Red Hat Directory Server 12

Red Hat Directory Server 보안

Directory Server의 보안 개선

Last Updated: 2024-07-10

Red Hat Directory Server 12 Red Hat Directory Server 보안

Directory Server의 보안 개선

법적 공지

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

초록

Red Hat Directory Server를 사용하여 LDAP 서비스의 보안을 개선합니다. 예를 들어 클라이언트와 디렉터리 서버 간의 연결을 암호화하고 Directory Server 데이터베이스에 암호화된 특성을 저장할 수 있습니다. 또한 복제 변경 로그를 암호화하고 인증을 구성하고 다른 보안 작업을 수행할 수도 있습니다.

차례

RED HAT DIRECTORY SERVER에 대한 피드백 제공

Red Hat의 문서 및 제품에 대한 의견을 제공해 주셔서 감사합니다. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오. 이렇게 하려면 다음을 수행합니다.

- Jira (계정 필요)를 통해 Red Hat Directory Server 설명서에 피드백을 제출하려면 다음을 수행합니다.
  1. Red Hat 문제 추적기 로 이동하십시오.
  2. **요약** 필드에 설명 제목을 입력합니다.
  3. **설명** 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
  4. 대화 상자 하단에서 **생성** 을 클릭합니다.
- Jira를 통해 Red Hat Directory Server 제품에 대한 피드백을 제출하기 위해 필요한 경우:
  1. Red Hat 문제 추적기 로 이동하십시오.
  2. **문제 생성** 페이지에서 **다음을** 클릭합니다.
  3. **Summary** 필드를 입력합니다.
  4. **Component** 필드에서 구성 요소를 선택합니다.
  5. 다음을 포함하여 **Description** 필드를 작성합니다.
     a. 선택한 구성 요소의 버전 번호입니다.
     b. 문제 또는 개선을 위한 제안을 재현하는 단계입니다.
  6. **생성**을 클릭합니다.

1장. DIRECTORY SERVER에 대한 TLS 암호화 연결 활성화

기본적으로 Red Hat Directory Server는 암호화 없이 LDAP 서비스를 제공합니다. 보안을 개선하기 위해 복제 환경의 클라이언트 또는 기타 호스트에서 암호화된 연결을 사용하도록 Directory Server에서 TLS를 구성할 수 있습니다. 그런 다음 포트 389 또는 포트 636의 LDAPS 프로토콜을 사용하여 보안 연결을 수행할 수 있습니다.

bind Distinguished Name(DN) 및 암호를 사용하거나 인증서 기반 인증을 사용하여 간단한 인증과 함께 TLS를 사용할 수 있습니다.

Directory Server의 암호화 서비스는 TLS 및 기본 암호화 함수 라이브러리인 Mozilla 네트워크 보안 서비스(NSS)에서 제공합니다. NSS에는 연방 정보 처리 표준(FIPS)이 인증된 소프트웨어 기반 암호화 토큰이 포함되어 있습니다.

1.1. DIRECTORY SERVER에 대한 암호화된 연결을 위한 다양한 옵션

암호화된 연결을 사용하여 Directory Server에 연결하려면 다음 프로토콜 및 프레임워크를 사용할 수 있습니다.

LDAPS

LDAPS 프로토콜을 사용하면 연결이 암호화 사용을 시작하고 성공하거나 실패합니다. 그러나 암호화되지 않은 데이터는 네트워크를 통해 전송되지 않습니다. 이러한 이유로 암호화되지 않은 LDAP를 통해ECDHE TLS를 사용하는 대신 LDAPS를 선호합니다.

LDAP를 통한TLS

클라이언트는 LDAP 프로토콜을 통해 암호화되지 않은 연결을 설정한 다음, TLS 명령을 보냅니다. 명령이 성공하면 추가 통신이 모두 암호화됩니다.

주의

TLS 명령이 실패하고 클라이언트가 연결을 취소하지 않으면 인증 정보를 포함한 모든 추가 데이터가 네트워크를 통해 암호화되지 않은 상태로 전송됩니다.

SASL

L(Simple Authentication and Security Layer) 프레임워크를 사용하면 Kerberos와 같은 외부

인증 방법을 사용하여 사용자를 인증할 수 있습니다.

1.2. DIRECTORY SERVER에서 NSS 데이터베이스 잠금 해제 방법

Directory Server는 CSR(인증서 서명 요청), 개인 키 및 인증서를 NSS(네트워크 보안 서비스) 데이터베이스에 저장합니다. 새 인스턴스를 설치하면 설치 프로그램이 NSS 데이터베이스를 자동으로 생성하고 임의의 암호로 보호합니다. 설치 프로그램은 다음 파일에 이 암호를 저장합니다.

- /etc/dirsrv/slapd-*instance_name*/pwdfile.txt: dsconf tls 명령은 이 파일을 사용하여 NSS 데이터베이스에 액세스합니다.
- /etc/dirsrv/slapd-*instance_name*/pin.txt: 이 파일에는 Directory Server가 시작될 때 NSS 데이터베이스의 잠금을 해제하는 토큰과 암호가 포함되어 있습니다.
  - 인스턴스를 시작할 때마다 Directory Server에서 NSS 데이터베이스 암호를 입력하라는 메시지를 표시하려면 이 파일을 제거합니다.
  - 암호를 입력하라는 메시지를 표시하지 않고 인스턴스가 자동으로 시작되도록 하려면 NSS 데이터베이스 암호를 변경할 때 이 파일을 유지하고 업데이트합니다.

/etc/dirsrv/slapd-*instance_name*/pin.txt 파일이 없으면 암호화가 활성화된 Directory Server를 시작하고 NSS 데이터베이스에 암호를 설정하면 동작은 다음과 같습니다.

- systemctl 또는 dsctl 유틸리티가 ns-slapd Directory Server 프로세스를 시작하는 경우 systemd 서비스에서 암호를 입력하라는 메시지를 표시하고 systemd-tty-ask-password-agent 유틸리티에 입력을 자동으로 전달합니다.

```
# dsctl instance_name start
Enter PIN for Internal (Software) Token: (press TAB for no echo)
```

- 드문 경우지만 ns-slapd Directory Server 프로세스가 systemctl 또는 dsctl 유틸리티에서 시작되지 않고 프로세스가 터미널에서 분리되는 경우 ns-slapd 는 wall 명령을 사용하여 모든 터미널에 메시지를 보냅니다.

```
Broadcast message from root@server (Fri 2021-01-01 06:00:00 CET):

Password entry required for 'Enter PIN for Internal (Software) Token:' (PID 1234).
Please enter password with the systemd-tty-ask-password-agent tool!
```

암호를 입력하여 다음을 수행합니다.

```
# systemd-tty-ask-password-agent
Enter PIN for Internal (Software) Token:
```

추가 리소스

- NSS 데이터베이스의 암호 변경

1.3. 명령줄을 사용하여 DIRECTORY SERVER에 대한 TLS 암호화 연결 활성화

TLS 암호화 또는 인증서 기반 인증을 사용하려면 NSS(네트워크 보안 서비스) 데이터베이스에서 인증서를 관리해야 합니다. 인스턴스를 만들 때 dscreate 유틸리티에서 /etc/dirsrv/slapd-*instance_name*/ 디렉터리에 이 데이터베이스를 자동으로 생성하고 강력한 암호로 보호합니다.

절차

1. 개인 키와 CSR(인증서 서명 요청)을 생성합니다. 외부 유틸리티를 사용하여 생성하려면 이 단계를 건너뜁니다.

   - 하나의 이름으로만 호스트에 연결할 수 있는 경우 다음을 입력합니다.

     ```
     # dsctl instance_name tls generate-server-cert-csr -s
     "CN=server.example.com,O=example_organization"
     ```

   - 여러 이름으로 호스트에 연결할 수 있는 경우:

     ```
     # dsctl instance_name tls generate-server-cert-csr -s
     "CN=server.example.com,O=example_organization" server.example.com
     server.example.net
     ```

     호스트 이름을 마지막 매개변수로 지정하면 명령에서 DNS:server.example.com, DNS:server.example.net 항목에 SAN(Subject Alternative Name) 확장을 CSR에 추가합니다.

-s *subject* 매개변수에 지정된 문자열은 RFC 1485에 따라 유효한 주체 이름이어야 합니다. 주체의 CN 필드가 필요하며 서버의 FQDN(정규화된 도메인 이름) 중 하나로 설정해야 합니다. 이 명령은 CSR을 /etc/dirsrv/slapd-*instance_name*/Server-Cert.csr 파일에 저장합니다.

2.

CSR을 **CA**(인증 기관)에 제출하여 발급된 인증서를 가져옵니다. 자세한 내용은 **CA** 설명서를 참조하십시오.

3.

CA에서 발급한 서버 인증서를 **NSS** 데이터베이스로 가져옵니다.

- **dsctl tls generate-server-cert-csr** 명령을 사용하여 개인 키를 생성한 경우 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security certificate add --file /root/instance_name.crt --name "server-cert" --primary-cert
```

--name _*certificate_nickname*매개변수에 설정한 인증서의 이름을 기록해 둡니다. 이후 단계에서 필요합니다.

- 외부 유틸리티를 사용하여 개인 키를 생성한 경우 서버 인증서와 개인 키를 가져옵니다.

```
# dsctl instance_name tls import-server-key-cert /root/server.crt /root/server.key
```

이 명령을 사용하려면 먼저 서버 인증서의 경로를 지정한 다음 개인 키 경로를 지정해야 합니다. 이 방법은 항상 인증서의 닉네임을 **Server-Cert** 로 설정합니다.

4.

CA 인증서를 **NSS** 데이터베이스로 가져옵니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate add --file /root/ca.crt --name "Example CA"
```

5.

CA 인증서의 **trust** 플래그를 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate set-trust-flags "Example CA" --flags "CT,,"
```

이렇게 하면 **TLS** 암호화 및 인증서 기반 인증에 대해 **CA**를 신뢰하도록 **Directory Server**가 구성됩니다.

6.

TLS를 활성화하고 **LDAPS** 포트를 설정합니다.

-

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-securePort=636 nsslapd-security=on
```

7. firewalld 서비스에서 LDAPS 포트를 엽니다.

```
# firewall-cmd --permanent --add-port=636/tcp
# firewall-cmd --reload
```

8. RSA 암호화 제품군을 활성화하고 NSS 데이터베이스 보안 장치 및 서버 인증서 이름을 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security rsa set --tls-
allow-rsa-certificates on --nss-token "internal (software)" --nss-cert-name Server-Cert
```

기본적으로 NSS 데이터베이스의 보안 장치 이름은 내부(소프트웨어)입니다.

9. 선택사항: 일반 텍스트 LDAP 포트를 비활성화합니다.

```
# dsconf inst security disable_plain_port
```

10. 인스턴스를 다시 시작

```
# dsctl instance_name restart
```

검증

- LDAPS 프로토콜을 사용하여 Directory Server에 대한 연결을 설정합니다. 예를 들어 쿼리를 실행합니다.

```
# ldapsearch -H ldaps://server.example.com:636 -D "cn=Directory Manager" -W -b
"dc=example,dc=com" -x -s base
```

ldap_sasl_bind(SIMPLE): LDAP 서버(-1) 오류와 함께 명령이 실패하면 디버그 수준 1로 명령을 다시 실행할 수 없습니다.

```
# ldapsearch -H ldaps://server.example.com:636 -D "cn=Directory Manager" -W -b
"dc=example,dc=com" -x -s base -d 1
```

다음 단계

- Directory Server에서 사용하는 CA 인증서를 Red Hat Enterprise Linux의 신뢰 저장소에 추가

- 선택 사항: NSS 데이터베이스의 암호 변경

- 선택 사항: 암호화 디렉터리 서버 지원 목록 업데이트

추가 리소스

- 명령줄을 사용하여 CA 신뢰 플래그 변경

1.4. 웹 콘솔을 사용하여 DIRECTORY SERVER에 대한 TLS 암호화 연결 활성화

웹 콘솔을 사용하여 TLS 암호화를 구성할 수 있습니다.

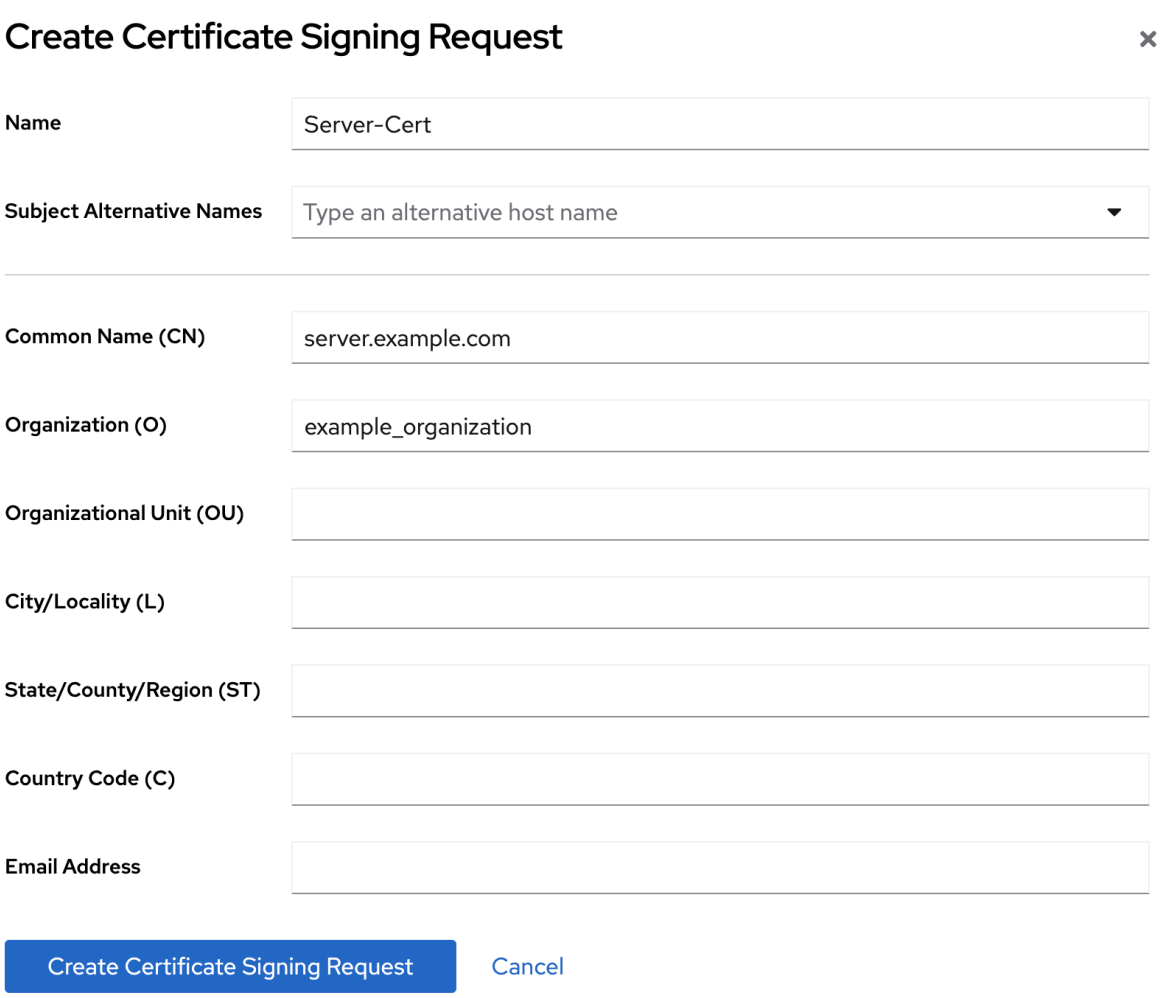
사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. Server → Security → Certificate Management → Certificate Signing Request 로 이동하여 Create Certificate Signing Request 를 클릭합니다.

2. CSR(인증서 서명 요청), CN(일반 이름) 및 조직(O)의 이름을 설정합니다.



여러 이름으로 호스트에 연결할 수 있는 경우 주체 대체 이름으로 대체 이름을 설정합니다.

3. 인증서 서명 요청 생성 을 클릭합니다.

4. **CSR** 텍스트를 보고 복사합니다.

   a. 확인할 **CSR**의 노드 옵션 아이콘을 클릭하고 **CSR** 보기를 선택합니다.

   b. **CSR** 콘텐츠를 복사합니다.

5. **CSR** 파일을 **CA**(인증 기관)에 제출하여 인증서 발급을 받습니다. 자세한 내용은 **CA** 설명서를 참조하십시오.

6. **CA**에서 인증서를 받으면 **Server** → **Security** → **Certificate Management** → **TLS** 인증서로

이동하여 서버 인증서 추가 를 클릭합니다.

7. 서버 인증서에 고유한 닉네임을 설정하고 발급된 인증서를 업로드하고 인증서 추가 를 클릭합니다.

   이후 단계에서 필요하므로 인증서 닉네임을 기억합니다.

8. **Server** → **Security** → **Certificate Management** → 신뢰할 수 있는 인증 기관으로 이동하여 **CA** 인증서 추가를 클릭합니다.

9. **CA** 인증서의 고유한 닉네임을 설정하고 **CA** 인증서 파일을 업로드하고 인증서 추가 를 클릭합니다.

10. 선택 사항: **Directory Server** 인스턴스 설치 중에 **TLS** 암호화를 활성화하지 않은 경우 활성화합니다.

    a. **Server** → **Security Settings** 로 이동하여 보안 스위치를 활성화합니다.

    b. 팝업 창에서 보안 사용을 클릭합니다.

    c. 보안 설정 페이지에서 **Save Configuration** 을 클릭합니다.

11. 보안 구성 페이지에서 서버 인증서 이름을 구성합니다.

    a. **Server** → **Security Configuration** 으로 이동합니다.

    b. 서버 인증서 이름 드롭다운 목록에서 서버 인증서 닉네임을 선택하고 **Save Configuration** 을 클릭합니다.

    c. 선택 사항: 드롭다운 목록에 인증서 닉네임이 표시되지 않으면 보안 설정 페이지를 새로 고치고 이전 단계를 다시 수행합니다.

12.

선택 사항: 636 이외의 LDAPS 포트를 사용하려면 서버 서버 → 설정으로 이동하여 LDAPS 포트를 설정한 다음 저장을 클릭합니다.

13. firewalld 서비스에서 LDAPS 포트를 엽니다.

```
# firewall-cmd --permanent --add-port=636/tcp
# firewall-cmd --reload
```

14. 선택 사항: 서버 → 보안 구성으로 이동하여 Secure Connections 필요 확인란을 선택한 다음 Save Configuration 을 클릭합니다.

Directory Server는 일반 텍스트 LDAP 포트를 비활성화합니다.

15. 오른쪽 상단에 있는 작업을 클릭하고 인스턴스 재시작 을 선택합니다.

다음 단계

- Directory Server에서 사용하는 CA 인증서를 Red Hat Enterprise Linux의 신뢰 저장소에 추가
- 선택 사항: NSS 데이터베이스의 암호 변경
- 선택 사항: 암호화 디렉터리 서버 지원 목록 업데이트

1.5. 인증서가 만료된 경우 DIRECTORY SERVER 작동 방식 관리

기본적으로 암호화가 활성화되고 인증서가 만료된 경우 Directory Server는 경고를 기록하고 서비스가 시작됩니다. 이 동작을 변경하려면 nsslapd-validate-cert 매개변수를 설정합니다. 다음 값으로 설정할 수 있습니다.

- 경고: Directory Server가 시작되고 만료된 인증서에 대한 경고를 /var/log/dirsrv/slapd-*instance_name*/error 로그 파일에 기록합니다. 이 설정은 기본 설정입니다.
- On: Directory Server에서 인증서의 유효성을 검사합니다. 인증서가 만료된 경우 인스턴스를 시작할 수 없습니다.

- **Off: Directory Server**는 인증서 만료 날짜를 확인하지 않습니다. 인스턴스가 시작되고 경고가 기록되지 않습니다.

사전 요구 사항

- **TLS** 암호화를 구성했습니다.

절차

- 다음 명령을 사용하여 **nsslapd-validate-cert** 매개변수를 변경합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-validate-cert=<value>
```

1.6. NSS 데이터베이스의 암호 변경

NSS(네트워크 보안 서비스) 데이터베이스의 암호를 변경할 수 있습니다. 예를 들어, 암호가 권한이 없는 사용자에게 알려진 경우 변경합니다.

사전 요구 사항

- 현재 **NSS** 데이터베이스 암호를 알고 있습니다.

Directory Server가 시작될 때 암호 파일을 사용하여 데이터베이스의 잠금을 해제하는 경우 암호는 **/etc/dirsrv/slapd-*instance_name*/pin.txt** 파일의 일반 텍스트에 암호화되지 않습니다.

절차

1. 다음 명령을 사용하여 **NSS** 데이터베이스 암호를 변경합니다.

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -W
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Password changed successfully.
```

2. NSS 데이터베이스 암호를 입력하라는 메시지가 표시되지 않고 암호 파일을 자동으로 시작하는 경우 이전 암호를 /etc/dirsrv/slapd-instance_name/pin.txt 의 새 암호로 교체하십시오.

- NSS 소프트웨어 암호화 모듈을 사용하는 경우 기본값입니다.

```
Internal (Software) Token:password
```

- HSM(Hardware Security Module)을 사용하는 경우:

```
name_of_the_token:password
```

검증

- 암호를 입력해야 하는 NSS 데이터베이스에서 작업을 수행합니다. 예를 들어 인스턴스의 개인 키를 나열합니다.

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -K
certutil: Checking token "NSS Certificate DB" in slot "NSS User Private Key and
Certificate Services"
Enter Password or Pin for "NSS Certificate DB":
< 0> rsa      72cb03f87381abfbb6b9e78234e2e4502ad1bfc0   NSS Certificate DB:Server-
Cert
```

새 암호를 입력한 후 명령에서 예상 출력을 표시하는 경우 암호 변경에 성공했습니다.

추가 리소스

- Directory Server에서 NSS 데이터베이스 잠금 해제 방법

1.7. NSS 데이터베이스 암호를 입력하라는 메시지가 표시되지 않고 인스턴스를 시작하도록 암호 파일 생성

새 인스턴스를 만들 때 설치 관리자는 /etc/dirsrv/slapd-instance_name/pin.txt 파일을 자동으로 생성하여 Directory Server가 네트워크 보안 서비스(NSS) 암호를 확인하지 않고 시작할 수 있도록 합니다. 그러나 이 파일을 제거하면 다시 생성할 수 있습니다.

주의

암호는 일반 텍스트로 저장됩니다. 서버가 보안되지 않은 환경에서 실행 중인 경우 암호 파일을 사용하지 마십시오.

사전 요구 사항

- NSS 데이터베이스 암호를 알고 있습니다.

절차

1. 다음 콘텐츠를 사용하여 /etc/dirsrv/slapd-instance_name/pin.txt 파일을 만듭니다.
   - NSS 소프트웨어 암호화 모듈을 사용하는 경우 기본값입니다.

     ```
     Internal (Software) Token:password
     ```

   - HSM(Hardware Security Module)을 사용하는 경우:

     ```
     name_of_the_token:password
     ```

2. 파일 권한을 설정합니다.

   ```
   # chown dirsrv:root /etc/dirsrv/slapd-instance_name/pin.txt
   # chmod 400 /etc/dirsrv/slapd-instance_name/pin.txt
   ```

검증

- 인스턴스를 다시 시작합니다.

  ```
  # dsctl instance_name restart
  ```

  시스템이 NSS 데이터베이스 암호를 묻지 않으면 Directory Server는 암호 파일을 사용합니다.

추가 리소스

- Directory Server에서 NSS 데이터베이스 잠금 해제 방법

1.8. RED HAT ENTERPRISE LINUX의 신뢰 저장소에 DIRECTORY SERVER에서 사용하는 CA 인증서 추가

Directory Server에서 TLS 암호화를 사용하도록 설정하면 CA에서 발급한 인증서를 사용하도록 인스턴스를 구성합니다. 클라이언트가 LDAPS 프로토콜 또는 LDAP를 통한 TLS 명령을 사용하여 서버에 대한 연결을 설정하는 경우 Directory Server는 이 인증서를 사용하여 연결을 암호화합니다. 클라이언트 유틸리티는 CA 인증서를 사용하여 서버의 인증서가 유효한지 확인합니다. 기본적으로 이 유틸리티는 서버 인증서를 신뢰하지 않는 경우 연결을 취소합니다.

