ISBN:

연구교육인증연합 정책 및 기술 프로파일

조진용, 공정욱, 장희진, 이경민

한국과학기술정보연구원

KAFE

목 차

KAFE

연구교육인증연합 정책 본문

– Korean Access Federation (KAFE) –

한국과학기술정보연구원

KAFE

연구교육인증연합 정책

2015. 10. 12. – 제정

제 1 장 용어 정의

본 정책 문서는 다음과 같은 용어를 사용한다.

1. ID 제공자: 사용자 정보를 관리하고 인증 결과 및 속성 정보를 다른 기관에 제공하는 서버(이하 “IdP”라고 한다)

2. 서비스 제공자: IdP 의 인증 결과 및 속성 정보를 이용하여 서비스를 제공하는 서버(이하 “SP”라고 한다)

3. 메타데이터: ID 페더레이션을 위해 필요한 IdP 와 SP 정보가 기록된 데이터

4. 속성: IdP 에서 관리되고 SP 에게 제공될 수 있는 사용자에 대한 정보

5. 협업응용서비스: 원격지에 위치한 연구자들이 온라인상에서 협업을 수행하기 위해 필요한 응용소프트웨어와 서비스를 통칭

6. ID 페더레이션: 사용자 인증 연계를 위하여 제공 서비스 및 사용자 정보의 교환에 동참하는 기관들의 연합

7. 회원기관: ID 페더레이션에 가입한 기관

8. 운영기관: 회원기관에게 정책적·기술적 기반구조를 제공하는 기관

9. 연합정책 본문: 본 문서에 명시된 규정 및 지침

10. 기술 프로파일: 특정 기술(예, SAML, eduroam 등)을 사용하여 정책 및 보증(assurance) 정보의 구체적 실현을 기술한 문서

11. 정보보호 프로파일: 개인정보 보호 및 정보시스템 보안에 대한 구체적 실현을 기술한 문서

12. 연합정책 문서: 연합정책 본문, 기술 프로파일, 정보보호 프로파일 등 규정과 지침을 기술한 문서

제 2 장 소개

한국과학기술정보연구원에서 주관하는 연구 및 교육 분야의 ID 페더레이션을 연구교육인증연합(이하 “인증연합”이라 한다)이라 한다. 영문 표기는 “KAFE”라고 한다.

인증연합은 별도로 정하는 기술 규격 및 운영 기준 등에 따라 운용된다.

본 정책 문서는 인증연합에 참여하는 회원기관의 의무와 권리를 정의한다. 본 규정은 인증연합에 참여하는 운영기관과 회원기관에 대하여 효력이 발생한다.

인증연합의 정책은 본 문서와 부록을 포함하는 모든 문서들로 구성된다. 부록의 목록은 www.coreen.or.kr 에 게시되어 있다.

제 3 장 거버넌스와 역할

3.1 관리기구

인증연합은 한국과학기술정보연구원 연구교육인증위원회(이하, "위원회"라 한다)에 의해 관리된다. 위원회는 인증연합 정책 문서들의 내용을 포함해 다음과 같은 책임을 갖는다.

1. 회원기관 가입 자격의 결정
2. 인증연합 가입 요청에 대한 가부 결정
3. 회원기관의 제명 승인
4. 타 ID 페더레이션과 상호 연계에 대한 평가와 결정
5. 인증연합의 추진 및 발전 방향에 대한 평가와 결정
6. 국내외 유관 기관과 공식적인 유대관계 유지
7. 인증연합 정책 개정의 승인
8. 인증연합과 관련된 기타 사항 결정

3.2 운영기관의 권리와 의무

인증연합은 한국과학기술정보연구원 국가과학기술연구망(KREONET, 이하 "운영기관")에 의해 운영된다. 운영기관은 인증연합 정책 문서들의 내용을 포함해 다음과 같은 책임을 갖는다.

1. 안전하고 신뢰성 있는 운영 관리 및 본 문서와 부록에 명시된 규정을 준수하는 서비스 제공
2. 회원 기관의 운용 담당자에 대한 기술 지원
3. ID 페더레이션 기술 센터의 역할 수행: 소프트웨어의 시험, 솔루션의 추천 및 문서화, 소프트웨어와 운영시스템에 대한 매뉴얼 제공 등

4. 국내외 이해 당사자들과 관계 유지, 연합 간 페더레이션 활동 및 선린 관계 차원의 협력

5. 잠재적 회원기관에 대한 인증연합의 홍보

또한 운영기관은 다음과 같은 권한을 갖는다.

1. 안전하고 신뢰성 있는 운영을 방해하는 회원기관에 대해 제명 발의

2. 인증연합 정책 개정의 발의

3. 회원기관 리스트와 적용된 기술 프로파일(Technical Profile)에 대한 정보 공개

4. 특정 기술 개요를 이용하는 회원기관에 대해 데이터의 일부를 공개

3.3 회원기관의 권리와 의무

회원기관은 다음과 같은 의무를 갖는다.

1. 회원기관은 운용 책임자와 운용 담당자를 임명해야 한다. 운용 책임자는 해당 기관이 설치하는 IdP 또는 SP 의 관리·운용에 책임이 있는 자로써 해당기관에 소속된 과장급 또는 부교수급 이상인 자로 한다.

2. 운용 책임자는 IdP 또는 SP 의 관리·운용에 관한 업무를 담당하는 운용 담당자를 임명해야 한다. 운용 책임자는 운용 담당자의 업무를 분담하는 다수의 운용 담당자를 임명할 수 있다.

3. 운영기관 및 회원기관들의 보안, 신뢰성 또는 명성에 악영향을 끼칠 수 있는 사건·사고의 경우 운영기관에게 보고해야하고, 사건·사고의 해결을 위해 운영기관 및 회원기관들과 협조해야 한다.

4. 회원기관들은 연합정책 본문에 명시된 조항들을 준수해야 한다.

5. 회원기관들은 기술 프로파일에 명시된 조항들을 준수해야 한다. 또한 기술 프로파일이 적용된 IT 시스템들에 대해 보안상 안전하게 유지·관리되도록 보장해야 한다.

6. 회원기관이 개인정보를 처리할 경우, 개인정보보호 관련 법령을 준수하고 개인정보 프로파일을 따라야 한다. 개인정보의 취급에 관하여 법령으로 정한 규정 이외에 위원회가 별도로 정하는 규정 등을 준수해야 한다.

7. 인증연합의 활성화를 목적으로, 운영기관이 공개하는 회원기관의 운용 담당자 이름 등에 대한 개인정보의 제 3 자 제공에 동의해야 한다.

제 4 장 회원기관 가입 자격

위원회는 인증연합의 가입자격 기준을 제시한다. 이 기준은 회원기관 가입자격 규정에 기술되어 있다.

인증연합에 가입 자격이 있는 자는 다음 각 호의 하나 이상에 해당하는 기관이 된다.

1. 한 대학 또는 단과 대학에서 IdP 또는 SP 를 구축하고자 하는 기관

2. 국공립 연구기관 및 연구 또는 지원을 목적으로 하는 법인 중 IdP 또는 SP 를 구축하고자 하는 기관

3. 제 1 호 또는 제 2 호의 기관에 협업응용서비스를 제공하는 것을 목적으로 SP 를 구축하려는 기관

4. 기타 인증연합에서 제공하는 협업응용서비스를 이용하는 것을 목적으로 인증연합에 참여가 필요하다고 위원회가 특별히 인정한 경우.

제 5 장 가입 및 탈퇴 절차

5.1 가입 절차

인증연합에 가입하고자 하는 기관은 소정의 가입 절차에 따라 위원회에 가입 승인을 요구하여야 한다. 가입 신청은 해당 기관의 장이 문서를 통해 실시한다. 위원회는 가입 자격이 적합하다고 인정되는 경우에 이를 승인한다. 가입 자격이 부적합하다고 판단될 경우 거부 사유를 명시해 가입 신청 기관에 통보한다.

5.2 탈퇴 절차

회원기관은 운영기관에 탈퇴를 요청함으로써 인증연합에서 탈퇴할 수 있다. 회원기관의 탈퇴는 일정 시간 이후에 인증연합의 기술 프로파일을 이용할 수 없음을 의미한다.

제 6 장 이용 약관

6.1 제명

인증연합 가입 신청 내용에 허위가 있다고 인정되는 경우, 인증연합 운영을 방해하고 신뢰를 해치는 행위를 실시했다고 인정되는 경우, 가입 자격에 해당되지 않게 되었다고 인정되는 경우, 연합정책 문서에 기술된 조항들을 준수하지 않은 경우, 위원회는 해당 회원기관에 대하여 인증연합 참여를 일시 중지하거나 인증연합에서 제명할 수 있다.

회원기관에서 상기 사항을 위반하였을 경우 운영기관은 5 일 이내에 위반한 사항에 대해 회원기관에게 공식적으로 경고한다. 회원기관이 경고 통지를 받은 후 60 일 이내에 소명하지

못할 경우, 위원회는 회원기관에 공식적으로 통지하고 인증연합 참여 중지 및 제명을 실시할 수 있다.

회원기관에 대한 제명은 회원기관이 사용한 기술 프로파일을 이용할 수 없음을 의미한다.

6.2 면책 조항

회원기관이 운영기관에게 결함의 수정, 비용의 반환 또는 피해 보상을 요구할 수 없는 결점이나 결함에 대해 운영기관과 위원회는 법적 책임을 지지 않는다.

운영기관과 위원회는 인증연합의 서비스에 대한 회원기관의 사용 결과로써 야기되는 어떠한 손실, 피해 또는 손해에 대해서 책임을 지지 않는다. 운영기관 직원의 중과실 또는 고의에 의한 경우는 제외한다.

운영기관과 위원회는 회원기관 또는 회원기관의 최종 사용자가 야기한 피해에 대해서 법적 책임을 지지 않는다.

인증연합 서비스의 사용, 서비스 정지 시간 또는 서비스의 사용과 관련 있는 그 밖의 쟁점으로 인해 운영기관과 위원회가 야기한 피해에 대해 회원기관은 법적 책임을 지지 않는다.

회원기관들 간에 문서로 합의된 사항이 없다면, 회원기관은 타 회원기관에 대해 법적 책임을 지지 않는다.

회원기관은 개인정보보호법 등 대한민국 법령을 준수해야 한다. 위원회와 운영기관은 회원기관과 최종 사용자가 인증연합 서비스를 이용함에 있어 관련 법령을 준수하지 못해 생긴 피해에 대하여 법적 책임을 지지 않는다.

운영기관과 위원회는 어떠한 부수적 또는 간접적(consequential or indirect) 피해에 대해서도 법적 책임을 지지 않는다.

타 ID 페더레이션과의 연계에 따른 정보의 교환은, 어떠한 페더레이션 회원기관과 운영기관 사이에서도 새로운 법적 의무나 권리를 발생시키지 않는다. 운영기관과 회원기관들은 각자의 관할 법령과 사법권에만 영향을 받는다.

6.3 분쟁 해결

인증연합 정책과 관련된 분쟁은 협상을 통해 우선 해결한다. 쟁점이 협상을 통해 해결되지 않을 경우, 법률에 따라 분쟁을 해결할 수 있다.

6.4 타 ID 페더레이션과의 연계

위원회는 타 ID 페더레이션과의 연계 협정에 참여할 수 있다. 자국내 법령과 페더레이션 정책들을 준수하는 타 페더레이션과 연계할 수 있음을 회원기관은 이해하고 인정해야 한다. 타 페더레이션이 준수하는 법령과 정책들은 위원회가 준수를 요구하는 법령 및 정책과 상이할 수 있다.

6.5 정책의 개정

운영기관은 인증연합의 정책 개정을 발의 할 수 있다. 발의된 정책은 위원회의 승인을 받아야 한다. 인증연합 정책이 정하는 것 이외에 인증연합의 운영에 필요한 사항은 위원회가 정한다. 개정된 인증연합 정책은 시행되기 최소 90 일 이전에 회원기관에 문서의 형태로 전달되어야 한다.

KAFE

기술프로파일

– Korean Access Federation (KAFE) –

한국과학기술정보연구원

목 차

기술프로파일

2015. 08. 20. – 초안

제 1 장 SAML 기술표준

KAFE 에서 이용되는 SAML(Security Assertion Markup Language) 표준은 OASIS 보안서비스기술위원회(Security Services Technical Committee)에서 규정한 다음의 표준에 기초하고 있다. 본 운용정책은 시험 페더레이션 기간에 한해 적용하며 정식 서비스 시행 시 수정될 수 있다.

