_priv`에서 권한 부여와 명령 간의 매핑을 계획할 때는 다음 지침을 고려합니다.

- 작업을 쉽게 역할에 할당하기 위해 작업을 논리 그룹으로 정의합니다.
- 하위 요소가 너무 많거나(10개 이상) 너무 적은(1) 작업 분기를 만들지 마십시오. 전체 트리가 지나치게 넓어 작업 그룹을 할당하기 어려워지거나 지나치게 길어 개별 작업 이름이 길어지고 사용하기 어려워지지 않도록 해야 합니다.
- 작업 이름의 마지막 요소는 작업(동사)으로 끝냅니다.
- 새 명령을 추가할 때 명확하게 배치될 수 있도록 작업을 정의합니다.

추가 명령을 구성하는 절차는 "추가 명령 권한 부여 및 권한 구성"을 참조하십시오.

9.4.4 HP-UX RBAC 제한 사항

다음은 HP-UX RBAC를 배포할 때 고려할 항목 목록입니다.

- HP-UX RBAC는 단일 사용자 모드를 지원하지 않으므로 단일 사용자 모드가 필요한 경우 root 계정을 사용할 수 있어야 합니다.
- Serviceguard는 HP-UX RBAC 및 `privrun`을 사용하여 Serviceguard 명령에 대한 액세스 권한을 부여하는 기능을 지원하지 않습니다. HP-UX RBAC 및 Serviceguard 클러스터에 대한 자세한 내용은 섹션 9.6.1.1을 참조하십시오.
- 모든 응용 프로그램과 마찬가지로 HP-UX RBAC에는 구획을 제어하는 규칙이 적용됩니다 (7장 참조). 구획과 함께 HP-UX RBAC를 사용할 때는 다음에 주의하십시오.
  - 구획 정의에 의해 제한되는 파일에서는 `privedit`를 실행할 수 없습니다.
  - 다른 응용 프로그램에 Fine-grained 권한을 제공하려면 응용 프로그램에 제공하려는 권한과 동일한 권한을 사용하여 `privrun` 명령이 실행되고 있어야 합니다. 기본적으로 `privrun`은 모든 권한을 사용하여 실행되도록 구성되어 있습니다(자세한 내용은 getfilexsec(1M) 참조). 그러나 때때로 이 기본 권한 집합이 제한될 수도 있습니다. 예를 들어, 구획이 권한을 허용하지 않도록 구성된 경우 `privrun` 자체에 권한이 없으므로 이 지정으로 인해 `privrun`이 응용 프로그램에 권한을 제공할 수 없습니다. 기본적으로 밀봉된 구획은 `POLICY` 복합 권한을 허용하지 않도록 구성되어 있습니다.
  - `privrun`이 구획에 있는 다른 응용 프로그램을 호출하려면 `privrun`에서 `CHANGECMPT` 권한을 어설션해야 합니다. `privrun`에서 `CHANGECMPT` 권한을 어설션할 수 없는 경우, 예를 들어 구획이 권한을 허용하지 않도록 구성된 경우에는 `privrun`이 실패합니다. 이 동작은 의도적인 것이며 밀봉된 구획의 개념을 보충하도록 설계되었습니다.

9.5 HP-UX RBAC 구성

HP-UX RBAC 구성은 다음 세 단계로 이루어진 프로세스입니다.

1. 역할을 구성합니다.
2. 권한 부여를 구성합니다.
3. 추가 명령을 구성합니다.

중요: 권한 부여는 HP-UX RBAC 관리 명령에 내장(하드코딩)되어 있으며 구성할 수 없습니다. 그러나 필수 HP-UX RBAC 관리 명령 권한 부여를 할당할 역할과 사용자를 구성할 수 있습니다.

HP-UX RBAC 관리 명령은 `setuid=0`이므로 `privrun` 명령으로 래핑하지 않아도 됩니다. HP-UX RBAC 관리 명령은 누가 호출하든 관계없이 root와 같은 권한을 사용하여 실행됩니다. 액세스 제어 확인에서 HP-UX RBAC 관리 명령을 사용할 수 있는 사람을 제한합니다.

권한 부여에 대한 자세한 내용은 각 HP-UX RBAC 관리 명령 맨페이지의 권한 부여 섹션을 참조하십시오.

섹션 9.5에서는 표 9-6의 예제 계획 결과와 사용자를 사용하여 HP-UX RBAC 관리 명령과 구성 프로세스를 보여 줍니다.

표 9-6 예제 계획 결과

사용자	역할	권한 부여 (참고: 객체는 *라고 가정)	일반적인 명령
chandrika, rwang	UserOperator	`hpux.user.*` `hpux.security.*`	`/usr/sbin/useradd` `/usr/sbin/usermod`
bdurant, prajessh	NetworkOperator	`hpux.network.*`	`/sbin/init.d/inetd`
luman	Administrator	`hpux.*` `company.customauth`	`/opt/customcmd`

9.5.1 역할 구성

사용자의 역할 구성은 다음 두 단계로 이루어진 프로세스입니다.

1. 역할을 만듭니다.
2. 사용자나 그룹에 역할을 할당합니다.

9.5.1.1 역할 만들기

`roleadm` 명령을 사용하여 역할을 만들고 사용자나 그룹에 할당합니다. 존재하지 않는 역할을 먼저 추가한 다음 이러한 역할에 사용자를 할당해야 합니다. 다음은 `roleadm` 명령 구문을 보여 줍니다.

```
roleadm add role [comments]
        | delete role
        | modify oldrolename newrolename
        | assign user role
        | assign "&group" role
        | revoke user [role]
        | revoke "&group" [role]
        | list [user=username][role=rolename][sys]
```

다음은 `roleadm` 명령 인수 목록과 간략한 설명입니다.

`add` `/etc/rbac/roles`에 있는 역할의 시스템 목록에 역할을 추가합니다.

`delete` `/etc/rbac/roles`에 있는 시스템 역할 목록에서 역할을 삭제합니다.

`modify`	역할과 관련된 세 개 데이터베이스 파일인 `/etc/rbac/roles`, `/etc/rbac/user_role` 및 `/etc/rbac/role_auth`에서 모두 역할 이름을 변경합니다.
`assign`	사용자나 그룹에 역할을 할당하고 `/etc/rbac/user_role`을 업데이트합니다.
`revoke`	사용자나 그룹에서 역할을 해지하고 `/etc/rbac/user_role`에서 항목을 제거합니다.
`list`	유효한 시스템 역할(sys)이나 사용자-역할 매핑을 표시합니다.

참고: 자세한 내용은 *roleadm*(1m) 맨페이지를 참조하십시오.

다음은 새 역할을 추가하는 `roleadm` 명령의 두 가지 예제입니다.

```
# roleadm add UserOperator
roleadm: added role UserOperator

# roleadm add NetworkOperator
roleadm: added role NetworkOperator
```

참고: HP-UX RBAC와 함께 제공된 기본 구성 파일에는 미리 구성된 Administrator 역할만 들어 있습니다. 기본적으로 Administrator 역할에는 모든 HP-UX 시스템 권한 부여(`hpux.*, *`)가 할당되며 root 사용자와 연관이 있습니다.

유효한 역할을 정의한 후 하나 이상의 사용자나 그룹에 할당할 수 있습니다. 만들어지지 않은 역할을 사용자에게 할당하려고 하면 역할이 없다는 오류 메시지가 표시됩니다.

9.5.1.2 사용자에게 역할 할당

역할 만들기와 역할 할당을 구분하면 다음과 같은 이점이 있습니다.

- 역할을 할당하기 전에 만들어야 하므로 역할 할당 중 역할 이름을 지정할 때 맞춤법 오류가 확인됩니다.
- 여러 사용자가 각 작업을 수행할 수 있습니다. 예를 들어, 같은 사용자가 역할 만들기와 역할 할당을 모두 수행하지 않아도 됩니다.

유효한 역할을 만든 후 다음 예제와 같이 `roleadm` 명령을 사용하여 해당 사용자에게 할당합니다.

```
# roleadm assign luman Administrator
roleadm assign done in /etc/rbac/user_role

# roleadm assign rwang UserOperator
roleadm assign done in /etc/rbac/user_role
```

`roleadm assign` 명령을 사용하여 역할을 사용자에게 할당한 후 `roleadm list` 명령을 사용하여 역할이 제대로 할당되었는지 확인할 수 있습니다. 예를 들면 다음과 같습니다.

```
# roleadm list
root: Administrator
luman: Administrator
rwang: UserOperator
```

참고: HP-UX RBAC는 DEFAULT라는 특수 사용자를 /etc/rbac/user_role 데이터베이스에 추가하는 기능을 제공합니다. DEFAULT 사용자에게 역할을 할당하면 시스템에 없는 모든 사용자에게 해당 역할이 할당됩니다.

9.5.1.3 그룹에 역할 할당

HP-UX RBAC를 사용하면 그룹에 역할을 할당할 수도 있습니다. roleadm assign user role 및 roleadm revoke user role과 같은 user 값을 사용하는 roleadm 명령 옵션을 사용하여 그룹과 역할을 관리할 수 있습니다.

사용자 값의 시작 부분에 앰퍼샌드(&)를 삽입하고 사용자 값을 따옴표로 묶어 roleadm 명령을 사용하여 그룹 및 역할 정보를 할당, 해지 또는 표시합니다. roleadm에서 그룹 이름 값과 앰퍼샌드(&)를 해석하려면 셸 이스케이프하거나 따옴표로 묶어야 합니다. 예를 들면 다음과 같습니다.

```
# roleadm assign "&groupname" role
```

9.5.2 권한 부여 구성

권한 부여 구성은 역할을 만들고 할당하는 것과 비슷합니다. 그러나 권한 부여에는 두 가지 요소인 작업과 객체가 있습니다. 자주 사용하는 객체인 * 와일드카드는 authadm 명령을 호출하는 동안 객체를 지정하지 않을 경우 사용되는 암시적 객체입니다. 대부분의 경우 작업이 모든 객체에 적용되도록 객체는 의도적으로 지정되지 않습니다. 객체를 지정하지 않는 방법은 래핑된 명령에 적용되는 권한 부여에 자주 사용합니다. 이는 명령 이름에서 작업 대상을 확인하기 어려울 수 있기 때문입니다.

이러한 객체 모호성의 예로 /usr/sbin/passwd 명령이 있습니다. passwd 명령은 /etc/passwd 파일, NIS 테이블, LDAP 항목 등 많은 리포지토리에서 작동할 수 있습니다. 명령줄만 보고 실제 객체를 확인할 수 없으므로 일반적으로 사용자가 모든 객체에 대해 작업을 수행하도록 하는 것이 가장 쉽습니다. 예를 들면 (hpux.security.passwd.change, *)와 같습니다.

참고: 기본 객체의 값을 구성할 수 있습니다. 기본적으로 객체를 지정하지 않으면 HP-UX RBAC는 * 와일드카드를 객체로 사용합니다. 그러나 /etc/default/security의 RBAC_DEFAULT_OBJECT= 매개 변수 값을 구성한 경우 HP-UX RBAC는 * 와일드카드 대신 이 값을 기본 객체로 사용합니다.

authadm 명령을 사용하여 HP-UX RBAC 데이터베이스에 있는 권한 부여 정보를 편집합니다. authadm 구문은 roleadm 구문과 비슷합니다. 다음은 authadm 명령 구문입니다.

```
authadm add operation[object[comments]]
        | delete operation[object]
        | assign role operation[object]
        | revoke [role=name][operation=name[object=name]]
        | list [role=name][operation=name[object=name][sys]
```

다음은 authadm 명령 인수 목록과 간략한 설명입니다.

add /etc/rbac/auths에 있는 유효한 권한 부여의 시스템 목록에 권한 부여를 추가합니다.

delete	/etc/rbac/auths에 있는 유효한 권한 부여의 시스템 목록에서 권한 부여를 삭제합니다.
assign	역할에 권한 부여를 할당하고 /etc/rbac/role_auth에 항목을 추가합니다.
revoke	역할에서 권한 부여를 해지하고 /etc/rbac/role_auth를 업데이트합니다.
list	시스템이나 역할별로 유효한 권한 부여를 나열하고 지정된 작업과 연관된 역할을 표시합니다.

중요: 별표 * 문자가 포함된 권한 부여를 할당할 때는 셸이 해석하지 않도록 다음 예제와 같이 와일드카드 문자를 따옴표로 묶어야 합니다.

다음은 표 9-6을 기반으로 권한 부여를 만들고 할당하는 예제입니다.

```
# authadm add 'company.customauth.*'
authadm added auth: (company.customauth.*,*)

# authadm assign Administrator 'company.customauth.*'
authadm added auth for role Administrator
```

authadm 명령에 list 인수를 사용하여 권한 부여 할당을 확인합니다. 예를 들면 다음과 같습니다.

```
# authadm list
Administrator: (hpux.*, *) (company.customauth.*, *)
```

9.5.3 추가 명령 권한 부여 및 권한 구성

기본 구성에 제공되지 않은 추가 명령을 정의해야 합니다. 명령을 실행하는 데 필요한 권한 부여가 이미 있어야 하고 역할에 할당되어야 합니다. 이렇게 하지 않은 경우 명령은 구성되지만 사용자에게 이 명령을 사용할 권한이 없습니다.

cmdprivadm 명령을 사용하여 명령의 권한 부여 및 권한 정보를 편집합니다. cmdprivadm 명령은 roleadm 및 authadm과 유사한 방식으로 작동하지만 privrun 데이터베이스에 명령 권한과 권한 부여를 추가하고 제거하는 것만 허용합니다.

다음은 cmdprivadm 명령 구문을 보여 줍니다.

```
cmdprivadm add cmd=full_path_name_of_a_command | full_path_name_of_a_file
                        |[op=operation]|[object=object]
                        |[ruid=ruid]|[euid=euid]
                        |[rgid=rgid]|[egid=egid]
                        |[compartment=compartment_label]
                        |[privs=comma_separated_privilege_list]
                        |[re-auth=pam_service_name]
                        |[flags=comma_separated_flags_list]
cmdprivadm delete cmd=full_path_name_of_a_command | full_path_name_of_a_file
                        |[op=operation]|[object=object]
                        |[ruid=ruid]|[euid=euid]
                        |[rgid=rgid]|[egid=egid]
                        |[compartment=compartment_label]
                        |[privs=comma_separated_privilege_list]
                        |[re-auth=pam_service_name]
                        |[flags=comma_separated_flags_list]
```

다음은 두 개의 주요 authadm 명령 인수 목록과 간략한 설명입니다.

add	/etc/rbac/cmd_priv 데이터베이스에 명령(또는 파일) 권한 부여 정보를 추가합니다.
delete	/etc/rbac/cmd_priv 데이터베이스에서 명령(또는 파일) 권한 부여 정보를 삭제합니다.

다음 예제에서는 가장 일반적인 cmdprivadm 인수를 보여 줍니다.

```
# cmdprivadm add cmd=/opt/customcmd \
op=companyname.customcommand ruid=0 euid=0 flags=edit \
/opt/customcmd::(companyname.customcommand,*):0/0/-1/-1::::edit
cmdprivadm added the entry to /etc/rbac/cmd_priv
```

앞의 예제와 같이 cmd_priv 파일 데이터베이스 파일에는 플래그 값 필드가 들어 있습니다. 명령이나 파일 권한 부여와 권한 정보를 구성할 때는 cmdprivadm 플래그 값을 고려해야 합니다.

privrun 명령은 하나의 정의된 플래그 KEEPENV를 인식합니다. KEEPENV 플래그가 특정 명령에 대해 cmd_priv 파일에 설정되어 있으면 privrun이 해당 특정 명령을 래핑할 때 환경 변수가 제거되지 않습니다.

privedit에 대해 플래그 값을 지정하여 privedit에서 파일을 편집할 수 있는지 여부를 나타낼 수 있습니다. 추가 플래그 값을 지정하여 privrun에서 명령을 실행할 수 있는지 여부를 나타낼 수 있습니다. 다음은 지원되는 플래그 값입니다.

flag=비어 있거나 다른 임의의 토큰	파일을 실행만 할 수 있고 편집할 수 없음을 나타냅니다.
flag=edit	파일을 편집 및 실행할 수 있음을 나타냅니다. 이 플래그는 주로 스크립트에 사용됩니다.
flag=noexec	파일을 실행할 수 없고 privedit를 사용하여 편집만 할 수 있음을 나타냅니다.

참고: 모든 cmdprivadm 인수에 대한 자세한 내용은 cmdprivadm(1M)을 참조하십시오. 대부분의 인수는 선택 사항이며, 인수를 지정하지 않을 경우 적절한 기본값이 채워집니다.

참고: /etc/rbac/cmd_priv 파일에 있는 기존 항목을 수정하려면 먼저 항목을 삭제하고 업데이트된 버전을 다시 추가해야 합니다. cmdprivadm을 사용하여 항목을 삭제하는 경우 인수가 필터 역할을 합니다. 예를 들어, cmdprivadm delete op=foo 명령은 작업이 foo인 모든 항목을 제거합니다. 따라서 cmdprivadm을 사용하여 항목을 삭제할 때는 충분한 인수를 지정하여 제거할 항목을 고유하게 식별해야 합니다.

9.5.4 Fine-grained 권한을 사용하여 HP-UX RBAC 구성

응용 프로그램은 시스템 호출을 사용하여 시스템의 리소스와 통신하고 운영 체제가 시스템 리소스에 액세스할 수 있도록 합니다. 특정 시스템 호출에서는 응용 프로그램이 운영 체제와 시스템 하드웨어에 액세스하는 데 특별한 높은 권한이 필요합니다.

Fine-grained 권한을 사용할 수 있기 전에는 특정 시스템 호출에 대한 특별한 높은 권한으로 UID=0이어야 했습니다. UID가 0이 아닌 경우 시스템 호출이 거부되고 응용 프로그램 오류가 반환되었습니다.

HP-UX RBAC 및 구체적으로 privrun 래퍼 명령을 통해 root가 아닌 사용자는 특정 응용 프로그램을 실행하는 데 필요한 UID=0이나 특별한 권한 수준을 얻을 수 있습니다. 특정 상황에서 특정 응용 프로그램을 실행하기 위해 root가 아닌 사용자에게 UID=0을 제공하는 것 외에도

HP-UX RBAC는 추가 권한을 사용하지만 UID=0 없이 응용 프로그램을 실행하기 위해 Fine-grained 권한을 사용할 수 있습니다.

