ACL

https://youtu.be/wbYPQtRYsnc

ACL

- 접근 제어 목록 (Access Control List)
- 특정 개체에 대한 접근 허가 목록
- XYZ 파일에 대한 ACL이 (Alice, delete)일 경우 Alice에게 XYZ 파일 삭제 권한 제공

ACL 적용 분야

- 파일 시스템

- 네트워킹

ACL - 파일 시스템

- setfacl 명령어를 사용하여 설정 가능
  - -m : 권한 수정
  - -x : 권한 삭제
  - -R : 하위 디렉토리까지 변경 (recursive)
  - -b : 지정한 권한 전부 제거

- ex) setfacl –m u:John:rwx /test
  - 유저 John에게 /test 디렉토리 rwx 권한 설정

ACL - 파일 시스템

- getfacl 명령어를 통해 설정된 ACL 권한 확인 가능

- $ getfacl /acct
- # file: file1
  # owner: Amy
  # group: acct
  user::rwx
  user:Arthur:rwx
  user:Ann:rwx
  group::r-x
  mask::rwx
  other::r-x

- 디렉토리 /acct에 대해 Arthur, Ann에 rwx가 적용된 모습
- mask : 해당 파일/디렉토리에 대해 가질 수 있는 최대 권한

ACL - 파일 시스템

- ACL 적용을 위해서는 파티션이 ACL을 지원하도록 마운트 해야 함

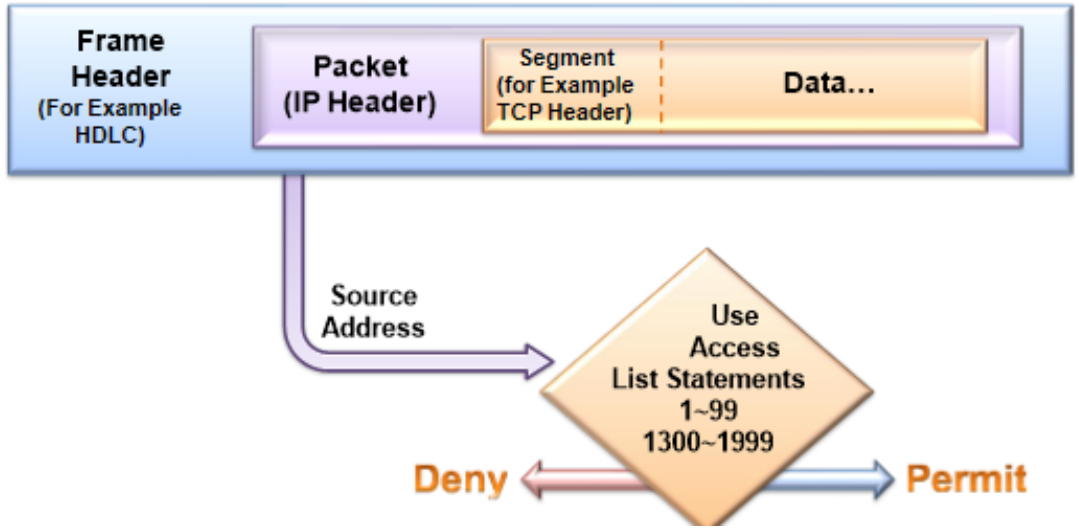
- CentOS 5.2에서는 설치 시 파티션에 ACL 자동 적용

ACL - 네트워킹

- Standard ACL (표준 ACL)
- Extended ACL (확장 ACL)
- Named Standard ACL
- Named Extended ACL
  - ACL 선언 시 번호가 아닌 사용자 설정값 사용