1.1 절 SAML v2.0 Core

SAML v2.0 표준 준수를 위해 필요한 기술적 요구사항을 규정하고 있다.
(http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)

1.2 절 SAML v2.0 Profiles

시스템들 간에 이용되는 식별자, 바인딩 지원, 인증서와 키들의 이용을 규정하고 있다.
(http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)

1.3 절 SAML v2.0 Metadata

메타데이터의 표준 표기법 작성 규칙을 규정하고 있다.
(http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)

제 2 장 프로토콜

본 지침은 KAFE 에 참여하는 ID 제공자 또는 서비스 제공자(이하 엔터티)들이 가급적 넓은 범주의 서비스를 제공하고 활용할 수 있도록 설계되어 있다. 이를 위해, KAFE 에 참여하는 모든 엔터티들은 KAFE 에서 규정한 표준 프로토콜의 이용을 권고한다. 표준 프로토콜은 인증 요청과 인증 응답에 대한 필요사항들을 반드시 만족해야 한다.

KAFE 는 표준 프로토콜의 준수를 위해 simpleSAMLphp SAML2.0 소프트웨어의 이용을 권장한다.

2.1 절 Authentication Request

HTTP-bound SAML 프로토콜의 인증 요청(Authentication Request) 메시지는 SAML 기술 표준 "SAML 2 Profiles" 4.1.3 및 4.1.4 에 정의된 Web Browser SSO 프로파일이 충족되도록 구현해야 한다.

2.2 절 Authentication Response

SAML assertions 를 포함하는 HTTP-bound 인증 응답(Authentication Response) 메시지들은 "SAML 2 Profiles" 4.1.3 및 4.1.4 에 정의된 Web Browser SSO 프로파일이 충족되도록 구현해야 한다.

또한 SAML 응답 메시지 또는 SAML assertion 중 하나는 서명되고, SAML assertion 은 암호화할 것을 권고한다.

2.3 절 simpleSAMLphp

simpleSAMLphp 는 SAML 소프트웨어 패키지이다. simpleSAMLphp 는 UNINETT 에 의해 개발 관리되고 있다(simplesamlphp.org). simpleSAMLphp 1.13 이상의 SAML ID 제공자, SAML 서비스 제공자 이용을 권장한다.

제 3 장 사용자 속성 정보

사용자 속성정보는 사용자의 권한을 부여하기 위해 개별 엔터티들이 이용하는 정보이다. KAFE 에서 사용되는 속성 정보는 본 문서의 부록 "지원되는 속성 정보 목록"을 참조한다.

3.1 절 속성 정보의 이용

[권고] KAFE 에 정의된 모든 사용자 속성정보는 고유 URI 를 가지고 있다. 각 엔터티들은 본 문서의 부록 "지원되는 속성 정보 목록"에서 사용자 속성 정보를 선택하여 이용하는 것을 권장한다. 만약 이용하려는 속성이 "지원되는 속성 정보 목록"에 존재하지 않는 경우 각 엔터티는 KAFE 에 새 속성의 추가를 신청할 수 있다. 신청된 새 속성 정보는 KAFE 에서 검토하고 "지원되는 속성 정보 목록"에 편입여부를 결정한다. KAFE 에 참여하지 않는 경우, "지원되는 속성 정보 목록"에 명시된 속성 이외의 속성 정보들을 이용할 수 있다.

3.2 절 속성 정보의 신뢰성

[권고] ID 제공자는 제공 중인 사용자 속성 정보에 대한 신뢰성을 보장해야 한다. ID 제공자 기관의 내부 구성원이 아닐 경우 속성 정보를 보장하지 않을 수 있다. 하지만 서비스 제공자에 대한 불법적, 악의적 접근을 방지하기 위해 해당 사용자에 대한 속성 관리를 수행해야 한다.

3.3 절 속성 정보의 확인

[권고] 서비스 제공자는 제공받은 모든 속성 정보들이 신뢰할 수 있는 기관(Trusted Authority)에서 전달받은 것인지 검증할 수 있는 방법을 확보해야 한다.

3.4 절 속성 정보의 등급 및 저장 기간

[필수] 서비스 제공자는 속성 정보의 등급을 사용자에게 명시해야 한다. 등급은 “필수(required)”, “권장(recommended)”, “선택(optional)”으로 분류하고 이용 목적과 함께 명확하게 기재해야 한다. 서비스 제공자는 속성 정보를 저장할 경우 저장 기간에 대해 명확하게 기재해야 한다.

3.5 절 사용자의 소속 범위

[필수] 속성 정보를 제공할 사용자의 소속 범위는 EntityID 에 기록된 도메인의 범위와 일치해야 한다. 예를 들어, EntityID 의 도메인 정보가 info.school.ac.kr 일 경우, ID 제공자는 정보통신공학과에 소속된 사용자의 속성정보만 제공해야 한다. 서비스 제공자는 ID 제공자의 메타데이터에 포함된 사용자 속성의 범위와 SAML assertion 에 기재되어 있는 범위를 비교하여 제공되는 범위를 판단해야 한다.

3.7 절 eduPersonTargetedID 의 규격

[필수] eduPersonTargetedID 는 NameQualifier, SPNameQualifier, Opaque ID 를 포함한다. 다음 규격을 준수한다.

● eduPersonTargetedID

<saml:NameID xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"

NameQualifier="[IDP의 entityID]" SPNameQualifier="[SP의 entityID]">

[opaque ID]</saml:NameID>

[조항추가 - 2015.9.17.]

제 4 장 메타데이터

KAFE 는 다음에 규정된 메타데이터를 이용한다.

4.1 절 메타데이터의 규격

[필수] SAML v2.0 메타데이터 규격(SAML v2.0 Metadata)에 따라야 한다.

4.2 절 메타데이터의 종류

[필수] KAFE 는 다음 두 종류의 메타데이터를 이용한다.

1. 엔터티 메타데이터
개별 엔터티들이 KAFE 에 제출한 메타데이터로, 각 엔터티들에 대한 정보를 포함

2. 페더레이션 메타데이터
KAFE 에 의해 생성된 메타데이터로, KAFE 에 참여하는 모든 엔터티들의 메타데이터를 포함

4.3 절 엔터티 메타데이터의 제출

[필수] KAFE 에 참여하는 모든 기관들은 제공자 엔터티들에 대한 엔터티 메타데이터를 KAFE 에 제출해야 한다.

4.4 절 엔터티 메타데이터의 내용

[필수] KAFE 에 참여하는 기관의 서버임을 인정하는 서버 인증서를 갱신하거나 기관의 메타데이터를 변경했을 경우, 해당 기관은 KAFE 에 최신 메타데이터 파일을 반드시 제출해야 한다.

[권고] 메타데이터에 개인정보의 입력이 필요한 부분은 예를 들어, 운용 담당자의 email 주소 등, 가급적 개인 정보가 나타나지 않도록 할 것을 권장한다.

[필수] KAFE 에 제출된 엔터티 메타데이터는 내부에 포함된 개인정보를 포함해 웹에 공개된다. 따라서 운용 책임자는 엔터티 메타데이터의 제출 시 또는 신청 시에 엔터티 메타 데이터에 포함된 개인정보에 대해 함께 공개한 것으로 간주하며 개인정보의 제 3 자 제공에 동의한 것으로 간주한다.

개별 기관에서 제출한 메타데이터 정보는 다음과 같은 목적에 이용된다.

- 엔터티 메타데이터에 포함된 항목들의 검증
- KAFE 의 관리 운영
- 페더레이션 메타데이터의 추가 및 갱신
- KAFE 회원 기관에 대한 페더레이션 메타데이터의 배포 또는 Web 을 통한 공개
- 탐색 서비스(Discovery Service), ID 제공자, 서비스 제공자의 등록

4.5 절 엔터티 메타데이터의 제공기관 관련 요소

[필수] ID 제공자는 KAFE 에 제출하는 엔터티 메타 데이터에 제공 기관관련 요소에 다음의 사항을 기재해야 한다. 기관 내에 여러 개의 엔터티들이 존재할 경우 각 엔터티들을 구분할 수 있어야 한다.

- OrganizationName
  : <md:OrganizationName xml:lang="en">name</md:OrganizationName>
- OrganizationDisplayName:
  : <md:OrganizationDisplayName xml:lang="en">name</md:OrganizationDisplayName>
- OrganizationURL
  : <md:OrganizationURL xml:lang="en">http://www.kreonet.net/</md:OrganizationURL>
- ContactPerson:
  <md:ContactPerson contactType="technical">
  <md:GivenName></md:GivenName>
  <md:SurName></md:SurName>
  <md:EmailAddress></md:EmailAddress>
  </md:ContactPerson>

[조항수정 및 추가 - 2015.11.6.]

4.7 절 엔터티 메타데이터의 개인정보보호정책 고지관련

[필수] 서비스 제공자는 개인정보처리방침을 고지하고 엔터티 메타데이터에 고지 URL 을 기재해야 한다. 고지되는 개인정보처리방침은 대한민국 개인정보보호법을 준수해야 한다.

● privacypolicy 기재 예시

'privacypolicy' => 'https://my.school.ac.kr/privacypolicy',

[조항추가 - 2015.9.16.]

4.8 절 엔터티 메타데이터의 엔터티 ID

[권고] 엔터티 메타데이터의 EntityID 는 다음의 규정을 따른다. 예를 들어, ID 제공 기관인 my.school.ac.kr 에서 simpleSAMLphp 소프트웨어를 이용할 경우 https://my.school.ac.kr/idp/simplesamlphp 로 EntityID 를 정의한다.

[기관 URL]/[IdP 또는 SP]/[SAML 2.0 소프트웨어]

4.9 절 페더레이션 메타데이터의 제출 및 공개

KAFE 는 제출된 모든 엔터티 메타데이터를 검증한 후, 페더레이션 메타데이터에 추가한다.

[필수] 제출된 메타데이터는 모든 회원기관에 공개된다.

페더레이션 메타데이터의 유효기간의 14 일이고, aggregator2 모듈의 validUntil 속성에 기재된다. 페더레이션 메타데이터 그룹의 이름과 공개 URL(제공 예정)은 다음과 같다.

● Name = "KAFE Federation"

● 공개 URL = https://metainfo.kreonet.net/kafe-metadata.xml

페더레이션 메타데이터는 "페더레이션 메타데이터의 제출 및 발간"의 공개 URL 을 통해 다운로드 받아 설치한다.

[조항수정 - 2015.11.6.]

4.10 절 페더레이션 메타데이터의 갱신

[권고] 오래된 페더레이션 메타데이터를 이용할 경우, 다른 사이트에 대한 접속 불가 및 보안 등급의 하락 등이 발생할 수 있다. KAFE 회원 기관은 페데레이션 메타데이터를 주기적으로 업데이트해야 한다.

페더레이션 메타데이터에 기재된 validUntil 속성값의 만료일 이전에 페더레이션 메타데이터가 갱신되어야 한다.

4.11 절 페더레이션 메타데이터의 검증

[권고] KAFE 회원 기관은 7.2 절의 내용을 참고해 다운받은 페더레이션 메타데이터의 유효성을 검증해야 한다.

[조항추가 - 2015.11.9.]

제 5 장 탐색 서비스

KAFE 는 모든 엔터티들이 최적의 방법으로 인증정보를 확인하기 위해 탐색 서비스(Discovery service)를 제공할 예정이다. 서비스 URL(제공 예정)은 다음과 같다.

● 서비스 URL = https://ds.kafe.coreen.or.kr/

제 6 장 기술 지원

KAFE 에 참여하는 개별 엔터티들은 본 문서에서 규정한 표준을 따르는 어떠한 SAML 소프트웨어라도 사용할 수 있다. SAML ID 제공자와 서비스 제공자를 구성하기 위해 필요하면 KAFE 에서 기술 지원이 가능하다. 상용 제품에 대해서는 기술 지원을 하지 않는다.

제 7 장 인증서의 사용

KAFE 는 각 엔터티의 신뢰성을 담보하기 위해 인증서를 이용한다.

7.1 절 페더레이션 메타데이터 서명용 인증서

[권고] KAFE 는 게시, 배포하는 페더레이션 메타 데이터에 대해 자체 서명인증서(Self-signed certificate)를 이용해 서명한다. 서명에 사용할 인증서 내용은 각 기관이 페더레이션 메타데이터의 서명을 검증할 목적으로 KAFE 에서 안전하게 배포해야 한다.

KAFE 는 해당 서명인증서를 웹을 통해 배포할 수 있다. 페더레이션 메타데이터에 이용된 서명인증서의 배포 URL 은 다음과 같다.

● 배포 URL = https://metainfo.kronet.net/

[조항추가 - 2015.11.9.]

7.2 절 페더레이션 메타데이터 서명용 인증서의 검증

[필수] 서명용 인증서의 fingerprint 가 다음과 같지 않을 경우, 개별 엔터티는 해당 서명용 인증서를 이용해서는 안 된다.

Fingerprint (SHA-256)
=9F::00

최신 fingerprint 값은 다음 웹사이트에서 확인할 수 있다.

● 배포 URL = https://metainfo.kronet.net/

[조항추가 - 2015.11.9.]

7.3 절 Certificate Authority

[권고] Certificate Authority(CA)에서 발급받은 인증서는 상호호환성의 문제를 야기할 수 있으므로 사용을 권장하지 않는다.

[조항추가 - 2015.11.9.]