HP-UX RBAC를 통해 선택한 권한 집합만 사용하고 각 사용자에 대해 다른 권한 집합을 사용하며 모두 UID=0 없이 실행되도록 명령을 구성할 수 있습니다. 예를 들어, 관리자는 여러 가지 권한을 사용하여 foobar 명령을 실행해야 하는 반면 일반 사용자는 foobar를 실행하는 데 훨씬 적은 권한이 필요할 수 있습니다.

Fine-grained 권한을 "시스템 호출 액세스 제어 확인 키"로 간주합니다. UID=0을 확인하는 대신 시스템 호출은 특정 권한을 확인합니다. 이러한 Fine-grained 권한은 시스템 호출을 "잠그고" 운영 체제와 하드웨어 리소스에 대한 응용 프로그램 액세스를 제어하는 기능을 제공합니다. 또한 권한을 Fine-grained 권한으로 분할하면 응용 프로그램을 실행하는 데 모든 권한이 필요하지 않으며 특정 권한이나 권한 집합만 있으면 됩니다. 특정 권한 집합을 사용하여 실행 중인 응용 프로그램 프로세스가 손상될 경우 프로세스가 UID=0을 사용하여 실행되는 동안 손상될 때보다 잠재적 손상이 훨씬 적습니다.

참고: Fine-grained 권한에 대한 자세한 내용은 privileges(5)를 참조하십시오.

cmdprivadm 명령과 privs 옵션을 사용하여 privrun에 대한 명령이 지정된 권한만 사용하여 래핑 및 실행되도록 구성합니다. 다음은 BASICROOT 복합 권한을 사용하여 실행되도록 /usr/bin/ksh 명령을 구성하고 (hpux.adm.mount, *) 권한 부여가 필요한 예제 cmdprivadm 명령입니다.

```
# cmdprivadm add cmd=/etc/mount op=hpux.adm.mount object='*' privs=BASICROOT
```

앞의 cmdprivadm 명령은 다음과 같이 /etc/rbac/cmd_priv 파일에 항목을 만듭니다.

```
#-------------------------------------------------------------------------------------------------------
# Command        : Args   :Authorizations       :U/GID :Cmpt   :Privs     :Auth  :Flags
#---------------:--------:---------------------:------:-------:----------:------:----------------------
/etc/mount       :dflt    :(hpux.adm.mount,*)   :///   :dflt   :BASICROOT :dflt  :
```

cmdprivadm을 사용하여 항목을 만들고 privrun을 사용하여 명령을 래핑하면 /etc/mount는 사용자에게 (hpux.adm.mount, *) 권한 부여가 없을 경우 UID=0 없이 BASICROOT 복합 Fine-grained 권한의 높은 권한을 사용하여 실행됩니다.

섹션 9.6.1에서 설명했듯이 privrun -p 명령 옵션은 -p 옵션에 의해 지정된 권한이 있는 /etc/rbac/cmd_priv 데이터베이스 파일의 항목하고만 일치합니다. privrun -p 옵션을 사용하여 권한을 지정하는 경우 privrun은 -p 지정 권한이 포함된 복합 권한 및 권한 그룹을 비롯하여 지정된 권한이 있는 모든 항목과 일치합니다. privrun 명령은 /etc/rbac/cmd_priv의 첫 번째 일치 항목에 따라 실행됩니다. 예를 들어, 다음은 예제 privrun -p 명령 및 /etc/rbac/cmd_priv에서 이 명령과 일치하는 항목 목록입니다.

명령은 다음과 같습니다.

```
# privrun -p MOUNT /etc/mount
```

이 명령은 다음과 같은 /etc/rbac/cmd_priv 항목과 일치합니다.

```
#-------------------------------------------------------------------------------------------------------
# Command        : Args   :Authorizations     :U/GID :Cmpt   :Privs
 :Auth :Flags
#---------------:--------:------------------:------:------:-------------------------------------:-----:-----
/etc/mount       :dflt    :(hpux.adm.mount,*) :///   :dflt   :PRIV_CHOWN, MOUNT
 :dflt :
/etc/mount       :dflt    :(hpux.*,nfs)       :///   :dflt   :MOUNT, PRIV_RTPRIO, PRIV_MLOCK
 :dflt :
/etc/mount       :dflt    :(hpux.adm.*,*)     :///   :dflt   :BASICROOT
 :dflt :
```

참고: privrun -p MOUNT /etc/mount 명령은 MOUNT 단순 권한이 미리 정의된 BASICROOT 복합 권한의 일부이기 때문에 BASICROOT 권한과 일치합니다. 단순 권한과 복합 권한에 대한 자세한 내용은 **privileges**(5) 맨페이지를 참조하십시오.

중요: privrun이 찾은 첫 번째 명시적 일치 항목에 따라 실행되기 때문에 /etc/rbac/cmd_priv의 항목 시퀀스는 중요합니다. 앞의 예제에서는 세 개 항목이 모두 privrun 명령과 일치하는 항목으로 간주되지만 privrun은 첫 번째 항목을 실행합니다. 명령과 권한 부여를 구성할 경우 항목 시퀀스에 주의하십시오. cmdprivadm 도구는 /etc/rbac/cmd_priv 파일의 아래쪽에 항목을 추가합니다.

9.5.5 구획을 사용하여 HP-UX RBAC 구성

HP-UX RBAC는 구획을 사용하여 특정 구획에서 실행되도록 응용 프로그램을 구성할 수도 있습니다. 구획을 사용하면 구획 규칙이 허용하도록 설정되지 않은 경우 프로세스가 해당 구획 밖에 있는 리소스에 액세스하거나 통신할 수 없도록 논리적으로 시스템을 여러 구획으로 분할할 수 있습니다.

다음은 /etc/cmpt/apache.rules 구획 규칙에 의해 정의된 apache 구획에서만 실행되도록 /sbin/init.d/hpws_apache 명령을 구성하는 예제 cmdprivadm 명령입니다.

```
# cmdprivadm add cmd='/sbin/init.d/hpws_apache -a start' \
op=hpux.network.service.start object=apache compartment=apache
```

앞의 cmdprivadm 명령은 다음과 같이 /etc/rbac/cmd_priv 파일에 항목을 만듭니다.

```
#-------------------------------------------------------------------------------------------------------------------
# Command                : Args   :Authorizations                    :U/GID   :Cmpt    :Privs :Auth
   :Flags
#------------------------:--------:----------------------------------:--------:--------:-------:-------
/sbin/init.d/hpws_apache  :start   :(hpux.network.service.start,apache) :///     :apache :dflt  :dflt
   :
```

cmdprivadm을 사용하여 항목을 만들고 privrun을 사용하여 명령을 매핑하면 권한이 부여된 사용자는 /sbin/init.d/hpws_apache -start 명령을 실행할 수 있으며, 이 명령은 apache 구획에서만 실행됩니다. 프로세스의 구획 태그는 apache이고 프로세스의 속성은 정의된 apache 구획 규칙을 따릅니다.

참고: 명령에 대해 구획을 구성하려면 cmdprivadm 명령만 사용합니다. cmdprivadm을 사용하지 않고 /etc/rbac/cmd_priv 데이터베이스 파일을 편집하지 마십시오.

/etc/rbac/cmd_priv 파일에 있는 기존 항목을 수정하려면 먼저 항목을 삭제하고 업데이트된 버전을 다시 추가해야 합니다. cmdprivadm을 사용하여 항목을 삭제하는 경우 인수가 필터 역할을 합니다. 예를 들어, cmdprivadm delete op=foo 명령은 작업이 foo인 모든 항목을 제거합니다. 따라서 cmdprivadm을 사용하여 항목을 삭제할 때는 충분한 인수를 지정하여 제거할 항목을 고유하게 식별해야 합니다.

9.6 HP-UX RBAC 사용

이 절에서는 privrun 및 privedit 명령을 실행하여 HP-UX RBAC를 작동하는 방법에 대해 설명합니다.

9.6.1 privrun 명령을 통해 권한을 사용하여 응용 프로그램 실행

privrun 명령을 통해 사용자는 호출하는 사용자와 연관된 권한 부여에 따라 다양한 권한을 사용하여 기존 응용 프로그램을 실행할 수 있습니다. 사용자는 기존 응용 프로그램을 명령줄 인수로 지정하여 privrun을 호출합니다. 그런 다음 privrun이 /etc/rbac/cmd_priv 데이터베이스를 참조하여 추가 권한을 사용하여 명령을 실행하는 데 필요한 권한 부여를 확인합니다. 사용자에게 필요한 권한 부여가 있으면 privrun은 해당 UID 또는 GID를 /etc/rbac/cmd_priv 데이터베이스에 지정된 대로 변경한 후 지정된 명령을 호출합니다.

다음은 privrun 명령 구문입니다.

```
privrun [options] command [args]
        | [-u eUID|username]
        | [-g eGID|groupname]
        | [-U rUID|username]
        | [-G rGID|groupname]
        | [-a (operation, object)]
        | [-c compartment]
        | [-p privilege[,privilege,privilege...]]
        | [-x]
        | [-v [-v]]
        | [-h]
        | [-t]
```

다음 목록에서는 각 privrun 명령 옵션에 대해 설명합니다.

-u 지정된 EUID 또는 사용자 이름과 연관된 EUID에 해당하는 EUID(유효 사용자 ID)가 포함된 항목하고만 일치합니다.

-g 지정된 EGID 또는 그룹 이름과 연관된 EGID에 해당하는 EGID(유효 그룹 ID)가 포함된 항목하고만 일치합니다.

-U 지정된 RUID 또는 사용자 이름과 연관된 RUID에 해당하는 RUID(실제 사용자 ID)가 포함된 항목하고만 일치합니다.

-G 지정된 RGID 또는 그룹 이름과 연관된 RGID에 해당하는 RGID(실제 그룹 ID)가 포함된 항목하고만 일치합니다.

-a 지정된 권한 부여가 필요한 항목하고만 일치합니다. 권한 부여는 /etc/rbac/cmd_priv 데이터베이스 파일에서 (작업, 객체) 쌍으로 정의됩니다. 지정된 권한 부여는 /etc/rbac/cmd_priv 파일에 있는 권한 부여와 정확하게 일치해야 하며 와일드카드는 지원되지 않습니다.

-c /etc/rbac/cmd_priv 데이터베이스 파일에 있는 지정된 구획과 일치합니다. 지정된 구획은 /etc/rbac/cmd_priv에 있는 구획과 정확하게 일치해야 합니다.

-p 지정된 권한을 /etc/rbac/cmd_priv 데이터베이스 파일에 있는 권한과 일치시킵니다. 권한을 여러 개 지정할 수 있습니다. 권한을 여러 개 지정하는 경우 각 권한을 쉼표로 구분합니다. privrun -p 옵션을 사용하여 권한을 지정하는 경우 privrun은 -p 지정 권한이 포함된 복합 권한 및 권한 그룹을 비롯하여 지정된 권한이 있는 모든 항목과 일치합니다. privrun 명령은 /etc/rbac/cmd_priv의 첫 번째 일치 항목에 따라 실행됩니다.

-x 권한 부여 또는 인증 확인이 실패한 경우에만 privrun의 동작을 수정하는 이탈(fall-through) 모드를 사용합니다. 오류 메시지를 표시하고 종료하는 대신 대상 명령이 실행되지만 추가 권한은 사용되지 않습니다. 대상 명령은 사용자가 privrun 없이 직접 명령을 실행한 것처럼 실행됩니다.

-v 상세 모드로 privrun을 호출합니다. 두 가지 -v 옵션을 지정하면 상세 수준이 증가합니다. 상세 수준이 증가하면 더 많은 정보가 인쇄됩니다.

-h privrun 도움말 정보를 인쇄합니다.

-t 구성 파일에 따라 모든 일반적인 권한 부여 및 인증 확인을 수행하여 원하는 privrun 호출이 성공하는지 확인하는 테스트 모드를 사용합니다. 유일한 차이점은 명령을 실행하는 대신 성공할 경우 privrun -t만 반환된다는 것입니다. 지정된 privrun 호출이 성공하는지 여부를 미리 보려면 이 옵션을 사용합니다.

다음은 기존 응용 프로그램을 래핑하는 가장 기본적인 privrun 사용 예제입니다. 이 경우 ipfstat 명령은 호출하는 사용자와 연관된 권한 부여에 따라 실행되기 위해 privrun 명령 인수로 실행됩니다.

```
# privrun ipfstat
```

로그인한 사용자에게 /etc/rbac/cmd_priv에 정의된 필요한 권한 부여가 있으면 privrun 래퍼 명령은 /etc/rbac/cmd_priv 항목에 정의된 권한(UID 및 GID)을 사용하여 기존 명령을 실행합니다.

동일한 명령에 대해 각각 다른 필수 권한 부여와 결과 권한을 가진 여러 개의 항목이 있을 수 있습니다. 이 경우 privrun은 /etc/rbac/cmd_priv 데이터베이스를 순차적으로 반복하고 사용자에게 권한이 부여된 첫 번째 명령을 실행합니다.

이 동작이 적합하지 않은 경우도 있습니다. 예를 들어, 모든 사용자가 passwd 명령을 실행하여 자신의 암호를 변경할 수 있지만 사용자 관리자가 실행할 경우 다른 사용자의 암호를 변경할 권한이 필요합니다. 모든 일반 사용자에 대한 항목이 사용자 관리자에 대한 항목보다 앞에 나열되어 있으면 이 항목이 먼저 실행되므로 사용자 관리자가 더 권한이 큰 버전을 실행할 수 없습니다.

이러한 경우를 위해 privrun에는 사용자가 원하는 권한을 지정할 수 있는 옵션이 있습니다. 지정된 권한(예: UID)과 일치하는 항목만 사용됩니다. 원하는 권한과 일치하는 항목이 없으면 privrun에서 오류 메시지를 반환합니다.

다음은 유효 UID가 0으로 설정된 항목하고만 일치하는 privrun의 예제 호출입니다.

```
# privrun -u 0 ipfstat
```

참고: privrun 명령 사용에 대한 자세한 내용은 *privrun*(1M) 및 *rbac*(5) 맨페이지를 참조하십시오.

9.6.1.1 Serviceguard 클러스터의 HP-UX RBAC

Serviceguard는 HP-UX RBAC 및 privrun을 사용하여 Serviceguard 명령에 대한 액세스 권한을 부여하는 기능을 지원하지 않습니다. Serviceguard 버전 A.11.16은 패키지 및 클러스터 구성 파일을 통해 액세스 제어 정책을 지정하고 Serviceguard 작업에 대한 클러스터 인식 정책을 제공하여 자체 RBAC(Role-Based Access Control)를 구현합니다. Serviceguard 작업의 RBAC(Role-Based Access Control)에는 Serviceguard 메커니즘을 사용해야 합니다. Serviceguard 액세스 제어 정책에 대한 자세한 내용은 최신 **Serviceguard 관리** 설명서를 참조하십시오.

Serviceguard 클러스터에서 Serviceguard 이외의 명령과 함께 HP-UX RBAC를 사용할 수 있습니다. 클러스터에 있는 모든 노드에 동일한 HP-UX RBAC 규칙이 적용되어야 합니다.

9.6.2 privedit 명령을 사용하여 액세스가 제어되는 파일 편집

privedit 명령을 사용하면 권한이 부여된 사용자가 일반적으로 파일 사용 권한이나 ACL로 인해 편집할 수 없는 파일을 편집할 수 있습니다. 명령을 호출하고 편집할 파일을 인수로 식별하면 privedit에서 privrun과 마찬가지로 /etc/rbac/cmd_priv 데이터베이스를 검사하여 지정된 파일을 편집하는 데 필요한 권한 부여를 확인합니다. 호출하는 사용자에게 파일을 편집할 권한이 있으면 privedit는 파일 복사본에 대해 편집기를 호출합니다.

참고: 파일을 편집하기 위해 privedit를 사용하여 편집기를 호출할 경우 편집기가 높은 권한을 사용하여 실행되지 않습니다. privedit에서 호출하는 편집기가 높은 권한을 사용하여 실행되지 않으므로 셸 이스케이프 같은 시도된 모든 작업은 사용자의 일반(높지 않은) 권한 집합을 사용하여 실행됩니다.

EDITOR 환경 변수를 설정하여 privedit에서 파일을 편집하는 데 사용하는 편집기를 지정할 수 있습니다. EDITOR 변수를 설정하지 않으면 privedit는 기본 편집기 vi를 사용합니다. privedit 명령줄을 통해 편집기에 인수를 전달할 수는 없습니다. 그러나 privedit를 호출하기 전에 편집기 관련 환경 변수를 설정하면 편집기가 해당 변수를 인식하고 지원합니다.

정규화된 파일 이름을 privedit 인수로 사용하여 편집할 파일을 식별합니다. 정규화된 파일 이름을 사용하지 않으면 privedit에서 지정된 파일 이름의 시작 부분에 현재 작업 디렉토리를 추가합니다. 편집할 파일을 지정하는 방법에 관계없이 모든 파일 이름은 privedit를 호출한 후 정규화됩니다. 또한 privedit 명령은 심볼릭 링크가 있는 파일을 인식하고 지원합니다.

privedit 명령은 한 번에 하나의 파일만 편집할 수 있습니다. 여러 개의 파일 이름을 privedit 인수로 지정하면 privedit는 지정된 첫 번째 파일을 편집하고 후속 파일 이름을 무시합니다. 다음은 privedit 명령 구문을 보여 줍니다.

```
privedit [option] fully-qualified-file-name
         |   [-a (operation, object)]
         |   [-v]
         |   [-h]
         |   [-t]
         |   [-x]
```

다음은 privedit 명령 옵션 목록과 간략한 설명입니다.

-a authorization	지정된 권한 부여가 있는 /etc/rbac/cmd_priv 파일 항목하고만 일치합니다.
-v	상세 모드로 privedit를 호출합니다.
-h	privedit 도움말 정보를 인쇄합니다.
-t	사용자에게 파일을 편집하는 데 필요한 권한 부여가 있는지 확인하고 결과를 보고합니다.
-x	권한 부여 확인이 실패하면 호출자의 원래 권한을 사용하여 파일이 편집됩니다.

다음은 privedit 명령을 통해 (hpux.sec.edit, secfile)의 특정 권한 부여를 사용하여 /etc/default/security 파일을 편집하는 예제입니다.

```
# privedit -a "(hpux.sec.edit, secfile)" /etc/default/security
```

참고: cmd_priv 데이터베이스에 있는 각 항목의 플래그 값은 privedit에서 파일을 편집할 수 있는지 여부를 나타냅니다. 플래그 및 privedit 명령 사용에 대한 자세한 내용은 "추가 명령 권한 부여 및 권한 구성" 및 *privedit*(1M) 맨페이지를 참조하십시오.