예 1.1. 클라이언트 유틸리티에서 CA 인증서를 사용하지 않는 경우 가능한 연결 오류

- dsconf

```
# dsconf -D "cn=Directory Manager" ldaps://server.example.com:636 config get
Error: {'desc': "Can't contact LDAP server", 'info': 'error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify failed (self signed
certificate in certificate chain)'}
```

- ldapsearch

```
# ldapsearch -H ldaps://server.example.com:636 -D "cn=Directory Manager" -W -b
"dc=example,dc=com" -x
Enter LDAP Password:
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
```

Red Hat Enterprise Linux의 클라이언트 유틸리티를 사용하여 인증서 Directory Server에서 사용하는지 확인하려면 운영 체제의 신뢰 저장소에 CA 인증서를 추가합니다.

사전 요구 사항

- NSS(네트워크 보안 서비스) 데이터베이스의 암호를 알고 있습니다.

  Directory Server 인스턴스를 설치하는 동안 생성된 암호를 계속 사용하는 경우 /etc/dirsrv/slapd-instance_name/pwdfile.txt 파일의 일반 텍스트에서 이 암호를 찾습니다.

절차

1. Directory Server에서 사용하는 CA 인증서의 로컬 사본이 없는 경우:

   a. 서버의 네트워크 보안 서비스(NSS) 데이터베이스에 인증서를 나열합니다.

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

Example CA                              C,,
Server-Cert                             u,u,u
```

   b. NSS 데이터베이스에서 CA 인증서의 닉네임을 사용하여 CA 인증서를 내보냅니다.

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L -n "Example CA" -a > /tmp/ds-ca.crt
```

2. CA 인증서를 /etc/pki/ca-trust/source/anchors/ 디렉터리에 복사합니다.

```
# cp /tmp/ds-ca.crt /etc/pki/ca-trust/source/anchors/
```

3. CA 신뢰 데이터베이스를 다시 빌드합니다.

```
# update-ca-trust
```

검증

- LDAPS 프로토콜을 사용하여 Directory Server에 대한 연결을 설정합니다. 예를 들어 쿼리를 실행합니다.

```
# ldapsearch -H ldaps://server.example.com:636 -D "cn=Directory Manager" -W -b "dc=example,dc=com" -x -s base
```

추가 리소스

- update-ca-trust(8) 매뉴얼 페이지

2장. 지원되는 TLS 프로토콜 버전 구성

Red Hat Enterprise Linux 9에서는 모든 시스템 전체의 암호화 정책 프로필이 TLS 1.2를 최소로 정의합니다. 따라서 이 TLS 버전은 Directory Server에서 최솟값입니다. 그러나 최신 TLS 버전을 지원하는 클라이언트만 있는 경우 보안을 높이기 위해 더 높은 프로토콜 버전을 최소로 설정할 수 있습니다.

2.1. 명령줄을 사용하여 최소 및 최대 TLS 프로토콜 버전 설정

명령줄을 사용하여 최소 및 최대 TLS 프로토콜을 모두 설정할 수 있습니다.

주의

최대 TLS 프로토콜을 설정하지 마십시오. 이렇게 하는 경우 클라이언트는 기본 표준보다 약한 TLS 프로토콜을 사용해야 할 수 있습니다. 최대 TLS 버전을 설정하지 않으면 Directory Server는 항상 지원되는 정식 버전을 사용합니다.

사전 요구 사항

- Directory Server에서 TLS 암호화를 활성화했습니다.

절차

1. 선택 사항: Directory Server에서 현재 활성화된 TLS 프로토콜을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security get | egrep -i
"sslVersionMin|sslVersionMax"
sslversionmin: TLS1.2
sslversionmax: TLS1.3
```

2. 최소 TLS 프로토콜을 설정합니다. 예를 들어 TLS 1.3으로 설정하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security set --tls-
protocol-min="TLS1.3"
```

매개변수를 TLS 1.2보다 낮은 값으로 설정할 수 없습니다. 이는 모든 RHEL 시스템 전체의 암호화 정책 프로필 중 최소값입니다.

3. 권장되지 않음: 지원되는 최고 TLS 프로토콜을 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security set --tls-protocol-max="TLS1.3"
```

--tls-protocol-max 를 --tls-protocol-min 보다 낮은 값으로 설정하면 Directory Server에서 최대 프로토콜을 최소값과 동일한 값으로 설정합니다.

항상 가중하게 지원되는 최대 TLS 버전으로 암호화 프로토콜을 사용하려면 --tls-protocol-max 를 설정하지 마십시오.

4. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

검증

1. 지원되는 TLS 프로토콜을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security get | egrep -i "sslVersionMin|sslVersionMax"
sslversionmin: TLS1.3
sslversionmax: TLS1.3
```

2. openssl 유틸리티를 사용하여 특정 TLS 프로토콜을 사용하여 보안 클라이언트 연결을 설정합니다.

```
# echo | openssl s_client -connect server.example.com:636 -tls1_3
...
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
...
```

2.2. 웹 콘솔을 사용하여 최소 및 최대 TLS 프로토콜 버전 설정

웹 콘솔을 사용하여 최소 및 최대 TLS 프로토콜을 모두 설정할 수 있습니다.

주의

최대 **TLS** 프로토콜을 설정하지 마십시오. 이렇게 하는 경우 클라이언트는 기본 표준보다 약한 **TLS** 프로토콜을 사용해야 할 수 있습니다. 최대 **TLS** 버전을 설정하지 않으면 **Directory Server**는 항상 지원되는 정식 버전을 사용합니다.

사전 요구 사항

- **Directory Server**에서 **TLS** 암호화를 활성화했습니다.
- 웹 콘솔에서 **Directory Server** 인스턴스에 로그인되어 있습니다.

절차

1. **Server → Security** 로 이동합니다.
2. 최소 **TLS** 버전 필드에서 최소 **TLS** 프로토콜을 설정합니다.
3. 권장되지 않음: 최대 **TLS** 버전 필드에서 지원되는 최고 **TLS** 프로토콜을 설정합니다.
4. **Save Settings** 을 클릭합니다.
5. 오른쪽 상단에 있는 작업을 클릭하고 인스턴스 재시작 을 선택합니다.

검증

- **openssl** 유틸리티를 사용하여 특정 **TLS** 프로토콜을 사용하여 보안 클라이언트 연결을 설정합니다.

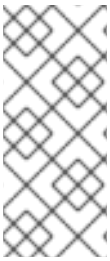
```
# echo | openssl s_client -connect server.example.com:636 -tls1_3
...
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
...
```

3장. 암호화된 연결에 LDAPS 또는TLS 필요

네트워크를 통해 암호화되지 않은 암호 전송을 방지하려면 서버에 연결할 때 사용자가 **LDAPS** 또는 **TLS** 암호화를 사용하도록 **Directory Server**를 구성할 수 있습니다.

3.1. LDAPS 또는TLS로 암호화된 연결만 허용하도록 명령줄을 사용하여 DIRECTORY SERVER 구성

기본적으로 **Directory Server**는 보안 위험인 암호화되지 않은 연결을 통한 바인딩 **DN** 및 암호를 사용하여 인증을 허용합니다. 인증서 기반 인증 또는 **SASL**과 같은 대체 보안 메커니즘을 사용할 수 없다고 가정합니다. 이 경우 **TLS** 또는**TLS**를 사용하여 서버에 인증할 때 암호화된 연결을 요구하도록 **Directory Server**를 구성할 수 있습니다.



참고

바인딩 작업에 대해 보안 연결이 필요한 경우 인증된 바인딩에만 적용됩니다. 익명 및 인증되지 않은 바인딩과 같은 암호 없이 작업을 바인딩하면 표준 연결을 진행할 수 있습니다.

사전 요구 사항

- 보안 바인딩을 사용하도록 복제 계약과 같은 기존 서버 간 연결을 구성했습니다.

절차

1. **nsslapd-require-secure-binds** 구성 매개변수를 **on** 으로 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-require-secure-binds=on
```

2. 선택 사항: **LDAPS**를 사용하려면 일반 텍스트 **LDAP** 포트를 비활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security
disable_plain_port
```

3. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

중요

이 기능을 활성화하면 모든 연결에 필요합니다. 예를 들어 복제 계약, 동기화 및 데이터베이스 체인 작업이 포함됩니다.

추가 리소스

- 인증 방법을 기반으로 액세스 정의

3.2. LDAPS 또는TLS로 암호화된 연결만 허용하도록 웹 콘솔을 사용하여 DIRECTORY SERVER 구성

기본적으로 Directory Server는 보안 위험인 암호화되지 않은 연결을 통한 바인딩 DN 및 암호를 사용하여 인증을 허용합니다. 인증서 기반 인증 또는 SASL과 같은 대체 보안 메커니즘을 사용할 수 없다고 가정합니다. 이 경우 TLS 또는TLS를 사용하여 서버에 인증할 때 암호화된 연결을 요구하도록 Directory Server를 구성할 수 있습니다.

참고

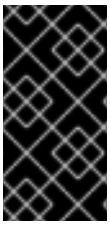
바인딩 작업에 대해 보안 연결이 필요한 경우 인증된 바인딩에만 적용됩니다. 익명 및 인증되지 않은 바인딩과 같은 암호 없이 작업을 바인딩하면 표준 연결을 진행할 수 있습니다.

사전 요구 사항

- 보안 바인딩을 사용하도록 복제 계약과 같은 기존 서버 간 연결을 구성했습니다.
- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. Server → Security Configuration (서버 보안 보안 구성)으로 이동하여 Require Secure Connections 옵션을 선택하고 Save Configuration 을 클릭합니다.
2. 선택 사항: LDAPS를 사용하려면 서버 서버 설정 일반 설정으로 이동하여 LDAP 포트 를 0 으로 설정하여 일반 텍스트 LDAP 포트를 비활성화합니다. 저장을 클릭합니다.
3. 오른쪽 상단에 있는 작업을 클릭하고 인스턴스 재시작 을 선택합니다.



중요

이 기능을 활성화하면 모든 연결에 필요합니다. 예를 들어 복제 계약, 동기화 및 데이터베이스 체인 작업이 포함됩니다.

추가 리소스

- 인증서 기반 인증 구성

4장. 암호화 디렉터리 서버 지원

암호화된 연결을 설정하려면 Directory Server 및 클라이언트 둘 다 하나 이상의 공통 암호가 필요합니다. 예를 들어 레거시 애플리케이션에 Directory Server에서 기본적으로 활성화되어 있지 않은 암호가 필요한 경우 이를 활성화할 수 있습니다.

4.1. 기본 암호화와 사용 가능한 암호의 차이점

구성에 개별 암호를 나열하는 대신 nsSSL3Ciphers 매개변수에서 다음 키워드 중 하나를 사용할 수 있습니다.

- default: 네트워크 보안 서비스(NSS)에서 활성화된 기본 암호를 나타냅니다. 목록을 표시하려면 다음을 입력합니다.

```
# /usr/lib64/nss/unsupported-tools/listsuites | grep -B1 --no-group-separator "Enabled"
```

default 키워드는 nsSSL3Ciphers 매개변수의 기본값입니다.

- 모두: Directory Server에서 지원되는 모든 암호를 나타냅니다. 목록을 표시하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ciphers list --supported
```

특정 암호만 활성화하려면 all 키워드를 사용합니다. 예를 들어 nsSSL3Ciphers 를 -all,+TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 로 설정하면 모든 암호를 비활성화하고 TLS_ECDSA_WITH_AES_256_GCM_SHA384만 사용하도록 디렉터리 서버를 구성합니다.

4.2. 약한 암호화

기본적으로 Directory Server는 약한 암호를 거부하고 이를 지원하도록 Directory Server를 구성해야 합니다.

다음과 같은 경우 암호가 약한 것으로 간주됩니다.

- export될 수 있습니다.

내보내기 가능한 암호는 암호화 이름에서 **EXPORT** 로 레이블이 지정됩니다. 예를 들어 **TLS_RSA_EXPORT_WITH_RC4_40_MD5** 에서 다음을 수행합니다.

- 이는 **3DES** 알고리즘보다 대칭적이고 약한 것입니다.

  대칭 암호는 암호화 및 암호 해독 모두에 동일한 암호화 키를 사용합니다.

- 키 길이는 **128**비트보다 짧습니다.

4.3. 암호 설정 DIRECTORY SERVER에서 명령줄 사용 지원

Directory Server에서 지원되는 암호 목록을 업데이트하려면 **nsSSL3Ciphers** 매개변수를 업데이트합니다.

사전 요구 사항

- **Directory Server**에서 **TLS** 암호화를 활성화했습니다.

절차

1. 활성화된 암호 목록을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ciphers list --enabled
```

2. 약한 암호를 활성화해야 하는 경우 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security set --allow-insecure-ciphers on
```

3. **nsSSL3Ciphers** 매개변수를 업데이트합니다. 예를 들어 **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384** 및 **TLS_ECDHE_WITH_AES_256_GCM_SHA384** 암호만 활성화하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ciphers set --"-all,+TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,+TLS_ECDHE_RSA_WITH_AE
```

```
S_256_GCM_SHA384"
```

-- 을 사용하여 쉘이 -all 의 - 문자를 명령에 대한 옵션으로 해석하지 않도록 합니다. 오류를 생성할 수 있으므로 \ 문자를 사용하여 이스케이프하지 마십시오. 그러면 다른 암호 선택이 생성됩니다.

4. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

검증

- 활성화된 암호 목록을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ciphers list
default
+TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
+TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
```

추가 리소스

- 기본 암호화와 사용 가능한 암호의 차이점
- 약한 암호화

4.4. 암호화 방식 설정 DIRECTORY SERVER에서 웹 콘솔 사용 지원

Directory Server 웹 콘솔의 Cipher preferences 메뉴에서 암호화 설정을 구성할 수 있습니다.

사전 요구 사항

- Directory Server에서 TLS 암호화를 활성화했습니다.
- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. 약한 암호를 활성화해야 하는 경우:

   a. **Server → Security Configuration** 으로 이동합니다.

   b. **Allow Weak Ciphers** 를 선택합니다.

   c. **Save Settings** 을 클릭합니다.

2. **Server → Security → Cipher preferences** 로 이동합니다.

3. 암호 설정을 업데이트합니다. 예를 들어 **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384** 및 **TLS_ECDHE_WITH_AES_256_GCM_SHA384** 암호화만 활성화하려면 다음을 수행합니다.

   a. **Cipher Suite** 필드에서 **No Ciphers** 를 선택합니다.

   b. **Allow Specific Ciphers** 필드에 **TLS_ECDSA_WITH_AES_256_CBC_SHA384** 를 입력합니다.

4. **Save Settings** 을 클릭합니다.

5. **Actions → Restart Instance** (인스턴스 다시 시작)를 클릭합니다.

검증

- **Server → Security → Cipher preferences** 로 이동합니다. **Enabled Ciphers** 목록에 활성화된 암호가 표시됩니다.

5장. CA 신뢰 플래그 변경

CA(인증 기관) 신뢰 플래그는 Directory Server가 CA 인증서를 신뢰하는 시나리오를 정의합니다. 예를 들어, 서버에 대한 TLS 연결 및 인증서 기반 인증에 대한 인증서를 신뢰하도록 플래그를 설정합니다.

5.1. 명령줄을 사용하여 CA 신뢰 플래그 변경

CA(인증 기관) 인증서에 다음 신뢰 플래그를 설정할 수 있습니다.

- C: 신뢰할 수 있는 CA
- T: 신뢰할 수 있는 CA 클라이언트 인증
- c: Valid CA
- P: 신뢰할 수 있는 피어
- P: 유효한 피어
- U : 개인 키

TLS, 이메일, 오브젝트 서명의 세 가지 범주로 쉼표로 구분된 신뢰 플래그를 지정합니다.

예를 들어 TLS 암호화 및 인증서 기반 인증에 CA를 신뢰하려면 신뢰 플래그를ECDHE ,.

사전 요구 사항

- CA 인증서를 NSS(네트워크 보안 서비스) 데이터베이스로 가져왔습니다.

절차

1. 다음 명령을 사용하여 **CA** 인증서의 신뢰 플래그를 변경합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate
set-trust-flags "Example CA" --flags "trust_flags"
```

검증

- **NSS** 데이터베이스의 모든 인증서를 표시합니다.

```
# certutil -d /etc/dirsrv/slapd-instance_name/ -L

Certificate Nickname                                         Trust Attributes
                                                             SSL,S/MIME,JAR/XPI

Example CA                                                   CT,,
```

추가 리소스

- **certutil(1)** 도움말 페이지

5.2. 웹 콘솔을 사용하여 CA 신뢰 플래그 변경

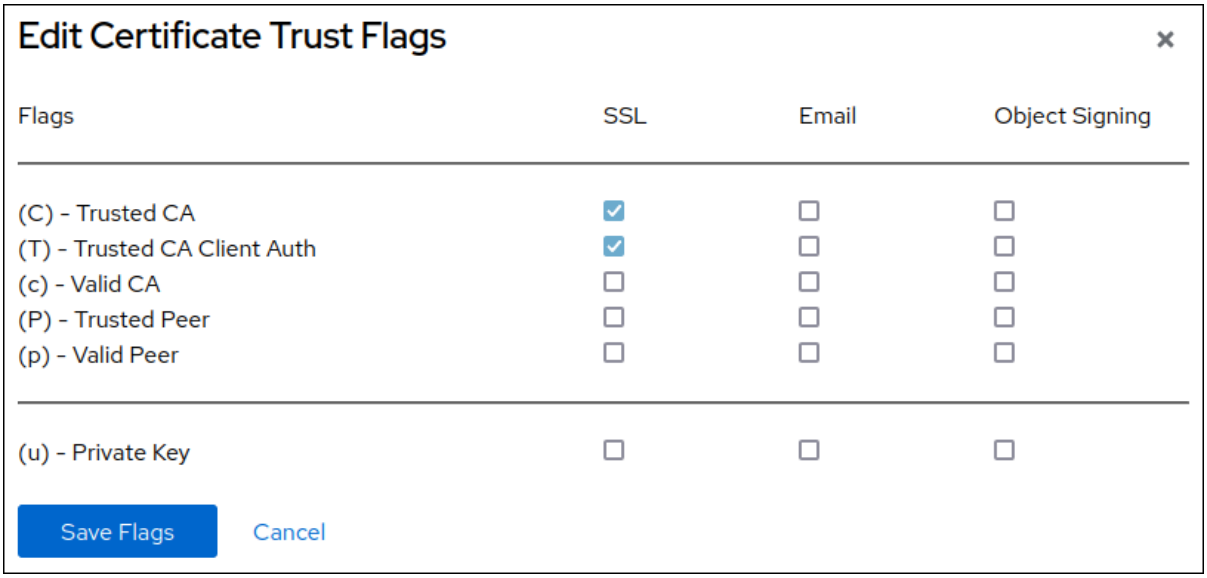
웹 콘솔을 사용하여 **CA** 신뢰 플래그를 변경할 수 있습니다.

사전 요구 사항

- **CA** 인증서를 **NSS(**네트워크 보안 서비스**)** 데이터베이스로 가져왔습니다.

절차

1. **Server** → **Security** → **Certificate Management** → 신뢰할 수 있는 인증 기관으로 이동합니다.
2. **CA** 인증서 옆에 있는 **...** 아이콘을 클릭하고 신뢰 **ECDHEs** 편집**)**을 선택합니다.
3. 신뢰 플래그를 선택합니다.



4. **저장을클릭합니다.**

검증

1. **Server → Security → Certificate Management → 신뢰할 수 있는 인증 기관으로 이동합니다.**

2. **신뢰 플래그를 표시하려면 CA 인증서 옆에 있는 >를 클릭합니다.**

6장. TLS 인증서 업데이트

TLS 인증서에는 만료 날짜 및 시간이 있습니다. 보안 연결을 지속적으로 제공하려면 만료되기 전에 Directory Server에서 서버 인증서를 갱신합니다.

6.1. 명령줄을 사용하여 TLS 인증서 업데이트

TLS 서버 인증서가 만료되기 전에 다음 절차를 따르십시오.

사전 요구 사항

- 특성 암호화는 구성되지 않습니다.
- TLS 인증서가 곧 만료됩니다.

절차

1. 개인 키와 CSR(인증서 서명 요청)을 생성합니다. 외부 유틸리티를 사용하여 생성하려면 이 단계를 건너뜁니다.

   - 하나의 이름으로만 호스트에 연결할 수 있는 경우 다음을 입력합니다.

     ```
     # dsctl instance_name tls generate-server-cert-csr -s
     "CN=server.example.com,O=example_organization"
     ```

   - 여러 이름으로 호스트에 연결할 수 있는 경우:

     ```
     # dsctl instance_name tls generate-server-cert-csr -s
     "CN=server.example.com,O=example_organization" server.example.com
     server.example.net
     ```

     호스트 이름을 마지막 매개변수로 지정하면 명령에서 DNS:server.example.com, DNS:server.example.net 항목에 SAN(Subject Alternative Name) 확장을 CSR에 추가합니다.

-s *subject* 매개변수에 지정된 문자열은 RFC 1485에 따라 유효한 주체 이름이어야 합니다. 주체의 CN 필드가 필요하며 서버의 FQDN(정규화된 도메인 이름) 중 하나로 설정해야 합니다. 이

명령은 **CSR**을 **/etc/dirsrv/slapd-*instance_name*/Server-Cert.csr** 파일에 저장합니다.

2. **CSR**을 **CA**(인증 기관)에 제출하여 발급된 인증서를 가져옵니다. 자세한 내용은 **CA** 설명서를 참조하십시오.

3. **CA** 인증서와 서버 인증서를 **/root/** 디렉터리에 저장합니다.

4. 다음 옵션 중 하나를 사용하여 **CA**에서 발급한 서버 인증서를 **NSS** 데이터베이스로 가져옵니다.

   - **dsctl tls generate-server-cert-csr** 명령을 사용하여 개인 키를 생성한 경우 다음을 입력합니다.

     ```
     # dsconf -D "cn=Directory Manager" ldap://server.example.com security certificate add --file /root/instance_name.crt --name "server-cert" --primary-cert
     ```

     --name _*certificate_nickname*_매개변수에 설정한 인증서의 이름을 기록해 둡니다. 이후 단계에서 필요합니다.

   - 외부 유틸리티를 사용하여 개인 키를 생성한 경우 서버 인증서와 개인 키를 가져옵니다.

     ```
     # dsctl instance_name tls import-server-key-cert /root/server.crt /root/server.key
     ```

     이 명령을 사용하려면 먼저 서버 인증서의 경로를 지정한 다음 개인 키 경로를 지정해야 합니다. 이 방법은 항상 인증서의 닉네임을 **Server-Cert** 로 설정합니다.

5. **CA** 인증서를 **NSS** 데이터베이스로 가져옵니다.

   ```
   # dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate add --file /root/ca.crt --name "Example CA"
   ```

6. **CA** 인증서의 **trust** 플래그를 설정합니다.

   ```
   # dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate set-trust-flags "Example CA" --flags "CT,,"
   ```

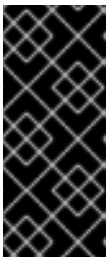
이렇게 하면 **TLS** 암호화 및 인증서 기반 인증에 대해 **CA**를 신뢰하도록 **Directory Server**가 구성됩니다.

7. 인스턴스를 중지합니다.

```
# dsctl instance_name stop
```

8. **/etc/dirsrv/slapd-*instance_name*/dse.ldif** 파일을 편집하고 속성을 포함한 다음 항목을 제거합니다.

- **CN=AES,cn=encrypted attribute keys,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config**
- **CN=3DES,cn=암호화 특성 키,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config**



중요

모든 데이터베이스의 항목을 제거합니다. **nsSymmetricKey** 속성이 포함된 항목이 **/etc/dirsrv/slapd-*instance_name*/dse.ldif** 파일에 남아 있으면 디렉터리 서버가 시작되지 않습니다.

9. 인스턴스를 시작합니다.

```
# dsctl instance_name start
```

7장. 인증서 기반 인증 구성

Directory Server는 LDAP 클라이언트의 인증서 기반 인증 및 복제 토폴로지와 같은 서버 간 연결을 지원합니다.

구성에 따라 클라이언트는 인증서를 사용하여 인증할 수도 있고 인증해야 합니다. 인증서의 subject 필드에 있는 속성을 기반으로 인증서를 확인한 후 서버는 디렉터리에서 사용자를 검색합니다. 검색에서 정확히 하나의 사용자 항목을 반환하는 경우 Directory Server는 이 사용자를 모든 추가 작업에 사용합니다. 선택적으로 인증에 사용되는 인증서가 사용자 항목의 userCertificate 속성에 저장된 DER(DER) 포맷된 인증서와 일치해야 함을 구성할 수 있습니다.