7.4 절 Certificate Authority

[필수] 개인키가 분실되었거나 유출되었을 경우에는 즉시 KAFE 에 통보하여야 하고 해당 인증서를 폐기해야 한다. 또한 새로운 인증서를 즉시 재발급받아야 한다.

[조항추가 - 2015.11.9.]

제 8 장 보안

[필수] 보안 관리를 위해 모든 참여 엔터티들은 다음 사항을 반드시 준수해야 한다.

8.1 절 사용자의 신원정보 관리

[필수] 제공되는 모든 사용자의 신원정보는 실제 사용자의 정보와 일치해야 한다. 각 엔터티의 사용자 신원정보는 유효 기간이 종료된 후 또는 사용자의 서비스 이용 의사 철회가 있는 경우 지체 없이 삭제되어야 한다.

8.2 절 사용자 ID 의 재사용

[권고] 사용 중이거나 이미 사용되었던 uid, eduPersonPrincipalName, eduPersonTargetedID 가 다른 사용자에게 할당되기 위해서는 마지막 사용일로부터 적어도 12 개월 이상 지나야 한다.

8.3 절 서비스 제공자의 사용자 ID 이용

[필수] 서비스 제공자는 중복된 사용자 ID 를 갖는 사용자들을 구분 가능하도록 충분한 기술적 조치를 강구해야 한다.

8.4 절 개인정보의 저장 및 목적 고지

[권고] 개인정보의 보호, 개인정보의 유지, 개인정보의 유출 방지를 위해, 서비스 제공자는 꼭 필요한 경우가 아니면 사용자 정보를 저장하지 않아야 한다.

[필수] 서비스 제공을 위해 사용자의 개인정보를 저장할 경우, 개인정보의 수집 목적, 항목, 보유 및 이용기간, 거부권 및 불이익 등이 사용자에게 고지되어야 한다.

8.5 절 개인정보제공에 대한 사용자 동의

[필수] 사용자의 개인정보가 처리될 때, 개인정보의 전송 시, 개인정보의 이용 고지, 이용 목적, 사용자 동의를 얻는 절차가 반드시 제공되어야 한다.

[필수] 각 엔터티는 사용자 동의 없이 제 3 자에게 개인정보를 제공할 수 없다. 개인정보의 제 3 자 제공을 위해서는 필수 고지 항목(제공받는 자, 목적, 항목, 보유 및 이용기간, 거부권 및 불이익)을 사용자에게 고지하고 정보 주체의 사전 동의를 받아야 한다.

8.6 절 로그파일의 저장

[필수] 서비스에 대한 접근 로그(acess log)는 최소 6 개월 이상 저장되어야 한다.

8.7 절 회원 기관의 책임

[필수] KAFE 에 참여하는 회원기관은 상호 협력하여 ID 페더레이션을 실현하도록 한다. 각 회원기관은 송신하는 정보의 신뢰성이나 정확성을 확보하기 위한 의무를 갖는다. 고의 또는 중대한 과실에 의한 경우를 제외하고 전송한 정보의 신뢰성이나 정확성이 미비하여 발생한 손해에 대해서는 회원기관이 책임을지지 않는다.

KAFE 와 별개로, 참여 기관 간 별도의 협정(agreement)에 의해 개별 엔터티들의 책임에 대해서 명문화 할 수 있다

부록
- 권장 속성정보 목록 -

1. uid

Name	uid
oid	urn:oid:0.9.2342.19200300.100.1.1
Description	computer system login names
Schema	RFC4519

Value or type	String
Multiple values	Single value
Remarks	예; "s9709015", "admin", and "student"

2. eduPersonTargetedID

Name	eduPersonTargetedID
oid	urn:oid:1.3.6.1.4.1.5923.1.1.1.10
Description	사용자 가명(가독 불가)
Schema	eduPerson Object Class Specification
Value or type	최대 256 바이트, 서비스 제공자마다 다른 persistent ID를 제공
Multiple values	Multiple
Remarks	예; "Kxxl8QLncKbguy5xjNLRSkdBc12="

3. sn

Name	sn
oid	urn:oid:2.5.4.4
Description	성(family name)
Schema	RFC4519
Value or type	String
Multiple values	Multiple
Remarks	예; Hong

4. givenName

Name	givenName
oid	urn:oid:2.5.4.42
Description	이름(first name)
Schema	RFC4519
Value or type	String
Multiple values	Multiple
Remarks	예; “Gil-dong”

5. displayName

Name	displayName
oid	urn:oid:2.16.840.1.113730.3.1.241
Description	화면표시 이름
Schema	RFC2798(inetOrgPerson)
Value or type	String
Multiple values	Single value
Remarks	예; "Kildong Hong"

6. mail

Name	mail
oid	urn:oid:0.9.2342.19200300.100.1.3
Description	Email 주소
Schema	RFC2798(inetOrgPerson)
Value or type	string@domain, 최대256 바이트
Multiple values	Single value
Remarks	예; "kildong_hong@kafe.or.kr"

7. eduPersonAffiliation

Name	eduPersonAffiliation
oid	urn:oid:1.3.6.1.4.1.5923.1.1.1.1
Description	사용자 직무 형태
Schema	eduPerson Object Class Specification
Value or type	"faculty", "staff", "student", "member", "employee", none
Multiple values	Multiple
Remarks	상위 6개 직무 타입이 값이 된다. 필요할 경우 타입을 추가할 수 있다. 예; "staff, member"

8. organizationName

Name	organizationName

oid	urn:oid:2.5.4.10
Description	기관명
Schema	RFC4519
Value or type	String
Multiple values	Single value
Remarks	예; "KISTI", "Korean Access Federation"

9. schacHomeOrganization

Name	schacHomeOrganization
oid	urn:oid:1.3.6.1.4.1.25178.1.2.9
Description	기관의 도메인 이름
Schema	RFC1035
Value or type	String
Multiple values	Single value
Remarks	예; "KISTI", "kafe.or.kr"

10. eduPersonPrincipalName

Name	eduPersonPrincipalName
oid	urn:oid:1.3.6.1.4.1.5923.1.1.1.6
Description	도메인 내 사용자 식별 정보
Schema	eduPerson Object Class Specification
Value or type	[unique and persistent identifier]@scope
Multiple values	Single value
Remarks	예; "kildong-home2015@kafe.or.kr"

11. eduPersonScopedAffiliation

Name	eduPersonScopedAffiliation
oid	SAML2: urn:oid:1.3.6.1.4.1.5923.1.1.1.9
Description	기관내 사용자 직무형태
Schema	eduPerson Object Class Specification

Value or type	String@scope
Multiple values	Multiple
Remarks	예; "staff@kafe.or.kr"

[최근 갱신: 2015-11-9 -draft v0.14]

KAFE

개인정보관리지침

– Korean Access Federation (KAFE) –

한국과학기술정보연구원

목 차

개인정보관리 지침

2015. 08. 20. – 초안

제 1 장 소개

KAFE 는 웹 응용서비스 등 정보자원의 제공자인 서비스 제공자와 사용자 인증 및 인가정보의 제공자인 교육·연구기관으로 구성된 연합체입니다.

KAFE 에 가입하면 교육·연구기관 간, 또는 교육·연구기관과 서비스 제공업체 간에 사용자 인증기능이 연계됩니다. 사용자 인증의 연계를 통해 교육·연구기관 내에서 웹 응용서비스 이용 시 싱글사인온(SSO, Single Sign On)이 실현됨과 동시에 다른 기관이나 상용서비스도 소속기관에 등록된 사용자 ID 와 비밀번호를 이용해 접근할 수 있습니다. 예를 들어, 다른 대학의 무선 LAN 을 자신의 소속기관에서 사용하는 사용자 ID 와 비밀번호를 이용해 접근할 수 있습니다.

ID 페더레이션에 있어서 제공자 간 신뢰 관계(Trusted Relationship)는 필수적입니다. 서비스 제공자는 각 기관에 등록된 사용자들을 제대로 관리하고 있는지 신뢰할 수 있어야 합니다. 악의적인 사용자가 서비스 제공자의 정보자원을 보안 침해할 수 있기 때문입니다. ID 제공자도 서비스 제공자에게 보낸 사용자 속성 정보가 악용되지 않는다는 것을 신뢰할 수 있어야 합니다. 개인정보의 유출의 위험이 있기 때문입니다. KAFE 는 ID 페더레이션의 관리주체(3rd-party Trusted Authority) 역할을 수행할 예정입니다. KAFE 에서는 참여 기관의 신뢰관계 구축을 위해 정책 수립과 정책 준수 여부를 정기적으로 조사할 예정입니다.

KAFE 는 2015 년 9 월부터 ID 페더레이션의 시험서비스를 실시하고 있습니다. 참여 기관의 수는 2015 년 9 월 현재 ID 제공자(Identity Provider: 교육 및 연구기관 등)가 2 개 기관, 서비스 제공자(Service Provider: 서비스 제공 측)이 7 건을 기록하고 있습니다. ID 제공자와 서비스 제공자는 계속해서 추가될 예정입니다.

제 2 장 KAFE 및 개인정보

KAFE 에 참여하면 사용자가 연계된 웹응용 서비스에 로그인 할 때마다 사용자 속성 정보가 서비스 제공자에게 전달됩니다. 속성정보에는 개인정보가 포함될 수 있습니다. 개인정보가 포함될 경우, 적절한 조치를 취하지 않으면 개인정보보호법의 [개인정보의 제공]과 [개인정보의 암호화] 등 관련 법령을 위반할 가능성이 있습니다. KAFE 는 이러한 법령 위반을 사전에 방지하기 위해 사용자 동의(예: uApprove, Consent) 방식을 제공하고 있으며 개인정보의 외부 송·수신시 암호화 조치를 강제하고 있습니다.

제 3 장 개인정보란?

대한민국 개인정보보호법에 의하면 개인정보란 살아있는 개인에 관한 정보로서 성명, 주민등록번호 및 영상 등을 통하여 개인을 알아볼 수 있는 정보를 의미합니다. 개별 정보만으로 특정 개인을 알아 볼 수 없더라도 다른 정보와 쉽게 결합하여 알아볼 수 있는 것들도 개인정보의 범주에 포함됩니다. 사용자 ID, email 주소, IP 주소 등도 개인정보의 범주에 포함됩니다. 보다 자세한 사항은 대한민국 법률 제 10465 호 개인정보보호법을 참조하십시오.

제 4 장 개인정보관리 지침

개인정보 관리 지침은 제공자 기관의 관리 감독 기관(정부 기관)에 따라 달라질 수 있으므로 법조항과 관리 감독 기관의 지침을 확인해야 합니다. 아래 표는 정부기관의 관리 가이드라인 중 일부를 발췌한 내용입니다.

해당 법 조항	점검 항목
제 15조 (개인정보의 수집·이용 동의)	– 회원 가입 시 동의 여부 – 개인정보 수집 시 동의 여부 – 정보주체 동의 시 필수고지항목 고지 여부 – 필수고지항목(4개) 내용의 적정 여부 4개: 목적, 항목, 보유 및 이용기간, 거부권 및 불이익
제 17조 (개인정보의 제공)	– 제 3자에게 개인정보 제공 시 정보주체의 동의 여부 – 정보주체 동의 시 필수 고지항목(5개) 고지 여부 – 필수 고지항목(5개) 내용의 적정 여부 5개: 제공받는 자, 목적, 항목, 보유 및 이용기간, 거부권 및 불이익
제 18조 (개인정보의 이용·제공 제한)	– 개인정보 수집 당시 정보주체의 이용, 제공 동의 범위를 초과하여 이용, 제공 여부 – 개인정보 제공 시 제공 목적범위 내 이용, 안전조치 실시, 목적 달성 후 파기 등 요청 여부 – 동의에 의한 목적 외 이용, 목적 외 제3자 제공시 필수 고지항목(5개) 고지 여부 – 필수 고지항목(5개) 내용의 적정 여부 – 5개: 제공받는 자, 목적, 항목, 보유 및 이용기간, 거부권 및 불이익
제 21조 (개인정보의 파기)	– 보유기간 경과, 처리 목적 달성 후 지체없이 개인정보 파기 여부 – 개인정보 파기 시 복구 또는 재생되지 않도록 조치 여부 – 임시파일 및 출력자료 등에 대한 즉시 파기 여부 – 법령에 따라 보존할 경우 별도 분리 보관 여부
제 23조 (민감정보의 처리 제한)	– 사상, 정치, 건강 등 민감정보의 동의에 의한 수집 및 제공 시 구분 동의 여부 – 정보주체 동의 시 필수 고지항목(수집 4개, 제공 5개) 고지 여부 – 필수 고지항목(4개 또는 5개) 내용의 적정 여부
제 24조	– 고유식별정보의 동의에 의한 수집 및 제공 시 구분 동의 여부

(고유식별정보의 처리 제한)	- 고유식별정보: 주민등록번호,여권번호,운전면허번호,외국인등록번호 주민등록번호 외 회원가입 방법 제공 여부
제 29조 (안전의무 조치)	- 비밀번호의 외부 송,수신 시 암호화 조치 여부 - 비밀번호의 내부 저장 시 일방향 암호화 조치 여부 - 바이오정보의 외부 송,수신 시 암호화 조치 여부 - 바이오정보의 내부 저장 시 암호화 조치 여부 - 고유식별정보의 외부 송,수신 시 암호화 조치 여부 - 고유식별정보의 인터넷과 내부망의 중간지점 저장시 암호화 조치여부 - 고유식별정보의 내부 저장 시 암호화 조치 또는 그에 상응하는 조치 적용 여부

제 5 장 속성정보와 개인정보

KAFE 에서 사용 권고 중인 사용자 속성들은 다음과 같습니다. 모든 속성 정보가 서비스 제공자에게 전달되는 것은 아닙니다. 서비스 제공자에 따라 개인정보가 필요 하지 않는 곳도 있을 수 있습니다.