9.6.3 ACPS를 사용하여 privrun 및 privedit 사용자 정의

HP-UX RBAC 기능은 privedit 및 privrun에서 사용자 권한 부여를 확인하는 방법을 사용자 정의하는 기능을 제공합니다. ACPS 모듈은 권한 부여 결정을 내려야 하는 응용 프로그램에 응답을 제공하는 사용자 정의 가능한 인터페이스입니다. ACPS 구성 파일 /etc/acps.conf는 ACPS의 다음 측면을 제어합니다.

- 액세스 결정을 위해 참조되는 모듈
- 모듈이 참조되는 시퀀스
- 모듈 응답을 결합하여 응용 프로그램에 결과를 반환하는 규칙

ACPS에 대한 자세한 내용은 섹션 9.3.1과 *acps.conf*(4), *acps*(3) 및 *rbac*(5)를 참조하십시오.

9.7 HP-UX RBAC 문제 해결

다음은 HP-UX RBAC 문제를 해결하고 디버깅하는 데 사용되는 주요 메커니즘 목록입니다.

- rbacdbchk 유틸리티는 HP-UX RBAC 데이터베이스 구문을 확인합니다.
- privrun -v 명령은 관련된 추가 정보를 보고합니다.

9.7.1 rbacdbchk 데이터베이스 구문 도구

가장 일반적인 버그는 HP-UX RBAC 데이터베이스를 수동으로 편집하여 구문상 잘못된 구성을 만들거나 데이터베이스 간에 일치하지 않는 구성(예: /etc/rbac/user_role에 있는 역할이 /etc/rbac/roles에는 정의되어 있지 않음)을 만드는 경우입니다. 이러한 일반적인 실수를 진단할 수 있도록 HP-UX RBAC에는 rbacdbchk 명령이 포함되어 있습니다. 이 명령은 전체 HP-UX RBAC 데이터베이스를 읽고 잘못되었거나 일치하지 않는 구성 항목이 있을 경우 경고문을 출력합니다.

```
# rbacdbchk
[/etc/rbac/user_role] chandrika: UserOperator
        invalid user
         The value 'chandrika' for the Username field is bad.

[/etc/rbac/cmd_priv]
/opt/cmd:dflt:(newop,*):0/0//:dflt:dflt:dflt:
        invalid command: Not found in the system
        The value '/opt/cmd' for the Command field is bad.

[Role in role_auth DB with no assigned user in user_role DB]
Rebooter:(hpux.admin.*, *)

[Invalid Role in user_role DB. Role 'UserOperator' assigned to user 'chandrika' does not exist in
the roles DB]
```

제대로 구성된 시스템에서는 rbacdbchk 명령이 오류가 없음을 나타내는 메시지를 생성하지 않습니다.

9.7.2 privrun -v 정보

문제를 감지하는 두 번째 방법은 -v 옵션(상세 모드)을 사용하여 privrun 명령을 실행하는 것입니다. 상세 모드에서 privrun은 입력 명령과 일치한 항목, 권한 부여 확인 상태 및 다른 관련된 데이터에 대한 추가 정보를 제공합니다. 대부분의 경우 이 출력을 통해 privrun이 실패한 원인을 찾을 수 있습니다. 상세 출력 수준을 높이려면 -v 옵션을 여러 번 지정합니다. 다음은 ipfstat 명령을 사용한 privrun -v 출력 예제입니다.

```
# privrun -v /sbin/ipfstat
privrun: user root intends to execute command /sbin/ipfstat
privrun: input entry: '/sbin/ipfstat:dflt:(,):///:dflt:dflt::'
privrun: found matching entry: '/sbin/ipfstat:dflt:(hpux.network.filter.readstat,*):0/0//:dflt:dflt::'
privrun: passed authorization check
privrun: attempting to set ruid/euid/rgid/egid to 0/0/-1/-1
privrun: current settings for ruid/euid/rgid/egid are 0/0/3/3
privrun: executing: /sbin/ipfstat
```

10 감사 관리

감사는 보안 문제를 분석하고 감지하기 위해 이벤트를 선택적으로 기록하는 것을 목적으로 합니다. 감사 데이터는 로그 파일에 기록됩니다. 따라서 감사 시스템은 시스템 악용을 방지하며 잠재적인 보안 취약점을 드러냅니다.

감사 시스템은 시스템의 객체에 대한 주체의 액세스 인스턴스를 기록하고 보호 메커니즘을 무시하려는 (반복) 시도와 잘못된 권한 사용을 감지하며 시스템의 잠재적 보안 취약점을 밝히는 데 도움이 됩니다.

사용자가 로그인하면 "감사 태그"라는 고유한 감사 세션 ID가 생성되어 사용자 프로세스와 연관됩니다. 감사 태그는 각 로그인 세션 동안 동일하게 유지됩니다. 사용자가 단일 세션 내에서 ID를 변경하더라도 모든 이벤트는 같은 감사 태그를 사용하여 기록되며 원래 로그인 사용자의 이름으로 고려됩니다.

감사 기록은 선택적인 보안 관련 시스템 이벤트에 대해 생성됩니다. 각 감사 기록에는 이벤트 정의, 발생 시기, 발생시킨 사용자 ID, 발생시킨 프로세스 ID 등에 대한 정보가 들어 있습니다.

감사 기록은 바이너리 형식으로 감사 로그/파일에 수집됩니다. HP-UX 11i v3 릴리즈의 HP-UX Auditing System은 둘 이상의 작성기 스레드를 사용하여 데이터를 파일에 기록할 수 있습니다. 각 작성기 스레드는 하나의 파일에 기록합니다. 이렇게 하면 데이터 처리량을 늘리는 데 도움이 됩니다. 따라서 파일 시스템의 감사 기록은 여러 개의 감사 파일이 들어 있는 디렉토리입니다.

감사 기록의 레코드는 파일 공간을 절약하기 위해 압축됩니다. 프로세스를 처음 감사할 때 전체 프로세스 수명 동안 일정하게 유지되는 정보가 들어 있는 PIR(Process Identification Record)이 감사 기록에 쓰여집니다. 여기에는 프로세스 ID, 부모 프로세스 ID, 감사 태그, 실제 사용자 ID, 실제 그룹 ID, 유효 사용자 ID, 유효 그룹 ID, 그룹 ID 목록, 유효 권한, 허용되는 권한 및 유지되는 권한, 구획 ID, 터미널 ID 등이 포함됩니다. PIR은 감사 기록당 프로세스별로 한 번만 입력됩니다.

이 장의 내용은 다음과 같습니다.

- 감사 구성 요소(섹션 10.1)
- 시스템 감사(섹션 10.2)
- 사용자 감사(섹션 10.3)
- 감사 이벤트(섹션 10.4)
- 감사 기록(섹션 10.5)
- 감사 로그 보기(섹션 10.6)
- 자체 감사(섹션 10.7)
- HP-UX RBAC 감사(섹션 10.8)

10.1 감사 구성 요소

HP-UX 11i의 감사 기능에는 구성 파일, 명령 및 맨페이지가 포함되어 있습니다. 이러한 구성 요소는 다음 절에 나열되어 있습니다.

10.1.1 명령

표 10-1에서는 각 감사 명령에 대해 간략하게 설명합니다.

표 10-1 감사 명령

명령	설명
audevent	이벤트 또는 시스템 호출 상태를 변경하거나 표시합니다.
audisp	감사 기록을 표시합니다.
audomon	감사 파일 모니터링 및 크기 매개 변수를 설정합니다.
audsys	감사를 시작하고 중지합니다. 감사 파일이나 디렉토리 정보를 설정하고 표시합니다.
userdbset	AUDIT_FLAG=1 옵션을 지정하여 감사할 사용자를 선택합니다.

10.1.2 감사 구성 파일

표 10-2에서는 감사 기능과 연관된 각 구성 파일에 대해 간략하게 설명합니다.

표 10-2 감사 구성 파일

파일	설명
/etc/audit/audit.conf	미리 정의된 이벤트 분류 정보가 들어 있는 파일입니다.
/etc/audit/auditr_site.conf	사이트별 이벤트 분류 정보가 들어 있는 파일입니다.
/etc/default/security	시스템 범위 감사 기본값이 들어 있는 파일입니다.
/var/adm/userdb	사용자별 감사 정보가 들어 있는 데이터베이스입니다.
/etc/rc.config.d/auditing	시스템을 다시 부팅할 때 감사가 시작되도록 지정하는 구성 정보가 들어 있는 파일입니다.

10.1.3 감사 맨페이지

표 10-3에서는 감사 기능과 연관된 각 맨페이지에 대해 간략하게 설명합니다.

표 10-3 감사 맨페이지

맨페이지	설명
audevent(1M)	audevent 기능과 구문에 대해 설명합니다.
audisp(1M)	audisp 기능과 구문에 대해 설명합니다.
audomon(1M)	audomon 기능과 구문에 대해 설명합니다.
audsys(1M)	audsys 기능과 구문에 대해 설명합니다.
userdbset(1M)	userdbset 기능과 구문에 대해 설명합니다.
audit.conf(4)	/etc/audit/audit.conf 파일에 대해 설명합니다.
audit(5)	HP-UX 감사에 대해 소개합니다.

10.2 시스템 감사

시스템에서 감사를 계획, 활성화 및 모니터링하려면 다음 절차를 사용합니다.

10.2.1 감사 구현 계획

감사 구현을 계획하려면 다음 단계를 수행하십시오.

1. 감사할 사용자를 결정합니다. 기본적으로 모든 사용자가 감사에 대해 선택됩니다.
2. 감사할 이벤트나 시스템 호출을 결정합니다. `audevent` 명령을 사용하여 현재 감사에 대해 선택된 이벤트 및 시스템 호출 목록을 표시합니다.

   이벤트와 시스템 호출을 프로파일로 그룹화할 수 있습니다.
3. 감사 로그 파일(감사 기록)을 시스템에 저장할 위치를 결정합니다. 감사 로그 파일 구성에 대한 자세한 내용은 섹션 10.5을 참조하십시오.
4. 감사 파일 아카이브 및 백업 전략을 만듭니다. 감사 파일은 많은 디스크 공간을 사용하는 경우가 많으며 신중하게 파일 관리를 계획하지 않으면 오버플로가 발생할 수 있습니다. `audomon` 명령에 `-X` 옵션을 사용하여 아카이브를 자동화합니다.

감사 구현을 계획하는 데 유용한 관리 및 감사 시스템 성능에 대한 자세한 내용은 섹션 10.2.5 및 섹션 10.2.6을 참조하십시오.

10.2.2 감사 활성화

시스템에서 감사를 활성화하려면 다음 단계를 수행하십시오.

1. `userdbset` 명령을 사용하여 감사할 사용자를 구성합니다. 사용자에 대해 감사를 구성하는 방법은 섹션 10.3을 참조하십시오.
2. `audevent` 명령을 사용하여 감사할 이벤트를 구성합니다. 예를 들어, MySitePolicy에 따라 감사하려면 다음 명령을 입력합니다.

   ```
   # audevent -P -F -r MySitePolicy
   ```

   MySitePolicy는 `/etc/audit/audit_site.conf` 파일에 정의되어 있어야 합니다.

   옵션 없이 `audevent` 명령을 사용하여 현재 감사에 대해 구성된 이벤트 및 시스템 호출 목록을 표시합니다.

   이벤트에 대해 감사를 구성하는 방법은 섹션 10.4를 참조하십시오.
3. `/etc/rc.config.d/auditing` 파일의 `audevent` 인수 매개 변수를 설정하여 시스템을 다시 부팅할 때 감사 시스템에서 현재 구성 매개 변수를 유지할 수 있게 합니다. 예를 들어, 2단계에서 구성된 매개 변수를 유지하려면 다음과 같이 매개 변수를 설정합니다.

   ```
   AUDEVENT_ARGS1 = -P -F -r MySitePolicy
   ```
4. `audsys` 명령을 사용하여 감사 시스템을 시작하고 감사 기록을 정의합니다.

   ```
   #audsys -n -c primary_audit_file -s 1000
   ```
5. `/etc/rc.config.d/auditing` 파일에서 로그 파일과 로그 파일 전환 매개 변수를 설정합니다. 다음 단계를 수행하십시오.
   a. `PRI_AUDFILE`을 주 감사 로그 파일의 이름으로 설정합니다.
   b. `PRI_SWITCH`를 주 감사 로그 파일의 최대 크기(KB)로 설정합니다. 감사 로깅은 이 크기에서 보조 로그 파일로 전환됩니다.

c. SEC_AUDFILE을 보조 로그 파일의 이름으로 설정합니다.

d. SEC_SWITCH를 보조 감사 로그 파일의 최대 크기(KB)로 설정합니다.

주 감사 로그 파일과 보조 감사 로그 파일 설정에 대한 자세한 내용은 섹션 10.5를 참조하십시오.

6. 아직 시작되지 않은 경우 audomon 데몬을 시작합니다. audomon 데몬은 현재 감사 기록의 확장을 모니터링하고 필요할 때마다 대체 감사 기록으로 전환됩니다. 예를 들면 다음과 같습니다.

```
#audomon -p 20 -t 1 -w 90 -X "/usr/local/bin/rcp_audit_trail hostname"
```

audomon 데몬 구성에 대한 자세한 내용은 섹션 10.5.2를 참조하십시오.

7. /etc/rc.config.d/auditing 파일에서 audomon 인수 매개 변수를 설정하여 시스템을 다시 부팅해도 현재 설정을 유지합니다.

8. /etc/rc.config.d/auditing 파일에서 AUDITING 플래그를 1로 설정하여 시스템을 부팅할 때 감사 시스템이 자동으로 시작될 수 있게 합니다.

10.2.3 감사 비활성화

시스템에서 감사를 비활성화하려면 다음 단계를 수행하십시오.

1. 다음 명령을 사용하여 시스템 감사를 중지합니다.

```
#audsys -f
```

2. /etc/rc.config.d/auditing 파일에서 AUDITING 플래그를 0으로 설정하여 시스템을 다시 부팅할 때 감사 시스템이 시작되지 않게 합니다.

3. (선택 사항) audomon 데몬을 중지하려면 다음을 입력합니다.

```
# kill `ps -e | awk '$NFS~ /audomon/ {print $1}'`
```

audomon 데몬을 다시 구성하려는 경우에만 이 단계를 사용합니다. audomon 데몬을 다시 구성하고 다시 시작하려면 섹션 10.2.2에 설명된 것처럼 6단계와 7단계를 수행합니다.

10.2.4 감사 파일 모니터링

감사 파일을 보고 모니터링 및 관리하려면 다음 단계를 수행하십시오.

1. audisp 명령을 사용하여 감사 로그 파일을 봅니다.

```
# audisp audit_file
```

audisp 명령 사용에 대한 자세한 내용은 "감사 로그 보기"를 참조하십시오.

2. /etc/rc.config.d/auditing 파일에서 감사 로그 파일 모니터 인수를 설정합니다. 2단계에서 사용한 값과 동일한 값을 설정합니다.

3. (선택 사항) 다음 명령을 사용하여 시스템 감사를 중지합니다.

```
#audsys -f
```

4. (선택 사항) /etc/rc.config.d/auditing 파일에서 AUDIT 플래그를 0으로 설정하여 다음에 시스템을 다시 부팅할 때 감사 시스템이 시작되지 않게 합니다.

10.2.5 성능 고려 사항

감사 작업은 시스템 오버헤드를 증가시킵니다. 성능이 중요할 경우 감사할 이벤트 및 사용자를 줄입니다. 이렇게 하면 성능에 대한 감사 작업의 영향을 줄일 수 있습니다.

10.2.6 감사 시스템 관리 지침

시스템을 관리하는 경우 다음 지침을 사용합니다.

- 보안 정책에 따라 감사 로그를 확인합니다. 온라인 감사 파일은 24시간 이상 유지되어야 하며 외부에 저장된 모든 감사 기록은 30일 이상 유지되어야 합니다.
- 늦은 시간에 이루어진 로그인, 로그인 실패, 시스템 파일에 대한 액세스 실패 및 보안 관련 작업 수행 실패와 같은 비정상적인 동작에 대한 감사 로그를 검토합니다.
- 매일 보관하여 감사 파일의 오버플로를 방지합니다.
- 새 릴리즈에서 종종 새 시스템 호출을 제공하므로 특히 HP-UX의 새 릴리즈를 설치한 후에는 정기적으로 현재 선택 가능한 이벤트를 업데이트합니다.
- 감사된 사용자를 정기적으로 업데이트합니다.
- 이벤트 또는 사용자 선택 사항에 대해 패턴을 따르거나 예약하지 마십시오.
- 현장 지침을 설정합니다. 이러한 지침을 결정하는 데 사용자 및 관리부서의 의견을 참조합니다.
- 감사 데이터 볼륨이 커질 경우 여러 개의 물리 디스크와 여러 개의 물리 I/O 카드로 이루어지도록 논리 볼륨의 감사 기록을 구성합니다. `audsys` 명령에 `-N` 옵션을 사용하여 감사 기록을 여러 개의 파일로 분할합니다.

10.3 사용자 감사

기본적으로 시스템 감사가 설정될 때 모든 사용자에 대한 감사 상태가 설정됩니다. 시스템에 추가된 새 사용자는 자동으로 감사됩니다.

사용자가 감사를 사용하여 HP-UX 시스템에서 수행하는 작업을 모니터링할 수 있습니다. 감사할 사용자를 변경하려면 다음 옵션 중 하나를 선택합니다.

- 모든 사용자 감사

  기본적으로 감사 시스템이 설정될 때 모든 사용자의 감사 상태는 켜기로 설정됩니다. 시스템에 추가된 새 사용자는 자동으로 감사됩니다.

  모든 사용자에 대해 감사가 해제되어 있으면 `/etc/default/security` 파일에서 `AUDIT_FLAG=1`을 설정합니다.

- 사용자 감사 안 함

  모든 사용자에 대해 감사를 해제하려면 다음 단계를 수행하십시오.

  1. 이미 감사 중인 사용자를 확인합니다. 확인하려면 다음 단계를 수행하십시오.
     a. `/etc/default/security` 파일의 `AUDIT_FLAG` 설정을 확인합니다.
     b. 다음 명령을 사용하여 사용자 데이터베이스에 저장된 `AUDIT_FLAG` 설정을 확인합니다.

        ```
        # userdbget -a AUDIT_FLAG
        ```

  2. `/etc/default/security` 파일에서 `AUDIT_FLAG=0`을 설정합니다.

- 특정 사용자 감사. 특정 사용자에 대해 감사를 구성하려면 다음 단계를 수행하십시오.
  1. `/etc/default/security` 파일에서 `AUDIT_FLAG=0`을 설정하여 모든 사용자에 대한 감사를 선택 취소합니다.