인증서 기반 인증 사용의 이점:

- 효율성 향상: 인증서 데이터베이스 암호로 인증된 다음 이후의 모든 바인딩 또는 인증 작업에 해당 인증서를 사용하는 것이 바인딩 고유 이름(DN) 및 암호를 반복적으로 제공하는 것보다 효율적입니다.
- 개선된 보안: 인증서 기반 인증은 인증서 기반 인증에서 공개 키 암호화를 사용하므로 비인증서 바인딩 작업보다 안전합니다. 공격자는 네트워크 전반에 걸쳐 바인딩 자격 증명을 인터셉트할 수 없습니다. 인증서 또는 장치가 손실되면 4.6.1이 없으면 소독되므로 피싱 공격과 같은 타사의 중단에 영향을 미칩니다.

7.1. 인증서 기반 인증 설정

사전 요구 사항

- Directory Server에서 TLS 암호화를 활성화했습니다.
- NSS(네트워크 보안 서비스) 데이터베이스에서 CA(인증 기관) 인증서에 대해ECDHE 플래그를 설정합니다.

절차

1. /etc/dirsrv/slapd-*instance_name*/certmap.conf 파일을 생성하여 인증서의 정보를 Directory Server 사용자로 매핑합니다.

```
certmap default       default
default:DNComps       dc
```

```
default:FilterComps     mail,cn
default:VerifyCert      on

certmap example         cn=Example CA
example:DNComps
```

이 구성을 사용하면 **cn=ECDHE CA** 에서 발급한 인증서의 경우 **DNComps** 매개변수가 이 발급자에게 비어 있게 설정되어 있으므로 **Directory Server**는 인증서 제목에서 기본 **DN**을 생성하지 않습니다. 또한 **FilterComps** 및 **VerifyCert** 에 대한 설정은 기본 항목에서 상속됩니다.

cn=ECDHE CA 와 다른 발행자 **DN**이 있는 인증서는 기본 항목의 설정을 사용하고 인증서 제목의 **cn** 특성을 기반으로 기본 **DN**을 생성합니다. 이를 통해 **Directory Server**는 전체 디렉터리를 검색하지 않고도 특정 **DN**에서 검색을 시작할 수 있습니다.

모든 인증서의 경우 **Directory Server**는 인증서의 제목에서 **mail** 및 **cn** 특성을 사용하여 검색 필터를 생성합니다. 그러나 메일 속성이 주체에 없는 경우 **Directory Server**는 제목에서 인증서의 **e** 속성 값을 자동으로 사용합니다.

2. 인증서 기반 인증을 활성화합니다. 예를 들어 인증서 기반 인증을 선택 사항으로 구성하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security set --tls-client-auth="allowed"
```

--tls-client-auth=required 옵션을 사용하여 인증서 기반 인증을 필수로 구성합니다.

3. 선택 사항: 인증서 기반 인증을 필요에 따라 구성한 경우 **nsslapd-require-secure-binds** 매개변수를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-require-secure-binds=on
```

이 설정을 사용하면 암호화되지 않은 연결을 사용하여 인증서 기반 인증을 바이패스할 수 없습니다.

4. 선택 사항: **Directory Server**에서 바인딩 요청의 자격 증명 대신 인증서의 **ID**를 사용해야 하는 경우 **EXTERNAL** 단순 인증 및 보안 계층(**ECDHEL**) 메커니즘을 사용하도록 **Directory Server**를 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-force-sasl-external=on
```

이 설정을 사용하여 Directory Server는 인증서의 ID보다 다른 바인딩 메서드를 무시합니다.

5. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

다음 단계:

- 인증 인증서가 사용자의 userCertificate 속성에 저장된 것과 일치해야하도록 Directory Server를 구성한 경우 사용자 항목에 인증서를 추가합니다. 자세한 내용은 다음을 참조하십시오. 7.2절. “사용자에게 인증서 추가”

추가 리소스

- Directory Server에 대한 TLS 암호화 연결 활성화
- CA 신뢰 플래그 변경
- certmap.conf(5) man page

7.2. 사용자에게 인증서 추가

인증서 기반 인증을 설정하면 인증에 사용된 인증서가 사용자의 userCertificate 바이너리 특성과 일치해야 하도록 서버를 구성할 수 있습니다. 이 기능을 활성화한 경우 영향을 받는 사용자의 인증서를 디렉터리 항목에 추가해야 합니다.

사전 요구 사항

- Directory Server에서 인증서 기반 인증을 활성화했습니다.
- 서버에서 신뢰하는 CA(인증 기관)에서 클라이언트 인증서를 발급했습니다.

- 클라이언트 인증서는 고유 인코딩 규칙(DER) 형식으로 되어 있습니다.

- 클라이언트 인증서는 서버의 /etc/dirsrv/slapd-instance_name/certmap.conf 에 설정된 요구 사항을 충족합니다.

절차

1. 인증서가 DER 형식이 아닌 경우 인증서를 변환합니다. 예를 들어 개인 정보 보호 강화 메일(PEM)에서 DER로 인증서를 변환하려면 다음을 입력합니다.

```
# openssl x509 -in /home/user_name/certificate.pem -out
/home/user_name/certificate.der -outform DER
```

2. 사용자의 userCertificate 속성에 인증서를 추가합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldaps://server.example.com -x

dn: uid=user_name,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate
userCertificate:< file:///home/user_name/example.der
```

검증

1. 인증서 기반 인증을 사용하여 사용자로 인증합니다.

   a. 다음 환경 변수를 CA 인증서의 해당 경로, 사용자 키 및 사용자 인증서로 설정합니다.

```
LDAPTLS_CACERT=/home/user_name/CA.crt
LDAPTLS_KEY=/home/user_name/user.key
LDAPTLS_CERT=/home/user_name/user.der
```

   또는 현재 사용자의 ~/.ldaprc 파일에 TLS_CACERT,TLS_KEY 및 TLS_CERT 매개변수를 설정합니다.

   b. 서버에 연결합니다.

```
# ldapwhoami -H ldaps://server.example.com -Y EXTERNAL
dn: uid=example,ou=people,dc=example,dc=com
```

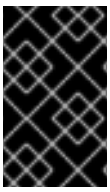
추가 리소스

- **ldap.conf(5)** 매뉴얼 페이지의 **TLS OPTIONS** 섹션

8장. 인증서 기반 인증을 사용하여 멀티 제공 복제 구성

두 **Directory Server** 인스턴스 간 복제를 설정하면 바인딩 **DN** 및 암호를 사용하여 복제 파트너에 인증하는 대신 인증서 기반 인증을 사용할 수 있습니다.

복제 토폴로지에 새 서버를 추가하고 인증서 기반 인증을 사용하여 새 호스트와 기존 서버 간에 복제 계약을 설정하여 이를 수행할 수 있습니다.



중요

인증서 기반 인증에는 **TLS** 암호화 연결이 필요합니다.

8.1. 인증서 기반 인증과 함께 복제 계약에 사용할 수 있도록 계정 및 바인딩 그룹 준비

복제 계약에 인증서 기반 인증을 사용하려면 먼저 계정을 준비하고 이러한 계정의 **userCertificate** 속성에 클라이언트 인증서를 저장합니다. 또한 이 절차에서는 나중에 복제 계약에 사용하는 바인딩 그룹을 생성합니다.

기존 호스트 **server1.example.com** 에서 다음 절차를 수행합니다.

사전 요구 사항

- **Directory Server**에서 **TLS** 암호화를 활성화했습니다.
- 클라이언트 인증서를 **/root/server1.der** 및 **/root/server2.der** 파일에 고유 인코딩 규칙**(DER)** 형식으로 저장했습니다.

  클라이언트 인증서 및 **CA**(인증 기관)에서 요청하는 방법에 대한 자세한 내용은 **CA** 설명서를 참조하십시오.

절차

1. 존재하지 않는 경우 **ou=services** 항목을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldaps://server1.example.com -x

dn: ou=services,dc=example,dc=com
```

```
objectClass: organizationalunit
objectClass: top
ou: services
```

2. cn=server1,ou=services,dc=example,dc=com 및 cn=server1,ou=services,dc=example,dc=com 과 같은 두 서버에 대한 계정을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldaps://server1.example.com -x

dn: cn=server1,ou=services,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: inetOrgPerson
sn: server1
cn: server1
userPassword: password
userCertificate:< file:///root/server1.der

adding new entry "cn=server1,ou=services,dc=example,dc=com"

dn: cn=server2,ou=services,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: inetOrgPerson
sn: server2
cn: server2
userPassword: password
userCertificate:< file:///root/server2.der

adding new entry "cn=server2,ou=services,dc=example,dc=com"
```

3. cn=repl_servers,dc=groups,dc=example,dc=com 과 같은 그룹을 생성합니다.

```
# dsidm -D "cn=Directory Manager" ldaps://server1.example.com -b
"dc=example,dc=com" group create --cn "repl_servers"
```

4. 두 개의 복제 계정을 그룹에 멤버로 추가합니다.

```
# dsidm -D "cn=Directory Manager" ldaps://server1.example.com -b
"dc=example,dc=com" group add_member repl_servers
"cn=server1,ou=services,dc=example,dc=com"

# dsidm -D "cn=Directory Manager" ldaps://server1.example.com -b
"dc=example,dc=com" group add_member repl_servers
"cn=server2,ou=services,dc=example,dc=com"
```

추가 리소스

- Directory Server에 대한 TLS 암호화 연결 활성화

8.2. 임시 복제 관리자 계정을 사용하여 새 서버 초기화

인증서 기반 인증은 디렉터리에 저장된 인증서를 사용합니다. 그러나 새 서버를 초기화하기 전에 **server2.example.com** 의 데이터베이스가 비어 있으며 연결된 인증서가 있는 계정이 존재하지 않습니다. 따라서 데이터베이스를 초기화하기 전에 인증서를 사용한 복제를 수행할 수 없습니다. 임시 복제 관리자 계정으로 **server2.example.com** 을 초기화하여 이 문제를 해결할 수 있습니다.

사전 요구 사항

- **server2.example.com** 에 Directory Server 인스턴스를 설치했습니다. 자세한 내용은 .inf 파일을 사용하여 명령줄에서 새 인스턴스 설정을 참조하십시오.
- **dc=example,dc=com** 접미사의 데이터베이스가 있습니다.
- **server1.example.com** 및 **server2.example.com** 의 Directory Server에서 **TLS** 암호화를 활성화했습니다.

절차

1. **server2.example.com** 에서 **dc=example,dc=com** 접미사에 대해 복제를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server2.example.com replication enable --suffix "dc=example,dc=com" --role "supplier" --replica-id 2 --bind-dn "cn=replication manager,cn=config" --bind-passwd "password"
```

이 명령은 **server2.example.com** 호스트를 **dc=example,dc=com** 접미사의 공급업체로 구성하고 이 호스트의 복제본 **ID**를 **2** 로 설정합니다. 또한 이 명령은 지정된 암호를 사용하여 임시 **cn=replication** 관리자인**cn=config** 사용자를 생성하고 이 계정에서 접미사에 대한 변경 사항을 이 호스트에 복제할 수 있습니다.

복제본 **ID**는 토폴로지의 모든 공급업체의 접미사에 대해 **1** 에서 **65534** 사이의 고유 정수여야 합니다.

2. **server1.example.com** 에서 :

a. 복제를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server1.example.com replication
enable --suffix="dc=example,dc=com" --role="supplier" --replica-id="1"
```

b. 인증을 위해 이전 단계의 임시 계정을 사용하는 임시 복제 계약을 생성합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server1.example.com repl-agmt create
--suffix="dc=example,dc=com" --host="server1.example.com" --port=636 --conn-
protocol=LDAPS --bind-dn="cn=Replication Manager,cn=config" --bind-
passwd="password" --bind-method=SIMPLE --init temporary_agreement
```

검증

1. 초기화에 성공했는지 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server1.example.com repl-agmt init-status
--suffix "dc=example,dc=com" temporary_agreement
Agreement successfully initialized.
```

추가 리소스

- Red Hat Directory Server 설치
- Directory Server에 대한 TLS 암호화 연결 활성화

8.3. 인증서 기반 인증을 사용하여 멀티 제공 복제 구성

인증서 기반 인증이 포함된 멀티 제공 복제 환경에서 복제본은 인증서를 사용하여 서로 인증합니다.

사전 요구 사항

- server1.example.com 및 server2.example.com 호스트 모두에서 인증서 기반 인증을 설정합니다.
- Directory Server는 클라이언트 인증서를 발급하는 CA(인증 기관)를 신뢰합니다.
-

클라이언트 인증서는 서버의 **/etc/dirsrv/slapd-instance_name/certmap.conf** 에 설정된 요구 사항을 충족합니다.

절차

1. **server1.example.com** 에서 :

   a. 임시 복제 계약 제거:

   ```
   # dsconf -D "cn=Directory Manager" ldaps://server1.example.com repl-agmt delete
   --suffix="dc=example,dc=com" temporary_agreement
   ```

   b. **cn=repl_servers,dc=groups,dc=example,dc=com** 바인딩 그룹을 복제 설정에 추가합니다.

   ```
   # dsconf -D "cn=Directory Manager" ldaps://server1.example.com replication set --
   suffix="dc=example,dc=com" --repl-bind-group
   "cn=repl_servers,dc=groups,dc=example,dc=com"
   ```

   c. 바인딩 그룹의 변경 사항을 자동으로 확인하도록 **Directory Server**를 구성합니다.

   ```
   # dsconf -D "cn=Directory Manager" ldaps://server1.example.com replication set --
   suffix="dc=example,dc=com" --repl-bind-group-interval=0
   ```

2. **server2.example.com** 에서 :

   a. 임시 복제 관리자 계정을 제거합니다.

   ```
   # dsconf -D "cn=Directory Manager" ldaps://server2.example.com replication
   delete-manager --suffix="dc=example,dc=com" --name="Replication Manager"
   ```

   b. **cn=repl_servers,dc=groups,dc=example,dc=com** 바인딩 그룹을 복제 설정에 추가합니다.

   ```
   # dsconf -D "cn=Directory Manager" ldaps://server2.example.com replication set --
   suffix="dc=example,dc=com" --repl-bind-group
   "cn=repl_servers,dc=groups,dc=example,dc=com"
   ```

c. 바인딩 그룹의 변경 사항을 자동으로 확인하도록 Directory Server를 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server2.example.com replication set --
suffix="dc=example,dc=com" --repl-bind-group-interval=0
```

d. 인증서 기반 인증을 사용하여 복제 계약을 생성합니다.

```
dsconf -D "cn=Directory Manager" ldaps://server2.example.com repl-agmt create --
suffix="dc=example,dc=com" --host="server1.example.com" --port=636 --conn-
protocol=LDAPS --bind-method="SSLCLIENTAUTH" --init server2-to-server1
```

3. server1.example.com 에서 인증서 기반 인증을 사용하여 복제 계약을 생성합니다.

```
dsconf -D "cn=Directory Manager" ldaps://server1.example.com repl-agmt create --
suffix="dc=example,dc=com" --host="server2.example.com" --port=636 --conn-
protocol=LDAPS --bind-method="SSLCLIENTAUTH" --init server1-to-server2
```

검증

1. 각 서버에서 초기화에 성공했는지 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server1.example.com repl-agmt init-status
--suffix "dc=example,dc=com" server1-to-server2
Agreement successfully initialized.

# dsconf -D "cn=Directory Manager" ldaps://server2.example.com repl-agmt init-status
--suffix "dc=example,dc=com" server2-to-server1
Agreement successfully initialized.
```

추가 리소스

- 인증서 기반 인증 설정
- CA 신뢰 플래그 변경

9장. 복제 변경 로그 암호화

공격자가 서버의 파일 시스템에 대한 액세스 권한을 얻는 경우 인스턴스의 보안을 강화하기 위해 복제 변경 로그를 암호화합니다.

변경 로그 암호화는 서버의 **TLS** 암호화 키와 동일한 **DestinationRule**을 사용하여 키 잠금을 해제합니다. 서버를 시작할 때 **BOOM**을 수동으로 입력하거나 **DestinationRule** 파일을 사용해야 합니다.

Directory Server는 임의로 생성된 대칭 암호화 키를 사용하여 변경 로그를 암호화하고 암호를 해독합니다. 서버는 구성된 각 암호에 대해 별도의 키를 사용합니다. 이러한 키는 서버의 **TLS** 인증서의 공개 키를 사용하여 래핑되며 결과 래핑된 키는 서버의 구성 파일에 저장됩니다. 속성 암호화의 효과적인 강점은 랩핑에 사용되는 서버 **TLS** 키의 힘과 동일합니다. 서버의 개인 키와 **DestinationRule**에 액세스하지 않으면 래핑된 사본에서 대칭 키를 복구할 수 없습니다.

9.1. 명령줄을 사용하여 변경 로그 암호화

복제 토폴로지의 보안을 늘리려면 공급자 및 허브에 대한 변경 로그를 암호화합니다. 이 절차에서는 **dc=example,dc=com** 접미사에 대해 변경 로그 암호화를 활성화하는 방법을 설명합니다.

사전 요구 사항

- 서버에는 **TLS** 암호화가 활성화되어 있습니다.
- 호스트는 복제 토폴로지의 공급업체 또는 허브입니다.

절차

1. 예를 들어 변경 로그를 **/tmp/changelog.ldif** 파일로 내보냅니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com replication export-changelog to-ldif -o /tmp/changelog.ldif -r "dc=example,dc=com"
```

2. **dc=example,dc=com** 접미사의 변경 로그 암호화를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com replication --suffix "dc=example,dc=com" --encrypt
```

3. /tmp/changelog.ldif 파일에서 변경 로그를 가져옵니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com replication import-changelog from-ldif -r "dc=example,dc=com" /tmp/changelog.ldif
```

4. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

검증

1. LDAP 디렉터리를 변경합니다(예: 항목 업데이트).

2. 인스턴스를 중지합니다.

```
# dsctl instance_name stop
```

3. 접미사 및 해당 데이터베이스를 나열합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix list
dc=example,dc=com (userroot)
```

변경 로그 암호화를 활성화한 데이터베이스의 이름을 기록해 둡니다.

4. 다음 명령을 입력하여 변경 로그의 일부를 표시합니다.

```
# dbscan -f /var/lib/dirsrv/slapd-instance_name/db/userroot/replication_changelog.db | tail -50
```

변경 로그가 암호화되면 암호화된 데이터만 표시됩니다.

5. 인스턴스를 시작합니다.

```
# dsctl instance_name start
```

추가 리소스

- Directory Server에 대한 TLS 암호화 연결 활성화

10장. 그룹의 멤버가 DIRECTORY SERVER를 백업하고 그룹 멤버 중 하나로 백업을 수행할 수 있도록 설정ENABLED MEMBERS OF A GROUP TO BACK UP DIRECTORY SERVER AND PERFORMING THE BACKUP AS ONE OF THE GROUP MEMBERS

그룹의 멤버에게 인스턴스를 백업하고 백업을 수행할 수 있는 권한이 있도록 구성할 수 있습니다. 이렇게 하면 백업 스크립트 또는 cron 작업에서 cn=Directory Manager 의 인증 정보를 설정할 필요가 없으므로 보안이 향상됩니다. 또한 그룹을 수정하여 백업 권한을 쉽게 부여하고 취소할 수 있습니다.

10.1. 그룹이 DIRECTORY SERVER를 백업하도록 설정

이 절차를 사용하여 cn=backup_users,ou=groups,dc=example,dc=com 그룹을 추가하고 이 그룹의 멤버가 백업 작업을 생성할 수 있도록 활성화합니다.

사전 요구 사항

- ou=groups,dc=example,dc=com 항목이 데이터베이스에 있습니다.

절차

1. cn=backup_users,ou=groups,dc=example,dc=com 그룹을 생성합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" group create --cn backup_users
```

2. cn=backup_users,ou=groups,dc=example,dc=com 그룹의 멤버가 백업 작업을 생성할 수 있는 ACL(액세스 제어 명령)을 추가합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com

dn: cn=config
changetype: modify
add: aci
aci: (target = "ldap:///cn=backup,cn=tasks,cn=config")(targetattr="*")
 (version 3.0 ; acl "permission: Allow backup_users
  group to create backup tasks" ; allow (add, read, search) groupdn
  = "ldap:///cn=backup_users,ou=groups,dc=example,dc=com";)
-
add: aci
aci: (target = "ldap:///cn=config")(targetattr = "nsslapd-bakdir ||
  objectClass") (version 3.0 ; acl "permission: Allow backup_users
  group to access bakdir attribute" ; allow (read,search)
  groupdn = "ldap:///cn=backup_users,ou=groups,dc=example,dc=com";)
```

3. 사용자를 생성합니다.

   a. 사용자 계정을 생성합니다.

   ```
   # dsidm -D "cn=Directory manager" ldap://server.example.com -b
   "dc=example,dc=com" user create --uid="example" --cn="example" --
   uidNumber="1000" --gidNumber="1000" --homeDirectory="/home/example/" --
   displayName="Example User"
   ```

   b. 사용자 계정에 암호를 설정합니다.

   ```
   # dsidm -D "cn=Directory manager" ldap://server.example.com -b
   "dc=example,dc=com" account reset_password
   "uid=example,ou=People,dc=example,dc=com" "password"
   ```

4. uid=example,ou=People,dc=example,dc=com 사용자를 cn=backup_users,ou=groups,dc=example,dc=com 그룹에 추가합니다.

   ```
   # dsidm -D "cn=Directory manager" ldap://server.example.com -b
   "dc=example,dc=com" group add_member backup_users
   uid=example,ou=People,dc=example,dc=com
   ```

검증

- cn=config 항목에 ACIs 세트를 표시합니다.

  ```
  # ldapsearch -o ldif-wrap=no -LLLx -D "cn=directory manager" -W -H
  ldap://server.example.com -b cn=config aci=* aci -s base
  dn: cn=config
  aci: (target = "ldap:///cn=backup,cn=tasks,cn=config")(targetattr="*")(version 3.0 ; acl
  "permission: Allow backup_users group to create backup tasks" ; allow (add, read,
  search) groupdn = "ldap:///cn=backup_users,ou=groups,dc=example,dc=com";)
  aci: (target = "ldap:///cn=config")(targetattr = "nsslapd-bakdir || objectClass")(version
  3.0 ; acl "permission: Allow backup_users group to access bakdir attribute" ; allow
  (read,search) groupdn = "ldap:///cn=backup_users,ou=groups,dc=example,dc=com";)
  ...
  ```

10.2. 일반 사용자로 백업 수행

cn=Directory Manager 대신 일반 사용자로 백업을 수행할 수 있습니다.

사전 요구 사항

- cn=backup_users,ou=groups,dc=example,dc=com 그룹의 멤버가 백업을 수행할 수 있도록 활성화했습니다.
- 백업을 수행하는 데 사용하는 사용자는 cn=backup_users,ou=groups,dc=example,dc=com 그룹의 멤버입니다.

절차

- 다음 방법 중 하나를 사용하여 백업 작업을 생성합니다.
  - dsconf backup create 명령을 사용합니다.

```
# dsconf -D "uid=example,ou=People,dc=example,dc=com"
ldap://server.example.com backup create
```

  - 작업을 수동으로 생성하면 다음을 수행합니다.

```
# ldapadd -D "uid=example,ou=People,dc=example,dc=com" -W -H
ldap://server.example.com

dn: cn=backup-2021_07_23_12:55_00,cn=backup,cn=tasks,cn=config
changetype: add
objectClass: extensibleObject
nsarchivedir: /var/lib/dirsrv/slapd-instance_name/bak/backup-
2021_07_23_12:55_00
nsdatabasetype: ldbm database
cn: backup-2021_07_23_12:55_00
```

검증

- 백업이 생성되었는지 확인합니다.

```
# ls -l /var/lib/dirsrv/slapd-instance_name/bak/
total 0
drwx------. 3 dirsrv dirsrv 108 Jul 23 12:55 backup-2021_07_23_12_55_00
...
```

추가 리소스

- 그룹이 Directory Server를 백업하도록 설정

11장. 그룹 멤버가 데이터를 내보내고 그룹 멤버 중 하나로 내보내기 수행 가능

그룹의 멤버에게 데이터를 내보낼 수 있는 권한이 있도록 구성할 수 있습니다. 이렇게 하면 더 이상 스크립트에서 **cn=Directory Manager** 의 인증 정보를 설정할 필요가 없으므로 보안이 향상됩니다. 또한 그룹을 수정하여 내보내기 권한을 쉽게 부여하고 취소할 수 있습니다.

11.1. 그룹에서 데이터 내보내기 활성화

이 절차를 사용하여 **cn=export_users,ou=groups,dc=example,dc=com** 그룹을 추가하고 이 그룹의 멤버가 내보내기 작업을 생성할 수 있도록 활성화합니다.