제공 속성	설명	개인정보 가능성
uid	사용자 ID(시험 서비스 기간에 한시적 적용)	O
eduPersonTargetedID	서비스 제공자 별 암호화된 사용자 고유번호	
displayName	화면표시 이름	O
mail	사용자 이메일 주소	O
eduPersonAffilation	사용자의 기관내 직무정보	
organizationName	사용자의 소속기관명	
schacHomeOrganization	사용자 소속기관의 최상위 도메인 이름	
eduPersonPrincipalName	도메인 내 사용자 ID 정보	O
eduPersonScopedAffilation	도메인 내 사용자 직무 정보	

제 6 장 개인정보의 제 3 자 제공에 대한 승인

KAFE 는 제 3 자 제공의 승인을 위해 사용자 동의(uApprove, Consent) 구조를 제공하고 있습니다. 사용자 동의 구조를 이용하면 사용자가 연계된 웹응용 서비스에 접속할 때 전송되는 속성 정보의 목록과 함께 제 삼자 제공을 승인할지 여부를 나타내는 메시지 상자가 실시간으로 나타납니다. 사용자가 정보 제공에 동의할 경우 개인정보의 제 3 자 제공에 대한 승인으로 판단합니다. 정보 제공에 동의하지 않을 경우 서비스에 접속할 수 없습니다.

제 7 장 KAFE 이용 시 유의점

KAFE 에 참여하면 개인 정보가 제 3 자에게 제공될 수 있습니다. 개인정보를 제 3 자에게 제공하기 위해서는 미리 사용자 동의를 얻어야 합니다. 하지만, 사용자 동의 구조를 이용하여 개별 사용자의 동의를 얻을 수 있습니다.

추가적으로 KAFE 에 참여하기 전에 다음 사항을 유의해야 합니다.

1. 개인정보보호법의 준수

2. 제공자 기관의 개인정보호정책과 KAFE 의 개인정보관리지침 간 정합성

3. 개인정보 활용에 대한 기관 내 승인

[최근 갱신: 2015-9-7 -draft v0.11]

KAFE

ID 제공자의 구축

– Korean Access Federation (KAFE) –

한국과학기술정보연구원

목 차

KAFE

SAML ID 제공자 시스템의 구축

2015. 08. 20. – 초안
2015.11.9.(draft v0.15) – 최종 갱신

제 1 장 설치 환경

본 설치매뉴얼은 simpleSAMLphp 1.13 버전을 이용해 Ubuntu 또는 CentOS 환경에서 SAML 2.0 IdP(Identity Provider)를 구축하는 방법을 기술한다. IdP 의 구축을 위해 다음과 같은 요구조건이 충족되어야 한다.

– LAMP 스택의 설치: Apache, MySQL, PHP 5.3 이상

– 공인인증서(SSL)의 설치

제 2 장 simpleSAMLphp 의 설치

2.1 절 simpleSAMLphp

simpleSAMLphp 는 UNINETT 에서 개발한 SAML v2.0 소프트웨어이다. IdP 또는 SP(Service Provider)로 설치가능하며 최신 버전은 1.13.2 이다(2015 년 9 월). 이하 IdP 또는 SP 는 SAML 2.0 IdP 또는 SAML 2.0 SP 를 의미한다. simplesamlphp.org 에서 추가적인 정보를 얻을 수 있다.

2.2 절 simpleSAMLphp 의 설치 환경

simpleSAMLphp 의 설치를 위한 서버 환경은 다음과 같다. 특별한 언급이 없는 한 ID 제공자용 서버는 Ubuntu 14.04 LTS(64 비트)를 이용한다.

– php, MySQL, httpd 가 설치

– IPv6 disable 권장

– selinux disable

– Linux 방화벽(iptables) 80/443(http/https) 포트 개방

– 시간 동기화를 위한 NTP 설정(NTP 서버: time.kriss.re.kr)

2.3 절 simpleSAMLphp 의 설치

simpleSAMLphp 의 구동을 위해 요구되는 소프트웨어 패키지를 설치한다. simpleSAMLphp 의 설치 경로는 /var/simplesamlphp 로 가정한다.

```
~# clear
```

```
~# sudo apt-get install php-date openssl php5-mcrypt
// 인증 소스로 LDAP를 이용하 경우
~# sudo apt-get install php5-ldap

//simplesamlphp 다운로드
~#sudo wget https://simplesamlphp.org/res/downloads/simplesamlphp-1.13.2.tar.gz

//압축해제 및 설치
~# sudo cp ./simplesamlphp-1.13.2.tar.gz    /var/
~# sudo cd   /var
~# sudo tar zxvf ./simplesamlphp-1.13.2.tar.gz
~# sudo mv ./simplesamlphp-1.13.2 ./simplesamlphp
```

※ CentOS 6.7 (php 5.5.30)에서 mcrypt 설치방법

```
//기존 php 5.5.30을 모두 지웠다고 가정한다.
~# rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
~# yum install php56w php56w-opcache php56w-mcrypt php56w-xml php56w-mysql
~# service httpd restart
```

2.4 절 Apache 서버 설정

아래 설정 방법은 HTTP(80 포트)에 대한 환경설정을 보여준다. HTTPS(443)에 대한 Apache 환경설정 방법은 생략한다. ※ ID 제공자 서버는 반드시 공인인증서를 설치하고 HTTPS(443)을 이용해야 한다.

```
~# sudo cd /etc/apache2/sites-available
~# sudo   nano   000-default.conf
// <VirtualHost *:80>을 찾아 아래와 같이  수정
<VirtualHost *:80>
        DocumentRoot    /var/www/html/
        Alias     /simplesaml       /var/simplesamlphp/www

~# sudo nano /etc/apache2/apache2.conf
// 다음과 같은 항목을 추가
<Directory /var/simplesamlphp/>
```

```
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

~# sudo service apache2 restart
```

2.5 절 simpleSAMLphp 의 구동환경 설정

아래와 같이 simpleSAMLphp 를 환경설정한다. 'secretsalt'는 다음 명령을 이용해 추출할 수 있다.

tr –c –d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32 count=1 2>/dev/null;echo

```
~# sudo cd  /var/simplesamlphp/config
~# sudo nano config.php
// 아래와 같이 환경설정 함
'baseurlpath' => 'simplesaml/',
'certdir' => 'cert/'
'loggingdir'        => 'log/',
'datadir' => 'data/',

// 다음 사항은 꼭 수정해야 함
'auth.adminpassword'      => '[관리용 패스워드 입력]',
'secretsalt' => '[secret salt 입력]',
'technicalcontact_name' => '[관리자 이름]',
'technicalcontact_email' => '[관리자 이메일]',
'language.default' => 'en',
'timezone' => 'Asia/Seoul',
```

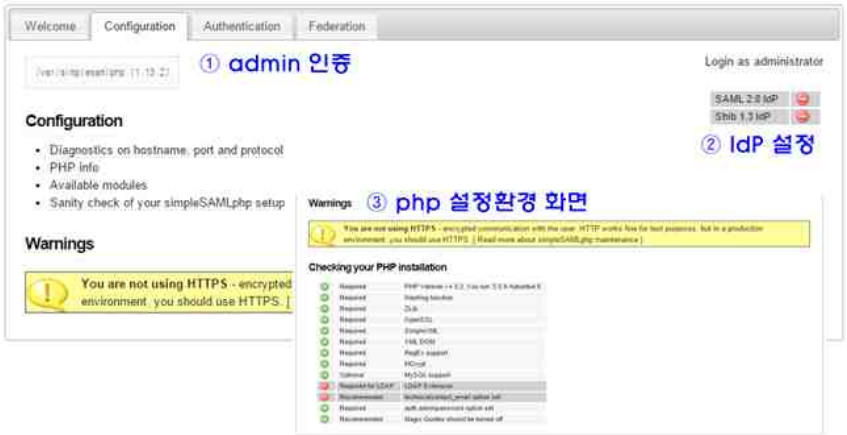
simpleSAMLphp 에서 제공하는 특정 모듈을 활성화하고 싶다면 [모듈명] 디렉토리에서 enable 파일을 생성한다. 다음 예는 LDAP 모듈을 활성화하기 위한 방법이다.

```
~# sudo   cd    /var/simplesamlphp/modules
~# sudo   cd   ./ldap
~# sudo   touch    enable
```

환경 설정이 완료되었다면 웹 브라우저를 이용해 http://[서버주소]/simplesaml 을 접속한다. 정상적으로 설치되었다면 아래와 같은 화면이 나타난다.

2.6 절 설정된 환경의 검증

Authentication 탭에서 admin 으로 로그인 한다. 현재 IdP 가 활성화되지 않은 상태이므로 ②와 같이 적색원이 보여야 한다. admin 으로 로그인한 후 ③과 같이 표시된다면 정상적으로 설치된 상태이다. 관리자 이메일과 LDAP Extension 을 설치했다면 모두 녹색원으로 표시되어야 한다.



제 3 장 SAML ID 제공자의 설치

3.1 절 ID 제공자 기능의 활성화

환경설정 파일을 수정해 IdP 기능을 활성화시킨다. SAML2.0 이외에 SAML1.0 을 지원하기 위해서는 'enable.shib13-idp'를 true 로 설정한다.

```
~# sudo  cd   /var/simplesamlphp/config
~# sudo  nano config.php
// 아래와 같이 수정한다.
'enable.saml20-idp' => true,
'enable.shib13-idp' => false,
'enable.adfs-idp' => false,
'enalbe.wsfed-sp'=> false,
'enable.authmemcookie'  => false,
```

위와같이 환경설정을 한 후 웹 브라우저를 이용해 접근했을 때 다음과 같이 SAML 2.0 IdP 항목이 녹색원으로 표시되어야 한다.

3.2 절 인증 소스의 선택

simpleSAMLphp 는 LDAP, LDAPMulti, SQL, Radius 등 다양한 사용자 인증 소스와의 연동을 지원한다. 더 자세한 사항은 simplesamlphp.org 를 참조한다.

본 매뉴얼에서는 exampleauth:UserPass 모듈을 이용해 사용자를 인증한다. 모듈을 다음과 같이 활성화시킨다. 이하 employee/employeepass(사용자 ID/비밀번호) 또는 student/studentpass(사용자 ID/비밀번호)로 사용자 인증이 가능하다.

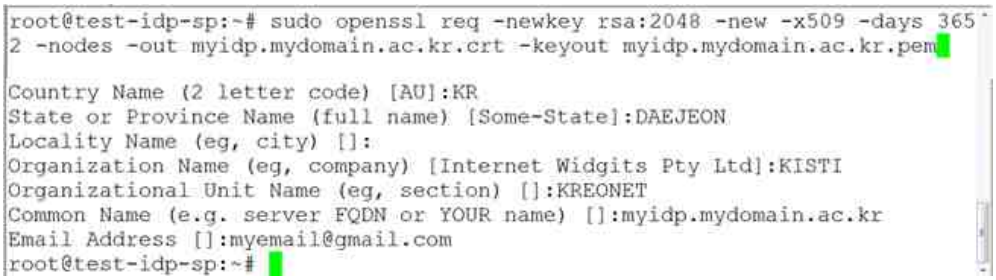
```
~# sudo  cd   /var/simplesamlphp/modules/exampleauth
~# sudo  touch enable


~# sudo cd /var/simplesamlphp/config
~# sudo nano authsources.php
// 아래와 같이 authsources.php를 수정한다.
```

```
/* (주석 제거)
'example-userpass' => array(
        'exampleauth:UserPass',
        'employee:employeepass'  => array(
                'uid'      => array('employee'),
                'eduPersonAffiliation' => array('member', 'employee'),
        ),
),
*/(주석 제거)
```

3.3 절 SSL 자가 인증서의 생성

※ 상용 CA(Certication Authority)에서 발급한 인증서는 SAML 호환성 문제를 야기할 수 있으므로 Self-signed certificate 를 이용한다.

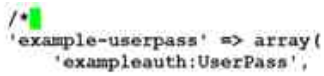
아래 [myidp.mydomain.ac.kr]은 IdP 구축자의 기관 환경에 맞게 적절히 변경해야 한다. Common Name (e.g., server FQDN or YOUR name)은 IdP 용 서버의 IP 주소 또는 도메인명으로 설정해야 한다.