2. 다음 명령을 사용하여 특정 사용자에 대해 감사를 구성합니다.

   # **/usr/sbin/userdbset -u user-name AUDIT_FLAG=1.**

감사 시스템이 아직 활성화되어 있지 않으면 audsys -n 명령을 사용하여 감사 시스템을 시작합니다. 감사 변경 사항은 다음에 사용자가 로그인할 때 적용됩니다.

10.4 감사 이벤트

이벤트는 파일 만들기, 파일 열기, 시스템에 로그인 등의 보안과 관련된 작업입니다. HP-UX 시스템의 이벤트를 감사하여 가능한 위반을 감지하고 보안을 향상시킬 수 있습니다. 그러나 더 많은 이벤트를 감사하도록 선택할수록 더 많은 시스템 리소스가 사용되며 시스템 성능에 미치는 영향이 커집니다. 보안 설계자는 비즈니스 요구와 적용되는 정부 규제를 기반으로 감사할 이벤트를 결정해야 합니다.

audevent 명령은 감사할 시스템 활동(감사 가능한 이벤트)을 지정하는 데 사용됩니다. 감사 가능한 이벤트는 쉽게 구성할 수 있도록 이벤트 범주와 프로파일로 분류됩니다. 이벤트 범주나 프로파일을 선택하면 해당 이벤트 범주나 프로파일과 연관된 모든 시스템 호출 및 자체 감사 이벤트가 선택됩니다. 감사 시스템을 설치하면 기본 이벤트 분류 정보 집합이 /etc/audit/audit.conf 파일에 제공됩니다. 사이트별 추가 분류 및 프로파일도 /etc/audit/audit_site.conf 파일에 정의되어 있을 수 있습니다.

참고:
최소한 다음 이벤트는 감사하는 것이 좋습니다.

- admin 이벤트
- login 이벤트
- moddac 자체 감사 이벤트
- execv, execve
- pset 이벤트

이러한 이벤트는 /etc/audit/audit.conf 파일에 **basic** 프로파일로 미리 정의되어 있습니다.

감사 시스템을 설정하기 전에 감사할 이벤트를 구성합니다. audevent 명령의 구문은 다음과 같습니다.

audevent [options]

audevent 명령에 자주 사용하는 옵션은 다음과 같습니다.

표 10-4 audevent 명령 옵션

audevent 옵션	설명
-e event	기록할 이벤트를 지정합니다.
-F	실패한 이벤트 작업을 기록합니다.
-l	전체 이벤트 유형 및 연관된 시스템 호출 목록을 표시합니다.
-P	성공한 이벤트 작업을 기록합니다.

표 10-4 audevent 명령 옵션 (계속)

audevent 옵션	설명
`-r profile`	기록할 이벤트 프로파일을 지정합니다. 프로파일은 `/etc/audit/audit.conf` 파일에 정의되어 있습니다.
`-S` or `-s system_call`	이벤트 또는 시스템 호출 감사 상태를 변경합니다.
옵션 없음	선택한 이벤트 또는 시스템 호출의 현재 상태를 표시합니다.

감사할 `admin`, `login` 및 `modaccess`를 구성하려면 다음 명령을 입력합니다.

```
# audevent -P -F -e admin -e login -e moddac
```

감사할 이벤트를 basic 프로파일에 구성하려면 다음 명령을 입력합니다.

```
# audevent -P -F -r basic
```

`Audit Success` 및 `Audit Failure`는 모두 성공 및 실패한 이벤트나 시스템 호출을 모니터링하는 이벤트 유형으로 설정됩니다. 이것이 시스템을 실행하는 데 권장되는 최소 이벤트 유형 선택 사항입니다.

일반적으로 이벤트가 감사 대상으로 선택되고 이벤트를 시작하는 사용자가 감사 대상으로 선택될 때 기록이 작성됩니다. 그러나 감사 대상으로 선택되지 않은 사용자가 세션을 시작하고 종료할 때 기록이 생성될 수도 있습니다. 이러한 기록은 사용자 선택 사항이 아니라 이벤트 선택 사항을 기반으로 하는 시스템 범위 정보로 간주됩니다. 자체 감사를 수행하는 프로그램에서 임의 결정에 따라 사용자 선택 사항을 무시할 수 있지만 이 기능은 자체 감사에서 사용하지 않는 것이 좋습니다.

10.5 감사 기록

모든 감사 데이터는 감사 기록에 작성됩니다. 일반 모드에서 파일 시스템의 감사 기록은 디렉토리로 표시되고 여러 개의 로그 파일로 구성되어 있습니다. 로그 파일 수는 데이터 로깅에 사용되는 작성기 스레드 수에 따라 달라집니다. 또한 디렉토리에 있는 하나 이상의 파일이 아니라 전체 디렉토리가 분석이나 표시에 사용되는 데이터를 나타냅니다. 일반 모드와 달리 HP-UX 11i v3 릴리즈에서는 하나의 파일로 표시되는 감사 기록을 생성하는 호환 모드도 제공됩니다. 호환 모드는 전적으로 이전 버전과의 호환성을 위해 지원되며 HP-UX 11i v3 이후 릴리즈에서는 더 이상 사용되지 않습니다. 자세한 내용은 *audsys*(1M) 맨페이지를 참조하십시오.

감사 시스템이 활성화된 경우 항상 감사 기록이 하나 이상 있어야 합니다. `audsys`를 사용하여 기록 이름과 기록의 다양한 속성을 지정할 수 있습니다. 현재 추적이 미리 정의된 용량(해당 AFS(Audit File Switch) 크기)에 도달하거나 이 파일이 있는 감사 파일 시스템이 미리 정의된 용량(해당 FSS(File Space Switch) 크기)에 도달할 경우 감사 하위 시스템에서 경고 메시지를 표시합니다. 현재 감사 기록의 AFS 또는 FSS에 도달하면 감사 하위 시스템은 보조 기록을 찾습니다. 보조 기록을 사용할 수 있는 경우 기록이 보조 기록으로 전환됩니다. 보조 기록이 지정되어 있지 않으면 감사 하위 시스템에서 기본 이름은 같지만 타임스탬프 확장이 다른 새 감사 기록을 만들고 기록을 시작합니다. 또한 audomon은 성공적으로 감사 기록을 전환한 후 실행되어 마지막 감사 기록을 처리하는 명령줄을 허용합니다. 사이트별 요구에 따라 데이터 백업, 아카이브, 오프사이트 이동, 정리 또는 데이터 보고 등이 처리에 포함될 수 있습니다. 자동 전환이 실패하면 적절한 관리자 작업을 요청하는 경고 메시지가 전송되며 현재 감사 기록이 계속 확장됩니다.

참고:

1. HP-UX 11i v3을 사용하면 전환할 보조 기록을 직접 명시적으로 지정할 필요가 없습니다. 감사 시스템에서 자동으로 감사 기록을 전환합니다.
2. 자동 전환이 실패하고 현재 감사 기록이 계속 증가하여 FSS 지점을 넘으면 시스템 정의 매개 변수 `minfree`에 도달할 수 있습니다. 이 지점에서 일반 사용자의 감사 가능한 모든 작업은 일시 중단됩니다. 감사 데이터를 보관하거나 공간이 있는 파일 시스템에 새로운 감사 로그 파일을 지정하여 시스템을 작업 상태로 되돌립니다.
3. 파일 시스템 공간이 다른 작업에 사용되고 있거나 선택한 파일 시스템이 선택한 AFS 크기에 대해 공간이 부족하면 Audit File Switch 지점보다 File Space Switch 지점에 먼저 도달할 수 있습니다.

감사 로그 파일을 위해 적당한 공간이 있는 파일 시스템을 선택합니다. `bdf` 명령을 사용하여 파일 시스템의 크기를 평가할 수 있습니다. 로그 파일을 최소한 다음 매개 변수로 구성하는 것이 좋습니다.

- 파일 시스템에 주 감사 로그 파일에 사용할 수 있는 공간이 5000KB 이상 있어야 합니다.
- 총 파일 공간의 20% 이상이 남아 있어야 합니다.

감사 로그 파일의 확장은 감사 데이터가 손실되지 않도록 감사 오버플로 모니터 데몬인 `audomon`에 의해 엄밀하게 모니터링됩니다.

10.5.1 감사 기록 구성

`audsys` 명령을 사용하여 감사 데이터를 수집할 주 로그 파일과 선택적인 보조 로그 파일을 지정합니다.

#**audsys -n -N2 -c *my_audit_trail* -s 5000**

이 예제에서는 감사 시스템을 시작하고 두 개의 작성기 스레드를 사용하여 `my_audit_trail` 디렉토리에 데이터를 기록합니다. AFS 크기는 5000KB로 설정됩니다. 자세한 내용은 *audsys*(1M)를 참조하십시오.

10.5.2 감사 기록 모니터링 및 관리

감사 오버플로 모니터 데몬(`audomon`)은 감사 기록을 모니터링하고 관리하는 데 사용됩니다. `audomon` 데몬은 시스템 부팅 시간에 감사가 시작될 때 자동으로 시작됩니다(`/sbin/init.d/auditing`에서 `AUDITING=1`). 권한이 부여된 사용자가 `audomon` 데몬을 시작할 수도 있습니다. 시작되면 `audomon` 데몬은 현재 감사 기록과 이 기록이 있는 파일 시스템의 용량을 모니터링합니다. `audomon` 데몬을 시작하는 데 사용되는 예제 명령은 다음과 같습니다.

audomon -p 20 -t 1 -w 90 -X "/user/local/bin/rcp_audit_trail hostname"

이 명령은 감사 시스템이 다음 명령을 사용하여 시작되었다고 가정하고 다음 동작으로 `audomon` 데몬을 시작합니다.

audsys -n -N 2 -c /var/.audit/my_trail -s 500

- `audomon`이 최소 1분 간격으로 휴면 상태를 유지합니다.
- 현재 감사 기록의 크기가 4500KB에 도달하거나 감사 기록이 있는 파일 시스템이 80% 차면 `audomon` 데몬이 현재 감사 기록에 데이터 쓰기를 중지하고 새 감사 기록 `/var/.audit/my_trail.yyyymmddHHMM`에서 기록을 시작합니다.

- 성공적으로 새 감사 기록으로 전환된 후 audomon 데몬은 다음 명령을 호출합니다.

  sh -c "/usr/local/bin/rcp_audit_trail hostname /var/.audit/my_trail"

  이 스크립트는 사이트와 관련이 있으며 이전 감사 기록을 복사하고 데이터 백업 또는 아카이브 기능을 수행하고 감사 보고서를 만드는 데 사용될 수 있습니다. audomon 데몬에 대한 자세한 내용은 *audomon*(1)을 참조하십시오.

주의:

- 감사 기록이 들어 있는 파일 시스템이 꽉 차면 감사 데이터를 생성하는 root가 아닌 프로세스는 커널에 갇히게 됩니다. 또한 root가 아닌 프로세스가 시스템 터미널에 연결되어 있으면 종료됩니다. 자세한 내용은 *audsys*(1M)의 WARNINGS 절을 참조하십시오.
- root 파일 시스템에 감사 기록을 보관하지 마십시오.

팁: 데이터 저장을 위한 장기 전략을 수행할 스크립트를 작성하고 -X 옵션을 사용하여 audomon 데몬에 전달하는 것이 좋습니다.

audomon 명령은 다음 인수를 사용합니다.

-p fss	감사 시스템이 보조 로그 파일로 전환되기까지 주 감사 로그 파일이 포함된 파일 시스템에 남아 있는 최소 공간 백분율입니다. 기본 fss 값은 20%입니다.
-t sp_freq	시스템이 감사 로그 파일 전환 지점에 대한 경고 메시지를 콘솔에 표시하는 최소 절전 모드 해제 간격(분)입니다. 기본 sp_freq 값은 1분입니다.
-w warning	경고 메시지가 콘솔로 전송되기까지 사용된 감사 로그 파일 공간 또는 사용된 최소 파일 시스템 여유 공간의 백분율입니다. 기본 warning 값은 90%입니다.
-X command	audomon이 감사 기록을 전환할 때마다 command가 실행됩니다.

자세한 내용은 *audomon*(1M)을 참조하십시오.

10.6 감사 로그 보기

감사 기능으로 인해 많은 데이터가 축적됩니다. audisp 명령을 사용하여 보려는 데이터를 선택합니다.

#**/usr/sbin/audisp *audit_trail***

audisp 명령에 사용할 수 있는 옵션은 다음과 같습니다.

-f	실패한 이벤트만 표시합니다.
-p	성공한 이벤트만 표시합니다.
-c system_call	선택한 시스템 호출을 표시합니다.
-t	지정된 시간 후에 발생한 이벤트를 표시합니다.
-s	지정된 시간 전에 발생한 이벤트를 표시합니다.
-u user-name	특정 사용자에 대한 정보를 표시합니다.
-l terminal-name	특정 터미널에 대한 정보를 표시합니다.
-e event-name	지정된 이벤트에 대한 정보를 표시합니다.

`> file-name`	지정된 파일에 출력을 씁니다.

용량이 큰 감사 로그를 사용하는 경우 보려는 기록을 준비하는 데 몇 분 정도 걸릴 수 있습니다. 감사 데이터를 볼 때는 다음과 같은 예외가 있다는 것에 주의하십시오.

- 감사 가능한 시스템 호출을 호출하는 프로그램에서 잘못된 매개 변수를 제공하면 감사 데이터가 부정확하게 표시될 수 있습니다. 감사 데이터를 보면 커널을 건너뛴 사용자 프로그램을 알 수 있습니다. 예를 들어, **`kill()`** 시스템 호출을 매개 변수 없이 호출하면 감사 기록의 매개 변수 부분에 예측할 수 없는 값이 생성됩니다.
- 파일 이름 인수를 갖는 시스템 호출에서는 장치 및 inode 정보가 제대로 기록되지 않을 수 있습니다. 호출이 성공적으로 완료되지 않으면 값은 `-1`이 됩니다.
- 이벤트 또는 시스템 호출 매개 변수를 변경하는 동안 수퍼유저를 감사하면 용량이 큰 감사 기록이 생성됩니다. 예를 들어, 감사할 이벤트 유형을 추가하면 추가될 새 이벤트 유형에 대한 것만이 아니라 감사가 활성화된 각 이벤트 유형 및 시스템 호출에 대한 기록이 생성됩니다.

10.6.1 audisp 명령 사용 예제

다음 예제에서는 `audisp` 명령을 사용하여 표시되는 감사 정보를 보여 줍니다.

- 화면에 로그 출력 표시:

  #**/usr/sbin/audisp *audit_trail***

- 로그 출력을 `/tmp/mylogoutput`으로 보내기:

  #**/usr/sbin/audisp *audit_trail* > /tmp/mylogoutput**

- 성공한 이벤트만 보기:

  #**/usr/sbin/audisp -p *audit_trail***

- 사용자 joe가 소유한 작업 보기:

  #**/usr/sbin/audisp -u joe *audit_trail***

- `ttypa` 터미널의 작업 보기:

  #**/usr/sbin/audisp -l ttypa *audit_trail***

- login 이벤트만 보기:

  #**/usr/sbin/audisp -e login *audit_trail***

10.7 자체 감사

일부 프로세스는 감사할 수 있는 일련의 작업을 호출합니다. 수집되는 감사 로그 데이터의 양을 줄이고 감사 로그 파일에 보다 의미 있는 내용을 기록하기 위해 이러한 프로세스 중 일부는 호출한 작업의 감사를 연기하고 발생한 프로세스를 설명하는 하나의 감사 로그 항목을 만들도록 프로그래밍되었습니다. 이런 방식으로 프로그래밍된 프로세스를 자체 감사 프로그램이라고 합니다. 자체 감사 프로그램을 사용하면 감사 로그 데이터가 단순화됩니다.

참고: 자체 감사 프로세스 목록은 시스템마다 다릅니다.

자체 감사 프로세스

자체 감사 기능이 있는 프로세스는 다음과 같습니다.

chfn	finger 항목 변경
chsh	로그인 셸 변경
login	로그인 유틸리티
newgrp	유효 그룹 변경
passwd	암호 변경
audevent	감사할 이벤트 선택
audisp	감사 데이터 표시
audsys	감사 시스템 시작 또는 중지
audusr	감사할 사용자 선택
init	실행 수준 변경, 사용자 로그오프
lpsched	라인 프린터 요청 예약
fbackup	융통성 있는 파일 백업
ftpd	파일 전송 프로토콜 데몬
remshd	원격 셸 서버 데몬
rlogind	원격 로그인 서버 데몬
telnetd	Telnet 서버 데몬
privrun	기존 응용 프로그램 호출[1]
privedit	권한이 부여된 사용자의 파일 편집 허용[1]
roleadm	역할 정보 편집[1]
authadm	권한 부여 정보 편집[1]
cmdprivadm	명령 권한 부여 및 권한 편집[1]

대부분의 자체 감사 프로그램은 하나의 이벤트 범주 아래에 감사 데이터를 생성합니다. 예를 들어, audsys 명령은 admin 이벤트 아래에 감사 데이터를 생성합니다. 일부 명령은 여러 개의 이벤트 범주 아래에 감사 데이터를 생성합니다. 예를 들어, init 명령은 login 및 admin 이벤트 아래에 데이터를 생성합니다.

10.8 HP-UX RBAC 감사

privrun, privedit, roleadm, authadm 및 cmdprivadm HP-UX RBAC 명령은 각각 감사 기록을 생성합니다. 각 감사 기록에 포함되어 있는 속성은 다음과 같습니다.

- 사용자 이름
- UID
- 역할
- 권한 부여(작업, 객체)
- 이벤트 시간
- 이벤트 결과(성공 또는 실패)

1. 자세한 내용은 9장을 참조하십시오.

10.8.1 HP-UX RBAC 기준 및 /etc/rbac/aud_filter 파일

HP-UX RBAC 버전 B.11.23.02 이상에서는 감사 필터 파일을 사용하여 감사할 특정 HP-UX RBAC 기준을 식별하는 기능을 지원합니다. /etc/rbac/aud_filter라는 필터 파일을 만들어 감사 기록을 생성할 특정 역할, 작업 및 객체를 식별할 수 있습니다. 감사 기록은 프로세스의 속성이 /etc/rbac/aud_filter에 있는 세 개 항목(역할, 작업 및 객체)과 모두 일치하는 경우에만 생성됩니다. 사용자 역할 및 연관된 권한 부여가 파일에 없거나 명시적으로 일치하지 않으면 역할-권한 부여와 관련된 감사 기록이 생성되지 않습니다.