절차

1. **cn=export_users,ou=groups,dc=example,dc=com** 그룹을 생성합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" group create --cn export_users
```

2. **cn=export_users,ou=groups,dc=example,dc=com** 그룹의 멤버가 내보내기 작업을 생성할 수 있는 **ACL(액세스 제어 명령)**을 추가합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com

dn: cn=config
changetype: modify
add: aci
aci: (target = "ldap:///cn=export,cn=tasks,cn=config")
 (targetattr="*") (version 3.0 ; acl "permission:
  Allow export_users group to export data" ;
  allow (add, read, search) groupdn
  = "ldap:///cn=export_users,ou=groups,dc=example,dc=com";)
-
add: aci
aci: (target = "ldap:///cn=config")(targetattr =
  "objectclass || cn || nsslapd-suffix || nsslapd-ldifdir")
  (version 3.0 ; acl "permission: Allow export_users
  group to access ldifdir attribute" ; allow
  (read,search) groupdn = "ldap:///cn=export_users,ou=groups,dc=example,dc=com";)
```

3. 사용자를 생성합니다.

   a. 사용자 계정을 생성합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" user create --uid="example" --cn="example" --
uidNumber="1000" --gidNumber="1000" --homeDirectory="/home/example/" --
displayName="Example User"
```

b. 사용자 계정에 암호를 설정합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" account reset_password
"uid=example,ou=People,dc=example,dc=com" "password"
```

4. uid=example,ou=People,dc=example,dc=com 사용자를 cn=export_users,ou=groups,dc=example,dc=com 그룹에 추가합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" group add_member export_users
uid=example,ou=People,dc=example,dc=com
```

검증

- cn=config 항목에 ACIs 세트를 표시합니다.

```
# ldapsearch -o ldif-wrap=no -LLLx -D "cn=directory manager" -W -H
ldap://server.example.com -b cn=config aci=* aci -s base
dn: cn=config
aci: (target = "ldap:///cn=export,cn=tasks,cn=config")(targetattr="*")(version 3.0 ; acl
"permission: Allow export_users group to export data" ; allow (add, read, search)
groupdn = "ldap:///cn=export_users,ou=groups,dc=example,dc=com";)
aci: (target = "ldap:///cn=config")(targetattr = "objectclass || cn || nsslapd-suffix ||
nsslapd-ldifdir")(version 3.0 ; acl "permission: Allow export_users group to access
ldifdir attribute" ; allow (read,search) groupdn =
"ldap:///cn=export_users,ou=groups,dc=example,dc=com";)
...
```

11.2. 일반 사용자로 내보내기 수행

cn=Directory Manager 대신 일반 사용자로 내보내기를 수행할 수 있습니다.

사전 요구 사항

- cn=export_users,ou=groups,dc=example,dc=com 그룹의 멤버가 데이터를 내보낼 수 있도록 활성화했습니다.

- 내보내기를 수행하는 데 사용하는 사용자는 **cn=export_users,ou=groups,dc=example,dc=com** 그룹의 멤버입니다.

절차

- 다음 방법 중 하나를 사용하여 내보내기 작업을 생성합니다.
  - **dsconf** 백엔드 내보내기 명령 사용:

```
# dsconf -D "uid=example,ou=People,dc=example,dc=com"
ldap://server.example.com backend export userRoot
```

  - 작업을 수동으로 생성하면 다음을 수행합니다.

```
# ldapadd -D "uid=example,ou=People,dc=example,dc=com" -W -H
ldap://server.example.com

dn: cn=userRoot-2021_07_23_12:55_00,cn=export,cn=tasks,cn=config
changetype: add
objectClass: extensibleObject
nsFilename: /var/lib/dirsrv/slapd-instance_name/ldif/None-userroot-
2021_07_23_12:55_00.ldif
nsInstance: userRoot
cn: export-2021_07_23_12:55_00
```

검증

- 백업이 생성되었는지 확인합니다.

```
# ls -l /var/lib/dirsrv/slapd-instance_name/ldif/*.ldif
total 0
-rw-------. 1 dirsrv dirsrv 10306 Jul 23 12:55 None-userroot-2021_07_23_12_55_00.ldif
...
```

추가 리소스

- 그룹에서 데이터 내보내기 활성화

12장. 액세스 제어 명령 관리

Directory Server가 요청을 수신할 때 바인딩 작업에서 사용자가 제공한 인증 정보와 디렉터리에 정의된 ACL(액세스 제어 명령)을 사용하여 요청된 항목 또는 속성에 대한 액세스를 허용하거나 거부합니다. 서버는 읽기,쓰기,검색 및 비교 와 같은 작업에 대한 권한을 허용하거나 거부할 수 있습니다. 사용자에게 부여된 권한 수준은 제공된 인증 정보에 따라 다릅니다.

Directory Server의 액세스 제어를 사용하면 ACI가 적용 가능한 시기에 대한 정확한 규칙을 설정할 수 있습니다.

- 전체 디렉터리, 하위 트리 또는 특정 항목의 경우
- 특정 사용자의 경우 특정 그룹 또는 역할에 속한 모든 사용자 또는 디렉터리의 모든 사용자
- IP 주소, IP 범위 또는 DNS 이름과 같은 특정 위치의 경우

  로드 밸런서는 위치별 규칙에 영향을 미칠 수 있습니다.

중요

복잡한 ACI는 읽고 이해하기 어렵습니다. 하나의 복잡한 ACI 대신 여러 개의 간단한 규칙을 작성하여 동일한 효과를 얻을 수 있습니다. 그러나 더 많은 ACI 수에 따라 ACI 처리 비용이 증가합니다.

12.1. ACI 배치

Directory Server는 디렉터리 항목에 다중 값의 운영 특성에 ACI(액세스 제어 명령) 를 저장합니다. ACI를 설정하려면 해당 디렉터리 항목에 aci 특성을 추가합니다. Directory Server는 ACI를 적용합니다.

- 하위 항목이 없는 경우 ACI가 포함된 항목에만 해당합니다. 예를 들어 클라이언트에서 uid=user_name,ou=People,dc=example,dc=com 오브젝트에 대한 액세스가 필요하고 ACI가 dc=example,dc=com 에만 설정되어 있고 하위 항목에는 이 ACI만 적용됩니다.

참고

추가 권한이 있는 ACIS는 나중에 생성된 하위 항목에도 적용됩니다.

- 하위 항목이 있는 경우 ACI 및 그 아래의 모든 항목에 포함된 항목으로 이동합니다. 결과적으로 서버는 지정된 항목에 대한 액세스 권한을 평가하면 요청된 항목과 디렉터리 접미사 간의 모든 항목에 대해 ACI와 항목 자체에 대한 ACIs를 확인합니다.

  예를 들어 ACI는 dc=example,dc=com 및 ou=People,dc=example,dc=com 항목에 설정됩니다. 클라이언트가 ACI 세트가 없는 uid=user_name,ou=People,dc=com 오브젝트에 액세스하려는 경우 Directory 서버는 먼저 ou=People,dc=com 항목의 ACI를 검증합니다. 이 ACI가 액세스 권한을 부여하면 평가가 중지되고 액세스 권한이 부여됩니다. 그렇지 않은 경우 Directory Server는 ou=People,dc=example,dc=com 의 ACI를 확인합니다. 이 ACI가 클라이언트에 성공적으로 권한을 부여하면 오브젝트에 액세스할 수 있습니다.

참고

rootDSE 항목의 ACIS 세트는 이 항목에만 적용됩니다.

항목에 생성된 ACI는 해당 항목에 직접 적용되지 않도록 설정할 수 있지만 아래 하위 트리의 일부 또는 모든 항목에 적용할 수 있습니다. 이 접근 방식의 장점은 일반 ACI를 디렉터리 트리에 더하여 트리에 있는 하위 항목에 영향을 줄 수 있다는 것입니다. 예를 들어 inetOrgPerson 개체 클래스를 포함하는 항목을 대상으로 하는 ACI는 organizationalUnit 항목 또는 locality 항목의 수준에서 만들 수 있습니다.

참고

일반 규칙을 높은 수준의 분기 포인트에 배치하여 디렉터리 트리의 ACI 수를 최소화합니다. 보다 구체적인 규칙의 범위를 제한하려면 가능한 한 밀접하게 항목을 리프트에 배치하십시오.

12.2. ACI의 구조

aci 속성은 다음 구문을 사용합니다.

```
(target_rule) (version 3.0; acl "ACL_name"; permission_rule bind_rules;)
```

- target_rule 은 액세스를 제어하는 항목, 속성 또는 항목 및 속성 집합을 지정합니다.

- 버전 **3.0** 은 **ACL(**액세스 제어 명령**)** 버전을 식별하는 필수 문자열입니다.
- *ACL 이름*"**AC l**를 설명하는 이름 또는 문자열을 설정합니다.
- **permission_rule** 은 읽기 또는 쓰기 와 같은 권한이 허용되거나 거부됩니다.
- **bind_rules** 는 액세스를 허용하거나 거부하기 위해 바인딩 중 일치해야 하는 규칙을 지정합니다.

권한 및 바인딩 규칙 쌍은 액세스 제어 규칙이라고 합니다.

지정된 대상에 대해 여러 액세스 제어를 효율적으로 설정하려면 각 대상에 대해 여러 액세스 제어 규칙을 설정할 수 있습니다.

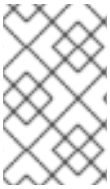
> (*target_rule*)(version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*; *permission_rule bind_rules*; ... ;)

12.3. ACl 평가

특정 항목에 대한 액세스 권한을 평가하기 위해 서버는 항목 자체 및 상위 항목에 있는 **ACL(**액세스 제어 명령**)** 목록을 **Directory Server**에 저장된 최상위 항목까지 백업합니다. **ACIS**는 특정 인스턴스에 대해 모든 데이터베이스에서 평가되지만 다른 인스턴스에서는 평가되지 않습니다.

Directory Server는 디렉터리 트리의 배치가 아닌 **ACI**의 의미 체계를 기반으로 이 **ACI** 목록을 평가합니다. 즉, 디렉터리 트리의 루트에 가까운 **ACI**가 디렉터리 트리의 리프에 더 가까운 **ACI**보다 우선하지 않습니다.

Directory Server에서 **ACI**의 거부 권한이 허용 권한보다 우선합니다. 예를 들어 디렉터리의 루트 수준에서 쓰기 권한을 거부하는 경우 다른 **ACI**가 이 권한을 부여했는지 여부에 관계없이 사용자가 디렉터리에 쓸 수 없습니다. 디렉터리에 특정 사용자 쓰기 권한을 부여하려면 사용자가 해당 디렉터리에 쓸 수 있도록 예외를 원래 거부 규칙에 추가해야 합니다.



참고

개선된 **ACI**를 위해 규칙을 거부하는 대신 세분화된 허용 규칙을 사용하십시오.

12.4. ACI의 제한

ACL(액세스 제어 명령)을 설정하면 다음과 같은 제한 사항이 적용됩니다.

- 디렉터리 데이터베이스가 여러 서버에 분산된 경우 ACI에서 사용할 수 있는 키워드에 다음 제한이 적용됩니다.
  - groupdn 키워드를 사용하는 그룹 항목에 따라 ACIS는 그룹 항목과 동일한 서버에 있어야 합니다.

    그룹이 동적인 경우 그룹의 모든 멤버가 서버에 항목이 있어야 합니다. 정적 그룹의 멤버 항목은 원격 서버에 있을 수 있습니다.
  - roledn 키워드를 사용하는 역할 정의에 따라 ACIS는 역할 정의 항목과 동일한 서버에 있어야 합니다. 역할을 가지려는 모든 항목도 동일한 서버에 있어야 합니다.

  그러나 대상 항목에 저장된 값과 바인딩 사용자의 항목에 저장된 값과 일치시킬 수 있습니다(예: userattr 키워드 사용). 이 경우 bind 사용자에게 ACI를 저장하는 서버에 항목이 없는 경우에도 액세스 권한이 정상적으로 평가됩니다.
- 다음 ACI 키워드에서는 서비스 클래스(CoS) 특성과 같은 가상 속성을 사용할 수 없습니다.
  - targetfilter
  - targattrfilters
  - userattr
- 액세스 제어 규칙은 로컬 서버에서만 평가됩니다. 예를 들어 LDAP URL에 서버의 호스트 이름을 ACI 키워드로 지정하면 URL이 무시됩니다.

12.5. 복제 토폴로지에서 DIRECTORY SERVER에서 ACI를 처리하는 방법

ACL(액세스 제어 명령)은 항목의 aci 속성에 저장됩니다. 따라서 ACI를 포함하는 항목이 복제된 데이

터베이스의 일부인 경우 ACI가 복제됩니다.

ACIS는 들어오는 LDAP 요청을 해결하는 서버에서 항상 평가됩니다. 소비자 서버가 업데이트 요청을 수신하면 공급 업체에서 요청을 서비스할 수 있는지 여부를 평가하기 전에 공급자 서버에 대한 설명을 반환합니다.

12.6. ACI 표시, 추가, 삭제 및 업데이트

ldapsearch 유틸리티를 사용하여 검색, ldapmodify 유틸리티를 사용하여 ACI(액세스 제어 지침)를 추가, 삭제 및 업데이트할 수 있습니다.

ACI 표시:

예를 들어 dc=example,dc=com 및 sub-entries에 ACI 세트를 표시하려면 다음을 입력합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -x -b
"dc=example,dc=com" -s sub '(aci=*)' aci
```

ACI 추가

예를 들어 ou=People,dc=example,dc=com 항목에 ACI를 추가하려면 다음을 입력합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="userPassword") (version 3.0; acl
  "Allow users updating their password";
  allow (write) userdn= "ldap:///self";)
```

ACI 삭제

ACI를 삭제하려면 다음을 수행합니다.

- 항목에 하나의 aci 속성만 설정되어 있거나 해당 항목에서 모든 ACI를 제거하려는 경우:

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: delete
delete: aci
```

- 항목에 여러 ACI가 있고 특정 ACI를 삭제하려면 정확한 ACI를 지정합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
delete: aci
aci: (targetattr="userPassword") (version 3.0; acl "Allow users
 updating their password"; allow (write) userdn= "ldap:///self";)
```

ACI 업데이트

ACI를 업데이트하려면 다음을 수행합니다.

- 기존 ACI를 삭제합니다.
- 업데이트된 설정으로 새 ACI를 추가합니다.

12.7. ACI 대상 정의

ACL(액세스 제어 명령)의 대상 규칙은 Directory Server가 ACI를 적용하는 항목을 정의합니다. 대상을 설정하지 않으면 ACI는 aci 특성 및 아래 항목이 포함된 항목에 적용됩니다.

ACI에서 다음과 같은 강조 표시된 부분은 대상 규칙입니다.

```
(target_rule)(version 3.0; acl "ACL_name"; permission_rule bind_rules;)
```

복잡한 ACI의 경우 Directory Server는 ACI에서 다양한 키워드를 사용하여 여러 대상 규칙을 지원합니다.

```
(target_rule_1)(target_rule_2)(...)(version 3.0; acl "ACL_name"; permission_rule bind_rules;)
```

여러 대상 규칙을 지정하는 경우 순서가 관련이 없습니다. ACI에서 다음 각 키워드를 한 번만 사용할 수 있습니다.

- 대상

- **targetattr**
- **targetattrfilters**
- **targetfilter**
- **target_from**
- **target_to**

12.7.1. 대상 규칙의 구문

대상 규칙의 일반적인 구문은 다음과 같습니다.

```
(keyword comparison_operator "expression")
```

- **keyword:** 대상의 유형을 설정합니다.
- **comparison_operator:** 유효한 값은 **=** 및 **!=** 이며 대상이 표현식에 지정된 오브젝트인지 여부를 나타냅니다.

주의

보안상의 이유로 다른 모든 항목 또는 속성에 대해 지정된 작업을 허용하므로 != 연산자를 사용하지 않는 것이 좋습니다. 예를 들면 다음과 같습니다.

```
(targetattr != "userPassword");(version 3.0; acl "example"); allow
(write) ... );
```

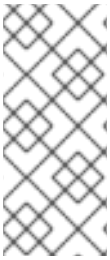
이전 예제에서는 사용자가 ACI를 설정한 Distinguished Name(DN) 아래의 userPassword 특성을 제외한 모든 속성을 설정, 업데이트 또는 삭제할 수 있습니다. 그러나 이를 통해 사용자는 예를 들어 이 속성에 대한 쓰기 액세스도 허용하는 추가 aci 특성을 추가할 수 있습니다.

- expression: 대상을 설정하고 따옴표로 구분해야 합니다. 표현식 자체는 사용하는 키워드에 따라 다릅니다.

12.7.2. 디렉터리 항목 대상 지정

Distinguished Name(DN) 및 그 아래의 항목을 기반으로 액세스를 제어하려면 ACL(액세스 제어 명령)에서 target 키워드를 사용합니다. target 키워드를 사용하는 대상 규칙은 DN을 표현식으로 사용합니다.

```
(target comparison_operator "ldap:///distinguished_name")
```



참고

대상 DN 또는 더 높은 수준의 DN에 키워드를 사용하여 ACI를 설정해야 합니다. 예를 들어 ou=People,dc=example,dc=com을 대상으로 지정하는 경우 ou=People,dc=example,dc=com 또는 dc=example,dc=com에 ACI를 설정해야 합니다.

예 12.1. target 키워드 사용

ou=People,dc=example,dc=com 항목에 저장된 사용자를 활성화하려면 해당 항목의 모든 속성을 검색하고 표시합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///ou=People,dc=example,dc=com") (version 3.0;
 acl "Allow users to read and search attributes of own entry"; allow (search, read)
 (userdn = "ldap:///self");)
```

target 키워드와 함께 와일드카드 사용

* 와일드카드 문자 대상 항목을 여러 개 사용할 수 있습니다.

다음 대상 규칙 예제에서는 **ou=People,dc=example,dc=com** 의 모든 항목과 일치하며, **uid** 속성이 문자 **a** 로 시작하는 값으로 설정됩니다.

```
(target = "ldap:///uid=a*,ou=People,dc=example,dc=com")
```

와일드카드 위치에 따라 규칙은 특성 값뿐만 아니라 전체 **DN**에도 적용됩니다. 따라서 와일드카드를 **DN** 부분 대신 사용할 수 있습니다.

예 12.2. 와일드카드를 사용하여 디렉터리 항목을 대상으로 지정

다음 규칙은 **dc=example,dc=com** 항목 자체에 저장된 항목뿐만 아니라 일치하는 **uid** 속성을 사용하여 **dc=example,dc=com** 트리의 모든 항목을 대상으로 합니다.

```
(target = "ldap:///uid=user_name*,dc=example,dc=com")
```

이전 대상 규칙은 다음과 같은 여러 항목과 일치합니다.

- uid=user_name,dc=example,dc=com
- uid=user_name,ou=People,dc=example,dc=com
- uid=user_name2,dc=example,dc=com

중요

디렉터리 Serverdoes는 DN의 접미사 부분에 있는 와일드카드를 지원하지 않습니다. 예를 들어 디렉터리 접미사가 dc=example,dc=com 인 경우 이 접미사의 와일드카드가 있는 대상을 사용할 수 없습니다 (target = "ldap:///dc=*.com ").

12.7.3. 대상 속성

ACL(액세스 제어 명령)의 액세스를 특정 속성으로 제한하려면 targetattr 키워드를 사용합니다. 예를 들어 이 키워드는 다음을 정의합니다.

- 읽기 작업에서는 클라이언트에 반환되는 속성입니다.
- 검색 작업에서는 검색할 속성입니다.
- 쓰기 작업에서는 오브젝트에 작성할 수 있는 속성을 지정합니다.
- 추가 작업에서는 새 오브젝트를 생성할 때 추가할 수 있는 속성

특정 상황에서는 다른 대상 키워드를 targetattr 과 결합하여 targetattr 키워드를 사용하여 ACI를 보호할 수 있습니다. 대상 규칙의 고급 사용을 참조하십시오.

중요

읽기 및 검색 작업에서는 기본 대상이 아닌 속성입니다. targetattr 키워드가 없는 ACI는 전체 항목에 영향을 미치는 권한이 있는 ACIs에만 유용합니다(예: 추가 또는 삭제).

targetattr 키워드를 사용하는 대상 규칙에서 여러 특성을 분리하려면 || 를 사용합니다.

```
(targetattr comparison_operator "attribute_1 || attribute_2 || ...")
```

표현식에 설정된 속성은 스키마에 정의해야 합니다.

표현식에 지정된 속성은 추가 대상 규칙으로 제한되지 않는 경우 ACI 및 그 아래의 모든 항목에 적용되

는 항목에 적용됩니다.

예 12.3. targetattr 키워드 사용

dc=example,dc=com 및 모든 하위 항목에 저장된 사용자를 활성화하려면 자체 항목에서 userPassword 특성을 업데이트합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap::server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "userPassword") (version 3.0;
 acl "Allow users updating own userPassword";
 allow (write) (userdn = "ldap:///self");)
```

targetattr 키워드와 함께 와일드카드 사용

* 와일드카드 문자를 사용하면 예를 들어 모든 특성을 대상으로 지정할 수 있습니다.

```
(targetattr = "*")
```

주의

보안상의 이유로 운영 특성을 포함하여 모든 속성에 대한 액세스를 허용하므로 targetattr 이 있는 와일드카드를 사용하지 마십시오. 예를 들어 사용자가 모든 속성을 추가하거나 수정할 수 있는 경우 사용자는 추가 ACI를 생성하고 자체 권한을 늘릴 수 있습니다.

12.7.4. LDAP 필터를 사용하여 항목 및 속성 지정

특정 조건과 일치하는 항목 그룹을 대상으로 하려면 LDAP 필터와 함께 targetfilter 키워드를 사용합니다.

```
(targetfilter comparison_operator "LDAP_filter")
```

filter 표현식은 표준 LDAP 검색 필터입니다.

예 12.4. targetfilter 키워드 사용

cn=Human Resources,dc=example,dc.com 그룹의 구성원에게 권한을 부여하려면 **department** 속성이 **Engineering** 또는 **Sales** 로 설정된 모든 항목을 수정합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetfilter = "(|(department=Engineering)(department=Sales)")
 (version 3.0; acl "Allow HR updating engineering and sales entries";
 allow (write) (groupdn = "ldap:///cn=Human Resources,dc=example,dc.com");)
```

targetfilter 키워드는 전체 항목을 대상으로 합니다. **targetattr** 키워드와 결합하는 경우 **ACI**(액세스 제어 명령)는 대상 항목의 특성 하위 집합에만 적용됩니다. 필터와 일치하는 항목의 특정 속성 지정을 참조하십시오.

참고

LDAP 필터를 사용하면 디렉터리에 분산된 항목 및 속성을 대상으로 할 때 유용합니다. 그러나 필터에서 액세스 권한을 관리하는 오브젝트의 이름을 직접 지정하지 않기 때문에 결과가 예기치 않은 경우가 있습니다. **ACI**가 대상으로 하는 항목 세트는 속성이 추가되거나 삭제되므로 변경될 수 있습니다. 따라서 **ACI**에서 **LDAP** 필터를 사용하는 경우 동일한 필터를 사용하여 동일한 항목 및 속성을 대상으로 하는지 확인합니다(예: **ldapsearch** 작업).

targetfilter 키워드와 함께 와일드카드 사용

targetfilter 키워드는 표준 **LDAP** 필터와 유사한 와일드카드를 지원합니다. 예를 들어 값이 **adm** 으로 시작되는 모든 **uid** 특성을 대상으로 지정하려면 다음을 사용합니다.

```
(targetfilter = "(uid=adm*) ...)
```

12.7.5. LDAP 필터를 사용하여 특성 값 지정

액세스 제어를 사용하여 특성의 특정 값을 대상으로 지정할 수 있습니다. 즉, 특성 값이 **ACL**(액세스 제어 명령)에 정의된 기준을 충족하는 경우 속성에 대한 권한을 부여하거나 거부할 수 있습니다. 특성 값에 따라 액세스를 허용하거나 거부하는 **ACI**를 값 기반 **ACI**라고 합니다. 이는 **ADD** 및 **DEL** 작업에만 적용됩니다. 검색 권한은 특정 값으로 제한할 수 없습니다.

값 기반 **ACI**를 생성하려면 다음 구문과 함께 **targattrfilters** 키워드를 사용합니다.

- 하나의 속성 및 필터 조합을 사용하는 한 가지 작업의 경우:

  (targattrfilters="*operation=attribute*:*filter*")

- 여러 속성 및 필터 조합이 있는 하나의 작업에 대해 다음을 수행합니다.

  (targattrfilters="*operation=attribute_1*:*filter_1* && *attribute_2*:*filter_2* ... && *attribute_m*:*filter_m*")

- 두 작업의 경우 각각 여러 속성 및 필터 조합이 있습니다.