자가 인증서를 생성한 후 환경설정을 계속한다.

```
~# sudo   cd    /var/simplesamlphp
~# sudo   mkdir cert
// 생성된 .cert 파일과 .pem 파일은 /var/simplesamlphp/cert 디렉토리로 이동
```

IdP 의 메타데이터를 수정해 자가 인증서를 등록한다. 인증소스 'example-userpass'는 앞서 authsources.php 에 등록한 인증소스의 이름과 일치해야 한다. 본 매뉴얼은 authsources.php 등록된 이름 'example-userpass'를 이용한다.



```
~# sudo   cd    /var/simplesamlphp/metadata
~# sudo   nano saml20-idp-hosted.php

// 아래와 같이 수정해 인증서를 등록한다.
'privatekey'        => '/var/simplesamlphp/cert/[인증서이름].pem',
'certificate'       => '/var/simplesamlphp/cert/[인증서이름].crt',

// 인증소스의 등록
'auth'    => 'example-userpass',
```

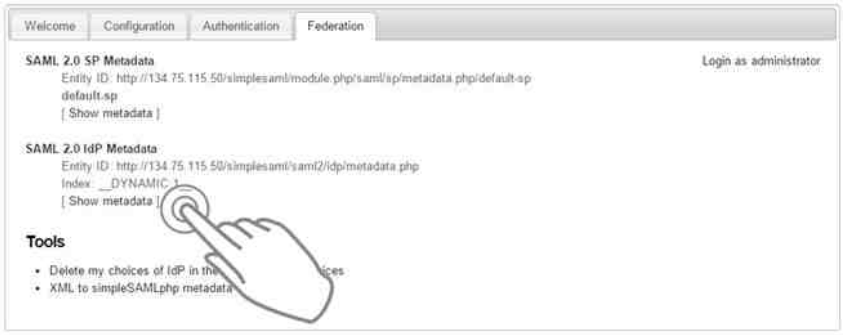
3.4 절 환경 설정의 검증

http://[서버주소]/simplesaml 에 접속해 설정된 SAML IdP 환경이 정상적으로 동작하는지 확인한다. Authentication 탭에서 Test configured authentication sources 를 선택하면 등록된 인증소스의 목록을 확인할 수 있다.

Example-userpass 를 선택한 후 student/studentpass(사용자 ID/비밀번호)를 이용해 로그인 하면 다음 그림과 같이 uid 와 Affiliation 정보가 화면표시된다.

3.5 절 ID 제공자의 메타데이터 확인

SAML 2.0 은 IdP 와 SP 간 신뢰관계를 확인하기 위해 메타데이터를 교환한다. IdP 와 SP 가 서로 통신하기 위해서는 상대방의 메타데이터 정보를 저장하고 있어야 한다. http://[서버주소]/simplesaml 에 접속해 Federation 탭을 선택하면 자신의 메타데이터 정보를 확인할 수 있다.



IdP 와 연계되는 상대 SP 의 SAML 소프트웨어가 simpleSAMLphp 일 경우 php 문법의 flat 포맷을 이용한다. 상대 SP 가 Shibboleth 를 이용할 경우 XML 포맷을 이용한다. 아래 그림과 같은 flat 포맷의 메타데이터 값을 상대 SP 와 교환해야 IdP-SP 간 통신이 가능하다.

제 4 장 ID 제공자와 사용자 DB 의 연동

지금부터 Ubuntu 대신 CentOS 기준으로 설명한다. Ubuntu 환경에서도 유사한 방법으로 환경설정이 가능하다.

4.1 절 MySQL 사용자 DB 이 구성

연동방법을 설명하기 위해 MySQL 데이터베이스에 사용자 계정을 생성한다. SQL 기반의 사용자 데이터베이스가 존재할 경우 본 절은 생략 가능하다. MySQL 에 로그인 한후 데이터베이스와 테이블을 생성한다.

```
mysql> create database user_db_test;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| geoip              |
| mysql              |
| performance_schema |
| test               |
| user_db_test       |
+--------------------+
6 rows in set (0.00 sec)

mysql> use user_db_test;
Database changed


mysql> use user_db_test;
Database changed
mysql> create table ex_users(
    -> pid int(10) not null auto_increment primary key,
    -> username varchar(30),
    -> password varchar(20));
Query OK, 0 rows affected (0.03 sec)


mysql> insert into ex_users values(1, 'student', 'student1234');
Query OK, 1 row affected (0.00 sec)

mysql> select * from ex_users
    -> ;
+-----+----------+-------------+
| pid | username | password    |
+-----+----------+-------------+
|   1 | student  | student1234 |
+-----+----------+-------------+
1 row in set (0.00 sec)
```

4.2 절 ID 제공자에 사용자 DB(인증소스) 등록

authsources.php 를 다음과 같이 수정한다. 인증소스의 이름을 dgist-userpass 로, 사용자 DB 의 이름은 user_db_test 로 가정한다. 이후 사용자 ID 와 비밀번호는 student/student1234 이다. 인증소스의 이름의 임의로 수정할 수 있으나 이름이 디렉토리 구조를 반영할 것을 권장한다.

```
~# cd /var/simplesamlphp/config/

~# nano authsources.php
```

```
// 아래와 같이 수정한다.
<?php
$config = array(
'dgist-userpass' => array(
        'dgist:CoreAuth',
        'dsn'=>'mysql:host=localhost;dbname=user_db_test',
        'username'=>'[MySQL 사용자 ID]',
        'password' =>'[MySQL 사용자 비밀번호'],
),
```

saml20-idp-hosted.php 파일을 수정한다.

```
~# cd /var/simplesamlphp/metadata/
~# nano saml20-idp-hosted.php

// 다음 부분을 찾아 수정한다.
/*
* Authentication source to use. Must be one that is configured in
*'config/authsources.php'.
*/
'auth' => 'dgist-userpass' ,
'userid.attribute' => 'uid',
```

Authentication sources 가 등록되었으므로 dgist:CoreAuth 라는 이름으로 인증 모듈을 구현한다.

4.3 절 dgist:CoreAuth 모듈의 생성

simpleSAMLphp 에서 모듈의 이름은 디렉토리 구조를 반영한다. 따라서 CoreAuth 모듈은 dgist 디렉토리 밑에 존재하게 된다. CentOS 에서 IdP 와 MySQL 데이터베이스를 연동시키기 위해서 PDO(PHP Data Objects) 모듈을 설치해야 한다.

```
~# yum install php-mysql 또는 yum install -enablerep mysql
~# service httpd restart

~# cd /var/simplesamlphp/modules
~# mkdir -p ./dgist/lib/Auth/Source
```

```
~# cd ./dgist/lib/Auth/Source
~# touch  CoreAuth.php

// dgist 모듈을 활성화 시킨다.
~# cd  /var/simplesamlphp/modules/dgist/
~# touch  default-enable
```

CoreAuth.php 파일을 수정한다.

```
~# cd /var/simplesamlphp/modules/dgist/lib/Auth/Source
~# nano  CoreAuth.php

<?php
class  sspmod_dgist_Auth_Source_CoreAuth extends  sspmod_core_Auth_UserPassBase {
        private $dsn;
        private $username;
        private $password;

        public function  __construct($info,  $config)  {
                parent::__construct($info,  $config);

                if(!is_string($config['dsn']))  {
                        throw new Exception('Error');
                }
                $this->dsn =  $config['dsn'];

                If(!is_string[$config['username']]) {
                        throw new Exception('Error');
                }
                $this->username = $config['username'];
                If(!is_string($config['password'])) {
                        throw new Exception('Error');
                }
```

```
                $this->password = $config['password']

private  function  checkPassword($db_in_pw, $user_in_pw) {
        return $user_in_pw === $db_in_pw;
}

protected function login($username, $password) {
                //$db = new PDO("mysql:host=localhost;dbname=user_db_test", "$this->username",
"$this->password");

                $db = new PDO($this->dsn, $this->username, $this->password);
                $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

                /*
                 * Ensure that we are operating with UTF-8 encoding
                 * This command is for MySQL. Other database may need different commands.
                 */
                $db->exec("SET NAMES 'utf8'");

                /*
                 * With PDO we use prepared statements. This saves us from having to escape
                 * the username in the database query.
                 */
                $st = $db->prepare("SELECT username, password, name, email, affi from ex_users
where username = '$usernam$
                if(!$st->execute()){
                        throw new Exception('Failed to Query database for the user');
                }
                $row = $st->fetch(PDO::FETCH_ASSOC);
                if (!$row) {
                        /* User not found */
                        SimpleSAML_Logger::warning('MyAuth:  Could  not  find  user  '  .
```

```
var_export($username, TRUE) . '.');
                        throw new SimpleSAML_Error_Error('WRONGUSERPASS');
                }
if (!$this->checkPassword($row['password'], $password)) {
                        SimpleSAML_Logger::warning('MyAuth: Could not find user ' .
var_export($password, TRUE) . '.');
                        throw new SimpleSAML_Error_Error('WRONGUSERPASS');
                }
                 $attributes = array(
                        'uid' => array($username),
                 );
                return $attributes;
        }
}
?>
```

4.4 절 dgist:CoreAuth 모듈의 검증

http://[서버주소]/simplesaml 에 접속해 dgist:CoreAuth 모듈이 정상동작하는지 확인한다. 정상적으로 설치되었다면 아래 그림처럼 dgist-userpass 가 존재한다.

dgist-userpass 를 선택하고 username 과 password 에 student/student1234 를 각각 입력한다. 정상적으로 설치되었으면 아래 그림처럼 사용자 속성값이 보인다.


Your attributes
User ID student
Logout
[Logout]
About simpleSAMLphp
This simpleSAMLphp thing is pretty cool. Where can I read more about it? You can find more information about simpleSAMLphp at the Feide RnD blog over at UNINETT.
Copyright © 2007-2014 Feide RnD

제 5 장 사용자 속성의 제어

개별 IdP 에서 제공하는 사용자 속성은 ID 제공자 기관에서 결정한다. KAFE(Korean Access Federation)에서는 8 가지 속성의 이용을 권장하고 있다. 개별 속성은 RFC4519, RFC2798, RFC4524, eduPerson, SCHAC 등에 정의되어 있다. KAFE 가 정의하는 속성은 다음과 같다. 보다 자세한 사항은 www.coreen.or.kr 을 참조한다.

제공 속성	설명	개인정보 가능성
uid	사용자 ID(시험 서비스 기간에 한시적 적용)	O
eduPersonTargetedID	서비스 제공자 별 암호화된 사용자 고유번호	
sn	성	O
givenName	이름	O
displayName	사용자의 화면표시 이름	O
mail	사용자 이메일 주소	O
eduPersonAffilation	사용자의 기관내 직무정보	
organizationName	사용자의 소속기관명	
schacHomeOrganization	사용자 소속기관의 최상위 도메인 이름	
eduPersonPrincipalName	도메인 내 사용자 ID 정보	O
eduPersonScopedAffilation	도메인 내 사용자 직무 정보	

5.1 절 MySQL 테이블의 확장 및 CoreAuth.php 수정

다음과 같이 데이터베이스 테이블을 확장한다.

```
Database changed
mysql> show tables;
+------------------------+
| Tables_in_user_db_test |
+------------------------+
| ex_users               |
+------------------------+
1 row in set (0.00 sec)

mysql> desc ex_users;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| pid      | int(10)     | NO   | PRI | NULL    | auto_increment |
| username | varchar(30) | YES  |     | NULL    |                |
| password | varchar(20) | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> alter table ex_users add column name varchar(30);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table ex_users add column email varchar(30);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table ex_users add column affi varchar(30);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0


mysql> desc ex_users;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| pid      | int(10)     | NO   | PRI | NULL    | auto_increment |
| username | varchar(30) | YES  |     | NULL    |                |
| password | varchar(20) | YES  |     | NULL    |                |
| name     | varchar(30) | YES  |     | NULL    |                |
| email    | varchar(30) | YES  |     | NULL    |                |
| affi     | varchar(30) | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

다음 그림처럼 표시될 수 있도록 테이블에 값을 추가한다.