권한이 부여된 사용자는 텍스트 편집기를 사용하여 /etc/rbac/aud_filter 파일을 편집하고 감사할 역할과 권한 부여를 지정할 수 있습니다. 각 권한 부여는 작업, 객체 쌍의 형식으로 지정됩니다. 역할과 연관된 모든 권한 부여는 하나의 항목으로 지정되어야 합니다. 각 줄에서 역할당 하나의 권한 부여만 지정할 수 있습니다. 그러나 * 와일드카드가 지원됩니다. /etc/rbac/aud_filter 파일에 대해 지원되는 항목과 형식은 다음과 같습니다.

```
role, operation, object
```

다음 목록에서는 각 /etc/rbac/aud_filter 항목에 대해 설명합니다.

역할 /etc/rbac/roles에 정의된 유효한 역할입니다. *를 지정하면 작업이 모든 역할에 액세스할 수 있습니다.

작업 객체에 대해 수행할 수 있는 특정 작업입니다. 예를 들어, hpux.printer.add는 프린터 추가 작업입니다. 또는 hpux.printer.*는 프린터 추가 또는 삭제 작업입니다. *를 지정하면 작업이 모든 작업에 액세스할 수 있습니다.

객체 사용자가 액세스할 수 있는 객체입니다. *를 지정하면 작업이 모든 객체에 액세스할 수 있습니다.

다음은 (hpux.passwd, /etc/passwd) 권한이 부여된 SecurityOfficer 역할 및 모든 객체에 대해 hpux.printer.add 작업을 수행하는 권한이 부여된 Administrator 역할의 감사 기록을 생성하는 방법을 지정하는 예제 /etc/rbac/aud_filter 항목입니다.

```
SecurityOfficer, hpux.passwd, /etc/passwd
Administrator, hpux.printer.add, *
```

참고: vi와 같은 편집기를 사용하여 직접 /etc/rbac/aud_filter 파일을 편집합니다. HP-UX RBAC 관리 명령은 /etc/rbac/aud_filter와 인터페이스되지 않습니다.

10.8.2 HP-UX RBAC 기준 감사 절차

다음 단계에서는 시스템에서 HP-UX RBAC 기준을 감사하는 감사 프로세스를 구성하는 방법에 대해 설명합니다.

1. 다음 명령을 사용하여 Administrator 이벤트에 대해 Passed 또는 Failed 이벤트를 감사하도록 시스템을 구성합니다.

   ```
   # audevent -PFe admin
   ```

2. 다음 명령을 사용하여 감사 출력 파일의 위치와 이름을 구성하고 시스템에서 감사를 활성화합니다.

   ```
   # audsys -n -c /tmp/aud.out -s 2048
   ```

3. HP-UX RBAC 명령을 실행합니다. 예를 들면 다음과 같습니다.

   ```
   # /usr/sbin/authadm add newauth
   ```

4. 다음 명령을 사용하여 감사 출력 파일을 열고 `authadm` 명령에 대한 기록을 검색합니다.

```
# audisp /tmp/aud.out |fgrep authadm
```

참고: HP-UX 시스템 감사에 대한 자세한 내용은 *audit*(5), *audevent*(1M), *audsys*(1M) 및 *audisp*(1M)를 참조하십시오.

A 트러스트된 시스템

이 부록에서는 트러스트된 시스템을 설정하고 관리하는 방법에 대해 설명합니다. 이 부록의 내용은 다음과 같습니다.

- 트러스트된 시스템 설치(섹션 A.1)
- 트러스트된 시스템 감사(섹션 A.2)
- 트러스트된 암호 및 시스템 액세스 관리(섹션 A.3)
- 트러스트된 백업 및 복구 지침(섹션 A.4)

참고: 트러스트된 시스템은 더 이상 사용되지 않습니다. HP-UX 11i v3은 이 제품을 지원하는 마지막 릴리즈입니다.

A.1 트러스트된 시스템 설치

트러스트된 시스템을 설치하려면 다음 단계를 수행하십시오.

1. 작업 현장에 적합한 전반적인 보안 정책을 수립합니다.
2. 시스템에 있는 모든 기존 파일에 보안 위험이 있는지 검사하고 문제를 해결합니다. 트러스트된 시스템으로 변환하기 전에 이 작업을 수행해야 합니다. 그 후에 정기적으로 또는 보안 문제가 의심될 때 파일을 조사합니다. 6장의 섹션 6.9를 참조하십시오.
3. 나중에 사용자 파일을 복구할 수 있도록 파일 시스템을 백업합니다. 또한 변환 전에 `/etc/passwd` 파일을 테이프에 백업해야 합니다.

   초기 백업 및 복구의 경우 HP-UX에서 제공하는 백업 및 복구 프로그램을 사용할 수 있습니다. 그러나 보안 기능이 구현된 후에는 ACL(액세스 제어 목록)을 보존하고 복원하는 `fbackup` 및 `frecover`만 사용합니다. 자세한 내용은 *fbackup*(1M) 및 *frecover*(1M)를 참조하십시오.
4. 트러스트된 시스템으로 변환합니다. 트러스트된 시스템으로 변환해도 다시 원래 상태로 되돌릴 수 있습니다.

   트러스트된 시스템으로 변환하려면 HP SMH를 실행하고 **System Security Policies**를 클릭합니다. `Convert to trusted system` 프롬프트가 표시됩니다. 확인 메시지가 표시될 수도 있습니다. **y**를 눌러 변환 프로세스를 시작합니다.

   트러스트된 시스템으로 변환하는 경우 변환 프로그램에서는 다음 작업을 수행합니다.

   - `/tcb/files/auth/`에 보호되는 새로운 암호 데이터베이스를 만듭니다.
   - 암호화된 암호를 `/etc/passwd` 파일에서 보호되는 암호 데이터베이스로 옮긴 다음 `/etc/passwd`의 암호 필드를 별표(*)로 바꿉니다.
   - 모든 사용자에게 암호를 사용하도록 요청합니다.
   - 각 사용자에 대해 감사 ID를 만듭니다. 감사 ID는 전체 사용자 내역에서 변경되지 않고 사용자를 고유하게 식별합니다. 감사 ID는 HP-UX 11i v3에서 트러스트된 시스템과 더불어 더 이상 사용되지 않으며 사용자가 성공적으로 새 로그인 세션을 시작할 때마다 동적으로 할당되는 감사 태그로 대체됩니다. 감사 태그에 대한 자세한 내용은 10장을 참조하십시오.
   - 모든 기존 사용자에 대해 감사 플래그를 설정합니다.
   - `at`, `batch` 및 `crontab` 입력 파일을 변환하여 제출자의 감사 ID를 사용합니다.
5. 감사 파일이 시스템에 있는지 확인합니다.

1. `swlist -l fileset`를 사용하여 설치된 파일 세트를 나열합니다. 감사 프로그램 파일이 들어 있는 `SecurityMon`이라는 파일 세트를 확인합니다. 목록을 줄이려면 # **`swlist -l fileset | grep Security`** 명령을 입력합니다.
2. 또한 다음 파일도 있는지 확인합니다(`SecurityMon`에 지정되어 있지 않음).
   - `/etc/rc.config.d/auditing`에는 감사를 제어하는 매개 변수가 들어 있습니다. SMH를 사용하거나 텍스트 편집기에서 직접 이 파일을 수정할 수 있습니다.
   - `/sbin/rc2.d/S760auditing`은 감사를 시작하는 스크립트입니다. 이 파일은 수정하지 마십시오.

6. 트러스트된 시스템으로 변환한 후 감사 하위 시스템을 사용하고 HP-UX 시스템을 트러스트된 시스템으로 실행할 수 있습니다.

참고: HP-UX 11i v3에서 감사 시스템은 트러스트된 시스템으로 변환되지 않고 시스템에서 작동하기도 합니다.

자세한 내용은 10장을 참조하십시오.

트러스트된 시스템에서 표준 시스템으로 다시 변환해야 하는 경우 SMH를 실행하고 **Auditing and Security** 창을 사용합니다. **Audited Events**, **Audited System Calls** 및 **Audited Users** 화면에서 모두 변환 취소 옵션을 제공합니다.

팁: 시스템이 트러스트된 시스템으로 변환되었는지 확인하는 한 가지 방법으로 `/tcb` 파일을 찾아볼 수 있습니다. 이 파일이 있으면 트러스트된 시스템입니다.

A.2 트러스트된 시스템 감사

트러스트된 시스템 감사는 트러스트된 모드로 변환되지 않은 시스템 감사와 매우 유사합니다. 감사에 대한 자세한 내용은 10장을 참조하십시오. 유일한 차이점은 감사할 사용자를 선택하는 방법입니다. 트러스트된 모드로 변환되지 않은 시스템에서는 `userdbset` 명령을 사용하여 감사할 사용자를 지정합니다. *userbdset*(1M) 및 *userdb*(4)를 참조하십시오. 연관된 속성은 `AUDIT_FLAG`라고 하며 *security*(4)에서 설명합니다. 트러스트된 시스템에서는 `audusr` 명령이 감사할 사용자를 지정합니다. 자세한 내용은 *audusr*(1M)를 참조하십시오.

A.3 트러스트된 암호 및 시스템 액세스 관리

암호는 가장 중요한 개별 사용자 식별 기호입니다. 시스템에서는 암호로 사용자를 인증하여 시스템에 액세스하도록 합니다. 암호는 사용하거나 저장하거나 알려질 때 보안상 취약해지므로 항상 비밀로 유지해야 합니다.

암호에 대한 자세한 내용은 2장을 참조하십시오.

보안 관리자의 책임

암호 보안에 대한 책임은 보안 관리자 및 시스템의 모든 사용자에게 있습니다. 보안 관리자가 수행하는 보안 작업은 다음과 같습니다.

- 새 사용자에 대해 임시 암호를 생성합니다. 처음 로그인할 때는 이 암호를 사용해야 합니다. 이 번호가 확인된 후 새 사용자에게 새 암호를 입력하라는 메시지가 표시됩니다.
- 표준 암호 파일인 `/etc/passwd` 및 트러스트된 데이터베이스 파일 `/tcb/files/auth/*`를 포함하여 모든 시스템 파일에서 적절한 사용 권한을 유지 관리합니다.

- 암호 변경기간 설정 기능을 설정합니다.
- 암호 재사용을 관리합니다.
- 만료된 암호, 시스템에 더 이상 액세스할 수 없는 사용자의 사용자 ID와 암호를 삭제하거나 무효화합니다.

사용자 책임

모든 사용자는 다음 규칙을 준수해야 합니다.

- 암호를 기억하고 항상 비밀을 유지합니다.
- 초기 암호를 즉시 변경하고 그 후에 정기적으로 암호를 변경합니다.
- 상태 변경 및 의심스러운 보안 위반을 모두 보고합니다.
- 암호를 입력할 때 아무도 보지 않도록 합니다.
- 계정이 있는 각 시스템에 대해 다른 암호를 선택합니다.

A.3.1 암호 파일

트러스트된 시스템에서는 `/etc/passwd` 파일, 보호되는 암호 데이터베이스 `/tcb/files/auth/`의 파일 등 여러 암호 파일을 유지 관리합니다("/tcb/files/auth/ 데이터베이스" 참조). 두 파일에 각 사용자에 대한 항목이 있으며, `login`에서는 두 항목을 확인하여 로그인 요청을 인증합니다.

모든 암호는 입력된 즉시 암호화되며 사용자의 보호되는 암호 데이터베이스 파일인 `/tcb/files/auth/user-char/user-name`에 저장됩니다. 암호화된 암호만 비교하는 데 사용됩니다.

모든 암호 파일에서 빈(null) 암호 필드를 허용하지 마십시오. 트러스트된 시스템에서는 `/etc/passwd`의 암호 필드가 무시됩니다. 암호가 비어 있는 사용자는 트러스트된 시스템에 로그인할 때 암호를 설정하도록 요구됩니다. 그러나 이 경우에도 보안 문제의 가능성이 있으므로 이 계정에 로그인하는 모든 사용자는 암호를 설정해야 합니다.

암호 파일은 직접 편집하지 마십시오. HP SMH, `useradd`, `userdel` 또는 `usermod`를 사용하여 암호 파일 항목을 수정합니다.

A.3.1.1 /etc/passwd 파일

트러스트된 시스템은 `/etc/passwd` 파일을 사용하여 로그인할 때 사용자를 식별합니다. 파일에는 HP-UX 시스템의 모든 계정에 대한 항목이 들어 있습니다. 각 항목은 콜론으로 구분된 일곱 개의 필드로 이루어져 있습니다. 트러스트된 시스템의 일반적인 `/etc/passwd` 항목은 다음과 같습니다.

```
robin:*:102:99:Robin Hood,Rm 3,x9876,408-555-1234:/home/robin:/usr/bin/sh
```

필드에는 다음 정보(순서대로 나열)가 들어 있으며 콜론으로 구분되어 있습니다.

1. 사용자(로그인) 이름입니다. 8자 이하로 구성됩니다 (예제에서는 `robin`).
2. 사용되지 않은 암호 필드입니다. 실제 암호 대신 별표로 이루어져 있습니다. (`*`)
3. 사용자 ID입니다. 0에서 `MAXINT-1`(2,147,483,646 또는 2^{31} -2) 사이의 정수(`102`)입니다.
4. `/etc/group`의 그룹 ID입니다. 0에서 `MAXINT-1` 사이의 정수입니다. (`99`)
5. 주석 필드입니다. 사용자의 전체 이름, 위치 및 전화 번호와 같은 정보를 식별하는 데 사용됩니다. 역사적인 이유로 `gecos` 필드라고도 합니다 (`Robin Hood,Rm 3,x9876,408-555-1234`)
6. 홈 디렉토리. 사용자의 초기 로그인 디렉토리 (`/home/robin`)입니다.

7. 로그인 프로그램 경로 이름입니다. 사용자가 로그인할 때 실행됩니다(/usr/bin/sh).

사용자는 chfn 명령으로 주석 필드(다섯 번째 필드)를 변경할 수 있으며 chsh 명령으로 로그인 프로그램 경로 이름(일곱 번째 필드)을 변경할 수 있습니다. 나머지 필드는 시스템 관리자가 설정합니다. 사용자 ID는 고유해야 합니다. 자세한 내용은 *chfn*(1), *chsh*(1), *passwd*(1) 및 *passwd*(4)를 참조하십시오. 사용자는 passwd를 사용하여 보호되는 암호 데이터베이스의 암호를 변경할 수 있습니다.

A.3.1.2 /tcb/files/auth/ 데이터베이스

시스템이 트러스트된 시스템으로 변환될 때 일반적으로 /etc/passwd의 두 번째 항목으로 지정되는 암호화된 암호는 보호되는 암호 데이터베이스로 이동되며 /etc/passwd 파일의 해당 자리는 별표로 지정됩니다.

보호되는 암호 데이터베이스 파일은 /tcb/files/auth/ 계층 구조에 저장됩니다. 사용자 인증 프로파일은 사용자 계정 이름의 첫 번째 문자에 따라 이러한 디렉토리에 저장됩니다. 예를 들어, 사용자 david에 대한 인증 프로파일은 /tcb/files/auth/d/david 파일에 저장됩니다.

트러스트된 시스템에서 주요 보안 요소는 수퍼유저만 액세스할 수 있는 보호되는 암호 데이터베이스에 들어 있습니다. HP SMH를 사용하여 암호 데이터 항목을 설정합니다. 사용자에 대해 암호 데이터가 설정되지 않은 경우 /tcb/files/auth/system/default 파일에 저장된 시스템 기본값을 기본값으로 사용합니다.

보호되는 암호 데이터베이스에는 사용자에 대한 많은 인증 항목이 들어 있습니다. 다음이 포함되어 있는 이러한 항목에 대한 자세한 내용은 *prpwd*(4)를 참조하십시오.

- 사용자 이름 및 사용자 ID
- 암호화된 암호
- 계정 소유자
- 지정된 사용자가 시스템을 부팅할 수 있게 허용하는 부팅 인증. *security*(4)를 참조하십시오.
- 사용자에 대한 감사 ID 및 감사 플래그(감사 설정 여부)
- 암호 변경 사이의 최소 시간
- 최대 암호 길이
- 암호 만료 시간. 이 시간이 지나면 암호를 변경해야 합니다.
- 암호 수명. 이 시간이 지나면 계정이 잠깁니다.
- 마지막으로 암호 변경이 성공 및 실패한 시간
- 계정이 만료될 절대 시간(날짜)
- 계정이 잠기기 전에 로그인 사이에 허용되는 최대 시간
- 경고 메시지가 표시될 만료 전 기간(일)
- 사용자가 암호를 생성하였는지 또는 시스템에서 생성되었는지 여부
- 일반 단어나 잘 알려진 용어가 암호로 사용되지 않도록 방지하는 일반적인 암호 검사
- 시스템에서 생성된 암호 유형
- Null 암호
- 계정 소유자가 아닌 경우 마지막으로 암호를 변경한 사용자의 사용자 ID
- 이 계정이 로그인에 사용될 수 있을 때의 시간 간격

- 마지막으로 성공 및 실패한 이 계정의 로그인과 관련된 터미널 또는 원격 호스트 ID
- 실패한 로그인 시도 횟수. 성공적으로 로그인되면 지워집니다.
- 계정이 잠기기까지 허용되는 최대 로그인 시도 횟수

A.3.2 암호 선택 사항 및 생성

트러스트된 시스템에서는 다음과 같은 암호 생성 옵션을 사용할 수 있습니다.

- 사용자 생성 암호

  암호 검사 옵션을 사용하여 로그인 및 그룹 이름을 사용했는지, 로그인 및 그룹 이름의 순서를 바꿔 사용했는지 및 거꾸로 사용했는지를 확인할 수 있습니다.

  새 암호는 이전 암호와 적어도 3자 이상 달라야 합니다.

- 문자 조합만 사용하는 시스템 생성 암호
- 문자, 숫자 및 문장 부호의 조합을 사용하는 시스템 생성 암호
- 발음할 수 있지만 의미 없는 음절을 사용하는 시스템 생성 암호

시스템에 대해 암호 생성 옵션을 설정할 수 있습니다. 또는 사용자별로 암호 생성 옵션을 설정하여 시스템 기본값을 재정의할 수 있습니다.

각 사용자에 대해 암호 생성 옵션을 하나 이상 설정해야 합니다. 한 사용자가 여러 옵션을 사용할 수 있으면 사용자가 암호를 변경할 때 암호 생성 메뉴가 표시됩니다.