  (targattrfilters="*operation_1=attribute_1_1*:*filter_1_1* && *attribute_1_2*:*filter_1_2* ... && *attribute_1_m*:*filter_1_m* , *operation_2=attribute_2_1*:*filter_2_1* && *attribute_2_2:filter_2_2* ... & *attribute_2_n*:*filter_2_n* ")

이전 구문 예제에서 작업을 추가 또는 **del** 로 설정할 수 있습니다. **attribute:filter** 조합은 필터와 필터가 적용되는 특성을 설정합니다.

다음은 **filter**와 일치해야 하는 방법을 설명합니다.

- 항목을 생성하고 필터가 새 항목의 속성에 적용되는 경우 해당 특성의 각 인스턴스가 필터와 일치해야 합니다.

- 항목을 삭제하고 필터가 항목의 속성에 적용되는 경우 해당 특성의 각 인스턴스도 필터와 일치해야 합니다.

- 항목 및 작업을 수정할 때 특성을 추가하면 해당 특성에 적용되는 **add** 필터가 일치해야 합니다.

- 작업에서 특성을 삭제하면 해당 특성에 적용되는 **del** 필터가 일치해야 합니다. 항목에 이미 존재하는 속성의 개별 값이 교체되면 **add** 및 **del** 필터가 모두 일치해야 합니다.

예 12.5. targattrfilters 키워드 사용

사용자가 **Admin** 역할을 제외한 자신의 항목에 역할을 추가할 수 있는 **ACI**를 만들고, **value**가 **ECDHE** 접두사로 시작하는 한, 전화 속성을 추가하려면 다음을 입력합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap::server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targattrfilters="add=nsroledn:(!(nsroledn=cn=Admin)) &&
 telephoneNumber:(telephoneNumber=123*)") (version 3.0;
 acl "Allow adding roles and telephone";
 allow (add) (userdn = "ldap:///self");)
```

12.7.6. 대상 소스 및 대상 DN

특정 상황에서 관리자는 사용자가 디렉터리 항목을 이동할 수 있도록 허용하려고 합니다. **ACL**(액세스 제어 명령)에서 **target_from** 및 **target_to** 키워드를 사용하여 사용자를 활성화하지 않고 작업의 소스 및 대상을 지정할 수 있습니다.

- **ACI**에 설정된 다른 소스에서 항목을 이동하려면 다음을 수행합니다.
- 항목을 **ACI**에 설정된 것과 다른 대상으로 이동하려면 다음을 수행합니다.
- 소스 **Distinguished Name(DN)**에서 기존 항목을 삭제하려면 다음을 수행합니다.
- 대상 **DN**에 새 항목을 추가하려면 다음을 수행합니다.

예 12.6. target_from 및 target_to 키워드 사용

uid=user,dc=example,dc=com 계정을 활성화하여 사용자 계정을 **cn=staging,dc=example,dc=com** 항목에서 **cn=people,dc=example,dc=com** 으로 이동하려면 다음을 입력합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap:server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target_from="ldap:///uid=*,cn=staging,dc=example,dc=com")
```

```
(target_to="ldap:///cn=People,dc=example,dc=com")
(version 3.0; acl "MODDN from"; allow (moddn))
userdn="ldap:///uid=user,dc=example,dc=com";)
```

ACIS는 정의된 하위 트리에만 적용됩니다. 이전 예에서 ACI는 dc=example,dc=com 하위 트리에만 적용됩니다.

target_from 또는 target_to 키워드가 설정되지 않은 경우 ACI는 소스 또는 대상과 일치합니다.

12.8. 대상 규칙의 고급 사용

여러 키워드를 결합하여 복잡한 대상 규칙을 만들 수 있습니다. 이 섹션에서는 대상 규칙의 고급 사용 예를 제공합니다.

12.8.1. 그룹 생성 및 유지 관리 권한 위임

특정 상황에서 관리자는 권한을 다른 계정 또는 그룹에 위임하려는 경우가 있습니다. 대상 키워드를 결합하면 이 요청을 해결하는 ACI(Secure Access Control instructions)를 만들 수 있습니다.

예 12.7. 그룹 생성 및 유지 관리 권한 위임

uid=user,ou=People,dc=example,dc=com" 계정을 활성화하여 ou=groups,dc=example,dc=com 항목에서 그룹을 생성하고 업데이트하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///cn=*,ou=Groups,dc=example,dc=com")
 (targattrfilters="add=objectclass:(|(objectclas=top)(objectclass=groupOfUniqueNames)))
 (targetattr="cn || uniqueMember || objectClass")
 (version 3.0; acl "example"; allow (read, search, write, add)
 (userdn = "ldap:///uid=test,ou=People,dc=example,dc=com");)
```

보안상의 이유로 이전 예제에서는 특정 제한 사항을 추가합니다. uid=test,ou=People,dc=example,dc=com 사용자

- top 및 groupOfUniqueNames 오브젝트 클래스를 포함해야 하는 오브젝트를 생성할 수 있습니다.

- account 와 같은 추가 오브젝트 클래스를 추가할 수 없습니다. 예를 들어 로컬 인증에 Directory Server 계정을 사용하여 root 사용자의 경우 0 과 같은 잘못된 사용자 ID를 사용하여 새 사용자를 만들 수 없습니다.

targetfilter 규칙을 사용하면 ACI 항목이 groupofuniquenames 개체 클래스의 항목에만 적용되고 targetattrfilter 규칙을 통해 다른 개체 클래스가 추가되지 않도록 합니다.

12.8.2. 항목 및 속성 모두 대상 지정

대상은 고유 이름(DN)에 따라 액세스를 제어합니다. 그러나 와일드카드와 targetattr 키워드와 함께 사용하는 경우 항목과 속성을 모두 대상으로 지정할 수 있습니다.

예 12.8. 항목 및 속성 모두 대상 지정

dc=example,dc=com 하위 트리의 모든 조직 단위에서 uid=user,ou=People,dc=example,dc.com 사용자를 활성화하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///cn=*,dc=example,dc=com")(targetattr="member" || "cn") (version 3.0;
 acl "Allow uid=user to search and read members of groups";
 allow (read, search) (userdn = "ldap:///uid=user,ou=People,dc=example,dc.com");)
```

12.8.3. 필터와 일치하는 항목의 특정 속성을 대상으로 지정

두 개의 대상 규칙에 targetattr 및 targetfilter 키워드를 결합하는 경우 필터와 일치하는 항목의 특정 특성을 대상으로 지정할 수 있습니다.

예 12.9. 필터와 일치하는 항목의 특정 속성을 대상으로 지정

cn=Engineering Admins,dc=example,dc=com 그룹의 멤버가 department 특성을 Engineering 으로 설정한 모든 항목의 jpeg ECDHE 및 manager 특성을 수정할 수 있도록 허용하려면 다음을 입력합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
```

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "jpegPhoto || manager")
 (targetfilter = "(department=Engineering)") (version 3.0;
 acl "Allow engineering admins updating jpegPhoto and manager of department
members";
 allow (write) (groupdn = "ldap:///cn=Engineering Admins,dc=example,dc=com");)
```

12.8.4. 단일 디렉터리 항목 지정

단일 디렉터리 항목을 대상으로 하려면 **targetattr** 및 **targetfilter** 키워드를 결합합니다.

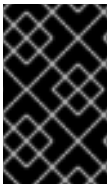
예 12.10. 단일 디렉터리 항목 지정

uid=user,ou=People,dc=example,dc=com 사용자를 활성화하려면 **ou=Engineering,dc=example,dc=com** 항목에서 **ou** 및 **cn** 속성을 읽고 검색합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=Engineering,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "ou || cn")
 (targetfilter = "(ou=Engineering)") (version 3.0;
 acl "Allow uid=user to search and read engineering attributes";
 allow (read, search) (userdn = "ldap:///uid=user,ou=People,dc=example,dc=com");)
```

이전 예제를 활성화하여 **ou=Engineering,dc=example,dc=com** 항목, **ou=Engineering,dc=example,dc=com** 의 하위 항목만 대상으로 지정하려면 **ou** 속성이 **Engineering** 으로 설정되어 있지 않아야 합니다.



중요

디렉터리 구조가 변경되면 이러한 종류의 **ACI**가 실패할 수 있습니다.

또는 바인딩 요청의 사용자 입력과 일치하는 바인딩 규칙을 대상 항목에 저장된 특성 값을 사용하여 생성할 수 있습니다. 값이 일치하는 값을 기반으로 액세스 정의를 참조하십시오.

12.9. ACI 권한 정의

권한 규칙은 ACL(액세스 제어 명령)과 액세스가 허용되거나 거부되는지와 관련된 권한을 정의합니다.

ACI에서 강조 표시된 부분은 권한 규칙입니다.

(*target_rule*) (version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*;)

12.9.1. 권한 규칙의 구문

권한 규칙의 일반 구문은 다음과 같습니다.

permission (*rights*)

- 권한: ACL(액세스 제어 명령)에서 권한을 허용하거나 거부하는지 여부를 설정합니다.
- rights: ACI가 허용하거나 거부할 수 있는 권한을 설정합니다. 권한 규칙에서 사용자 권한을 참조하십시오.

예 12.11. 권한 정의

ou=People,dc=example,dc=com 항목에 저장된 사용자를 활성화하려면 해당 항목의 모든 속성을 검색하고 표시합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///ou=People,dc=example,dc=com") (version 3.0;
 acl "Allow users to read and search attributes of own entry"; allow (search, read)
 (userdn = "ldap:///self");)
```

12.9.2. 권한 규칙에 대한 사용자 권한

권한 규칙의 권한은 부여되거나 거부되는 작업을 정의합니다. ACI에서는 다음 권한 중 하나 또는 여러 개를 설정할 수 있습니다.

표 12.1. 사용자 권한

right	설명
read	사용자가 디렉터리 데이터를 읽을 수 있는지 여부를 설정합니다. 이 권한은 LDAP의 검색 작업에만 적용됩니다.
쓰기	속성을 추가, 수정 또는 삭제하여 사용자가 항목을 수정할 수 있는지 여부를 설정합니다. 이 권한은 LDAP의 **수정** 및 **modrdn** 작업에 적용됩니다.
add	사용자가 항목을 만들 수 있는지 여부를 설정합니다. 이 권한은 LDAP의 **add** 작업에만 적용됩니다.
delete	사용자가 항목을 삭제할 수 있는지 여부를 설정합니다. 이 권한은 LDAP의 **삭제** 작업에만 적용됩니다.
search	사용자가 디렉터리 데이터를 검색할 수 있는지 여부를 설정합니다. 반환된 데이터를 검색 결과의 일부로 보려면 **검색** 및 **읽기** 권한을 할당합니다. 이 권한은 LDAP의 검색 작업에만 적용됩니다.
비교	사용자가 제공하는 데이터를 디렉터리에 저장된 데이터와 비교할 수 있는지 여부를 설정합니다. **비교** 권한을 사용하면 디렉터리에서 성공 또는 실패 메시지를 반환하여 조회에 응답하지만 사용자는 항목 또는 속성 값을 볼 수 없습니다. 이 권한은 LDAP의 비교 작업에만 적용됩니다.
selfwrite	사용자가 그룹에서 고유 이름(DN)을 추가하거나 삭제할 수 있는지 여부를 설정합니다. 이 권한은 그룹 관리에만 사용됩니다.
proxy	지정된 DN이 다른 항목의 권한을 사용하여 대상에 액세스할 수 있는지 여부를 설정합니다. **프록시** 권한은 ACL 범위 내에서 부여되며, 권한을 부여한 사용자 또는 그룹은 모든 Directory Server 사용자로 명령을 실행할 수 있습니다. 특정 사용자에게 **프록시** 권한을 제한할 수 없습니다. 보안상의 이유로 디렉터리의 가장 대상 수준에서 프록시를 사용하는 ACI를 설정합니다.
all	**프록시** 를 제외한 모든 권한을 설정합니다.

12.9.3. LDAP 작업에 필요한 권한

This section describes the rights you must grant to users depending on the type of LDAP operation you want to authorize them to perform.

- **항목 추가:**
  - **추가할 항목에 대한 추가 권한을 부여합니다.**
  - **항목의 각 속성 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 targattrfilters 키워드를 사용하여 제한할 수 있습니다.**

- 항목 삭제:
  - 삭제할 항목에 삭제 권한을 부여합니다.
  - 항목의 각 속성 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 **targattrfilters** 키워드를 사용하여 제한할 수 있습니다.
- 항목의 속성 수정:
  - 특성 유형에 대한 쓰기 권한을 부여합니다.
  - 각 특성 유형의 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 **targattrfilters** 키워드를 사용하여 제한할 수 있습니다.
- 항목의 **RDN** 수정:
  - 항목에 대한 쓰기 권한을 부여합니다.
  - 새 **RDN**에 사용되는 속성 유형에 대한 쓰기 권한을 부여합니다.
  - 이전 **RDN**을 삭제할 수 있는 권한을 부여하려면 이전 **RDN**에서 사용되는 속성 유형에 대한 쓰기 권한을 부여합니다.
  - 새 **RDN**에 사용되는 속성 유형의 값에 대한 쓰기 권한을 부여합니다. 이 권한은 기본적으로 부여되지만 **targattrfilters** 키워드를 사용하여 제한할 수 있습니다.
- 특성 값 비교:
  - 특성 유형에 대한 비교 권한을 부여합니다.

- 항목 검색:
  - 검색 필터에 사용되는 각 속성 유형에 대한 검색 권한을 부여합니다.
  - 항목에 사용된 속성 유형에 대한 읽기 권한을 부여합니다.

12.10. ACI 바인딩 규칙 정의

ACL(액세스 제어 명령)의 바인딩 규칙은 Directory Server가 ACI를 적용하도록 충족해야 하는 필수 바인딩 매개 변수를 정의합니다. 예를 들어 다음을 기반으로 바인딩 규칙을 설정할 수 있습니다.

- DNS
- 그룹 멤버십 또는 할당된 역할
- 항목이 바인딩해야 하는 위치
- 바인딩 중 사용해야 하는 인증 유형
- 바인딩이 발생한 시간 또는 일

ACI에서 다음과 같이 강조 표시된 부분은 bind 규칙입니다.

> (*target_rule*) (version 3.0; acl "*ACL_name*"; *permission_rule bind_rules*;)

12.10.1. 바인딩 규칙의 구문

바인딩 규칙의 일반 구문은 다음과 같습니다.

> *keyword comparison_operator "expression"*

- **keyword:** 바인딩 작업 유형을 설정합니다.

- **comparison_operator:** 유효한 값은 **=** 및 **!=** 이며 대상이 표현식에 지정된 오브젝트인지 여부를 나타냅니다. 키워드가 추가 비교 연산자를 지원하는 경우 해당 섹션에 언급됩니다.

- **expression:** 표현식을 설정하고 따옴표로 구분해야 합니다. 표현식 자체는 사용하는 키워드에 따라 다릅니다.

12.10.2. 사용자 기반 액세스 정의

userdn 키워드를 사용하면 하나 이상의 **DN**에 따라 액세스 권한을 부여하거나 거부할 수 있으며 다음 구문을 사용할 수 있습니다.

> **userdn** *comparison_operator* **"***ldap:///distinguished_name* **||** *ldap:///distinguished_name* **|| ..."**

표현식의 **DN**을 다음과 같이 설정합니다.

- **DN:** **userdn** 키워드가 있는 **DN** 사용에서 확인할 수 있습니다.

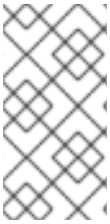
- **LDAP** 필터: **LDAP** 필터와 함께 **userdn** 키워드 사용을 참조하십시오.

- **anyone alias:** **anonymous** 액세스 부여를 참조하십시오.

- **all alias:** 인증된 사용자에 대한 액세스 권한 부여를 참조하십시오.

- 자체 별칭: 사용자가 자신의 항목에 액세스할 수 있도록 활성화 를 참조하십시오.

- 상위 별칭: 사용자의 하위 항목에 대한 액세스 설정을 참조하십시오.



참고

LDAP URL 내에서 호스트 이름 또는 포트 번호를 지정하지 마십시오. **URL**은 항상 로컬 서버에 적용됩니다.

userdn 키워드와 함께 **DN** 사용

userdn 키워드를 **DN**(고유 이름)으로 설정하여 일치하는 항목에만 **ACI**를 적용합니다. 여러 항목을 일치시키려면 **DN**의 * 와일드카드를 사용합니다.

DN과 함께 **userdn** 키워드를 사용하면 다음 구문과 일치해야 합니다.

```
userdn comparison_operator ldap:///distinguished_name
```

예 12.12. userdn 키워드와 함께 DN 사용

uid=admin,ou=People,dc=example,dc=com 사용자를 활성화하려면 **ou=People,dc=example,dc=com** 항목에 있는 다른 모든 사용자의 **manager** 속성을 읽습니다.

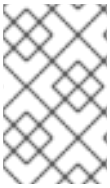
```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="manager") (version 3.0; acl "Allow uid=admin reading manager attribute";
 allow (search, read) userdn = "ldap:///uid=admin,ou=People,dc=example,dc=com";)
```

LDAP 필터와 함께 **userdn** 키워드 사용

사용자에게 권한을 동적으로 허용하거나 거부하려면 **LDAP** 필터와 함께 **userdn** 키워드를 사용합니다.

```
userdn comparison_operator "ldap:///distinguished_name??scope?(filter)"
```



참고

LDAP 필터는 * 와일드카드를 지원합니다.

예 12.13. LDAP 필터와 함께 userdn 키워드 사용

department 속성이 human Resources 로 설정된 사용자가 ou=People,dc=example,dc=com 항목에 있는 사용자의 homePostalAddress 특성을 업데이트하도록 하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="homePostalAddress") (version 3.0;
 acl "Allow HR setting homePostalAddress"; allow (write)
 userdn = "ldap:///ou=People,dc=example,dc=com??sub?(department=Human
Resources)";)
```

익명 액세스 권한 부여

특정 상황에서는 관리자가 디렉터리의 데이터에 대한 익명 액세스를 구성하려고 합니다. 익명 액세스는 다음을 제공하여 디렉터리에 바인딩할 수 있음을 나타냅니다.

- DN 및 암호 바인딩되지 않음
- 유효한 바인딩 DN 및 암호

익명 액세스를 구성하려면 bind 규칙에서 userdn 키워드와 함께 ldap:///anyone 표현식을 사용합니다.

```
userdn comparison_operator "ldap:///anyone"
```

예 12.14. 익명 액세스 권한 부여

인증이 없는 사용자를 활성화하여 ou=People,dc=example,dc=com 항목의 sn,givenName 및 telephoneNumber 속성을 읽고 검색하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H __ldap://server.example.com -x`
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="sn" || targetattr="givenName" || targetattr = "telephoneNumber")
 (version 3.0; acl "Anonymous read, search for names and phone numbers";
 allow (read, search) userdn = "ldap:///anyone")
```

인증된 사용자에게 액세스 권한 부여

특정 상황에서 관리자는 익명 바인딩을 제외하고 **Directory Server**에 성공적으로 바인딩할 수 있는 모든 사용자에게 권한을 부여하려고 합니다. 이 기능을 구성하려면 **bind** 규칙에 **userdn** 키워드와 함께 **ldap:///all** 표현식을 사용합니다.

```
userdn comparison_operator "ldap:///all"
```

예 12.15. 인증된 사용자에게 액세스 권한 부여

인증된 사용자가 **ou=example,ou=groups,dc=example,dc=com** 그룹의 멤버로 추가하고 제거할 수 있도록 하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=example,ou=Groups,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="member") (version 3.0;
 acl "Allow users to add/remove themselves from example group";
 allow (selfwrite) userdn = "ldap:///all")
```

사용자가 자신의 항목에 액세스할 수 있도록 허용

사용자에게 자신의 항목에 대한 액세스를 허용하거나 거부하는 **ACI**를 설정하려면 **bind** 규칙의 **userdn** 키워드와 함께 **ldap:///self** 표현식을 사용합니다.

```
userdn comparison_operator "ldap:///self"
```

예 12.16. 사용자가 자신의 항목에 액세스할 수 있도록 허용

ou=People,dc=example,dc=com 항목에서 사용자를 활성화하여 자체 **userPassword** 특성을 업데이트하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="userPassword") (version 3.0;
 acl "Allow users updating their password";
 allow (write) userdn = "ldap:///self")
```

사용자의 하위 항목에 대한 액세스 설정

바인딩 DN이 대상 항목의 상위인 경우에만 항목에 대한 액세스 권한이 부여되거나 거부되도록 지정하려면 바인딩 규칙에 userdn 키워드와 함께 self:///parent 표현식을 사용합니다.

```
userdn comparison_operator "ldap:///parent"
```

예 12.17. 사용자의 하위 항목에 대한 액세스 설정

cn=user,ou=People,dc=example,dc=com 사용자를 활성화하려면 cn=example,cn=user,ou=People,dc=example,dc=com 과 같은 자체 하위 항목의 manager 속성을 업데이트합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=user,ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="manager") (version 3.0;
 acl "Allow cn=user to update manager attributes";
 allow (write) userdn = "ldap:///parent")
```

12.10.3. 그룹 기반 액세스 정의

그룹 기반 액세스 제어 명령(ACI)을 사용하면 그룹에 사용자를 추가하거나 제거하여 액세스를 관리할 수 있습니다. 그룹 멤버십을 기반으로 하는 ACI를 구성하려면 groupdn 키워드를 사용합니다. 사용자가 지정된 그룹 중 하나 또는 여러 그룹의 멤버인 경우 ACI는 일치합니다.

groupdn 키워드를 사용할 때 Directory Server는 다음 특성을 기반으로 그룹 멤버십을 확인합니다.

- 멤버
- uniqueMember
- memberURL
- memberCertificateDescription

groupdn 키워드를 사용하여 규칙을 바인딩하려면 다음 구문을 사용합니다.

```
groupdn comparison_operator "ldap:///distinguished_name || ldap:///distinguished_name || ..."
```

표현식에서 고유 이름(DN)을 다음과 같이 설정합니다.

- A DN. groupdn 키워드를 사용하여 DN 사용을 참조하십시오.
- LDAP 필터. LDAP 필터와 함께 groupdn 키워드 사용을참조하십시오.

하나의 바인딩 규칙에 여러 DN을 설정하면 인증된 사용자가 이러한 그룹 중 하나의 멤버인 경우 Directory Server는 ACI를 적용합니다. 사용자를 여러 그룹의 멤버로 설정하려면 여러 groupdn 키워드를 사용하고 부울 및 연산자를 사용하여 결합합니다. 자세한 내용은 부울 Operator를 사용하여 바인딩 규칙 결합 을 참조하십시오.

참고

LDAP URL 내에서 호스트 이름 또는 포트 번호를 지정하지 마십시오. URL은 항상 로컬 서버에 적용됩니다.

groupdn 키워드를 사용하여 DN 사용

그룹 멤버에 ACI를 적용하려면 groupdn 키워드를 그룹의 DN으로 설정합니다.

DN으로 설정된 groupdn 키워드는 다음 구문을 사용합니다.

```
groupdn comparison_operator ldap:///distinguished_name
```

예 12.18. groupdn 키워드와 함께 DN 사용

cn=example,ou=Groups,dc=example,dc=com 그룹의 멤버를 활성화하려면 ou=People,dc=example,dc=com 에서 항목의 manager 속성을 검색하고 읽습니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
```

```
add: aci
aci: (targetattr="manager") (version 3.0;
 acl "Allow example group to read manager attribute";
 allow (search, read) groupdn = "ldap:///cn=example,ou=Groups,dc=example,dc=com";)
```

LDAP 필터와 함께 groupdn 키워드 사용

groupdn 키워드와 함께 LDAP 필터를 사용하면 인증된 사용자가 ACI와 일치하도록 필터 검색에서 반환하는 그룹 중 하나 이상의 멤버여야 함을 정의할 수 있습니다.

LDAP 필터가 포함된 groupdn 키워드는 다음 구문을 사용합니다.

```
groupdn comparison_operator "ldap:///distinguished_name??scope?(filter)"
```

참고

LDAP 필터는 * 와일드카드를 지원합니다.

예 12.19. LDAP 필터와 함께 groupdn 키워드 사용

manager 속성이 example으로 설정된 dc=example,dc=com 및 subtrees 그룹 멤버를 활성화하려면 ou=People,dc= example,dc=com 에서 항목의 homePostalAddress 를 업데이트합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="homePostalAddress") (version 3.0;
 acl "Allow manager=example setting homePostalAddress"; allow (write)
 userdn = "ldap:///dc=example,dc=com??sub?(manager=example)";)
```

12.10.4. 값 일치에 따른 액세스 정의

바인딩 규칙에 userattr 키워드를 사용하여 디렉터리와 대상 항목에 바인딩하는 데 사용된 항목 간에 일치해야 하는 속성을 지정합니다.

userattr 키워드는 다음 구문을 사용합니다.