```
mysql> select * from ex_users;
+-----+----------+-------------+---------+---------------------+---------+
| pid | username | password    | name    | email               | affi    |
+-----+----------+-------------+---------+---------------------+---------+
|   1 | student  | student1234 | my name | my_name@dgist.ac.kr | student |
+-----+----------+-------------+---------+---------------------+---------+
1 row in set (0.00 sec)
```

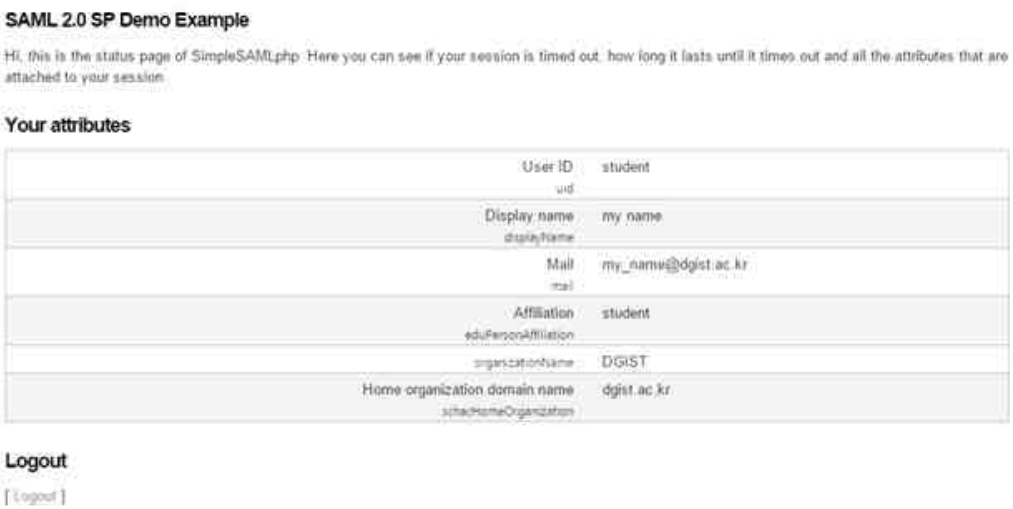
CoreAuth.php 파일을 수정한다.

```
~# cd  /var/simplesamlphp/modules/dgist/lib/Auth/Source

~# nano CoreAuth.php


// 다음과 같이 수정

/*

 * With PDO we use prepared statements. This saves us from having to escape
```

```
 * the username in the database query
 */

$st = $db->prepare("SELECT username, password, name, email, affi    from
ux_users   where   username = '$username'");
// CoreAuth.php 하단의 $attributes 변수를 다음과 같이 수정
$attributes = array(
        'uid' => array($username),
        'displayName'     => array($row['name']),
        'mail'              => array($row['email']),
        'eduPersonAffiliation'       => array($row['affi']),
        'organizationName'        => array('DGIST'),
        'schacHomeOrganization' => array('dgist.ac.kr'),
);
```

5.2 절 검증

http://[서버주소]/simplesaml 에 접속해 속성값들이 화면 표시되는지 확인한다.



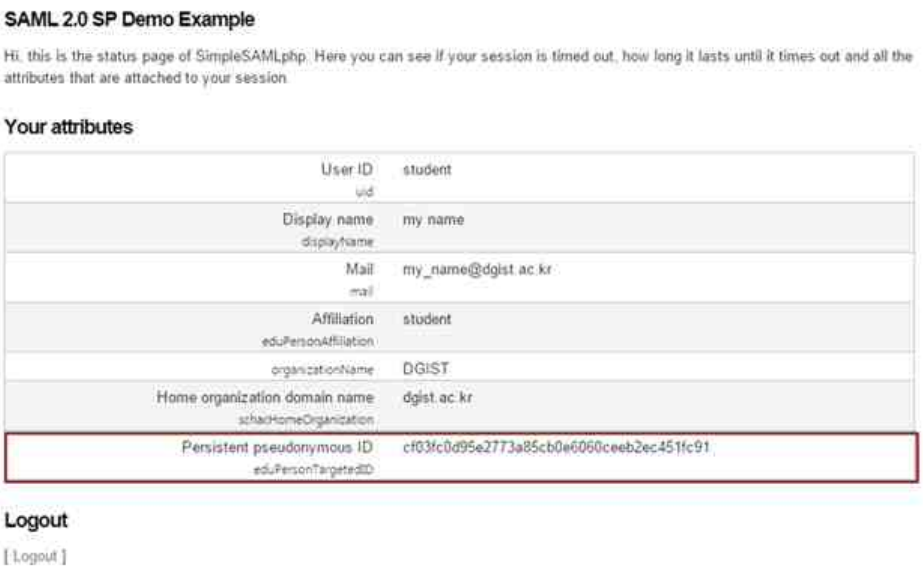
5.3 절 eduPersonTargetedID 값의 생성 및 검증

saml20-idp-hosted.php 에서 userid.attribute 이 설정을 확인하고 config.php 파일을 수정한다.

※ 6.2 절 oid 설정 후 적용해야 함

```
~# nano /var/simplesamlphp/metadata/saml20-idp-hosted.php
// 'userid.attribute' => 'uid' 로 설정되어 있는지 확인
'authproc' => array(
        60 => array(
            'class' => 'core:TargetedID',
            'nameId' => TRUE,
        ),
    ),
    'AttributeNameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
    'attributeencodings' => array(
        'urn:oid:1.3.6.1.4.1.5923.1.1.1.10' => 'raw', /* eduPersonTargetedID with oid NameFormat. */
    ),
```

수정이 완료되었다면 웹 브라우저를 이용해 simplesamlphp 에 접근한 후 사용자 로그인을 수행한다. [주의] IdP 서버에서 로그인했을 경우에는 eduPersonTargetedID 속성값이 화면표시 되지 않는다. SP 와 ID 연계한 후, SP 서버에서 로그인했을 경우 속성값이 아래와 같이 출력된다.



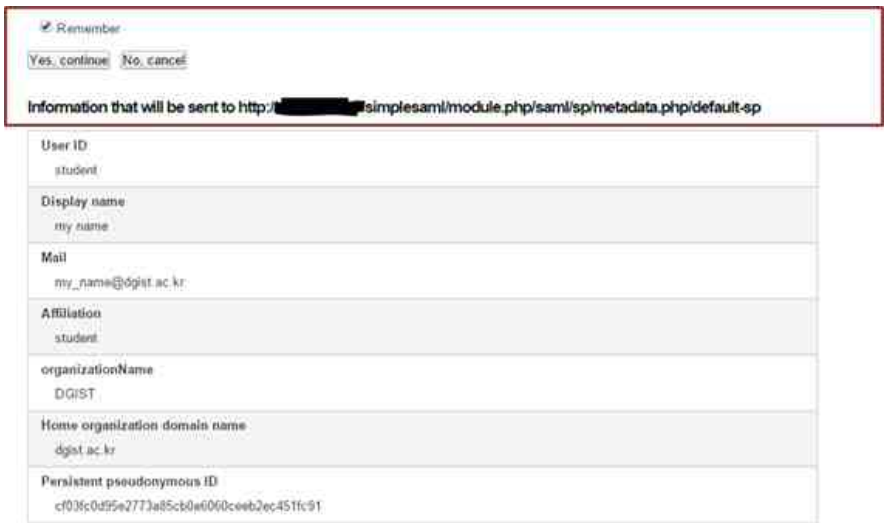
5.4 절 Consent 모듈의 활성화 및 검증

Consent 모듈은 속성정보를 서비스 제공자에게 전달하기 전에 사용자 동의를 구하는 기능을 갖는다. 개인정보보호법 제 17 조(개인정보의 제공)에 의하면 제 3 자에게 개인정보를 제공할 경우에는 해당 사용자의 동의가 반드시 필요하다.

```
//Consent 모듈을 활성화
```

```
~# cd    /var/simplesamlphp/modules/consent
~# touch   enable

//config.php 파일의 수정
~# cd   /var/simplesamlphp/config
~# nano config.php

// 90번 배열의 주석 제거
90 => array(
        'class'    => 'consent:Consent',
        'store'    => 'consent:Cookie',
        'focus'    => 'yes',
        'checked'          => TRUE
),
```

[주의] IdP 서버에서 로그인했을 경우에는 Consent 화면이 보이지 않는다. SP 와 ID 연계한 후, SP 서버에서 로그인했을 경우 아래와 같이 Consent 화면이 출력된다.



제 6 장 메타데이터의 설정

6.1 절 제공기관 관련정보의 설정

IdP 의 메타데이터에 제공기관 관련정보를 설정한다. 제공해야 할 정보는 3 가지이다.

```
~# cd  /var/simplesamlphp/metadata
~# nano saml20-idp-hosted.php
  'OrganizationName' => array(
      'en' => '[my school]',
  ),
  'OrganizationDisplayName' => array(
      'en' => '[my school]',
  ),
  'OrganizationURL' => array(
      'en' => '[https://my.school.ac.kr/]',
  ),
```

6.2 절 oid 설정

KAFE 에서는 사용자 속성이름의 naming 방식으로 oid 포맷을 권장한다. 다음과 같이 ID 제공자의 메타데이터를 수정해 oid 포맷을 지원할 수 있도록 설정한다.

```
~# cd  /var/simplesamlphp/metadata
~# nano saml20-idp-hosted.php


 'attributes.NameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
 'authproc' => array(
            ...
            100 => array(
                    'class' => 'core:AttributeMap',
                    'name2oid',
            ),
            ...
 ),
```

6.3 절 메타데이터 signing

메타데이터의 무결성 검증을 위한 signing 을 한다.

```
~# nano /var/www/simplesamlphp/config/config.php
...
'metadata.sign.enable' => true,
...
```

6.4 절 Assertion 암호화

Assertion 메시지를 암호화한다.

```
~# nano /var/www/simplesamlphp/metadata/saml20-idp-hosted.php
...
'assertion.encryption' => true,
...
```

6.5 절 메타데이터 수집 자동화

Shibboleth 의 경우 SAML 2.0 Metadata XML Format 을 통한 자동화된 메타데이터 관리 설정이 가능하다. OpenConext 의 경우, 페더레이션에 속한 모든 엔터티들의 메타데이터(a SAML 2.0 document)를 다운로드할 수 있는 URL 을 제공한다.

SimpleSAMLphp 를 이용해 특정 URL 에서 메타데이터 문서를 다운로드하고 파싱하는 방법은 다음과 같다.

```
~# cd /var/simplesamlphp
~# touch modules/cron/enable
~# cp modules/cron/config-templates/*.php config/
~# touch modules/metarefresh/enable
~# cp modules/metarefresh/config-templates/*.php config/
```

https://fedinfo.kreonet.net/metadata/federation/KAFE-testfed/metadata.xml 에서 KAFE Test Fededration 메타데이터를 제공한다. 콘솔에서 다음 명령을 통해 metafresh 모듈을 검증할 수 있다. -s 는 콘솔에 결과를 출력하기 위한 옵션이다. 정상적으로 출력되는지 확인한다.

```
~# cd /var/simplesamlphp/modules/metarefresh/bin
./metarefresh.php -s https://fedinfo.kreonet.net/metadata/federation/KAFE-testfed/metadata.xml
```

cron 모듈을 환경설정한다.

```
~# cd /var/simplesamlphp/config
~# nano module_cron.php


$config = array (
    'key' => '충분히 길게 secret key를 설정',
    'allowed_tags' => array('daily', 'hourly', 'frequent'),
    'debug_message' => TRUE,
    'sendemail' => TRUE,
);
```

웹 브라우저로 http(s)://[Base URL]/simplesaml/module.php/cron/croninfo.php 에 접속한다. 화면에 출력되는 정보를 이용해 crontab 을 설정한다.


```
~# crontab -e
// 위 'Cron result page'의 # Run cron: [frequent] 앞 부분까지 총 4라인을 복사해 넣는다.
```

metarefresh 모듈을 설정한다.

```
~# cd /var/simplesamlphp/config
```

```
~# nano config-metarefresh.php
  $config = array('sets' => array(
     'kafe' => array(
          'cron' => array('hourly'),
        'sources' => array(
              array(
            'src' => 'https://fedinfo.kreonet.net/signedmetadata/federation/KAFE-testfed/metadata.xml',
              ),
          ), // 'sources'
          'expireAfter' => 60*60*24*4, // maximum 4 days cache time
          'outputDir' => 'metadata/metadata-kafe-test/',
          'outputFormat' => 'flatfile',
      ), // 'kafe'
    ));
```

메타데이터의 저장을 위해 디렉토리를 생성하고 권한을 설정한다.

```
~# cd /var/simplesamlphp/metadata
~# mkdir metadata-kafe-test
// Ubuntu에서는 아래와 같이
~# chown www-data metadata-kafe-test
// CentOS에서는 아래와 같이
~# chown apache.apache metadata-kafe-test
```

수집한 메타데이터를 이용할 수 있도록 simpleSAMLphp 를 환경설정 한다.

```
~# cd /var/simplesamlphp/config
~# nano config.php

 'metadata.sources' => array(
            array('type' => 'flatfile', 'directory' => 'metadata/metadata-kafe-test'),
            array('type' => 'flatfile'),
 ),
```

http(s)://[Base URL]/simplesaml/로 접속해서 'Federation' 탭을 선택하고 Tools 의 'Metarefresh: fetch metadata 를 클릭한다. 오류없이 다운로드 되는지 확인한다(metadata/metadata-kafe-test 디렉토리에 idp 와 sp 들이 포함되어 있는지 확인한다).

제 7 장 사용자인터페이스(Theme)의 변경

7.1 절 새로운 Theme 의 생성

Theme 은 로그인용 팝업화면의 GUI(Graphical User Interface)이다. Theme 생성을 위해 다음과 같이 디렉토리를 생성하고 파일을 복사한다.