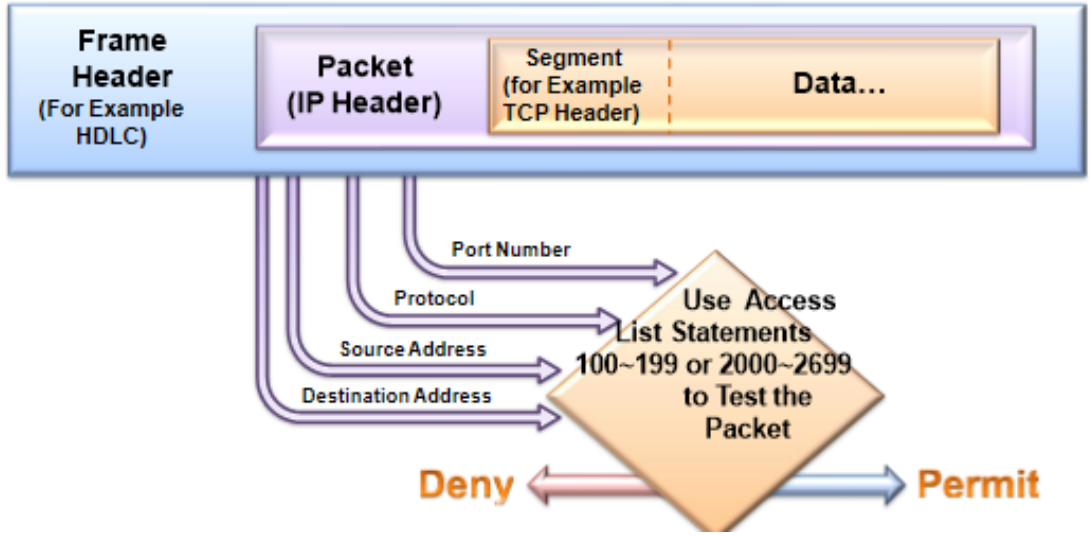
ACL - 네트워킹 - Standard ACL

- 출발지 IP 주소만 참조하여 패킷 필터링
- 1~99, 1300~1999 번호 사용
- 단점 : 패킷 자체를 막기 때문에 특정 서비스를 막고 싶어도 다른 서비스들까지 모두 막게 됨



ACL - 네트워킹 - Extended ACL

- 출발지, 목적지 IP 주소, TCP, UDP, 포트번호를 참조하여 패킷 필터링
- 100~199, 2000~2699 번호 사용



Java Spring

- 자바 웹 프레임워크 Spring의 보안 Spring Security는 ACL 설정 전용 모듈 지원

- Todo 리스트에 대한 예제 (모델 클래스명 : Todo)
  - 새로운 todo가 생성될 경우 접근 권한을 설정하는 코드

```
ObjectIdentity oid = new ObjectIdentityImpl(Todo.class, todo.getId());
MutableAcl acl = mutableAclService.createAcl(oid);
acl.insertAce(0, READ, new PrincipalSid(todo.getOwner()), true);
acl.insertAce(1, WRITE, new PrincipalSid(todo.getOwner()), true);
acl.insertAce(2, DELETE, new PrincipalSid(todo.getOwner()), true);
acl.insertAce(3, READ, new PrincipalSid(“ADMIN”), true);
acl.insertAce(4, WRITE, new PrincipalSid(“ADMIN”), true);
acl.insertAce(5, DELETE, new PrincipalSid(“ADMIN”), true);
```

Java Spring

- 어노테이션을 통해 더 간단하게 구현 가능
- 서비스 구현 클래스에 설정
- save 메소드 구현 예시

```
@Override
@PreAuthorize(“hasAuthority(‘USER’)”)
public void save(Todo todo) {
    // save code
}
```

Hyperledger Fabric

- 정책에 따른 리소스 접근 관리
- configtx.yaml에서 설정
- Signature Policy와 ImplicitMeta Policy로 나뉨

Hyperledger Fabric - Signature Policy

- 정책을 충족시키기 위해 로그인 해야 하는 사용자 식별

```
Policies:
  MyPolicy:
    Type: Signature
    Rule: “OR(‘Org1.peer’, ‘Org2.peer’)”
```

- MyPolicy 정책에 대해 Org1의 피어나 Org2의 피어 하나의 서명만 있어도 시행 가능

- AND, OR, NOutOf 함수 사용 가능

- 다양한 정책 설정 가능

Hyperledger Fabric - ImplicitMeta Policy

- Signature Policy에서 정의된 정책에 대해 보다 깊은 정의 가능

- 역할에 따른 구분 가능
  - Admins: 네트워크의 주요 운영 역할
  - Writers: 원장 업데이트 역할
  - Readers: 원장 조회 역할

```
Policies: &ApplicationDefaultPolicies
    Readers:
        Type: ImplicitMeta
        Rule: "ANY Readers"
    Writers:
        Type: ImplicitMeta
        Rule: "ANY Writers"
    Admins:
        Type: ImplicitMeta
        Rule: "MAJORITY Admins"
    MyPolicy:
        Type: Signature
        Rule: "OR('SampleOrg.admin')"
```

Q & A