```
userattr comparison_operator "attribute_name#bind_type_or_attribute_value
```

-

자세한 내용은 다음을 참조하십시오.

- USERDN 바인딩 유형 사용
- GROUPDN 바인딩 유형 사용
- ROLEDN 바인딩 유형 사용
- SELFDN 바인딩 유형 사용
- LDAPURL 바인딩 유형 사용
- userattr 키워드와 함께 사용

중요

기본적으로 **Directory Server**는 생성된 항목에 대한 액세스 권한을 평가합니다. 그러나 동일한 수준의 사용자 개체를 방지하기 위해 **Directory Server**는 **userattr** 키워드를 사용할 때 **ACI**(액세스 제어 명령)를 설정하는 항목에 추가 권한을 부여하지 않습니다. 이 동작을 구성하려면 **parent** 키워드와 함께 **userattr** 키워드를 사용하고 수준 **0**에서 추가 권한을 부여합니다.

ECDHE에 대한 자세한 내용은 값 일치 값을 기반으로 액세스 정의를 참조하십시오.

USERDN 바인딩 유형 사용

바인딩 사용자 고유 이름(**DN**)이 속성에 저장된 **DN**과 일치하는 경우 **USERDN** 바인딩 유형을 사용합니다.

USERDN 바인딩 유형의 **userattr** 키워드에는 다음 구문이 필요합니다.

```
userattr comparison_operator "attribute_name#USERDN"
```

예 12.20. USERDN 바인딩 유형 사용

관리자에게 해당 담당자의 **phone Number** 속성에 모든 권한을 부여하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "telephoneNumber")
 (version 3.0; acl "Manager: telephoneNumber";
 allow (all) userattr = "manager#USERDN";)
```

이전 **ACI**는 **ou=People,dc=example,dc=com** 의 항목에서 작업을 수행하는 사용자의 **DN**이 이 항목의 **manager** 속성에 저장된 **DN**과 일치하는 경우 **true**로 평가됩니다.

GROUPDN 바인딩 유형 사용

바인딩 사용자 **DN**이 속성에 그룹 세트의 멤버인 경우 **ACI**를 적용하려면 **GROUPDN** 바인딩 유형을 사용합니다.

GROUPDN 바인딩 유형의 **userattr** 키워드에는 다음 구문이 필요합니다.

```
userattr comparison_operator "attribute_name#GROUPDN"
```

예 12.21. GROUPDN 바인딩 유형 사용

사용자에게 **ou=Socialrule,ou=Groups,dc=example,dc=com** 항목에서 소유한 그룹 항목을 삭제할 수 있는 권한을 부여하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=Social Committee,ou=Groups,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ou=Social Committee,ou=Groups,dc=example,dc=com)
 (targattrfilters="del=objectClass:(objectClass=groupOfNames)")
 (version 3.0; acl "Delete Group";
 allow (delete) userattr = "owner#GROUPDN";)
```

이전 **ACI**는 작업을 수행하는 사용자의 **DN**이 **owner** 속성에 지정된 그룹의 멤버인 경우 **true**로 평가됩니다.

지정된 그룹은 동적 그룹일 수 있으며 그룹의 **DN**은 데이터베이스의 접미사 아래에 있을 수 있습니다. 그러나 서버에 의한 이러한 유형의 **ACI** 평가는 매우 리소스 집약적입니다.

대상 항목과 동일한 접미사가 있는 정적 그룹을 사용하는 경우 더 나은 성능을 위해 다음 표현식을 사용합니다.

```
userattr comparison_operator "ldap:///distinguished_name?attribute_name#GROUPDN"
```

ROLEDN 바인딩 유형 사용

바인딩 사용자가 속성에 지정된 역할에 속하는 경우 **ACI**를 적용하려면 **ROLEDN** 바인딩 유형을 사용합니다.

ROLEDN 바인딩 유형의 **userattr** 키워드에는 다음 구문이 필요합니다.

```
userattr comparison_operator "attribute_name#ROLEDN"
```

예 12.22. ROLEDN 바인딩 유형 사용

cn=Administrators,dc=example,dc=com 역할의 사용자를 활성화하려면 **ou=People,dc=example,dc=com** 에서 항목의 **manager** 속성을 검색하고 읽습니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (version 3.0; acl "Allow example role owners to read manager attribute";
 allow (search, read) userattr = manager#ROLEDN;)
```

지정된 역할은 데이터베이스의 접미사 아래에 있을 수 있습니다. 필터링된 역할을 사용하는 경우 이 유형의 **ACI** 평가에서는 서버에서 많은 리소스를 사용합니다.

정적 역할 정의를 사용하고 있으며 역할 항목이 대상 항목과 동일한 접미사 아래에 있는 경우 성능 향상을 위해 다음 표현식을 사용합니다.

SELFDN 바인딩 유형 사용

SELFDN 바인딩 유형을 사용하면 바인딩된 사용자의 DN이 항목의 단일 값 속성에 설정된 경우 권한을 부여할 수 있습니다.

SELFDN 바인딩 유형의 userattr 키워드에는 다음 구문이 필요합니다.

```
userattr comparison_operator "attribute_name#SELFDN"
```

예 12.23. SELFDN 바인딩 유형 사용

사용자가 ipatokenOwner 속성에 bind 사용자의 DN이 설정된 ipatokenuniqueid=*,cn=otp,dc=example,dc=com 항목을 추가하도록 활성화하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=otp,dc=example,dc=com
changetype: modify
add: aci
aci: (target = "ldap:///ipatokenuniqueid=*,cn=otp,dc=example,dc=com")
 (targetfilter = "(objectClass=ipaToken)")(version 3.0;
 acl "token-add-delete"; allow (add) userattr = "ipatokenOwner#SELFDN";)
```

LDAPURL 바인딩 유형 사용

바인딩 DN이 대상 항목의 속성에 지정된 필터와 일치하는 경우 ACL을 적용하려면 LDAPURL 바인딩 유형을 사용합니다.

LDAPURL 바인딩 유형의 userattr 키워드에는 다음 구문이 필요합니다.

```
userattr comparison_operator "attribute_name#LDAPURL"
```

예 12.24. LDAPURL 바인딩 유형 사용

aciurl 속성이 ldap:///ou=People,dc=example,dc=com?one?(uid=user*) 로 설정된 사용자 오브젝트에 읽기 및 검색 권한을 부여하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
```

```
aci: (targetattr = "*")
 (version 3.0; acl "Allow read,search "; allow (read,search)
 (userattr = "aciurl#LDAPURL);)
```

userattr 키워드와 함께 사용

userattr 키워드를 사용하여 대상 항목과 바인딩하는 데 사용되는 항목을 연결하면 **ACI**는 지정된 대상에만 적용되며 해당 항목 아래의 항목에는 적용되지 않습니다. 특정 상황에서 관리자는 대상 항목 아래의 **ACI**의 애플리케이션을 여러 수준으로 확장하려고 합니다. 이는 **parent** 키워드를 사용하고 **ACI**를 상속해야 하는 대상 아래의 수준 수를 지정하여 가능합니다.

userattr 키워드를 **parent** 키워드와 함께 사용하는 경우 구문은 다음과 같습니다.

```
userattr comparison_operator
"parent[inheritance_level].attribute_name#bind_type_or_attribute_value
```

- **ECDHE_level:** 대상에서 **ACI**를 상속한 수준 수를 나타내는 쉼표로 구분된 목록입니다. 대상 항목 아래에 **5** 단계 **(0,1,2,3,4)**를 포함할 수 있습니다. **0(0)**은 대상 항목을 나타냅니다.
- **attribute_name: userattr** 또는 **groupattr** 키워드가 대상으로 하는 속성입니다.
- **bind_type_or_attribute_value: USERDN** 과 같은 특성 값 또는 바인딩 유형을 설정합니다.

예를 들면 다음과 같습니다.

```
userattr = "parent[0,1].manager#USERDN"
```

바인딩 **DN**이 대상 항목의 **manager** 특성과 일치하는 경우 이 바인딩 규칙은 **true**로 평가됩니다. 바인딩 규칙이 **true**로 평가될 때 부여된 권한은 대상 항목과 그 아래의 모든 항목에 적용됩니다.

예 12.25. userattr 키워드와 함께 사용

사용자가 **cn=Profiles,dc=example,dc=com** 항목을 읽고 검색할 수 있도록 하려면 소유자 속성에 사용자 **DN**이 설정된 첫 번째 수준과 **cn=mail,cn=Profiles,dc=example,dc=com** 및 **cn=news,dc=example,dc= com**을 포함하는 하위 항목의 첫 번째 수준을 표시합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x`
```

```
dn: cn=Profiles,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="*") (version 3.0; acl "Profile access",
 allow (read,search) userattr="parent[0,1].owner#USERDN" ;)
```

12.10.5. 특정 IP 주소 또는 범위에서 액세스 정의

bind 규칙의 **ip** 키워드를 사용하면 특정 **IP** 주소 또는 **IP** 주소 범위에서 액세스 권한을 부여하거나 거부할 수 있습니다.

ip 키워드를 사용한 바인딩 규칙은 다음 구문을 사용합니다.

```
ip comparison_operator "IP_address_or_range"
```

예 12.26. 바인딩 규칙에서 **IPv4** 주소 범위 사용

192.0.2.0/24 네트워크에서 **dc=example,dc=com** 항목으로의 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny 192.0.2.0/24"; deny (all)
 (userdn = "ldap:///anyone") and (ip != "192.0.2.");)
```

예 12.27. 바인딩 규칙에서 **IPv6** 주소 범위 사용

2001:db8::/64 네트워크에서 **dc=example,dc=com** 항목으로의 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny 2001:db8::/64"; deny (all)
 (userdn = "ldap:///anyone") and (ip != "2001:db8::");)
```

12.10.6. 특정 호스트 또는 도메인의 액세스 정의

bind 규칙의 **dns** 키워드를 사용하면 특정 호스트 또는 도메인의 액세스 권한을 부여하거나 거부할 수 있습니다.

주의

Directory Server에서 **DNS**를 사용하여 정규화된 도메인 이름(**FQDN**)에 연결 **IP** 주소를 확인할 수 없는 경우 서버는 이 클라이언트에 대한 **dns** 바인딩 규칙에 **ACI**(액세스 제어 명령)를 적용하지 않습니다.

DNS를 사용하여 클라이언트 **IP** 주소를 확인할 수 없는 경우 대신 **ip** 키워드 및 **IP** 주소를 사용합니다. 특정 **IP** 주소 또는 범위에서 액세스 정의를 참조하십시오.

dns 키워드를 사용하여 규칙을 바인딩하려면 다음 구문을 사용합니다.

```
dns comparison_operator "host_name_or_domain_name"
```

예 12.28. 특정 호스트의 액세스 정의

client.example.com 호스트에서 **dc=example,dc=com** 항목으로의 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "*") (version 3.0;acl "Deny client.example.com"; deny (all)
 (userdn = "ldap:///anyone") and (dns != "client.example.com");)
```

예 12.29. 특정 도메인에서 액세스 정의

example.com 도메인 내의 모든 호스트에서 **dc=example,dc=com** 항목으로의 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "") (version 3.0;acl "Deny example.com"; deny (all) (userdn =
"ldap:///anyone") and (dns != ".example.com");)
```

12.10.7. 연결에서 특정 수준의 보안 필요

연결 보안은 보안 강화 요인(SSF)에 따라 결정되는데, 이는 운영을 처리하는 데 필요한 최소의 주요 강점을 설정합니다. 바인딩 규칙에 **ssf** 키워드를 사용하여 연결이 일정 수준의 보안을 사용하도록 설정할 수 있습니다. 따라서 암호 변경과 같이 암호화된 연결을 통해 작업을 강제 수행할 수 있습니다.

모든 작업에 대한 SSF의 값은 TLS 연결과 SASL 바인딩 간의 값보다 높습니다. 즉, 서버가 TLS를 통해 실행되도록 구성되어 SASL/GSSAPI에 대해 복제 계약이 구성된 경우 작업의 SSF는 사용 가능한 암호화 유형이 더 안전합니다.

ssf 키워드를 사용한 바인딩 규칙은 다음 구문을 사용합니다.

```
ssf comparison_operator key_strength
```

다음과 같은 비교 연산자를 사용할 수 있습니다.

- = (예:)
- ! (Not equal to)
- < 이하만 해당)
- > (greater than)
- what (less than or equal to)

- >= (greater than 또는 equal to)

key_strength 매개변수가 0 으로 설정된 경우 LDAP 작업에는 보안 작업이 필요하지 않습니다.

예 12.30. 연결에서 특정 수준의 보안 필요

dc=example,dc=com 항목에서 사용자를 구성하려면 SSF가 128 이상인 경우에만 userPassword 속성만 업데이트할 수 있습니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr = "userPassword") (version 3.0;
 acl "Allow users updating own userPassword";
 allow (write) (userdn = "ldap:///self") and (ssf >= "128");)
```

12.10.8. 특정 요일에 액세스 정의

바인딩 규칙의 dayofweek 키워드를 사용하면 요일에 따라 액세스 권한을 부여하거나 거부할 수 있습니다.

참고

Directory Server는 서버의 시간을 사용하여 클라이언트의 시간이 아니라 ACI(액세스 제어 명령)를 평가합니다.

dayofweek 키워드와 함께 규칙을 바인딩하려면 다음 구문을 사용합니다.

```
dayofweek comparison_operator "comma-separated_list_of_days"
```

예 12.31. 특정 요일에 대한 액세스 권한 부여

토요일 및 데네임의 서버에 바인딩하기 위해 uid=user,ou=People,dc=example,dc=com 사용자 항목에 대한 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
```

```
dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (version 3.0; acl "Deny access on Saturdays and Sundays";
 deny (all)
 (userdn = "ldap:///uid=user,ou=People,dc=example,dc=com") and
 (dayofweek = "Sun,Sat");)
```

12.10.9. 특정 시간에 액세스 정의

바인딩 규칙의 **timeofday** 키워드를 사용하면 일 시간에 따라 액세스를 허용하거나 거부할 수 있습니다.

참고

Directory Server는 서버의 시간을 사용하여 클라이언트의 시간이 아닌 **ACI**(액세스 제어 명령)를 평가합니다.

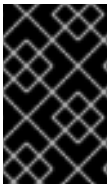
timeofday 키워드를 사용하여 규칙을 바인딩하려면 다음 구문을 사용합니다.

```
timeofday comparison_operator "time"
```

다음과 같은 비교 연산자를 사용할 수 있습니다.

- = (예:)
- ! (Not equal to)
- & lt; 이하만 해당)
- & gt; (greater than)
- what (less than or equal to)

- >= (greater than 또는 equal to)



중요

timeofday 키워드를 사용하려면 시간을 24시간 형식으로 지정해야 합니다.

예 12.32. 특정 시간에 액세스 권한 정의

6pm과 0am 사이에 서버에 바인딩할 uid=user,ou=People,dc=example,dc=com 사용자 항목에 대한 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
 aci: (version 3.0; acl "Deny access between 6pm and 0am";
 deny (all)
 (userdn = "ldap:///uid=user,ou=People,dc=example,dc=com") and
 (timeofday >= "1800" and timeofday < "2400");)
```

12.10.10. 인증 방법을 기반으로 액세스 정의

bind 규칙의 authmethod 키워드는 클라이언트가 서버에 연결할 때 클라이언트가 사용해야 하는 인증 방법을 설정하여 ACL(액세스 제어 명령)을 적용합니다.

authmethod 키워드를 사용하여 규칙을 바인딩하려면 다음 구문을 사용합니다.

```
authmethod comparison_operator "authentication_method"
```

다음 인증 방법을 설정할 수 있습니다.

- None: 인증이 필요하지 않으며 익명 액세스를 나타냅니다. 이는 기본값입니다.
- simple: 클라이언트는 디렉터리에 바인딩할 사용자 이름과 암호를 제공해야 합니다.
-

SSL: 클라이언트는 데이터베이스, 스마트 카드 또는 기타 장치에서 **TLS** 인증서를 사용하여 디렉터리에 바인딩해야 합니다. 인증서 기반 인증에 대한 자세한 내용은 인증 방법에 따라 액세스 정의를 참조하십시오.

- **SASL:** 클라이언트는 **Simple Authentication and Security Layer (ECDHEL)** 연결을 통해 디렉터리에 바인딩해야 합니다. 바인딩 규칙에서 이 인증 방법을 사용할 때 **EXTERNAL** 과 같은 **SASL** 메커니즘을 추가로 지정합니다.

예 12.33. EXTERNAL SASL 인증 방법을 사용하여 연결에 대해서만 액세스 활성화

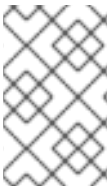
연결에서 인증서 기반 인증 방법 또는 **SASL**을 사용하지 않는 경우 서버에 대한 액세스를 거부하려면 다음을 수행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x`

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (version 3.0; acl "Deny all access without certificate"; deny (all)
 (authmethod = "none" or authmethod = "simple");)
```

12.10.11. 역할을 기반으로 액세스 정의

bind 규칙의 **roledn** 키워드를 사용하면 하나 이상의 역할 세트가 있는 사용자에게 액세스 권한을 부여하거나 거부할 수 있습니다.



참고

Red Hat은 역할 대신 그룹을 사용하는 것이 좋습니다.

roledn 키워드를 사용한 바인딩 규칙은 다음 구문을 사용합니다.

```
roledn comparison_operator "ldap:///distinguished_name || ldap:///distinguished_name || ..."
```

고유 이름(DN)에 쉼표가 포함된 경우 백슬래시로 쉼표를 이스케이프합니다.

예 12.34. 역할을 기반으로 액세스 정의

nsRole 속성에 cn=Human Resources,ou=People,dc=example,dc=com 역할이 설정된 사용자를 활성화하려면 ou=People,dc=example,dc=com 에서 항목의 manager 속성을 검색하고 읽습니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="manager") (version 3.0;
 acl "Allow manager role to update manager attribute";
 allow (search, read) roledn = "ldap:///cn=Human
Resources,ou=People,dc=example,dc=com";)
```

12.10.12. 부울 연산자를 사용하여 바인딩 규칙 결합

복잡한 바인딩 규칙을 만들 때 AND,OR, NOT 부울 연산자를 사용하면 여러 키워드를 결합할 수 있습니다.

부울 연산자와 결합된 바인딩 규칙에는 다음 구문이 있습니다.

```
bind_rule_1 boolean_operator bind_rule_2...
```

예 12.35. 부울 연산자를 사용하여 바인딩 규칙 결합

cn=Administrators,ou=Groups,dc=example,com 및 cn=
Operators,ou=Groups,dc=example,com]의 멤버인 사용자를 구성하려면 group can [command]'read,search,update, and delete entries in ou=People,dc=example,dc=com:

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: ou=People,dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///ou=People,dc=example,dc=com") (version 3.0;
 acl "Allow members of administrators and operators group to manage users";
 allow (read, search, add, write, delete)
 groupdn = "ldap:///cn=Administrators,ou=Groups,dc=example,com" AND
 groupdn = "ldap:///cn=Operators,ou=Groups,dc=example,com";)
```

Directory Server에서 부울 연산자를 평가하는 방법

Directory Server는 다음 규칙을 사용하여 부울 연산자를 평가합니다.

- 왼쪽에서 오른쪽으로 모든 표현입니다.

  다음 예제에서는 **bind_rule_1** 이 먼저 평가됩니다.

  (*bind_rule_1*) OR (*bind_rule_2*)

- 가장 먼저 가장 가까운 접착식에서 가장 외부 표현식까지.

  다음 예제에서 **bind_rule_2** 는 먼저 평가되고 **bind_rule_3** 은 두 번째입니다.

  (*bind_rule_1*) OR ((*bind_rule_2*) AND (*bind_rule_3*))

- **AND** 또는 **OR** 연산자가 아닌 경우

  다음 예제에서 *bind_rule_2* 는 먼저 평가됩니다.

  (*bind_rule_1*) AND NOT (*bind_rule_2*)

  AND 및 **OR** 연산자는 우선 순위가 없습니다.

13장. FIPS 모드에서 DIRECTORY SERVER 실행

Directory Server는 연방 정보 처리 표준 (FIPS)을 완전히 지원합니다. Directory Server를 실행하는 경우 FIPS 모드에서 실행하면 보안 관련 설정이 변경됩니다. 예를 들어 SSL은 자동으로 비활성화되어 TLS 1.2 및 1.3 암호화만 사용됩니다.

13.1. FIPS 모드 활성화

FIPS(Federal Information Processing Standard) 모드에서 Directory Server를 사용하려면 RHEL 및 Directory Server에서 모드를 활성화합니다.

사전 요구 사항

- RHEL에서 FIPS 모드를 활성화했습니다.

절차

1. NSS(네트워크 보안 서비스) 데이터베이스에 FIPS 모드를 활성화합니다.

```
# modutil -dbdir /etc/dirsrv/slapd-instance_name/ -fips true
```

2. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

검증

- NSS 데이터베이스에 FIPS 모드가 활성화되어 있는지 확인합니다.

```
# modutil -dbdir /etc/dirsrv/slapd-instance_name/ -chkfips true
FIPS mode enabled.
```

이 명령은 모듈이 FIPS 모드 인 경우 FIPS 모드를 활성화합니다.

13.2. 추가 리소스

- 연방 정보 처리 표준 (FIPS)
- 시스템을 FIPS 모드로 전환

14장. 암호 기반 계정 잠금 정책 구성

암호 기반 계정 잠금 정책은 공격자가 사용자의 암호를 반복적으로 추측하지 못하도록 합니다. 지정된 횟수의 바인딩 실패 후 사용자 계정을 잠그도록 계정 잠금 정책을 구성할 수 있습니다.

암호 기반 계정 잠금 정책이 구성된 경우 **Directory Server**는 사용자 항목의 다음 속성에 잠금 정보를 유지합니다.

- **passwordRetryCount** 실패한 바인딩 시도 수를 저장합니다. **directory Server**는 사용자가 **retryCountResetTime** 의 시간보다 나중에 디렉터리에 바인딩되는 경우 값을 재설정합니다. 이 속성은 사용자가 처음으로 바인딩하지 못하면 존재합니다.

- **retryCountResetTime: passwordRetryCount** 속성이 재설정된 시간을 저장합니다. 이 속성은 사용자가 처음으로 바인딩하지 못하면 존재합니다.

- **AccountUnlockTime:** 사용자 계정이 잠금 해제된 시간을 저장합니다. 이 속성은 계정이 처음 잠긴 후 존재합니다.

14.1. 구성된 최대 시도에 도달하거나 초과할 때 계정을 잠글지 여부 구성

관리자는 **Directory Server**가 실패한 로그인 시도 시 계정을 잠글 때 다음 동작 중 하나를 구성할 수 있습니다.

- 제한이 초과된 경우 서버 잠금 계정입니다. 예를 들어 제한이 **3**번 시도로 설정되면 네 번째 실패 시도**(n+1)** 후에 잠금이 발생합니다. 또한 네 번째 시도에 성공하면 **Directory Server**가 계정을 잠그지 않습니다.

  기본적으로 **Directory Server**는 기존 **LDAP** 클라이언트에서 자주 사용하는 이 레거시 암호 정책을 사용합니다.

- 제한에 도달한 경우 서버 잠금 계정입니다. 예를 들어 제한이 **3**번으로 설정되면 세 번째 시도에 실패한 후 서버가 계정을 잠급니다**(n)**.

  최신 **LDAP** 클라이언트는 종종 이러한 동작을 예상합니다.

다음 절차에서는 레거시 암호 정책을 비활성화하는 방법을 설명합니다. 정책을 변경한 후 **Directory**

Server는 구성된 제한에 도달한 사용자에 대한 로그인 시도를 차단합니다.

사전 요구 사항

- 계정 잠금 정책을 구성하셨습니다.

절차

- 레거시 암호 정책을 비활성화하고 계정에 도달한 경우 계정을 잠급니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
passwordLegacyPolicy=off
```

검증

1. passwordmaxfailure 설정 값을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy get
passwordmaxfailure
passwordmaxfailure: 2
```

2. 유효하지 않은 암호를 사용하여 바인딩하려고 하면 **passwordmaxfailure** 에 설정된 값보다 한 번 더 늘어납니다.