```
~# cd /var/simplesamlphp/modules/dgist
~# mkdir -p ./themes/www/dgistidp/default/includes
~# cd /var/simplesamlphp/modules/dgist/themes/www/dgistidp/default/includes
~# cp /var/simplesamlphp/templates/includes/*.php ./
~# cd /var/simplesamlphp/modules/dgist/themes/www/dgistidp/
~# mkdir core
~# cd ./core
~# cp /var/simplesamlphp/modules/themefeidernd/themes/feidernd/core/loginuserpass.php ./
```

config.php 파일을 수정한다.

```
~# cd /var/simplesamlphp/config
~# nano config.php
// 'theme.use'를 다음과 같이 수정
  'theme.use' => 'dgist:dgistidp',
//참조; 디렉토리이름과 동일해야함(/var/simplesamlphp/modules/dgist/themes/dgistidp/core
```

스타일 파일(.css)을 복사하고 loginuserpass.php 파일을 수정한다.

```
~# cd /var/simplesamlphp/modules/dgist/www
~# cp /var/simplesamlphp/modules/themefeidernd/www/feidernd.css ./
~# cd /var/simplesamlphp/modules/dgist/themes/dgistidp/core
~# nano loginuserpass.php
// 'themefeidernd/feidernd.css'를 'dgist/feidernd.css'로 변경한다.
  <?php echo simpleSAML_Module::getModuleURL('dgist/feidernd.css'); ?>
```

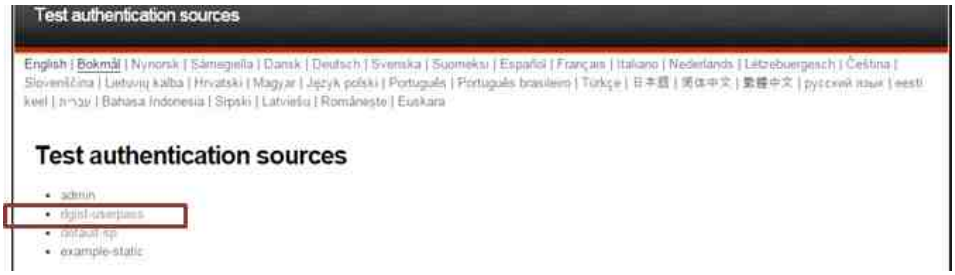
로고파일을 복사하고 loginuserpass.php 파일을 수정한다.

```
~# cd /var/simplesamlphp/modules/dgist/www
~# cp /var/simplesamlphp/modules/themefeidernd/www/ssplogo-fish-only-s.png ./
```

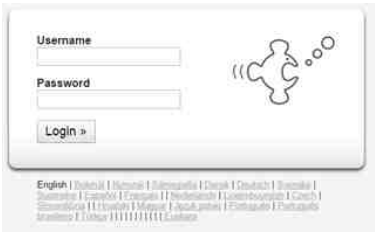
```
~# cd /var/simplesamlphp/modules/dgist/themes/dgistidp/core
~# nano loginuserpass.php
// 'themefeidernd/ssplogo-fish-only-s.png'를 'dgist/ssplogo-fish-only-s.png'로 변경한다.
  <?php echo simpleSAML_Module::getModuleURL('dgist/ssplogo-fish-only-s.png'); ?>
```

7.2 절 새로운 Theme 의 검증

http://[서버주소]/simplesaml 에 접속해 Test authentication sources 화면의 dgist-userpass 를 선택한다.



theme 이 정상적으로 적용되었으면 아래와 같은 화면이 출력된다.



7.3 절 KREONET IdP 용으로 Theme 변경하기

다음과 같은 KREONET IdP 용 GUI 로 Theme 을 변경할 수 있다.

Theme 변경을 위해 2 개의 파일(inthremes.tar.gz, inwww.tar.gz)이 필요하다. coreen@kreonet.net 에 요청해 KREONET IdP 용 Theme 을 확보한다.

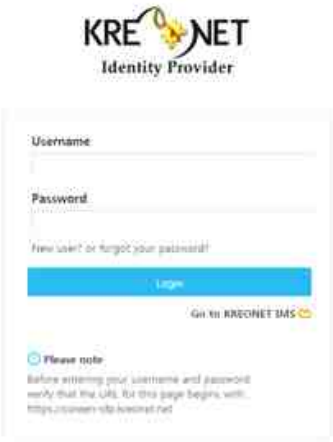
```
// inwww.tar.gz 파일이 /var/simplesamlphp/www에 존재하는 것으로 가정한다.
~# cd /var/simplesamlphp/www/
~# tar zxvf inwww.tar.gz

// inthemes.tar.gz 파일이 /var/simplesamlphp/modules/dgist/themes에 존재하는 것으로 가정한다.
~# cd /var/simplesamlphp/modules/dgist/themes
~# tar zxvf inthemes.tar.gz
~# mv dgistidp dgistidp_old
~# mv kreonet dgistidp
```

loginuserpass.php 파일을 수정한다.

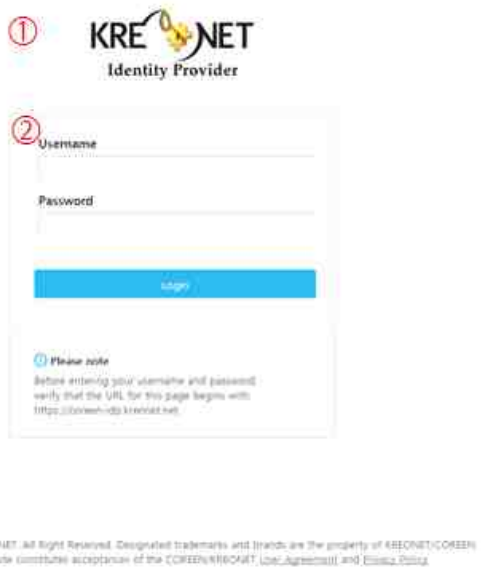
```
~# cd /var/simplesamlphp/modules/dgist/themes/dgistidp/core
~# nano loginuserpass.php
// 아래 적색 박스의 내용을 적절히 수정한다.
```

```
        echo htmlspecialchars($this->t('{errors:descr_' . $this->data['errorcode'] . '}', $thi
    span>
```

http://[IdP 서버주소]/simplesaml 에 접속해 Test authentication sources 화면의 dgist-userpass 를 선택한다.



정상적으로 설정되었으면 위와 같은 화면이 출력된다.



변경된 Theme 은 크게 ①기관 BI, ②로그인, ③Footer 로 구성된다. 개별요소의 변경을 원하면 다음 파일을 찾아 내용을 수정한다.

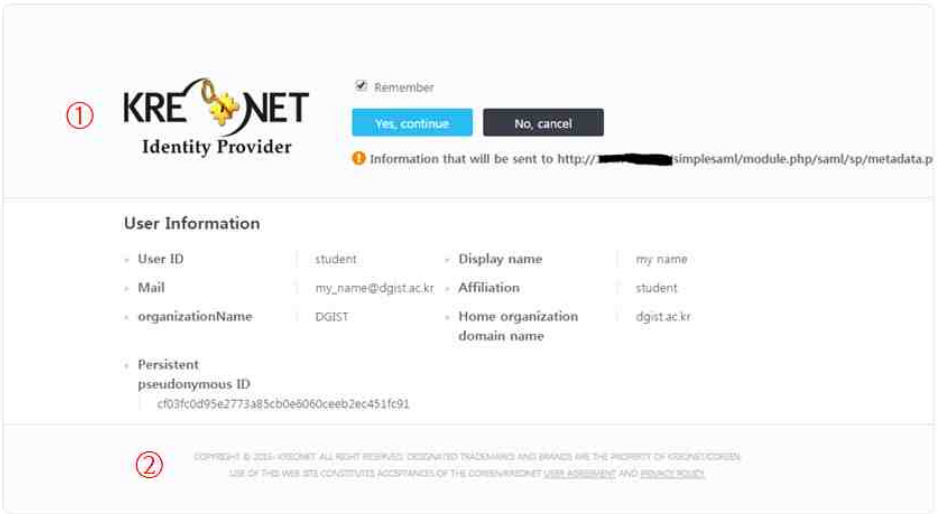
①기관 BI	이미지 위치	/var/simplesamlphp/www/images/korean/common/kreonet.gif
	css파일 위치	/var/simplesamlphp/www/css/korean/common.css (‘kreonet.gif‘가 위치한 라인의 내용 편경)
②로그인	/var/simplesamlphp/modules/dgist/themes/dgistidp/core/loginuserpass.php	
③Footer	/var/simplesamlphp/www/user_layout_footer.php	

제 8 장 Consent 화면의 변경

KREONET IdP 용 Consent 를 적용하기 위해서는 inconsent.tar.gz 파일이 필요하다. coreen@kreonet.net 에 요청해 해당 파일을 확보한다.

```
// inconsent.tar.gz 파일이 /var/simplesamlphp/modules/consent에 존재하는 것으로 가정
~# cd /var/simplesamlphp/modules/consent
~# mv templates templates_old
~# tar zxvf inconsent.tar.gz
```

Consent 화면의 변경여부를 확인하기 위해서 http://[SP 서버주소]/simplesaml 에 접속해서 로그인을 수행한다. 해당 SP 는 IdP 와 ID 연계되어 있어야 한다. http://[IdP 서버주소]/simplesaml 을 이용하면 적용된 Consent 화면이 나타나지 않는다. KREONET IdP 용 Consent 가 정상적으로 적용되면 아래 그림과 같은 화면이 출력된다.



①기관 BI 및 ②Footer 의 변경을 위해 다음 표에 명시된 파일을 수정한다.

①기관 BI	이미지 위치	/var/simplesamlphp/www/images/korean/common/login_kreonet.gif
	css파일 위치	/var/simplesamlphp/www/css/korean/common.css ('login_kreonet.gif'가 위치한 라인의 내용 편경)
②Footer	/var/simplesamlphp/www/user_layout_footer.php	

제 9 장 보안 설정

9.1 절 페이지 접근제어

simplesamlphp 를 설치하면 일반사용자가 웹 브라우저를 통해 IdP 또는 SP 의 metadata, phpinfo 등 보안정보에 접근할 가능성이 있다. simplesamlphp 를 설치한 후 해당 정보들을 은닉하기 위해 config.php 파일을 수정해야 한다. default themes 의 userloginpass.php 파일을 수정해 사용자 인터페이스를 변경할 수 있다.

```
~# clear
~# cd /var/simplesamlphp/config/config.php
//아래와 같이 protectindexpage와 protectmetadata 값을 true로 변경한다.
 'admin.protectindexpage' => true,
 'admin.protectmetadata' => true,
```

SSP 의 관리자페이지 노출 취약점을 해결하기 위해 apache 설정을 변경해 줘야 한다. CentOS 6.5 기준으로 IdP 가 SSL 이 적용되어 있을 때 다음과 같이 설정한다.

```
~# clear
~# cd /etc/httpd/conf.d
~# nano ssl.conf
//</VirtualHost> 앞에 다음과 같이 추가한다. 설치환경에 맞게 적절히 수정되어야 한다.
ReWriteEngine On
RewriteCond %{REQUEST_URI} ^/simplesaml/module.php/core/loginuserpass.php [NC]
RewriteCond %{QUERY_STRING} ^AuthState=(.*)as_login(.*)AuthId(.*)admin(.*) [NC]
// 접속가능한 IP의 주소를 설정한다 (아래 예; 192.168.0.*)
RewriteCond %{REMOTE_ADDR} !^192\.168\.0\.[0-9] +
RewriteCond ^ - [F]
```

9.2 절 SSP 의 보안 강화 사항

showerrors 항목을 false 로 해서 오류가 발생했을 때 노출되지 않아야 할 오류정보(stacktrace 는 시스템 정보를 노출함)가 사이트에 노출되는 것을 방지한다. 또한, admin 비밀번호를 설정해 SSP 의 정보가 노출되지 않도록 한다.

```
~# nano /var/simplesamlphp/config/config.php
//5.1과 동일
 'admin.protectindexpage' => true,
//추가 또는 수정
 'showerrors' => false,
```

쿠키 보안을 위해, 평문 연결(plain text connection, Non-TLS)일 때 쿠키 정보가 전송되는 것을 막고 자바스크립트가 쿠키에 접근하는 것을 막아야 한다. TLS(https connection)를 반드시 이용해야 한다. 쿠키 보안 설정을 하지 않으면 Cross Site Scripting 공격에 취약할 수 있다.

```
~# nano /var/simplesamlphp/config/config.php
// 다음과 같이 수정
 'session.cookie.secure' => true,
 'session.phpsession.httponly' => true,
```

SSP 가 'redirect'를 할 도메인 이름을 설정한다. 다음과 같이 empty array 로 설정하면 SSP 가 자동으로 신뢰하는 도메인으로만 redirect 한다.

```
~# nano /var/simplesamlphp/config/config.php
// 다음과 같이 수정
 'trusted.url.domains' => array(),
```

SHA1(보안취약) 대신 SHA-256 으로 이용한다. ID 제공자는 saml20-idp-hosted.php 를, SP 제공자는 authsources.php 를 수정한다. SSP 1.12 이상의 버전에서는 config 파일에 반영되어 있으므로 주석표시만 제거한다.