A.3.3 암호 변경기간 설정 기능

각 사용자에 대해 암호 변경기간 설정 기능을 활성화하거나 비활성화할 수 있습니다. 암호 변경기간 설정 기능이 활성화되어 있으면 시스템에서 암호에 대해 다음을 유지 관리할 수 있습니다.

최소 시간 암호 변경 사이에 필요한 최소 시간입니다. 이는 사용자가 암호를 변경했다가 새 암호를 기억하기 귀찮아서 즉시 이전 암호로 되돌리는 것을 방지합니다.

만료 시간 이 시간이 지나면 로그인할 때 사용자가 암호를 변경해야 합니다.

경고 시간 만료 전에 경고 메시지가 표시되는 시간입니다.

수명 암호가 변경되지 않은 경우 암호와 관련된 계정이 잠기는 시간입니다. 계정이 잠긴 후에는 시스템 관리자만 잠금을 해제할 수 있습니다. 잠금이 해제된 후에는 암호를 변경해야 사용자가 계정에 로그인할 수 있습니다.

암호가 변경될 때 만료 시간 및 수명 값이 다시 설정됩니다. 수명이 0이면 암호 변경기간 설정 기능이 지정되지 않습니다. 이 경우 다른 암호 만기일 기능 시간은 적용되지 않습니다.

A.3.4 암호 내역 및 암호 재사용

시스템 범위에서 암호 내역 기능을 활성화하여 사용자가 이전 암호를 재사용하지 못하도록 할 수 있습니다.

`/etc/default/security` 파일에서 `PASSWORD_HISTORY_DEPTH` 속성을 정의하여 암호 재사용 검사를 활성화합니다.

```
PASSWORD_HISTORY_DEPTH=n
```

여기서 `n`은 검사할 이전 암호 수를 지정하는 정수입니다.

사용자가 암호를 변경하면 현재 암호부터 시작하여 `n`개의 이전 암호에 대해 새 암호를 검사합니다. 시스템에서 일치 항목을 찾으면 새 암호를 거부합니다. `n`이 `2`이면 사용자가 두 암호를 서로 번갈아 사용할 수 없습니다.

자세한 내용은 *passwd*(1) 및 *security*(4)를 참조하십시오.

A.3.5 시간 기반 액세스 제어

트러스트된 시스템에서는 각 사용자에 대해 로그인 허용 시간대와 요일을 지정할 수 있습니다. 사용자가 허용된 액세스 시간 외에 로그인을 시도하면 이벤트가 기록되고(로그인 실패 및 성공에 대한 감사가 설정된 경우) 로그인이 종료됩니다. 수퍼유저는 허용된 액세스 시간 외에 로그인할 수 있지만 이벤트는 기록됩니다. 액세스 시간 사용 범위는 사용자에 대한 보호되는 암호 데이터베이스에 저장되며 HP SMH를 사용하여 설정할 수 있습니다. 범위가 끝난 후 로그인한 사용자는 기록되지 않습니다.

A.3.6 장치 기반 액세스 제어

트러스트된 시스템의 각 MUX 포트 및 전용 DTC 포트에 액세스할 수 있는 사용자 목록을 지정할 수 있습니다. 장치에 대한 목록이 비어 있으면 모든 사용자가 액세스할 수 있습니다.

장치 액세스 정보는 트러스트된 시스템의 각 터미널 장치에 대한 항목이 들어 있는 장치 지정 데이터베이스인 `/tcb/files/devassign`에 저장됩니다. 항목의 필드에는 장치를 사용할 수 있는 사용자 목록이 나열됩니다.

트러스트된 시스템에서 터미널 로그인 정보는 각 터미널에 다음 데이터를 제공하는 터미널 제어 데이터베이스인 `/tcb/files/ttys`에 저장됩니다.

- 장치 이름
- 터미널에 마지막으로 로그인에 성공한 사용자의 사용자 ID
- 터미널에 마지막으로 로그인에 성공한 시간
- 터미널에 마지막으로 로그인에 실패한 시간
- 터미널이 잠기기 전에 연속적으로 로그인에 실패한 횟수
- 터미널 잠금 플래그

이러한 트러스트된 시스템 데이터베이스에는 수퍼유저만 액세스할 수 있으며 HP SMH를 사용하여 항목을 설정할 수 있습니다. *devassign*(4) 및 *ttys*(4)를 참조하십시오.

A.3.7 트러스트된 시스템 데이터베이스 조작

다음 맨페이지에서 설명하는 라이브러리 루틴을 사용하여 암호 파일 및 다른 트러스트된 시스템 데이터베이스의 정보에 액세스할 수 있습니다.

getdvagent(3)	`/tcb/files/devassign`의 장치 항목을 조작합니다.
getprdfent(3)	`/tcb/files/auth/system/default`의 시스템 기본값을 조작합니다.
getprpwent(3)	`/tcb/files/auth/`에서 암호 항목을 가져옵니다.
getprtcent(3)	터미널 제어 데이터베이스, `/tcb/files/ttys`를 조작합니다.
getpwent(3C)	`/etc/passwd`에서 암호 항목을 가져옵니다.
putpwent(3C)	`/etc/passwd`에 암호 파일 항목을 씁니다.
getspwent(3X)	이전 버전과의 호환성을 위해 제공된 `/tcb/files/auth/`에서 암호 항목을 가져옵니다.
putspwent(3X)	이전 버전과의 호환성을 위해 제공된 `/tcb/files/auth/`에 암호 항목을 씁니다.
putprpwnam(3)	`/tcb/files/auth/`에 암호 파일 항목을 씁니다.

A.4 트러스트된 백업 및 복구 지침

트러스트된 시스템에서 백업 및 복구할 때의 지침은 다음과 같습니다.

- `fbackup` 및 `frecover`만 사용하여 선택적으로 파일을 백업하고 복구합니다. `fbackup` 및 `frecover`를 사용하는 경우에만 ACL(액세스 제어 목록)이 유지됩니다. ACL을 구현하지 않는 시스템에서 사용할 파일을 백업하고 복구할 경우에는 이러한 명령에 `-A` 옵션을 사용합니다. 자세한 내용은 *fbackup*(1M) 및 *frecover*(1M)를 참조하십시오.
- 다른 시스템에 파일을 복구할 경우 두 시스템에서 사용자의 사용자 이름과 그룹 이름이 일치해야 합니다.
- 백업 미디어는 민감한 요소이므로 필요한 경우에만 미디어에 대한 액세스를 허용합니다.
- 백업 테이프에 레이블을 붙인 다음 안전하게 보관합니다. 외부 저장소에 보관하는 것이 보안상 가장 좋습니다. 최소 6개월 동안 아카이브를 보관한 다음 미디어를 다시 사용합니다.
- 재사용하기 전에 적절한 절차에 따라 자기 미디어를 지워 데이터를 제거합니다.
- 매일 증분 백업을 수행하고 매주 전체 백업을 수행합니다.

  조직 내 정보 흐름에 백업 일정을 맞춥니다. 예를 들어, 주로 사용하는 데이터베이스가 금요일마다 업데이트되면 금요일 밤에 주별 백업을 예약할 수 있습니다.
- 일정에 따라 모든 파일을 백업해야 하는 경우에는 백업하기 전에 모든 사용자에게 로그오프하도록 요청합니다. `fbackup`을 사용하는 경우에는 백업이 수행될 때 파일이 변경되고 있으면 경고 메시지가 표시됩니다.
- 최근 백업의 로그 파일을 조사하여 백업하는 동안 발생한 문제를 식별합니다. 백업 로그 파일에 제한적인 사용 권한을 설정합니다.
- `frecover` 명령은 파일 덮어쓰기를 허용합니다. 그러나 파일에는 파일이 백업될 때 설정된 사용 권한 및 ACL이 유지됩니다.
- 사전에 복구 프로세스를 테스트하여 응급 상황에 데이터 전체를 복구할 수 있는지 확인해야 합니다.
- 다른 시스템으로부터 파일을 복구할 때 새 시스템에 사용자 및 그룹이 없으면 `chown` 명령을 실행하여 현재 상주하고 있는 시스템에 사용자 ID 및 그룹 ID를 설정할 수 있습니다. 지정된 그룹이 없는 새 시스템에 파일을 복구하면 파일은 `frecover`를 실행하고 있는 사용자의 그룹 소유권을 갖게 됩니다. 서로 다른 시스템에서 소유자 및 그룹 이름의 의미가 다르면 복구 결과를 예측할 수 없게 됩니다.
- 정전으로 파일이 손실되지는 않습니다. 그러나 정전 후 파일이 손실되었다고 보고한 사용자가 있으면 백업 테이프로부터 파일을 복원하기 전에 `/lost+found`를 찾아보십시오.
- 복구할 테이프의 내용을 확인하려면 `frecover`에 `-I` 옵션을 사용하여 테이프에 있는 파일의 인덱스를 미리 봅니다. 이제 파일 시스템의 기존 사용 권한이 백업으로 온전한 상태를 유지하게 됩니다. `frecover`를 사용할 경우 파일의 사용 권한에서 읽기가 금지되어 있으면 파일을 읽을 수 없습니다.
- `/etc/passwd` 또는 `/tcb/files`에 있는 파일과 같이 중요한 파일은 제자리에 복구하지 마십시오. 대신 임시 디렉토리(`/tmp`를 사용하면 안 됨)에 파일을 복원하고 이 디렉토리 사용 권한을 `drwx------`로 지정하여 다른 사용자가 사용할 수 없게 합니다. 복원된 파일을 대체할 파일과 비교하여 필요한 만큼 변경합니다.
- 시스템을 복구할 때 감사가 자동으로 활성화되지 않으므로 `audsys` 명령을 사용하여 감사를 설정해야 합니다.

B 기타 보안 제품

이 부록에서는 HP-UX에 사용할 수 있는 추가 보안 제품에 대해 설명합니다. 이러한 제품은 HP Software Depot(http://www.hp.com/go/softwaredepot)에서 무료로 다운로드할 수 있습니다.

B.1 HP-UX HIDS

보안 관리자는 HP-UX HIDS(Host Intrusion Detection System)를 사용하여 다음과 같이 네트워크 내의 공격을 능동적으로 모니터링, 감지 및 대응할 수 있습니다.

- 기존 공격 시나리오와 아직 알려지지 않은 일부 시나리오로부터 보호합니다. 다양한 데이터 원본에서 시스템 작업에 대한 정보를 검사하여 보안 문제나 악용을 나타낼 수 있는 패턴을 찾습니다. 잘못된 작업에는 해커의 시스템 침입 또는 방해 시도, 파괴적인 "내부자" 활동 또는 바이러스 확산 시도 등이 포함될 수 있습니다.
- 제품이 네트워크 내의 로컬 호스트 수준 보안을 향상시키는 것을 감지합니다. 네트워크에 구성된 각 호스트 시스템에서 원하지 않거나 잠재적으로 손상을 가할 수 있는 침입을 자동으로 모니터링합니다. 선택 취소하면 주요 시스템의 가용성이 손실되거나 시스템 무결성이 손상될 수 있습니다. HP-UX HIDS는 많은 유형의 이용에 대해 경고를 생성합니다.
- 다른 침입 감지 시스템과 달리 기존 공격 시나리오와 알려지지 않은 시나리오 둘 다로부터 지속적으로 보호합니다. HIDS는 감지 템플릿을 사용하여 침입을 감지합니다. 감지 템플릿은 권한이 없는 시스템 작업의 기본 유형이나 엔터프라이즈 네트워크에서 자주 발견되는 보안 공격을 식별하는 데 사용되는 빌딩 블록입니다.
- 공격의 전조인 의심스러운 작업이 있을 경우 알려줍니다. 반면 다른 침입 감지 시스템은 전적으로 운영자가 시작한 시스템 로그 파일 분석에 의존합니다. 일반적으로 운영자는 하루를 마감할 때 시스템 로그 파일을 분석합니다. 이러한 공격 분석 지연으로 인해 시스템이 상당 시간 동안 손상될 수 있습니다.

B.2 보안 패치

HP-UX SWA(Software Assistant)는 HP-UX 시스템에서 패치 관리 및 보안 정보 관리를 통합 및 간소화하는 명령줄 기반 도구입니다. SWA 도구는 2007년 1월 HP-UX 릴리즈부터 사용할 수 있는 새 제품으로 SPC(Security Patch Check)를 대신하며, HP에서 게시한 HP-UX 소프트웨어 관련 보안 정보로 최신 상태를 유지하는 데 사용되는 HP 권장 유틸리티입니다.

HP는 시스템에 대해 권한이 부여되지 않은 root 액세스를 허용하는 알려진 보안 문제를 해결하기 위한 최신 소프트웨어 패치를 제공합니다. 자세한 내용은 1장을 참조하십시오.

B.3 HP-UX IPFilter

HP-UX IPFilter는 IP 패킷을 필터링하여 시스템에 드나드는 패킷을 제어하는 시스템 방화벽입니다. 시스템의 노출 지점 수를 줄여 보안을 유지하는 역할을 합니다.

B.4 HP-UX Secure Shell

HP-UX Secure Shell은 해싱을 사용하여 데이터 무결성을 보장하고 보안 터널링 기능, 포트 전달 및 클라이언트의 개인 키를 유지 관리하는 SSH 에이전트를 제공합니다.

HP-UX Secure Shell을 사용하여 네트워크를 통해 다른 시스템에 안전하게 로그인하고 원격 시스템에서 명령을 실행하고 한 시스템의 파일을 다른 시스템으로 이동할 수 있습니다. HP-UX Secure Shell은 `rlogin`, `rsh`, `rcp`, `ftp` 및 `telnet`과 같은 비보안 명령을 대체하는 명령 집합

을 제공합니다. 또한 HP-UX Secure Shell은 다음과 같은 보안 위험으로부터 네트워크를 보호합니다.

IP 스푸핑	권한 없이 컴퓨터에 액세스하는 데 사용되는 기술입니다. 침입자는 메시지가 트러스트된 호스트에서 전송된 것처럼 해당 IP 주소를 사용하여 메시지를 컴퓨터로 보냅니다.
도청	시스템에서 암호, 신용 카드 번호 또는 기업 비밀을 검색합니다.
가로채기	공격자가 통신 당사자 간에 전송되는 데이터를 검사하고 수정할 수 있도록 네트워크 통신을 가로채는 데 사용되는 기술입니다.

용어

3DES Triple Data Encryption Standard의 약어로, 매번 다른 56비트 키를 사용하여 세 번 데이터를 암호화하는 대칭 키 블록 암호화 알고리즘입니다(키에 168비트 사용). 3DES는 대량 데이터 암호화에 적합합니다.

AAA 서버 Authentication, Authorization, and Accounting 서버의 약어입니다. AAA 서버는 네트워크 진입점에서 사용자 네트워크 액세스의 인증, 권한 부여 및 계정 서비스를 제공합니다. HP-UX는 RADIUS 프로토콜과 Diameter Base 프로토콜을 기반으로 하는 AAA 서버를 제공합니다.

ACL Access Control List의 약어로, 사용자나 다른 프린시펄이 액세스할 수 있는 리소스와 허용되는 액세스 유형을 정의하는 목록 또는 데이터베이스입니다.

AES Advanced Encryption Standard의 약어로, 대칭 키 블록 암호화 알고리즘입니다. HP-UX IPSec는 128비트 키를 사용하여 AES를 지원합니다. AES는 대량 데이터 암호화에 적합합니다.

AH Authentication Header의 약어입니다. AH는 데이터 무결성과 시스템 수준 인증을 제공하며 중계 방지 기능을 제공할 수 있습니다. AH는 IPsec 프로토콜 집합의 일부입니다.

Authentication Header AH을(를) 참조

bastion host 침입자로부터 내부 네트워크를 보호하는 컴퓨터 시스템입니다. 방화벽 및 강화된 시스템을 참조하십시오.

CA Certificate Authority의 약어로, 사용자를 인증하고 인증서를 발행하는 트러스트할 수 있는 제3자입니다. 사용자의 공개 키와 인증서에 있는 다른 보안 관련 정보 간의 바인딩을 트러스트할 수 있게 하는 것은 물론 CA는 개인 키를 사용하여 인증서 정보에 디지털 서명합니다.

challenge-response 인증 인증자가 임의 값(challenge)을 인증되는 사용자나 프린시펄에 보내는 인증 형식입니다. 사용자는 challenge 값 및 인증자와 이전에 설정한 MD5 해시 값 등의 공유 비밀 값을 기반으로 응답을 보냅니다.

일반적인 암호 교환과 달리 challenge-response 대화는 매번 달라지므로 침입자가 사용자 응답을 재생하여 인증을 얻을 수 없습니다.

chroot jail 프로세스 및 해당 프로세스의 사용자가 액세스할 수 있는 파일과 디렉토리를 제한하는 방법입니다. 프로세스는 지정된 기본 디렉토리(root)에서 시작하고 root 디렉토리 위의 디렉토리나 파일에 액세스할 수 없습니다.

CRL Certificate Revocation List의 약어입니다. 인증서는 시작 날짜/시간과 만료 날짜/시간으로 정의된 특정 수명과 함께 발행됩니다. 그러나 키 값이 손상된 경우 등 인증서를 해지해야 하는 경우가 발생할 수 있습니다. 이러한 경우 인증 기관에서 인증서를 해지할 수 있습니다. CA에서 정기적으로 업데이트 및 발행하고 인증서 사용자가 사용할 수 있는 CRL에 인증서 일련 번호를 포함시키면 됩니다. CA를 참조하십시오.

Data Encryption Standard DES를 참조하십시오.

DES Data Encryption Standard의 약어로, 대칭 키 블록 암호화에 56비트 키를 사용합니다. DES는 대량 데이터 암호화에 적합합니다.

DES는 크래킹되었습니다(DES를 사용하여 인코딩된 데이터가 제3자에 의해 디코딩됨).

Diameter Base RADIUS 프로토콜을 기반으로 AAA(Authentication, Authorization, and Accounting) 서비스를 제공하는 프로토콜입니다. Diameter 프로토콜은 RADIUS와 동일한 기능을 제공하지만 안정성, 보안 및 인프라가 향상되었습니다. RADIUS를 참조하십시오.

Diffie-Hellman 대칭 키를 생성하는 공개 키 방법으로, 양쪽이 공개적으로 값을 교환하고 동일한 대칭 키를 생성할 수 있습니다. prime `p`와 generator `g`로 시작하며 공개적으로 알려질 수 있습니다. 일반적으로 이러한 숫자는 잘 알려진 Diffie-Hellman 그룹에서 제공됩니다). 양쪽 모두 개인 값(`a` 및 `b`)을 선택하고 공개 값(`g**a mod p` 및 `g**b mod p`)을 생성합니다. 양쪽이 공개 값을 교환합니다. 그런 다음 각자 개인 값과 상대방의 공개 값을 사용하여 동일한 대칭 키 `(g**a)**b mod p` 및 `(g**b)**a mod p`를 생성합니다. 후속 통신에서 두 키는 모두 `g**(a*b) mod p`로 평가됩니다.

man-in-the-middle 또는 제3자 공격(스푸핑)을 방지하려면 Diffie-Hellman 방법을 인증과 조합해야 합니다. 예를 들어, Diffie-Hellman을 인증서나 사전 공유 키 인증과 함께 사용할 수 있습니다.