```
# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Constraint violation (19)
 additional info: Exceed password retry limit. Please try later.
```

기존 암호가 비활성화된 상태에서 **Directory Server**는 두 번째 시도 후 계정을 잠겼고 **ldap_bind: Constraint violation-19)** 오류로 인해 추가 시도가 차단됩니다.

추가 리소스

- 명령줄을 사용하여 암호 기반 계정 잠금 정책 구성

14.2. 명령줄을 사용하여 암호 기반 계정 잠금 정책 구성

잘못된 암호로 로그인 반복된 바인드 시도를 차단하려면 암호 기반 계정 잠금 정책을 구성합니다.

중요

Directory Server에서 구성된 최대 시도 횟수를 초과할 때 계정을 잠그는지의 동작은 레거시 암호 정책 설정에 따라 달라집니다.

절차

1. 선택 사항: 레거시 암호 정책이 활성화되었는지 여부를 식별합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config get
passwordLegacyPolicy
passwordLegacyPolicy: on
```

2. 암호 잠금 정책을 활성화하고 최대 실패 수를 2 로 설정합니다.

```
# [command]`dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy
set --pwdlockout on --pwdmaxfailures=2
```

레거시 암호 정책을 활성화하면 Directory Server는 바인딩 시도에 실패한 세 번째 시도 후 계정을 잠급니다(-pwdmaxfailures 매개변수 + 1).

dsconf pwpolicy set 명령은 다음 매개변수를 지원합니다.

- --pwdlockout: 계정 잠금 기능을 활성화하거나 비활성화합니다. 기본값: off.
- --pwdmaxfailures: Directory Server가 계정을 잠그기 전에 허용되는 최대 바인딩 시도 수를 설정합니다. 기본값: 3.

레거시 암호 정책 설정이 활성화된 경우 이 잠금은 나중에 한 번 시도합니다. 기본값: 3.

- **--pwdresetfailcount: Directory Server**가 사용자 항목에서 **passwordRetryCount** 속성을 재설정하기 전의 시간(초)을 설정합니다. 기본값: 600 초(10분).

- **--pwdlockoutduration:** 계정 잠금 시간을 초 단위로 설정합니다. **--pwdunlock** 매개변수를 **off** 로 설정하면 이 매개변수가 무시됩니다. 기본값: 3600 초(1시간).

- **--pwdunlock:** 일정 시간 후에 잠긴 계정의 잠금 해제 여부를 설정하거나 관리자가 수동으로 잠금 해제할 때까지 비활성화 상태를 유지합니다. 기본값: 설정.

검증

- **--pwdmaxfailures** 매개변수에서 설정한 값보다 잘못된 암호를 두 번 사용하여 바인딩하려고 합니다.

```
# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Constraint violation (19)
        additional info: Exceed password retry limit. Please try later.
```

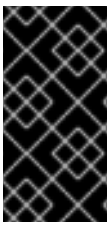
기존 암호가 활성화되면 **Directory Server**에서 제한을 초과한 후 계정을 잠겼고 **ldap_bind: Constraint violation-19)** 오류로 인해 추가 시도가 차단됩니다.

추가 리소스

- 레거시 암호 정책 구성

14.3. 웹 콘솔을 사용하여 암호 기반 계정 잠금 정책 구성

잘못된 암호로 로그인 반복된 바인드 시도를 차단하려면 암호 기반 계정 잠금 정책을 구성합니다.



중요

Directory Server에서 구성된 최대 시도 횟수를 초과할 때 계정을 잠그는지의 동작은 레거시 암호 정책 설정에 따라 달라집니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. 선택 사항: 레거시 암호 정책이 활성화되었는지 여부를 식별합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config get
passwordLegacyPolicy
passwordLegacyPolicy: on
```

이 설정은 웹 콘솔에서 사용할 수 없습니다.

2. 데이터베이스 암호 정책 글로벌 정책 계정 잠금 으로 이동합니다.

3. **Enable Account Lockout** 을 선택합니다.

4. 잠금 설정을 구성합니다.

- 계정 잠금: **Directory Server**가 계정을 잠그기 전에 허용되는 최대 바인딩 시도 횟수를 설정합니다.

-

Time Until Failure Count Resets: Directory Server가 사용자 항목에서 **passwordRetryCount** 속성을 재설정하기 전의 시간(초)을 설정합니다.

- **Time Until Account Unlocked:** 계정 잠금 시간을 초 단위로 설정합니다. **Do Not Lockout Account Forever** 를 비활성화하면 이 매개변수가 무시됩니다.

- **Do Not Lockout Account Forever:** 일정 시간 후에 잠긴 계정의 잠금을 해제할지 또는 관리자가 수동으로 잠금 해제할 때까지 잠긴 계정의 잠금을 해제해야 하는지 여부를 활성화하거나 비활성화합니다.

5. 저장을 클릭합니다.

검증

- 유효하지 않은 암호를 사용하여 바인딩하려고 하면 실패한 로그인 수에 설정한 값보다 두 번 이상 바인딩하십시오.

```
# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)

# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w invalid-password -b
"dc=example,dc=com" -x
ldap_bind: Constraint violation (19)
        additional info: Exceed password retry limit. Please try later.
```

기존 암호가 활성화되면 **Directory Server**에서 제한을 초과한 후 계정을 잠겼고 **ldap_bind: Constraint violation-19)** 오류로 인해 추가 시도가 차단됩니다.

추가 리소스

- 레거시 암호 정책 구성

15장. 익명 바인딩 비활성화

사용자가 자격 증명을 제공하지 않고 **Directory Server**에 연결을 시도하는 경우 이 작업을 **anonymous bind** 라고 합니다. 익명 바인딩은 사용자가 먼저 인증하지 않고도 디렉터리에서 전화 번호를 찾는 등 검색 및 읽기 작업을 단순화합니다. 그러나 계정이 없는 사용자가 데이터에 액세스할 수 있기 때문에 익명 바인딩도 보안 위험이 될 수 있습니다.

주의

기본적으로 검색 및 읽기 작업을 위해 **Directory Server**에서 익명 바인딩이 활성화됩니다. 이를 통해 루트 디렉터리 서버 항목(DSE)과 같은 구성 항목뿐만 아니라 사용자 항목에 무단으로 액세스할 수 있습니다.

15.1. 명령줄을 사용하여 익명 바인딩 비활성화

보안을 강화하기 위해 익명 바인딩을 비활성화할 수 있습니다.

절차

- **nsslapd-allow-anonymous-access** 구성 매개변수를 **off** 로 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-allow-anonymous-access=off
```

검증

- 사용자 계정을 지정하지 않고 검색을 실행합니다.

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -x
ldap_bind: Inappropriate authentication (48)
 additional info: Anonymous access is not allowed
```

15.2. 웹 콘솔을 사용하여 익명 바인딩 비활성화

보안을 강화하기 위해 익명 바인딩을 비활성화할 수 있습니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. 서버 설정 고급 설정으로 이동합니다.
2. **Allow Anonymous Access** 매개변수를 **off** 로 설정합니다.
3. 저장을 클릭합니다.

검증

- 사용자 계정을 지정하지 않고 검색을 실행합니다.

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -x
ldap_bind: Inappropriate authentication (48)
        additional info: Anonymous access is not allowed
```

16장. 복제 환경의 모든 서버에서 계정 잠금 동기화

Directory Server는 계정 잠금 속성을 로컬로 저장합니다. 여러 서버가 있는 환경에서는 공격자가 계정 잠금 수에 도달한 다음 다른 서버에서 계속될 때까지 공격자가 하나의 서버에 로그인하지 못하도록 이러한 속성에 대한 복제를 구성합니다.

16.1. DIRECTORY SERVER에서 복제 환경에서 암호 및 계정 잠금 정책을 처리하는 방법

Directory Server에서는 다음과 같이 암호 및 계정 잠금 정책을 적용합니다.

- 데이터 공급자에 암호 정책이 적용됩니다.
- 계정 잠금 정책은 복제 토폴로지의 모든 서버에 적용됩니다.

Directory Server는 다음 암호 정책 특성을 복제합니다.

- passwordMinAge
- passwordMaxAge
- passwordExp
- passwordWarning

그러나 기본적으로 Directory Server는 일반 계정 잠금 속성을 복제하지 않습니다.

- passwordRetryCount
- retryCountResetTime

- **accountUnlockTime**

공격자가 계정 잠금 수에 도달한 다음 다른 서버에서 계속될 때까지 공격자가 하나의 서버에 로그인하지 못하도록 하려면 이러한 계정 잠금 속성을 복제합니다.

추가 리소스

- 계정 잠금 속성을 복제하도록 Directory Server 구성

16.2. 계정 잠금 속성을 복제하도록 DIRECTORY SERVER 구성

passwordRetryCount,retryCountResetTime 또는 **accountUnlockTime** 특성을 업데이트하는 계정 잠금 정책 또는 암호 정책을 사용하는 경우 모든 서버에서 해당 값이 동일하게 되도록 **Directory Server**를 구성합니다.

복제 토폴로지의 모든 공급업체에서 이 절차를 수행합니다.

사전 요구 사항

- 언급된 속성 중 하나 이상을 업데이트하는 계정 잠금 정책 또는 암호 정책을 구성하셨습니다.
- 복제 환경에서 **Directory Server**를 사용합니다.

절차

1. 암호 정책 특성 복제를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdisglobal="on"
```

2. 부분 복제를 사용하는 경우 복제에서 제외된 특성 목록을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com repl-agmt get --suffix
"dc=example,dc=com" example-agreement | grep "nsDS5ReplicatedAttributeList"
```

기본 설정을 사용하여 출력이 표시되지 않으며 Directory Server는 계정 잠금 속성을 복제합니다. 그러나 명령에서 다음 예와 같이 제외된 속성 목록을 반환하는 경우 특성 목록을 확인합니다.

```
nsDS5ReplicatedAttributeList: (objectclass=*) $ EXCLUDE accountUnlockTime
passwordRetryCount retryCountResetTime example1 example2
```

이 예에서 계정UnlockTime,passwordRetryCount, retryCountResetTime 잠금 정책 속성은 복제에서 제외되며 두 개의 다른 속성과 함께 합니다.

3. 이전 명령의 출력에 계정 잠금 속성이 나열된 경우 부분 복제 설정을 잠금 정책 특성 이외의 속성만 포함하도록 업데이트합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com repl-agmt set --suffix
"dc=example,dc=com" --frac-list "example1 example2" example-agreement
```

검증

1. 잘못된 암호를 사용하여 사용자로 검색을 수행합니다.

```
# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w "invalid-password" -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)
```

2. 사용자의 passwordRetryCount 특성을 표시합니다.

```
# ldapsearch -H ldap://server.example.com -D "cn=Directory Manager" -W -b
"uid=example,ou=People,dc=example,dc=com" -x passwordRetryCount
...
dn: uid=example,ou=People,dc=example,dc=com
passwordRetryCount: 1
```

3. 복제 토폴로지의 다른 서버에서 이전 명령을 실행합니다. passwordRetryCount 특성 값이 동일하면 Directory Server가 해당 특성을 복제합니다.

추가 리소스

- 암호 기반 계정 잠금 정책 구성

17장. 시간 기반 계정 잠금 정책 구성

계정 정책 플러그인을 사용하여 다음과 같은 다양한 시간 기반 잠금 정책을 구성할 수 있습니다.

- 마지막으로 성공적으로 로그인할 때 일정 시간 동안 계정을 자동으로 비활성화
- 계정을 생성한 후 일정 시간 자동 비활성화
- 암호 만료 후 일정 시간 자동 비활성화
- 계정 비활성 및 암호 만료 모두에서 계정 자동 비활성화

17.1. 마지막으로 성공적으로 로그인할 때 일정 시간 동안 계정을 자동으로 비활성화

다음 절차에 따라 **21**일 이상 로그인하지 않는 **dc=example,dc=com** 항목에서 사용자를 비활성화하는 시간 기반 잠금 정책을 구성합니다.

예를 들어 회사에서 계정을 삭제하고 관리자가 계정을 삭제하는 것을 잊어버리는 경우 **Directory Server**는 일정 시간 후에 계정을 비활성화하는 것을 확인하는 계정 비활성 기능입니다.

절차

1. 계정 정책 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy enable
```

2. 플러그인 구성 항목을 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy config-entry set "cn=config,cn=Account Policy Plugin,cn=plugins,cn=config" --always-record-login yes --state-attr lastLoginTime --alt-state-attr 1.1 --spec-attr acctPolicySubentry --limit-attr accountInactivityLimit
```

이 명령은 다음 옵션을 사용합니다.

- **--always- history-login 예:** 로그인 시간을 로깅할 수 있습니다. 이는 **acctPolicySubentry** 특성이 설정되지 않은 경우에도 계정 정책과 함께 클래스 서비스(CoS) 또는 역할을 사용하는 데 필요합니다.

- **--state-attr lastLoginTime:** 계정 정책 플러그인이 사용자의 **lastLoginTime** 속성에 마지막 로그인 시간을 저장하도록 구성합니다.

- **--Alt-state-attr 1.1:** 대체 속성을 사용하여 기본 속성이 존재하지 않는지 확인합니다. 기본적으로 **Directory Server**는 **createTimestamp** 특성을 대안으로 사용합니다. 그러나 이로 인해 **Directory Server**에서 계정에 **lastLoginTime** 특성이 설정되지 않고 **createTimestamp**가 구성된 비활성 기간보다 오래된 경우 **Directory Server**가 기존 사용자를 자동으로 로그아웃합니다. 대체 특성을 비활성화하면 **Directory Server**에서 다음에 로그인할 때 **lastLoginTime** 속성이 사용자 항목에 자동으로 추가됩니다.

- **--spec-attr acctPolicySubentry: Directory Server**를 구성하여 **acctPolicySubentry** 속성이 설정된 항목에 정책을 적용합니다. **CoS** 항목에서 이 속성을 구성합니다.

- **--limit-attr accountInactivityLimit:** 계정의 **accountInactivityLimit** 속성에서 비활성 시간을 저장하도록 구성합니다.

3. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

4. 계정 비활성화 정책 항목을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=Account Inactivation Policy,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: accountpolicy
accountInactivityLimit: 1814400
cn: Account Inactivation Policy
```

accountInactivityLimit 속성의 값은 마지막 로그인 후 **Directory Server**가 계정 **1814400** 초

(21일)를 비활성화합니다.

5. **CoS 템플릿 항목을 생성합니다.**

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=TemplateCoS,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: cosTemplate
acctPolicySubentry: cn=Account Inactivation Policy,dc=example,dc=com
```

이 템플릿 항목은 계정 비활성화 정책을 참조합니다.

6. **CoS 정의 항목을 생성합니다.**

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=DefinitionCoS,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=TemplateCoS,dc=example,dc=com
cosAttribute: acctPolicySubentry default operational-default
```

이 정의 항목은 CoS 템플릿 항목을 참조하고 acctPolicySubentry 속성이 cn=Account Inactivation Policy,dc=example,dc=com 으로 설정된 값을 사용하여 각 사용자 항목에 표시되도록 합니다.

검증

1. **사용자의 lastLoginTime 속성을 구성한 비활성 시간보다 오래된 값으로 설정합니다.**

```
# ldapmodify -H ldap://server.example.com -x -D "cn=Directory Manager" -W

dn: uid=example,ou=People,dc=example,dc=com
changetype: modify
replace: lastLoginTime
lastLoginTime: 20210101000000Z
```

2. 다음 사용자로 디렉터리에 연결을 시도합니다.

```
# ldapsearch -H ldap://server.example.com -x -D
"uid=example,ou=People,dc=example,dc=com" -W -b "dc=example,dc=com"
ldap_bind: Constraint violation (19)
 additional info: Account inactivity limit exceeded. Contact system administrator to
reset.
```

Directory Server에서 액세스를 거부하고 이 오류를 반환하는 경우 계정 비활성이 작동합니다.

추가 리소스

- 비활성 제한에 도달한 계정 재활성화

17.2. 계정을 생성한 후 일정 시간 자동 비활성화

다음 절차에 따라 dc=example,dc=com 항목에서 해당 계정을 생성한 후 60일이 지나면 만료되도록 구성합니다.

예를 들어 계정 만료 기능을 사용하여 외부 작업자 계정을 생성한 후 일정 시간 동안 잠급니다.

절차

1. 계정 정책 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy
enable
```

2. 플러그인 구성 항목을 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy
config-entry set "cn=config,cn=Account Policy Plugin,cn=plugins,cn=config" --always-
record-login yes --state-attr createTimestamp --alt-state-attr 1.1 --spec-attr
acctPolicySubentry --limit-attr accountInactivityLimit
```

이 명령은 다음 옵션을 사용합니다.

- **--always- history-login 예:** 로그인 시간을 로깅할 수 있습니다. 이는 **acctPolicySubentry** 특성이 설정되지 않은 경우에도 계정 정책과 함께 클래스 서비스(CoS) 또는 역할을 사용하는 데 필요합니다.

- **--state-attr createTimestamp:** 계정 정책 플러그인이 **createTimestamp** 속성 값을 사용하여 계정이 만료되었는지 여부를 계산합니다.

- **--Alt-state-attr 1.1:** 대체 속성을 사용하여 기본 속성이 존재하지 않는지 확인합니다.

- **--spec-attr acctPolicySubentry: Directory Server**를 구성하여 **acctPolicySubentry** 속성이 설정된 항목에 정책을 적용합니다. **CoS** 항목에서 이 속성을 구성합니다.

- **--limit-attr accountInactivityLimit:** 계정 만료 정책 항목의 **accountInactivityLimit** 속성에서 최대 기간을 저장하는 것을 구성합니다.

3. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

4. 계정 만료 정책 항목을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=Account Expiration Policy,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: accountpolicy
accountInactivityLimit: 5184000
cn: Account Expiration Policy
```

accountInactivityLimit 속성의 값은 계정이 생성된 후 **5184000** 초(60일) 만료되도록 구성합니다.

5. **CoS** 템플릿 항목을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x
```

```
dn: cn=TemplateCoS,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: cosTemplate
acctPolicySubentry: cn=Account Expiration Policy,dc=example,dc=com
```

이 템플릿 항목은 계정 만료 정책을 참조합니다.

6. CoS 정의 항목을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=DefinitionCoS,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=TemplateCoS,dc=example,dc=com
cosAttribute: acctPolicySubentry default operational-default
```

이 정의 항목은 CoS 템플릿 항목을 참조하고 acctPolicySubentry 속성이 cn=Account Expiration Policy,dc=example,dc=com 으로 설정된 값을 사용하여 각 사용자 항목에 표시되도록 합니다.

검증

- 60일 전에 createTimestamp 속성이 설정된 dc=example,dc=com 항목에 저장된 사용자로 디렉터리에 연결을 시도합니다.

```
# ldapsearch -H ldap://server.example.com -x -D "uid=example,dc=example,dc=com" -
W -b "dc=example,dc=com"
ldap_bind: Constraint violation (19)
 additional info: Account inactivity limit exceeded. Contact system administrator to
reset.
```

Directory Server에서 액세스를 거부하고 이 오류를 반환하는 경우 계정 만료가 작동합니다.

추가 리소스

- 비활성 제한에 도달한 계정 재활성화

17.3. 암호 만료 후 일정 시간 자동 비활성화

28일 이상 암호를 변경하지 않는 **dc=example,dc=com** 항목에서 사용자를 비활성화하는 시간 기반 잠금 정책을 구성하려면 다음 절차를 따르십시오.

사전 요구 사항

- 사용자는 해당 항목에 **passwordExpirationTime** 속성이 설정되어 있어야 합니다.

절차

1. 암호 만료 기능을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
passwordExp=on
```

2. 계정 정책 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy
enable
```

3. 플러그인 구성 항목을 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy
config-entry set "cn=config,cn=Account Policy Plugin,cn=plugins,cn=config" --always-
record-login yes --always-record-login-attr lastLoginTime --state-attr
non_existent_attribute --alt-state-attr passwordExpirationTime --spec-attr
acctPolicySubentry --limit-attr accountInactivityLimit
```

이 명령은 다음 옵션을 사용합니다.

- **--always- history-login** 예**:** 로그인 시간을 로깅할 수 있습니다. 이는 **acctPolicySubentry** 특성이 설정되지 않은 경우에도 계정 정책과 함께 클래스 서비스**(CoS)** 또는 역할을 사용하는 데 필요합니다.
- **--always- records-login-attr lastLoginTime:** 계정 정책 플러그인이 사용자의 **lastLoginTime** 속성에 마지막 로그인 시간을 저장하도록 구성합니다.

- **--state-attr *non_existent_attribute*: 계정 정책을 존재하지 않는 dummy 특성 이름으로 평가하는 데 사용되는 기본 시간 특성을 설정합니다.**

- **--Alt-state-attr 'passwordExpirationTime: 확인할 대체 속성으로 passwordExpirationTime 특성을 사용하도록 플러그인을 구성합니다.**

- **--spec-attr acctPolicySubentry: Directory Server를 구성하여 acctPolicySubentry 속성이 설정된 항목에 정책을 적용합니다. CoS 항목에서 이 속성을 구성합니다.**

- **--limit-attr accountInactivityLimit: 계정 정책 항목의 accountInactivityLimit 속성이 마지막 암호 변경 후 계정이 비활성화된 시간을 저장하도록 구성합니다.**

4. **인스턴스를 다시 시작합니다.**

```
# dsctl instance_name restart
```

5. **계정 비활성화 정책 항목을 생성합니다.**

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=Account Inactivation Policy,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: accountpolicy
accountInactivityLimit: 2419200
cn: Account Inactivation Policy
```

accountInactivityLimit 속성의 값은 암호가 변경된 후 Directory Server가 계정 2419200 초(28일)를 비활성화합니다.

6. **CoS 템플릿 항목을 생성합니다.**

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=TemplateCoS,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
```

```
objectClass: extensibleObject
objectClass: cosTemplate
acctPolicySubentry: cn=Account Inactivation Policy,dc=example,dc=com
```

이 템플릿 항목은 계정 비활성화 정책을 참조합니다.

7. CoS 정의 항목을 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=DefinitionCoS,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=TemplateCoS,dc=example,dc=com
cosAttribute: acctPolicySubentry default operational-default
```

이 정의 항목은 CoS 템플릿 항목을 참조하고 acctPolicySubentry 속성이 cn=Account Inactivation Policy,dc=example,dc=com 으로 설정된 값을 사용하여 각 사용자 항목에 표시되도록 합니다.

검증

1. 사용자의 passwordExpirationTime 속성을 구성한 비활성 시간보다 오래된 값으로 설정합니다.

```
# ldapmodify -H ldap://server.example.com -x -D "cn=Directory Manager" -W

dn: uid=example,ou=People,dc=example,dc=com
changetype: modify
replace: passwordExpirationTime
passwordExpirationTime: 20210101000000Z
```

2. 다음 사용자로 디렉터리에 연결을 시도합니다.

```
# ldapsearch -H ldap://server.example.com -x -D
"uid=example,ou=People,dc=example,dc=com" -W -b "dc=example,dc=com"
ldap_bind: Constraint violation (19)
 additional info: Account inactivity limit exceeded. Contact system administrator to
reset.
```

Directory Server에서 액세스를 거부하고 이 오류를 반환하는 경우 계정 비활성이 작동합니다.

추가 리소스

- 비활성 제한에 도달한 계정 재활성화

17.4. 계정 비활성 및 암호 만료 모두에서 계정 자동 비활성화

checkAllStateAttrs 설정을 사용하여 사용자가 인증할 때 계정 비활성 및 암호 만료를 모두 적용할 수 있습니다. 기본적으로 **checkAllStateAttrs** 가 플러그인 구성 항목에 없거나 이 매개변수를 **no** 로 설정하면 플러그인에서 **state** 특성 **lastLoginTime** 을 확인합니다. 항목에 속성이 없으면 플러그인은 대체 상태 특성을 확인합니다.

기본 **state** 속성을 존재하지 않는 속성으로 설정하고, 플러그인에서 **passwordExpirationtime** 속성을 기반으로 만료를 처리하려는 경우 대체 상태 속성을 **passwordExpirationtime** 으로 설정할 수 있습니다. 이 매개변수를 활성화하면 기본 상태 특성이 확인되고 계정이 정상이면 대체 상태 특성을 확인합니다.

이는 **passwordExpirationtime**이 비활성 제한을 초과하는 경우 계정 정책 플러그인이 계정을 완전히 비활성화한다는 점에서 암호 정책의 암호 만료와 다릅니다. 암호 정책 만료를 사용하는 동안 사용자는 계속 로그인하여 암호를 변경할 수 있습니다. 계정 정책 플러그인은 사용자가 아무 작업도 수행하지 못하도록 완전히 차단하고 관리자가 계정을 재설정해야 합니다.

절차

1. 플러그인 구성 항목을 생성하고 설정을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy config-entry set "cn=config,cn=Account Policy Plugin,cn=plugins,cn=config" --always-record-login yes --state-attr lastLoginTime --alt-state-attr 1.1 --spec-attr acctPolicySubentry --limit-attr accountInactivityLimit --check-all-state-attrs yes
```

2. 서버를 다시 시작하여 새 플러그인 구성을 로드합니다.

```
# dsctl instance_name restart
```

주의

checkAllStateAttrs 설정은 대체 상태 속성이 **passwordExpiratontime** 으로 설정된 경우에만 작동하도록 설계되었습니다. **createTimestamp** 로 설정하면 바람직하지 않은 결과가 발생할 수 있으며 항목이 잠길 수 있습니다.