EAP Extensible Authentication Protocol의 약어로, 암호, Kerberos 및 challenge-response 프로토콜을 포함하여 여러 가지 인증 방법과 프로토콜을 사용하기 위한 프레임워크를 제공하는 프로토콜입니다.

Encapsulating Security Payload ESP을(를) 참조

ESP Encapsulating Security Payload의 약어로, IPsec 프로토콜 집합의 일부입니다. ESP는 기밀성(암호화)과 중계 방지 서비스를 제공합니다. 선택적인 ESP 인증 필드(인증된 ESP)를 사용하거나 인증 헤더 메시지에 중첩되어 인증과 함께 사용해야 합니다. 인증된 ESP는 데이터 출처 인증 및 비연결성(connectionless) 무결성도 제공합니다. 터널 모드에서 사용할 경우 ESP는 제한된 트래픽 흐름 기밀성을 제공합니다.

Extensible Authentication Protocol EAP을(를) 참조

Fine-grained 권한 특정 하위 수준 작업을 수행하는 권한(예: 특정 시스템 호출을 실행하는 권한)입니다.

HMAC Hashed Message Authentication Code의 약어입니다. MAC를 참조하십시오.

IKE IKE(Internet Key Exchange) 프로토콜은 IPsec 프로토콜 집합의 일부입니다. IKE는 사용할 암호화 및/또는 인증 서비스를 확인하기 위해 IPsec ESP 또는 AH 프로토콜 교환 전에 사용됩니다. 또한 IKE는 ESP 및 AH에서 사용하는 대칭(공유) 암호화 키의 배포와 업데이트를 관리합니다. ESP 및 AH를 참조하십시오.

IPSec 정책 IPSec 정책은 데이터의 안전한 전송을 위해 준수되는 규칙을 지정합니다. 일반적으로 IPSec 정책에는 패킷 필터 정보와 동작이 포함되어 있습니다. 패킷 필터는 패킷에 대한 정책을 선택하는 데 사용되고 이 정책을 사용하여 동작이 패킷에 적용됩니다.

Kerberos 클라이언트 또는 서버 응용 프로그램에 강력한 인증을 제공하는 네트워크 인증 프로토콜입니다. Kerberos를 사용하면 사용자가 네트워크를 통해 암호화되지 않은 암호를 전송하지 않고도 자신을 인증할 수 있습니다.

LDAP(Lightweight Directory Access Protocol)

LDAP 프로토콜은 네트워크 디렉토리 액세스를 제공합니다. LDAP는 OSI X.500 디렉토리 서비스와 유사한 디렉토리 구조를 사용하지만 데이터를 문자열로 저장하고 OSI 네트워크 스택 대신 TCP/IP 네트워크 스택을 사용합니다.

MAC

MAC(Message Authentication Code)는 인증 알고리즘 응용 프로그램에 의해 비밀 키와 함께 파생되는 메시지에 대한 인증 태그이며, 체크섬이라고도 합니다. MAC는 디지털 서명과 달리 의도된 수신자만 확인할 수 있도록 동일한 키를 사용하여 계산되고 확인됩니다.

HMACS(Hash function-based MAC)는 해시 기능과 함께 키를 사용하여 메시지에 추가되는 체크섬을 생성합니다. 예를 들어, 키가 지정된 MD5 메시지 인증 방법이 있습니다.

MAC는 블록 암호화에서 파생될 수도 있습니다. 데이터는 DES CBC를 사용하여 메시지 블록으로 암호화되고 암호화 텍스트의 최종 블록은 체크섬으로 사용됩니다. DES-CBC MAC는 널리 사용되는 미국 및 국제 표준입니다.

man-in-the-middle 공격

제3자 공격을(를) 참조

MD5

Message Digest-5의 약어로, RSA에서 개발된 인증 알고리즘입니다. MD5는 128비트 키를 사용하는 128비트 메시지 다이제스트를 생성합니다. IPSec는 메시지 다이제스트를 96비트로 자릅니다.

NAT

Network Address Translation의 약어로, 내부 전용 네트워크의 여러 시스템이 하나의 공개 인터넷 IP 주소를 공유할 수 있는 방법입니다. NAT 게이트웨이는 내부 네트워크에서 공개 인터넷으로 패킷을 전달할 때 내부 IP 주소와 포트를 공개 IP 주소로 바꾸고(변환) 반환 경로에 대해 역변환을 수행합니다.

out-of-band 키 교환

직접 대면이나 전화 통화와 같이 일반 컴퓨터 통신 채널 외부에서 보안 통신 채널을 사용한 키 교환입니다.

PAM

Pluggable Authentication Module의 약어로, 시스템 관리자가 시스템 로그인 유틸리티와 같은 HP-UX 유틸리티에 대해 인증, 계정 관리, 세션 관리 및 암호 관리 서비스를 구성할 수 있는 인증 프레임워크입니다.

PFS(Perfect Forward Secrecy)

Perfect Forward Secrecy를 사용하면 키 하나가 노출될 경우 해당 키에 의해 보호되는 데이터에 대한 액세스만 허용됩니다.

Pluggable Authentication Module

PAM을(를) 참조

RADIUS

RADIUS(Remote Authentication Dial-In User Service) 프로토콜은 네트워크 서비스에 대한 액세스를 관리하기 위해 널리 사용되고 구현됩니다. RADIUS는 인증, 권한 부여 및 계정 작업을 수행하기 위해 네트워크 액세스 장치와 AAA(Authentication, Authorization, and Accounting) 서버 간의 정보 교환 표준을 정의합니다. RADIUS AAA 서버는 인증(사용자 이름 및 암호 확인), 전달할 서비스 유형을 지정하는 구성 정보 및 사용자 액세스를 제한할 수 있는 적용할 정책을 위해 사용자 프로파일을 관리할 수 있습니다.

RADIUS 프로토콜은 인증 교환을 위한 프레임워크만 제공하며 수많은 인증 방법과 함께 사용할 수 있습니다.

RBAC

Role-Based Access Control의 약어로, 시스템 리소스, 명령 및 시스템 호출에 대한 Fine-grained 액세스를 제공하는 HP-UX 메커니즘입니다. 역할에 사용자가 할당되고 사용자에게 역할에 따른 액세스 권한이 부여됩니다.

RSA Rivest, Shamir, and Adelman의 약어로, 개인 정보 보호(암호화) 및 인증(서명)을 위해 사용할 수 있는 공개-개인 키 암호화 시스템입니다. 암호화를 위해 시스템 A는 시스템 B의 공개 키를 사용하여 암호화된 데이터를 보낼 수 있습니다. 시스템 B의 개인 키로만 데이터를 해독할 수 있습니다. 인증을 위해 시스템 A는 시스템 A의 개인 키로 암호화된 디지털 서명, 다이제스트 또는 해시를 사용하여 데이터를 보냅니다. 서명을 확인하기 위해 시스템 B는 시스템 A의 공개 키를 사용하여 서명을 해독하고 해독된 해시 또는 다이제스트를 메시지에 대해 계산하는 다이제스트 또는 해시와 비교합니다.

SASL Simple Authentication and Security Layer의 약어로, 연결 기반 네트워크 응용 프로그램에 인증 서비스를 추가하는 데 사용되는 프로토콜입니다. SASL API는 프로그래머가 공통 인터페이스를 사용하여 여러 개의 인증 서비스에 액세스할 수 있는 융통성 있는 프레임워크를 제공합니다.

Secure Shell SSH을(를) 참조

Secure Sockets Layer SSL을(를) 참조

SHA1 Secure Hash Algorithm-1의 약어로, 160비트 키를 사용하여 160비트 메시지 다이제스트를 생성하는 인증 알고리즘입니다.

SSH Secure Shell의 약어로, 원격 로그인, 파일 전송 및 원격 명령 실행의 보안 대체 항목을 제공하는 네트워크 서비스 집합입니다. 또한 SSH는 보안 터널링 기능, 포트 전달 및 클라이언트의 개인 키를 유지 관리하는 SSH 에이전트를 제공합니다.

SSL Secure Sockets Layer의 약어로, 네트워크 데이터를 암호화하는 데 사용되는 프로토콜입니다. SSL 프로토콜은 데이터 스택에서 TCP 위에 있습니다. SSL는 공개/개인 키를 사용하여 프린시펄을 인증하고 개인(공유) 키를 교환합니다. 그런 다음 SSL은 개인 키를 사용하여 데이터를 암호화합니다.

VPN Virtual Private Network의 약어입니다. 글로벌 인터넷과 같은 공개 네트워크 내의 개인 네트워크입니다. VPN은 터널을 사용하여 효율적으로 물리 네트워크 내에 별도의 논리 네트워크를 만들기 때문에 가상 네트워크입니다. VPN은 외부 사용자가 전송되는 데이터를 보거나 수정할 수 없으므로 개인 네트워크입니다. 또한 호스트 ID 인증을 사용하는 VPN은 IP 주소 스푸핑으로부터 보호합니다.

감사 보안 문제를 분석하고 감지하기 위해 이벤트를 선택적으로 기록한 것입니다. HP-UX 감사 시스템은 사용자와 프로세스를 감사할 메커니즘을 제공합니다.

강화된 시스템 원치 않는 사용자나 유해한 침입 응용 프로그램으로부터 네트워크를 보호하는 장벽으로 사용되며 최소 운영 체제 기능, 사용자 및 응용 프로그램으로 구성된 컴퓨터 시스템입니다. bastion host라고도 합니다.

개인 키 암호화 공유 키 암호화을(를) 참조

객체 시스템, 파일, 프린터, 터미널, 데이터베이스 레코드 등의 시스템 또는 네트워크 리소스입니다. 권한 부여 컨텍스트에서 권한 부여는 객체에 대한 주체의 작업에 부여됩니다.

공개 키 암호화 기계적으로 관련된 두 개의 키(예: k1 및 k2)를 사용하는 암호화 방법으로, k1을 사용하여 암호화된 데이터는 k2를 사용해서만 해독할 수 있습니다. 또한 대부분의 알고리즘은 k1 소유자만 k2로 해독할 수 있는 데이터를 제대로 암호화할 수 있도록 합니다.

키 하나는 소유자에게만 알려진 개인 키여야 하지만 두 번째 키는 널리 알려질 수 있으므로(공개) 키 배포를 쉽게 관리할 수 있습니다. 공개 키 암호화는 계산

	측면에서 비용이 높기 때문에 대량 데이터 암호화에는 효율적이지 않습니다. 대신 공개 키 암호화는 일반적으로 데이터 인증에 사용됩니다. 비대칭 키 암호화(두 키가 같지 않음) 또는 공개-개인 키 암호화라고도 합니다.
공개-개인 키 암호화	개인 키 암호화을(를) 참조
공유 키 암호화	양쪽이 데이터 암호화 또는 인증을 위해 같은 키를 사용(양쪽이 같은 키 공유)하는 암호화 방법입니다. 데이터 개인 정보 보호나 인증을 제공하기 위해 양쪽 당사자만 키 값을 알 수 있습니다(개인 키여야 함). 공유 키 암호화는 공개-개인 키 암호화보다 데이터 암호화에 더 효율적이므로 대량 데이터 암호화에 사용되는 경우가 많습니다. 그러나 공유 키를 배포하거나 설정하려면 out-of-band 키 교환(예: 직접 대면을 통한 구두상 교환), Diffie-Hellman 교환 또는 기타 메커니즘이 필요합니다. 개인 키 암호화 또는 대칭 키 암호화라고도 합니다.
과도적 트러스트 관계	다른 트러스트된 엔터티를 통해 트러스트 관계를 확장합니다. A와 B가 모두 C를 트러스트하면 A와 B는 C를 통해 과도적 트러스트 관계를 사용하여 서로를 트러스트할 수 있습니다. 계층 구조에서 A와 B는 공통 root에 대한 트러스트 체인을 설정할 수 있는 경우과도적 트러스트 관계를 설정할 수 있습니다.
구획	시스템의 여러 구성 요소를 서로 분리하는 방법입니다. 제대로 구성된 경우 구획은 HP-UX 시스템과 이 시스템에 있는 데이터를 효과적으로 보호할 수 있습니다.
권한	컴퓨터 시스템에서 작업을 수행할 권한입니다.
권한 부여	액세스 제어 정보를 평가하고 주체(사용자, 호스트, 장치 또는 컴퓨터 네트워크의 다른 엔터티)가 특정 리소스나 객체에 대해 작업을 수행할 수 있는지 확인하는 프로세스입니다. 권한 부여는 일반적으로 주체의 ID가 인증된 후 수행됩니다. RBAC 컨텍스트에서 권한 부여는 구체적으로 주체와 작업 연결을 나타내고 **사용 권한**이라고도 합니다. RBAC를 참조하십시오.
대칭 키 암호화	공유 키 암호화을(를) 참조
디지털 서명	디지털 서명은 공개/개인 키 쌍을 사용하는 키가 지정된 해시 알고리즘의 변형입니다. 보낸 사람은 개인 키와 데이터를 입력으로 사용하여 디지털 서명 값을 만듭니다.
방화벽	원치 않는 사용자나 유해한 침입 응용 프로그램으로부터 네트워크를 보호하는 장벽으로 사용되는 하나 이상의 장치 또는 컴퓨터 시스템입니다. bastion host 및 강화된 시스템을 참조하십시오.
버퍼 오버플로 공격	프로세스 오류를 발생시키거나 프로세스가 악성 코드를 실행하도록 하여 시스템을 공격하는 방법입니다. 이 공격은 일반적으로 스택의 입력 버퍼를 오버플로하여 수행됩니다. 이 경우 메모리 위반 또는 프로세스를 종료시키거나 프로세스가 악성 코드를 실행하도록 하는 기타 오류가 발생합니다. 스택 버퍼 오버플로 공격을 참조하십시오.
보안 인증서	인증서을(를) 참조
비대칭 키 암호화	공개 키 암호화을(를) 참조
사전 공유 키	두 시스템에서 암호화나 인증을 위해 동의한 암호화 값입니다. 이 키는 일반적으로 out-of-band 키 교환(예: 구두상, 직접 대면 교환)을 사용하여 컴퓨터 데이터 통신 전에 교환됩니다. 공유 키 암호화를 참조하십시오.

상태 저장 패킷 필터	상위 계층 프로토콜 필드와 TCP 연결 상태 등의 상태 정보를 사용하는 패킷 필터링 유형입니다.
섀도 패스워드	사용자 암호에 추가 보안을 제공하는 구조입니다. 섀도 패스워드 구조(spwd)에는 암호화된 사용자 암호 및 passwd 구조와 함께 사용되는 기타 정보가 들어 있습니다. 섀도 패스워드 구조는 일반적으로 권한이 부여된 사용자만 읽을 수 있는 파일에 저장됩니다.
서비스 거부 공격	시스템이 요청을 처리할 수 없도록 네트워크 패킷에 응답하지 못하게 하는 공격입니다. 서비스 거부 공격은 취약한 시스템에 많은 리소스를 사용하는 거짓 요청을 쇄도하여 구현할 수 있습니다. 서비스 거부 공격은 스푸핑된 호스트(스푸퍼가 가장하는 IP 주소를 가진 호스트)가 스푸퍼와 스푸퍼가 액세스하려는 시스템 간의 교환에 참여하지 못하도록 하기 위해 호스트 스푸핑과 함께 사용되는 경우가 많습니다.
수동 키	IPSec에 대해 수동으로 구성된 암호화 키입니다. IKE(Internet Key Exchange) 프로토콜 대신 사용되어 암호화 키와 IPSec SA(Security Association)에 대한 기타 정보를 생성합니다.
스택 버퍼 오버플로 공격	프로세스가 악성 코드를 실행하도록 하여 시스템을 공격하는 방법입니다. 이 공격은 일반적으로 스택의 입력 버퍼를 오버플로하여 악성 코드를 삽입한 다음 악성 코드를 실행하도록 스택 포인터를 수정하여 수행됩니다. 버퍼 오버플로 공격을 참조하십시오.
암호화	특정 정보 소유자가 디코딩할 수 있도록 일반 데이터(또는 일반 텍스트)를 인코딩하는 프로세스입니다.
암호화	읽을 수 있는 형식의 데이터를 개인 정보 보호를 위해 읽을 수 없는 형식으로 변환하는 프로세스입니다. 암호화 기능은 일반적으로 데이터와 암호화 키(값 또는 비트 시퀀스)를 입력으로 사용합니다.
역할	역할에 할당된 사용자에게 지정된 권한 및 책임과 관련해서 연관된 의미를 가진 조직 컨텍스트 내의 작업 기능입니다.
역할 기반 액세스 제어	RBAC을(를) 참조
이벤트	파일 만들기, 파일 열기, 시스템에 로그인 등의 작업입니다.
인증	주체(사용자, 호스트, 장치 또는 컴퓨터 네트워크의 다른 엔터티)의 ID를 확인하는 프로세스입니다. 인증은 시스템의 리소스에 대한 액세스를 허용하는 전제 조건이 되기도 합니다. 또는 데이터 무결성이나 데이터를 보낸 사람의 ID를 확인하는 프로세스입니다.
인증 기관	CA을(를) 참조
인증서	보안 인증서는 공개 키를 프린시펄(특정 사용자, 시스템, 장치 또는 기타 엔터티)과 연관(또는 바인딩)시킵니다. 보안 인증서는 사용자가 트러스트하는 CA(인증 기관)라는 엔터티에 의해 발행되며, 해당 개인 키 소유자(사용자, 장치 또는 기타 엔터티)의 ID를 보증하거나 확인합니다. CA는 CA의 공개 키를 사용하여 인증서를 확인할 수 있도록 CA의 개인 키를 사용하여 인증서에 디지털 서명합니다. 자주 사용하는 공개 키 인증서 형식은 ISO(International Organization for Standardization) X.509 표준 버전 3입니다.
인증서 해지 목록	CRL을(를) 참조
작업	하나 이상의 객체에 대한 특정 액세스 모드입니다(예: 파일에 쓰기). 권한 부여 컨텍스트에서 권한 부여는 객체에 대한 주체의 작업에 부여됩니다.

제**3**자 공격
: 제3자 공격에서 공격자는 공격을 받는 A와 B 간의 패킷을 가로챕니다. A와 B는 서로 메시지를 교환하고 있다고 가정하지만 실제로 제3자와 메시지를 교환하고 있습니다. 공격자는 A의 ID를 가장하여 B와 메시지를 교환하고 B의 ID를 가장하여 A와 메시지를 교환합니다. man-in-the-middle 공격이라고도 합니다.

제약
: 프로세스의 액세스 권한을 제한하는 메커니즘 또는 메커니즘 집합입니다.

  RBAC 컨텍스트에서 제약은 필수 액세스 제어 및 Fine-grained 권한의 조합입니다. RBAC를 참조하십시오.

주체
: 컴퓨터 네트워크의 사용자, 호스트, 장치 또는 기타 엔터티입니다. 권한 부여 컨텍스트에서 권한 부여 결정이 필요한 객체에 대한 작업 개시자입니다.

패킷 필터
: 네트워크 패킷을 선택하거나 제한하는 데 사용되는 필터입니다. 패킷 필터는 네트워크 패킷 특징을 지정합니다. 일반적으로 패킷 필터는 소스 및 대상 IP 주소, 상위 계층 프로토콜(예: TCP 또는 UDP) 및 TCP 또는 UDP 포트 번호를 지정합니다. 패킷 필터에서 IPv6 헤더 유형, 상위 계층 메시지 유형(예: ICMP 메시지 유형) 및 TCP 연결 상태와 같은 기타 패킷 필드를 정의할 수도 있습니다.

프린시펄
: 사용자, 시스템, 장치 또는 기타 엔터티입니다.

필터
: 원치 않는 객체를 차단하는 메커니즘이나 액세스가 허용 또는 거부되는 객체를 지정하는 매개 변수입니다. 일반적으로 필터는 원치 않는 네트워크 패킷을 차단하는 데 사용됩니다(패킷 필터).

색인

심볼

/dev 특수 장치 파일
  보안 고려 사항, 101
/etc/d_passwd 파일
  사용하여 액세스 제어, 49
/etc/default/security, 22
/etc/dialups 파일
  사용하여 액세스 제어, 49
/etc/ftpd/ftpusers 파일
  액세스 변경, 71
/etc/group 파일, 171
/etc/inetd.sec 파일, 73
/etc/pam_user.conf 파일, 31
/etc/pam.conf 파일, 31
  시스템 범위 구성, 32
/etc/passwd 파일, 169–171
  변경, 37
  복구, 24
  응용 프로그램 사용자 계정, 26
  의사 계정 예제, 39
  제한된 계정, 26
  형식, 38
/etc/rbac/aud_filter, 166
/etc/rbac/cmd_priv, 147
  항목, 149
/etc/security.dsc 파일, 41
/etc/shadow 섀도 패스워드 파일, 38
/sbin/rc2.d/S760auditing, 170
/tcb/files/auth/ 보호된 암호 데이터베이스, 170–171
/tcb/files/auth/*/*, 169, 172, 174
/tcb/files/ttys, 174
/tmp, 175
/var.adm/userdb 파일, 42, 65
/var/adm/inetd.sec 파일
  구성, 73

A

ACL
  HFS 설정, 91
  JFS 설정, 94
  JFS와 HFS 비교, 100
  기본 JFS 항목, 98
  및 NFS, 101
  설정, 91
  최소 JFS 변경 예제, 96
  트러스트된 시스템 백업/복구, 175
AES(Advanced Encryption Standard), 179
AH(Authentication Header)
  정의, 179
at 명령, 169
audisp 명령
  감사 로그 출력 보기, 163
authadm, 145
  구문, 145
  예제, 146

B

Bastille (참조 HP-UX Bastille)
Bastion Host, 53
batch, 169
btmp 파일
  실패한 로그인 추적, 29

C

CA(인증 기관)
  정의, 179
  정의됨, 179
CDE 로그인 관리자
  로그인, 28
CDE 잠금 관리자
  구성, 48
chfn, 172
chmod 명령
  class 항목에 대한 영향, 96
  파일 액세스 권한 변경, 88
chown, 24, 172, 175
chroot jail, 83
chsh, 172
cmdprivadm, 146
  구문, 146
  예제, 147
CRL(Certificate Revocation List), 179, 185
crontab, 169

D

DES(Data Encryption Standard), 179–180, 184
Diffie-Hellman, 180
  그룹, 180

E

ESP(Encapsulating Security Payload)
  정의, 180
/etc/ftpd/ftpusers, 71
/etc/inetd.sec, 73
/etc/passwd, 24

F

fbackup 명령, 24
  트러스트된 백업, 175
Fine-grained 권한, 123
  구성, 147
frecover 명령, 24

트러스트된 복구, 175
fsck 명령
파일 손상 수정, 90
FTP
보안 유지, 70
익명 ftp 보안 유지, 70
ftpd 서버, 71

G
getacl 명령
ACL 보기, 96
getdvagent 함수, 174
getfilexsec 명령, 112, 124
getprdfent 함수, 174
getprocxsec 명령, 112, 124
getprpwent 함수, 174
getprtcent 함수, 174
getpwent 함수, 174
getspwent 함수, 174
GSS-API
SSH, 81

H
HFS, 91
HFS ACL
JFS ACL과 비교, 100
및 NFS, 101
설정, 91
작동하는 명령 및 호출, 92
High Performance File System
HFS 참조, 91
HP-UX Bastille, 53
구성 파일
만들기, 54, 56
복제, 54
다운로드, 53
대화형 사용, 54
변경 사항 되돌리기, 58
보고서
생성, 54
파일 액세스, 59
비대화형 사용, 56
사용
대화식, 54
비대화형, 56
설치, 53
수행할 작업 목록
사용, 58
수행, 55
제거, 60
팁 및 문제 해결, 59
파일
로그 파일 보기, 57
위치 및 설명, 58
편차
bastille_drift 사용, 54
구성 편차 파일 액세스, 59
상태 비교, 54
HP-UX RBAC
감사, 165
구성 요소, 135
구성 파일, 136
구획 구성, 149
기본 사용자, 144
맨페이지, 137
명령, 137
래핑, 143
문제 해결, 153
아키텍처, 138
작업, 139
HP-UX 설치
보안 고려 사항, 21
보안 패치 설치, 23
부팅 중 보안 문제 방지, 21
설치 시간 보안 옵션 설정, 23
설치 후 보안 팁, 23

I
IKE(Internet Key Exchange)
프로토콜, 180
inetd 데몬
TCP wrappers 및, 73
개요, 72
보안 유지, 73
IPSec 정책
정의, 180

J
JFS, 91
JFS ACL
HFS ACL과 비교, 100
setacl 명령을 사용하여 변경, 99
기본 항목 사용, 98
및 NFS, 101
설정, 94
최소 변경 예제, 96
Journaled File System
JFS 참조, 91

L
last 명령
사용 예제, 29
LDAP 디렉토리 서버
저장된 암호 보안 유지, 40
Logical Volume Manager
LVM 참조, 102
login 명령, 28
login 프로세스
설명, 28
lost+found 디렉토리, 24, 175
LVM, 102

M
MAC, 181

N
NFS, 104
  NFS 마운트된 파일 보호, 105
  및 ACL, 101
  서버 보안 유지, 105
  클라이언트 보안 유지, 105
NIS
  저장된 암호 보안 유지, 40

P
PAM
  개요, 30
  사용자 인증, 30
  시스템 범위 구성, 32
PAM 라이브러리, 31
PAM 서비스 모듈, 30
PAM 인증
  로그인 예제, 34
passwd 명령, 172
  예제, 37
PASSWORD_HISTORY_DEPTH 매개 변수, 173
PFS(Perfect Forward Secrecy)
  정의, 181
privedit, 152
  구문, 152
  옵션, 152
privrun, 150
  -p, 148
  구문, 150
  예제, 151
  옵션, 150
  작업, 139
prpwd, 172
putprpwnam 함수, 174
putpwent 함수, 174
putspwent 함수, 174

R
random number generator, 83
roleadm, 143
  구문, 143
  예제, 144
root
  단점, 133
root 계정
  보호, 51
root 액세스
  Restricted SMH Builder 사용, 51
  검토, 52
  모니터링, 51
  얻기, 22
RPC
  및 TCP wrappers, 74
RSA 암호화 시스템, 181–182
rsh 명령
  시스템 액세스 제한, 26

S
Sec00Tools 보안 수준, 23
Sec20MngDMZ 보안 수준, 23
Sec30DMZ 보안 수준, 23
Secure Shell
  SSH 참조, 76
SecurityMon 파일 세트, 170
setacl 명령
  ACL 변경, 96
  JFS ACL 변경, 99
setfilexsec 명령, 112, 124
setgid 프로그램, 24, 175
  관리, 43
setuid 프로그램, 24, 175
  관리, 43
SIS, 74
Software Assistant
  사용, 23
SSH, 40
  GSS-API, 81
  HP-UX 시스템, 82
  scp 클라이언트 실행, 79
  sftp 클라이언트 실행, 79
  ssh 클라이언트 실행, 78
  strong random number generator, 83
  TCP Wrappers 지원, 83
  공개 키 기반 인증, 81
  권한 모드 실행, 79
  기능, 77
  소프트웨어 구성 요소, 77
  실행, 78
  암호 인증, 81
  암호화, 77
  연관된 기술, 82
  원격 세션 보안 유지, 76
  인증, 77, 80
  포트 전달, 77
SSH-1 프로토콜, 82
SSH-2 프로토콜, 82
strong random number generator, 83
swlist 명령, 170

T
TCP wrappers
  및 SSH, 83
  스푸핑 방지, 73
TMOUT 변수
  구성, 48

U
umask 명령
  기본 파일 사용 권한 변경, 89

userdbset 명령
  사용자 속성 정의 예제, 43

V
/var/adm/inetd.sec, 73

W
who 명령
  사용자 로그인 정보 얻기, 30
wtmp 파일
  성공한 로그인 추적, 29
WU-FTPD, 71

ㄱ
감사
  기본 프로파일, 160
  명령, 156
  복구 후 설정, 24
  사용자, 155
  활성화, 156
감사 ID(aid), 169, 172–173
감사 로그 파일, 161
  기존 덮어쓰기, 163
  데이터 단순화, 164
  보기, 163
감사 이벤트, 160
  유형, 161
감사 플래그, 173
게스트 계정
  모니터링, 26
고유한 사용자 이름
  중요성, 28
고정 비트
  설정, 89
공개 키 기반 인증
  SSH에서 사용, 81
  및 호스트 기반 인증, 81
관리 도메인
  관리, 75
구획, 107
  IPC 규칙, 115
  구조 계획, 109
  권한 제한 규칙, 118
  규칙 만들기, 112
  규칙 수정, 112
  네트워크 규칙, 116
  네트워크 인터페이스 규칙, 118
  문제 해결, 119, 128
  파일 시스템 규칙, 114
  활성화, 110
권한 부여, 134, 170
  객체, 134
  구성, 145
  작업, 134
권한 부여 번호, 170
그룹 ID 프로그램 설정
  setgid 프로그램 참조, 43
그룹 ID(gid), 171
그룹 계정
  관리, 27

ㄴ
내역
  암호, 173
네트워크 관리, 76
  관리 도메인 관리, 75
  파일 보안 제어, 104
네트워크 제어 파일
  사용 권한 확인, 76, 104
네트워크 제어 파일의 사용 권한 확인, 76
논리 볼륨
  보안 고려 사항, 102

ㄷ
단일 사용자 모드
  부팅 정보, 22
도메인
  관리 관리, 75
디렉토리 액세스
  보안 유지, 89
디스크 파티션
  보안 고려 사항, 102

ㄹ
로그 파일
  감사, 161
로그인 명령, 171
로그인 배너
  보안, 50
로그인 추적 파일, 29

ㅁ
만료 시간
  암호 변경기간 설정, 173
매개 변수
  PASSWORD_HISTORY_DEPTH, 173
명령
  swlist, 170
  로그인, 171
모뎀 액세스
  관리에 대한 보안 지침, 48
모바일 연결
  보안 유지, 49

ㅂ
백업
  보안 지침, 23
  트러스트된 시스템, 169, 175
백업 미디어
  보안, 175
보안 강화 (참조 HP-UX Bastille)

보안 수준
  설치 중 선택, 23
보안 패치
  설치, 23
보안 패치 설치
  Software Assistant 사용, 23
보조 감사 로그 파일, 162
보호된 암호 데이터베이스
  /tcb/files/auth/, 170–171
복구
  보안 지침, 23
부팅
  부팅 중 보안 문제 방지, 21
부팅 인증
  사용, 22
부팅 프로세스
  얻기, 22

ㅅ
사용 권한
  네트워크 제어 파일 확인, 76, 104
사용자 ID 프로그램 설정
  setuid 프로그램 참조, 43
사용자 ID(uid), 171, 173
사용자 계정
  제한됨, 26
사용자 보안
  관리, 25
사용자 액세스
  관리, 25
사용자 이름
  고유하게 만들기, 28
사용자 인증
  PAM 로그인 예제, 34
  PAM 사용, 30
  로그인 중, 27
사전 공유 키
  정의, 181–185
섀도 패스워드, 38
선택 및 생성, 173
수명
  암호 변경기간 설정, 173
수퍼유저 액세스
  Restricted SMH Builder 사용, 51
  검토, 52
  모니터링, 51
  보호, 51
스택 버퍼 오버플로 차단, 46
스푸핑
  TCP Wrappers를 사용하여 방지, 73
  정의, 71
시간 기반 액세스 제어, 174
시스템 관리
  FTP 보안 유지, 70
  HP-UX 보안 설치, 21
  HP-UX 파일 시스템 보안, 87
  inetd 보안 유지, 73
  PAM을 사용하여 사용자 인증, 30
  root 액세스 보호, 51
  setuid 및 setgid 프로그램 관리, 43
  감사 지침, 159
  관리 도메인 관리, 75
  네트워크의 파일 보안 제어, 104
  로그인 배너 보안 유지, 50
  로그인 중 사용자 인증, 27
  무인 워크스테이션 및 터미널 보호, 46
  백업 지침, 24
  보안 문제 , 21
  보안 속성 정의, 40, 64
  보안 패치 설치, 23
  부팅 인증을 사용하여 권한 없는 액세스 방지, 22
  사용자 감사, 155
  사용자 액세스 관리, 25
  설치 시간 보안 옵션 설정, 23
  스택 버퍼 오버플로 공격 방지, 46
  안전하게 파일 시스템 마운트 및 마운트 해제, 103
  암호 관리, 35
  원격 액세스 관리, 48
  인터넷 서비스 보안 유지, 69
시스템 보안
  보안 속성 정의, 40, 64
시스템 속성
  정의, 40, 64
시스템 실행 수준
  변경, 47
  액세스 제어, 47
시스템 액세스
  원격에 대한 보안 지침, 48
실행 수준
  변경, 47
  액세스 제어, 47

ㅇ
안전하게 파일 시스템 마운트, 103
안전하게 파일 시스템 마운트 해제, 103
암호, 173
  관리, 35
  내역, 173
  데이터베이스, 169–170, 172
    /tcb/files/auth/, 170–171
  무결성, 171
  변경기간 설정, 170, 172–173
    만료 시간, 173
    수명, 173
    최소 시간, 173
  보안, 170
  생성, 173
  암호화, 171
  암호화된 필드, 171–172
  인증
    SSH에 사용, 81
  재사용, 173
  종류, 173
  좋은 암호의 기준, 36

파일
보호된 암호 데이터베이스, 169–170, 172
필드, 171
항목
조작, 174
암호 관리, 35
암호화
정의, 180, 183–184
암호화된 암호 필드, 171–172
액세스
시간 기반 액세스, 173–174
암호, 173
장치 기반 액세스, 174
터미널 제어, 173
액세스 제어 목록
ACL 참조, 91
액세스 제어 정책 전환, 136
사용자 정의, 153
인터페이스, 136
역할
구성, 143
그룹, 145
기본값, 144
작성에 대한 지침, 141
워크스테이션
무인 보호, 46
원격 세션
SSH를 사용하여 보안 유지, 76
원격 세션 보안 유지, 76
원격 액세스
관리에 대한 보안 지침, 48
원격 액세스 서비스, 69
개요, 69
원격 프로시저 호출
RPC 참조, 74
의사 계정
예제, 39
익명 FTP
보안 유지, 70
인증
PAM 로그인 예제, 34
PAM 사용, 30
SSH에 사용, 80
로그인 중, 27
부팅 사용, 22
인터넷 데몬
inetd 데몬 참조, 72
인터넷 서비스, 69
개요, 69
임시 계정
비활성화, 26

ㅈ
자체 감사 프로그램, 164
작업
작성에 대한 지침, 142
잠금 도구 (참조 HP-UX Bastille)
장치 기반 액세스 제어, 174
장치 지정 데이터베이스
트러스트된 시스템, 174
재사용
암호, 173
전원 공급 중단, 24, 175
파일 손실, 24
전화
보안 유지, 49
주 감사 로그 파일, 162

ㅊ
최소 시간
암호 변경기간 설정, 173

ㅌ
터미널
무인 보호, 46
화면 잠금 구성, 48
터미널 액세스, 173
터미널 장치 파일
보호, 47
터미널 제어 데이터베이스
트러스트된 시스템, 174
트러스트된 시스템
데이터베이스, 174
변환, 169–170
트러스트된 암호, 173
트러스트된 암호 데이터베이스, 174
트러스트됨, 173

ㅍ
파일
/etc/group, 171
/etc/passwd, 169–171
파일 보안
/dev 특수 파일의 고려 사항, 101
NFS 마운트된 파일 보호, 105
네트워크에서 제어, 104
디스크 파티션 및 논리 볼륨 보호, 102
사용자 계정과 관련된 파일 보호, 90
파일 액세스 제어, 87
파일 세트
SecurityMon, 170
파일 소유권
설정, 89
파일 손상
fsck 명령을 사용하여 검색 및 수정, 90
파일 시스템
마운트 및 마운트 해제에 대한 보안 지침, 103
파일 액세스
액세스 권한 설정, 88
파일 액세스 관리, 87
패스워드
섀도, 38
패치 설치

Software Assistant 사용, 23
필터
정의, 180, 184–185

ㅎ
함수
getdvagent, 174
getprdfent, 174
getprpwent, 174
getprtcent, 174
getpwent, 174
getspwent, 174
putprpwnam, 174
putpwent, 174
putspwent, 174
호스트 기반 인증
SSH에 사용, 81
및 공개 키 기반 인증, 81
화면 잠금
구성, 48