18장. 비활성 제한에 도달한 계정 재활성화

Directory Server가 비활성 제한에 도달했기 때문에 계정을 비활성화한 경우 관리자가 계정을 다시 활성화할 수 있습니다.

18.1. 계정 정책 플러그인에 의해 비활성화된 계정을 다시 활성화

dsconf 계정 잠금 해제 명령을 사용하거나 비활성화된 사용자의 lastLoginTime 속성을 수동으로 업데이트하여 계정을 다시 활성화할 수 있습니다.

사전 요구 사항

- 비활성화된 사용자 계정.

절차

- 다음 방법 중 하나를 사용하여 계정을 다시 활성화합니다.
  - dsconf 계정 unlock 명령 사용:

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" account unlock
"uid=example,ou=People,dc=example,dc=com"
```

  - 사용자의 lastLoginTime 속성을 최근 타임스탬프로 설정하여 다음을 수행합니다.

```
# ldapmodify -H ldap://server.example.com -x -D "cn=Directory Manager" -W

dn: uid=example,ou=People,dc=example,dc=com
changetype: modify
replace: lastLoginTime
lastLoginTime: 20210901000000Z
```

검증

- 다시 활성화한 사용자로 인증합니다. 예를 들어 검색을 수행합니다.

```
# ldapsearch -H ldap://server.example.com -x -D
"uid=example,ou=People,dc=example,dc=com" -W -b "dc=example,dc=com -s base"
```

사용자가 인증할 수 있는 경우 계정이 다시 활성화됩니다.

19장. 잠금 정책을 설정하지 않고 마지막 로그인 시간 추적

계정 정책 플러그인을 사용하여 만료 시간 또는 비활성 기간을 설정하지 않고도 사용자 로그인 시간을 추적할 수 있습니다. 이 경우 플러그인은 **lastLoginTime** 속성을 사용자 항목에 추가합니다.

19.1. 마지막 로그인 시간을 기록하도록 계정 정책 플러그인 구성

다음 절차에 따라 사용자 항목의 **lastLoginTime** 속성에 사용자의 마지막 로그인 시간을 기록합니다.

절차

1. 계정 정책 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy enable
```

2. 플러그인 구성 항목을 생성하여 로그인 시간을 기록합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin account-policy config-entry set "cn=config,cn=Account Policy Plugin,cn=plugins,cn=config" --always-record-login yes --state-attr lastLoginTime
```

이 명령은 다음 옵션을 사용합니다.

- **--always- history-login 예:** 로그인 로깅을 사용할 수 있습니다.
- **--state-attr lastLoginTime:** 계정 정책 플러그인이 사용자의 **lastLoginTime** 속성에 마지막 로그인 시간을 저장하도록 구성합니다.

3. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

검증

1. **Directory Server에 사용자로 로그인합니다. 예를 들어 검색을 실행합니다.**

```
# ldapsearch -H ldap://server.example.com -x -D
"uid=example,ou=People,dc=example,dc=com" -W -b "dc=example,dc=com"
```

2. **이전 단계에서 사용한 사용자의 lastLoginTime 속성을 표시합니다.**

```
# ldapsearch -H ldap://server.example.com -x -D "cn=Directory Manager" -W -b
"uid=example,ou=people,dc=example,dc=com" lastLoginTime
...
dn: uid=example,ou=People,dc=example,dc=com
lastLoginTime: 20210913091435Z
```

lastLoginTime 특성이 존재하고 Directory Server가 해당 값을 업데이트한 경우 마지막 로그인 시간 기록이 작동합니다.

20장. DIRECTORY MANAGER 계정에서 액세스 제어 설정

제한되지 않은 관리 사용자는 유지 관리 관점에서 의미가 있습니다. 디렉터리 관리자는 유지 관리 작업을 수행하고 문제에 대응하기 위해 높은 수준의 액세스가 필요합니다.

그러나 **Directory Manager** 사용자의 성능으로 인해 특정 수준의 액세스 제어가 관리자 권한으로 수행되는 공격을 방지하기 위해 권장될 수 있습니다.

20.1. DIRECTORY MANAGER 계정의 액세스 제어 정보

Directory Server는 디렉터리 트리에만 일반 액세스 제어 명령을 적용합니다. **Directory Manager** 계정의 권한은 하드 코딩되며 이 계정을 바인딩 규칙에서 사용할 수 없습니다. **Directory Manager** 계정에 대한 액세스를 제한하려면 **RootDN** 액세스 제어 플러그인을 사용합니다.

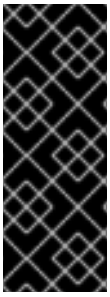
이 플러그인의 기능은 표준 **ACL**(액세스 제어 명령)과 다릅니다. 예를 들어 대상(**Directory Manager** 항목) 및 허용된 권한(모든 권한)과 같은 특정 정보가 적용되지 않습니다. **RootDN** 액세스 제어 플러그인의 용도는 이 사용자가 수행할 수 있는 작업을 제한하지 않고 위치 또는 시간에 따라 **Directory Manager**로 로그인할 수 있는 사람을 제한하여 보안 수준을 제공하는 것입니다.

이러한 이유로 플러그인 설정은 다음만 지원합니다.

- 특정 날짜 및 특정 시간 범위에서 액세스를 허용하거나 거부할 수 있는 시간 기반 액세스 제어
- 정의된 **IP** 주소, 서브넷 및 도메인에서 액세스를 허용하거나 거부하는 **IP** 주소 규칙
- 호스트 액세스 규칙: 특정 호스트, 도메인 및 하위 도메인에서 액세스를 허용하거나 거부합니다.

Directory Manager에 대해 설정할 수 있는 액세스 제어 규칙은 하나만 있습니다. 이는 플러그인 항목에 있으며 전체 디렉터리에 적용됩니다.

일반 **ACI**와 마찬가지로 거부 규칙은 허용 규칙보다 우선 순위가 높습니다.



중요

Directory Manager 계정에 적절한 수준의 액세스 권한이 있는지 확인합니다. 이 관리자는 **24**시간 내에 유지보수 작업을 수행하거나 장애에 대응해야 할 수 있습니다. 이 경우 너무 제한적인 시간 또는 일 규칙을 설정하면 **Directory Manager** 사용자가 디렉터리를 적절하게 관리하지 못할 수 있습니다.

20.2. 명령줄을 사용하여 ROOTDN 액세스 제어 플러그인 구성

기본적으로 **RootDN** 액세스 제어 플러그인이 비활성화됩니다. **Directory Manager** 계정의 권한을 제한하려면 플러그인을 활성화하고 구성합니다.

절차

1. **RootDN** 액세스 제어 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin root-dn enable
```

2. 바인딩 규칙을 설정합니다. 예를 들어 **Directory Manager** 계정이 **IP** 주소 **192.0.2.1** 을 사용하여 호스트에서 오후 **6**시부터 오후 **9**시 사이에만 로그인할 수 있도록 하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin root-dn set --
open-time=0600 --close-time=2100 --allow-ip="192.0.2.1"
```

설정할 수 있는 전체 매개변수 목록은 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin root-dn set --
help
```

3. 인스턴스를 다시 시작합니다.

```
# dsctl instance_name restart
```

검증

- 허용되는 시간 범위를 벗어나거나 허용되지 않는 호스트에서 **cn=Directory Manager** 로 쿼리를 수행합니다.

```
[user@192.0.2.2]$ ldapsearch -D "cn=Directory Manager" -W -H
```

```
ldap://server.example.com -x -b "dc=example,dc=com"
Enter LDAP Password:
ldap_bind: Server is unwilling to perform (53)
 additional info: RootDN access control violation
```

Directory Server에서 액세스를 거부하면 플러그인이 예상대로 작동합니다.

20.3. 웹 콘솔을 사용하여 ROOTDN 액세스 제어 플러그인 구성

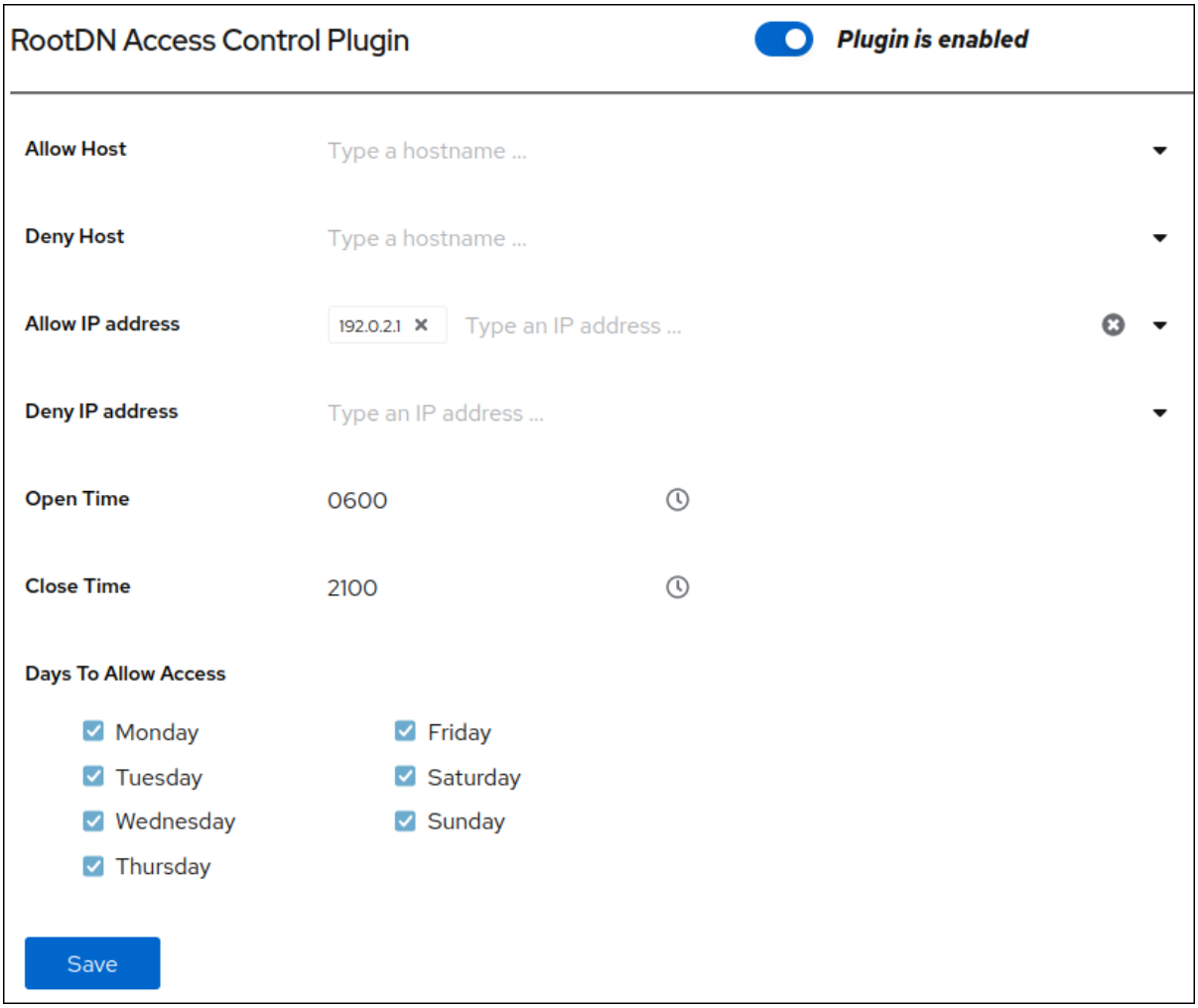
기본적으로 RootDN 액세스 제어 플러그인이 비활성화됩니다. Directory Manager 계정의 권한을 제한하려면 플러그인을 활성화하고 구성합니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. Plugins → RootDN 액세스 제어 로 이동합니다.
2. 플러그인을 활성화합니다.
3. 요구 사항에 따라 필드를 작성합니다.



4. 저장을 클릭합니다.

5. 오른쪽 상단에 있는 작업을 클릭하고 인스턴스 재시작 을 선택합니다.

검증

- 허용되는 시간 범위를 벗어나거나 허용되지 않는 호스트에서 **cn=Directory Manager** 로 쿼리를 수행합니다.

```
[user@192.0.2.2]$ ldapsearch -D "cn=Directory Manager" -W -H
ldap://server.example.com -x -b "dc=example,dc=com"
Enter LDAP Password:
ldap_bind: Server is unwilling to perform (53)
        additional info: RootDN access control violation
```

Directory Server에서 액세스를 거부하면 플러그인이 예상대로 작동합니다.

21장. 특성 암호화 관리

Directory Server는 디렉터리에서 중요한 데이터에 대한 액세스를 보호하는 여러 메커니즘을 제공합니다. 그러나 기본적으로 서버는 데이터베이스에 암호화되지 않은 데이터를 저장합니다. 매우 민감한 정보의 경우 공격자가 데이터베이스에 액세스할 수 있는 잠재적 위험이 심각한 위험이 될 수 있습니다.

관리자는 특성 암호화 기능을 사용하여 데이터베이스에서 암호화된 정부 식별 번호와 같은 중요한 데이터로 특정 특성을 저장할 수 있습니다. 접미사로 활성화하면 인덱스 데이터도 이러한 속성의 모든 인스턴스는 데이터베이스에 이 속성에 저장된 모든 항목에 대해 암호화됩니다. 접미사에 대한 속성 암호화를 활성화할 수 있습니다. 전체 서버에 대해 이 기능을 활성화하려면 서버의 각 접미사마다 속성 암호화를 활성화해야 합니다. 특성 암호화는 **eq** 및 **pres** 인덱싱과 완전히 호환됩니다.

중요

항목 고유 이름(DN) 내에서 사용하는 모든 속성은 효율적으로 암호화할 수 없습니다. 예를 들어 **uid** 특성을 암호화하도록 구성한 경우 해당 값은 **DN**이 아닌 항목에서 암호화됩니다.

```
dn: uid=demo_user,ou=People,dc=example,dc=com
...
uid::Sf04P9nJWGU1qiW9JJCGRg==
```

21.1. DIRECTORY SERVER에서 특성 암호화에 사용하는 키 디렉터리

특성 암호화를 사용하려면 **TLS**를 사용하여 암호화된 연결을 구성해야 합니다. Directory Server는 특성 암호화에 서버의 **TLS** 암호화 키와 동일한 **DestinationRule** 입력 방법을 사용합니다.

서버는 임의로 생성된 대칭 암호화 키를 사용하여 특성 데이터를 암호화하고 암호를 해독합니다. 서버는 서버의 **TLS** 인증서의 공개 키를 사용하여 이러한 키를 래핑합니다. 결과적으로 속성 암호화의 효과적인 강점은 서버 **TLS** 키의 성능보다 클 수 없습니다.

주의

서버의 개인 키에 액세스하지 않으면 랩된 사본에서 대칭 키를 복구할 수 없습니다. 따라서 서버의 인증서 데이터베이스를 정기적으로 백업하십시오. 키를 손실하면 더 이상 데이터베이스에 저장된 데이터의 암호를 해독하고 암호화할 수 없습니다.

21.2. 명령줄을 사용하여 속성 암호화 활성화

이 절차에서는 명령줄을 사용하여 **userRoot** 데이터베이스 의 **phoneNumber** 속성에 대한 속성 암호화를 활성화하는 방법을 설명합니다. 절차를 수행한 후 서버는 이 속성의 기존 값과 새 값을 **AES** 암호화로 저장합니다.

사전 요구 사항

- **Directory Server**에서 **TLS** 암호화를 활성화했습니다.

절차

1. **userRoot** 데이터베이스를 내보냅니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend export -E
userRoot
```

서버는 내보내기를 /var/lib/dirsrv/*slapd-instance_name*/ldif/ 디렉터리에 **LDIF** 파일에 저장합니다. **-E** 옵션은 내보내기 중에 이미 암호화된 속성을 해독합니다.

2. **phone Number** 특성에 **AES** 암호화를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend attr-encrypt --
add-attr telephoneNumber dc=example,dc=com
```

3. 인스턴스를 중지합니다.

```
# dsctl instance_name stop
```

4. **LDIF** 파일을 가져옵니다.

```
# dsctl instance_name ldif2db --encrypted userRoot /var/lib/dirsrv/slapd-
instance_name/ldif/None-userroot-2022_01_24_10_28_27.ldif
```

--encrypted 매개변수를 사용하면 스크립트가 가져오기 중에 암호화하도록 구성된 속성을 암호화할 수 있습니다.

5. 인스턴스를 시작합니다.

```
# dsctl instance_name start
```

추가 리소스

- Directory Server에 대한 TLS 암호화 연결 활성화

21.3. 웹 콘솔을 사용하여 특성 암호화 활성화

이 절차에서는 웹 콘솔을 사용하여 **userRoot** 데이터베이스 의 **phoneNumber** 속성에 대한 속성 암호화를 활성화하는 방법을 보여줍니다. 절차를 수행한 후 서버는 이 속성의 기존 값과 새 값을 **AES** 암호화로 저장합니다.

웹 콘솔의 내보내기 및 가져오기 기능은 암호화된 속성을 지원하지 않습니다. 따라서 명령줄에서 다음 단계를 수행해야 합니다.

사전 요구 사항

- Directory Server에서 TLS 암호화를 활성화했습니다.
- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. **userRoot** 데이터베이스를 내보냅니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend export -E
userRoot
```

서버는 내보내기를 **/var/lib/dirsrv/slapd-*instance_name*/ldif/** 디렉터리에 **LDIF** 파일에 저장합니다. **-E** 옵션은 내보내기 중에 이미 암호화된 속성을 해독합니다.

2. 웹 콘솔에서 **Database** → **Suffixes** → ***suffix_entry*** → **Encrypted attributes s**로 이동합니다.

3. 암호화할 속성을 입력하고 속성 추가를 클릭합니다.

4. 작업 메뉴에서 인스턴스 중지 를 선택합니다.

5. 명령줄에서 **LDIF** 파일을 가져옵니다.

```
# dsctl instance_name ldif2db --encrypted userRoot /var/lib/dirsrv/slapd-instance_name/ldif/None-userroot-2022_01_24_10_28_27.ldif
```

--encrypted 매개변수를 사용하면 스크립트가 가져오기 중에 암호화하도록 구성된 속성을 암호화할 수 있습니다.

6. 웹 콘솔에서 **Actions** 메뉴를 열고 **Start Instance** 를 선택합니다.

추가 리소스

- Directory Server에 대한 TLS 암호화 연결 활성화

21.4. 속성 암호화를 활성화한 후 일반적인 고려 사항

데이터베이스에 이미 있는 데이터에 대한 암호화를 설정한 후 다음 사항을 고려하십시오.

- 암호화되지 않은 데이터는 서버의 데이터베이스 페이지 풀 백업 파일에 유지될 수 있습니다. 이 데이터를 제거하려면 다음을 수행합니다.

  a. 인스턴스를 중지합니다.

  ```
  # dsctl instance_name stop
  ```

  b. /var/lib/dirsrv/slapd-*instance_name*/db/guardian 파일을 제거합니다.

  ```
  # **rm /var/lib/dirsrv/slapd-instance_name/db/guardian``
  ```

c. 인스턴스를 시작합니다.

```
# dsctl instance_name start
```

- 암호화를 활성화하고 데이터를 성공적으로 가져온 후 암호화되지 않은 데이터로 **LDIF** 파일을 삭제합니다.

- 디렉터리 서버는 복제 로그 파일을 암호화하지 않습니다. 이 데이터를 보호하려면 복제 로그를 암호화된 디스크에 저장합니다.

- 서버의 메모리(**RAM**)에 있는 데이터는 암호화되지 않으며 스왑 파티션에 일시적으로 저장할 수 있습니다. 이 데이터를 보호하려면 암호화된 스왑 공간을 구성합니다.

중요

암호화되지 않은 데이터가 포함된 파일을 삭제하더라도 특정 상황에서 이 데이터를 복원할 수 있습니다.

21.5. 특성 암호화에 사용되는 TLS 인증서 업데이트

특성 암호화는 서버의 **TLS** 인증서를 기반으로 합니다. **TLS** 인증서를 업데이트하거나 교체한 후 속성 암호화가 실패하지 않도록 하려면 다음 절차를 따르십시오.

사전 요구 사항

- 특성 암호화를 구성했습니다.

- **TLS** 인증서가 곧 만료됩니다.

절차

1. **userRoot** 데이터베이스를 내보냅니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend export -E
userRoot
```

서버는 내보내기를 **/var/lib/dirsrv/*slapd-instance_name*/ldif/** 디렉터리에 **LDIF** 파일에 저장합니다. **-E** 옵션은 내보내기 중에 이미 암호화된 속성을 해독합니다.

2.

개인 키와 **CSR(**인증서 서명 요청**)**을 생성합니다. 외부 유틸리티를 사용하여 생성하려면 이 단계를 건너뜁니다.

- 하나의 이름으로만 호스트에 연결할 수 있는 경우 다음을 입력합니다.

```
# dsctl instance_name tls generate-server-cert-csr -s
"CN=server.example.com,O=example_organization"
```

- 여러 이름으로 호스트에 연결할 수 있는 경우:

```
# dsctl instance_name tls generate-server-cert-csr -s
"CN=server.example.com,O=example_organization" server.example.com
server.example.net
```

호스트 이름을 마지막 매개변수로 지정하면 명령에서 **DNS:server.example.com, DNS:server.example.net** 항목에 **SAN(Subject Alternative Name)** 확장을 **CSR**에 추가합니다.

-s *subject* 매개변수에 지정된 문자열은 **RFC 1485**에 따라 유효한 주체 이름이어야 합니다. 주체의 **CN** 필드가 필요하며 서버의 **FQDN(**정규화된 도메인 이름**)** 중 하나로 설정해야 합니다. 이 명령은 **CSR**을 **/etc/dirsrv/slapd-*instance_name*/Server-Cert.csr** 파일에 저장합니다.

3.

CSR을 **CA(**인증 기관**)**에 제출하여 발급된 인증서를 가져옵니다. 자세한 내용은 **CA** 설명서를 참조하십시오.

4.

CA에서 발급한 서버 인증서를 **NSS** 데이터베이스로 가져옵니다.

- **dsctl tls generate-server-cert-csr** 명령을 사용하여 개인 키를 생성한 경우 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security certificate
add --file /root/instance_name.crt --name "server-cert" --primary-cert
```

--name *_certificate_nickname*매개변수에 설정한 인증서의 이름을 기록해 둡니다. 이

후 단계에서 필요합니다.

- 외부 유틸리티를 사용하여 개인 키를 생성한 경우 서버 인증서와 개인 키를 가져옵니다.

```
# dsctl instance_name tls import-server-key-cert /root/server.crt /root/server.key
```

이 명령을 사용하려면 먼저 서버 인증서의 경로를 지정한 다음 개인 키 경로를 지정해야 합니다. 이 방법은 항상 인증서의 닉네임을 **Server-Cert** 로 설정합니다.

5. **CA** 인증서를 **NSS** 데이터베이스로 가져옵니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate add --file /root/ca.crt --name "Example CA"
```

6. **CA** 인증서의 **trust** 플래그를 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com security ca-certificate set-trust-flags "Example CA" --flags "CT,,"
```

이렇게 하면 **TLS** 암호화 및 인증서 기반 인증에 대해 **CA**를 신뢰하도록 **Directory Server**가 구성됩니다.

7. 인스턴스를 중지합니다.

```
# dsctl instance_name stop
```

8. **/etc/dirsrv/slapd-*instance_name*/dse.ldif** 파일을 편집하고 속성을 포함한 다음 항목을 제거합니다.

- **CN=AES,cn=encrypted attribute keys,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config**

- **CN=3DES,cn=암호화 특성 키,cn=*database_name*,cn=ldbm database,cn=plugins,cn=config**

중요

모든 데이터베이스의 항목을 제거합니다. **nsSymmetricKey** 속성이 포함된 항목이 **'/etc/dirsrv/slapd-*instance_name*/dse.ldif** 파일에 남아 있으면 디렉터리 서버가 시작되지 않습니다.

9. **LDIF** 파일을 가져옵니다.

```
# dsctl instance_name ldif2db --encrypted userRoot /var/lib/dirsrv/slapd-instance_name/ldif/None-userroot-2022_01_24_10_28_27.ldif
```

--encrypted 매개변수를 사용하면 스크립트가 가져오기 중에 암호화하도록 구성된 속성을 암호화할 수 있습니다.

10. 인스턴스를 시작합니다.

```
# dsctl instance_name start
```