```
// ID 제공자일 경우에 해당
~# nano /var/simplesamlphp/metadata/saml20-idp-hosted.php
// 다음과 같이 주석제거
'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
```

제 10 장 기타 설정

10.1 절 Metadata 의 EntityID 변경

KAFE 회원기관은 EntityID 를 다음과 같은 방법으로 설정해야 한다. IdP 또는 SP 의 URL 이 https://z.ac.kr 이며 SAML 소프트웨어로 simplesamlphp 를 이용한다고 가정한다.

ID 제공자: https://z.ac.kr/idp/simplesamlphp

서비스 제공자: https://z.ac.kr/sp/simplesamlphp

~# cd /var/simplesamlphp/metadata

~# nano saml20-idp-hosted.php

// 아래 적색박스부분에 EntityID를 기록한다.

```
 GNU nano 2.0.9                          File: saml20-idp-hosted.php

<?php
/**
 * SAML 2.0 IdP configuration for simpleSAMLphp.
 *
 * See: https://simplesamlphp.org/docs/stable/simplesamlphp-reference-idp-hosted
 */

$metadata['https://[redacted]/idp/simplesamlphp'] = array(
        /*
         * The hostname of the server (VHOST) that will use this SAML entity.
         *
         * Can be '__DEFAULT__', to use this entry by default.
         */
        'host' => '__DEFAULT__',

        /* X.509 key and certificate. Relative to the cert directory. */
        'privatekey' => '/var/simplesamlphp/cert/test.kreonet.net.pem',
        'certificate' => '/var/simplesamlphp/cert/test.kreonet.net.crt',
```

10.2 절 기타

Single Logout	메타데이터의 수정	metadata/saml20-idp-hosted.php	'logouttype' => 'iframe'
	레이아웃 수정	simplesamlphp/modules/core/templates/logout-iframe.php	

KAFE

서비스 제공자의 구축

– Korean Access Federation (KAFE) –

한국과학기술정보연구원

목 차

SAML 서비스 제공자 시스템의 구축

2015. 09. 09. – 초안
2015.11.10.(draft v0.16) – 최종 갱신

제 1 장 설치 환경

본 설치매뉴얼은 simpleSAMLphp 1.13 버전을 이용해 Ubuntu 또는 CentOS 환경에서 SAML 2.0 SP(Service Provider)를 구축하는 방법을 기술한다. SP 의 구축을 위해 다음과 같은 요구조건이 충족되어야 한다.

– LAMP 스택의 설치: Apache, MySQL, PHP 5.3 이상

– 공인인증서(SSL)의 설치

제 2 장 simpleSAMLphp 의 설치

2.1 절 simpleSAMLphp

simpleSAMLphp 는 UNINETT 에서 개발한 SAML v2.0 소프트웨어이다. IdP(Identity Provider) 또는 SP(Service Provider)로 설치가능하며 최신 버전은 1.13.2 이다(2015 년 9 월). 이하 IdP 또는 SP 는 SAML 2.0 IdP 또는 SAML 2.0 SP 를 의미한다. simplesamlphp.org 에서 추가적인 정보를 얻을 수 있다.

2.2 절 simpleSAMLphp 의 설치 환경

simpleSAMLphp 의 설치를 위한 서버 환경은 다음과 같다. 특별한 언급이 없는 한 서비스 제공자용 서버는 Ubuntu 14.04 LTS(64 비트)를 이용한다.

– php, MySQL, httpd 가 설치

– IPv6 disable 권장

– selinux disable

– Linux 방화벽(iptables) 80/443(http/https) 포트 개방

– 시간 동기화를 위한 NTP 설정(NTP 서버: time.kriss.re.kr)

2.3 절 simpleSAMLphp 의 설치

simpleSAMLphp 의 구동을 위해 요구되는 소프트웨어 패키지를 설치한다. simpleSAMLphp 의 설치 경로는 /var/simplesamlphp 로 가정한다.

```
~# clear
```

```
~# sudo apt-get install php-date openssl php5-mcrypt
// 인증 소스로 LDAP를 이용하는 경우
~# sudo apt-get install php5-ldap

//simplesamlphp 다운로드
~#sudo wget https://simplesamlphp.org/res/downloads/simplesamlphp-1.13.2.tar.gz

//압축해제 및 설치
~# sudo cp ./simplesamlphp-1.13.2.tar.gz    /var/
~# sudo cd   /var
~# sudo tar zxvf ./simplesamlphp-1.13.2.tar.gz
~# sudo mv ./simplesamlphp-1.13.2 ./simplesamlphp
```

※ CentOS 6.7 (php 5.5.30)에서 mcrypt 설치방법

```
//기존 php 5.5.30을 모두 지웠다고 가정한다.
~# rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
~# yum install php56w php56w-opcache php56w-mcrypt php56w-xml php56w-mysql
~# service httpd restart
```

2.4 절 Apache 서버 설정

아래 설정 방법은 HTTP(80 포트)에 대한 환경설정을 보여준다. HTTPS(443)에 대한 Apache 환경설정 방법은 생략한다.

※ ID 제공자 서버는 반드시 공인인증서를 설치하고 HTTPS(443)을 이용해야 한다.

```
~# sudo cd /etc/apache2/sites-available
~# sudo   nano   000-default.conf
// <VirtualHost *:80>을 찾아 아래와 같이  수정
<VirtualHost *:80>
        DocumentRoot    /var/www/html/
        Alias     /simplesaml       /var/simplesamlphp/www

~# sudo nano /etc/apache2/apache2.conf
// 다음과 같은 항목을 추가
<Directory /var/simplesamlphp/>
```

```
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

~# sudo service apache2 restart
```

2.5 절 simpleSAMLphp 의 구동환경 설정

아래와 같이 simpleSAMLphp 를 환경설정한다. 'secretsalt'는 다음 명령을 이용해 추출할 수 있다.

tr –c –d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32 count=1 2>/dev/null;echo

```
~# sudo cd  /var/simplesamlphp/config
~# sudo nano config.php
// 아래와 같이 환경설정 함
'baseurlpath' => 'simplesaml/',
'certdir' => 'cert/'
'loggingdir'        => 'log/',
'datadir' => 'data/',

// 다음 사항은 꼭 수정해야 함
'auth.adminpassword'      => '[관리용 패스워드 입력]',
'secretsalt' => '[secret salt 입력]',
'technicalcontact_name' => '[관리자 이름]',
'technicalcontact_email' => '[관리자 이메일]',
'language.default' => 'en',
'timezone' => 'Asia/Seoul',
```

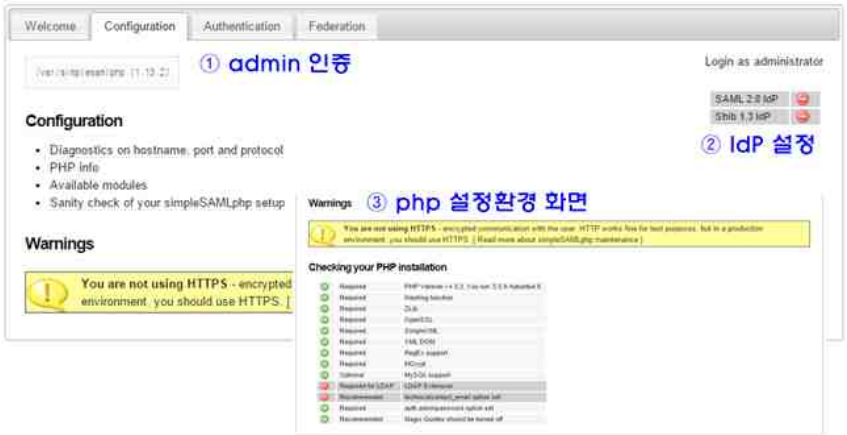
simpleSAMLphp 에서 제공하는 특정 모듈을 활성화하고 싶다면 [모듈명] 디렉토리에서 enable 파일을 생성한다. 다음 예는 LDAP 모듈을 활성화하기 위한 방법이다.

```
~# sudo  cd  /var/simplesamlphp/modules
~# sudo  cd  ./ldap
~# sudo  touch  enable
```

환경 설정이 완료되었다면 웹 브라우저를 이용해 http://[서버주소]/simplesaml 을 접속한다. 정상적으로 설치되었다면 아래와 같은 화면이 나타난다.

2.6 절 설정된 환경의 검증

Authentication 탭에서 admin 으로 로그인 한다. 현재 IdP 가 활성화되지 않은 상태이므로 ②와 같이 적색원이 보여야 한다. admin 으로 로그인한 후 ③과 같이 표시된다면 정상적으로 설치된 상태이다. 관리자 이메일과 LDAP Extension 을 설치했다면 모두 녹색원으로 표시되어야 한다.

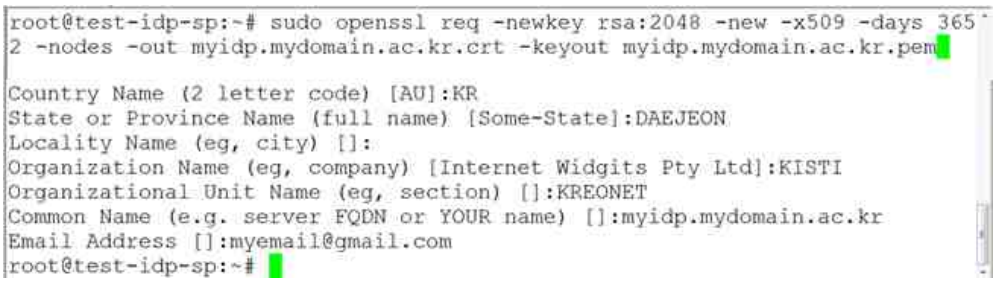


제 3 장 SAML 서비스 제공자의 설치

3.1 절 SSL 자가 인증서의 생성

※ 상용 CA(Certication Authority)에서 발급한 인증서는 SAML 호환성 문제를 야기할 수 있으므로 Self-signed certificate 를 이용한다.

아래 [myidp.mydomain.ac.kr]은 SP 구축자의 기관 환경에 맞게 적절히 변경해야 한다. Common Name (e.g., server FQDN or YOUR name)은 SP 용 서버의 IP 주소 또는 도메인명으로 설정해야 한다.



자가 인증서를 생성한 후 환경설정을 계속한다.

```
~# sudo   cd    /var/simplesamlphp
~# sudo   mkdir cert
// 생성된 .cert 파일과 .pem 파일은 /var/simplesamlphp/cert 디렉토리로 이동
```

SP 의 메타데이터를 수정해 자가 인증서를 등록한다.

```
~# sudo   cd    /var/simplesamlphp/config
~# sudo   nano   authsources.php

// 'default-sp' => array에 아래와 같이 수정해 인증서를 등록한다.
'privatekey'        => '/var/simplesamlphp/cert/[인증서이름].pem',
'certificate'       => '/var/simplesamlphp/cert/[인증서이름].crt',
```

3.2 절 메타데이터 설정 및 IdP 메타데이터의 등록

SP 의 entityID 를 설정한다. entityID 는 http(s)://domain_name/sp/saml_software 의 형식에 따른다.

```
~# sudo   cd    /var/simplesamlphp/config
~# sudo   nano   authsources.php
```

```
// 'default-sp' => array에 아래와 같이 entityID 값을 변경한다.
'entityID'          => 'https://myidp.mydomain.ac.kr/sp/simplesamlphp',
```

SP 와 IdP 를 ID 연계하기 위해서, IdP 와 SP 간 각각의 메타데이터를 교차 등록해야 한다. SP 에 IdP 의 메타데이터를 등록하는 방법은 다음과 같다. IdP 의 메타데이터 파일을 확보하고 있다고 가정한다.

```
~# sudo  cd   /var/simplesamlphp/metadata
~# sudo  nano  saml20-idp-remote.php

// 아래 그림(예시)처럼 IdP의 메타데이터를 추가한다.
```

※ simpleSAMLphp 기반의 IdP 를 보유하고 있다고 가정했을 때, http://[IdP 서버주소]/simplesaml 에 접근해 Federation 탭을 클릭하면 IdP 의 메타데이터 정보를 확인할 수 있다.

위 그림처럼 SP 에 IdP 의 메타데이터를 등록하고 등록된 메타데이터에 이름 'name'을 추가한다. 등록된 IdP 의 메타데이터에 'name'이 이미 등록되어 있는 경우에는 'name' 등록을 생략한다. 'name'의 내용은 사용자가 SP 에서 IdP 를 선택할 때(IdP discovery), 목록의 형태로 보여진다.

```
~# sudo  cd   /var/simplesamlphp/metadata
~# sudo  nano  saml20-idp-remote.php

// 아래 그림(예시)처럼 IdP의 메타데이터를 추가한다.
$metadata['https://[IdP 주소]/idp/simplesamlphp'] = array(
  'name' => array(
          'en' => '[IdP의 이름]',
  ),
```

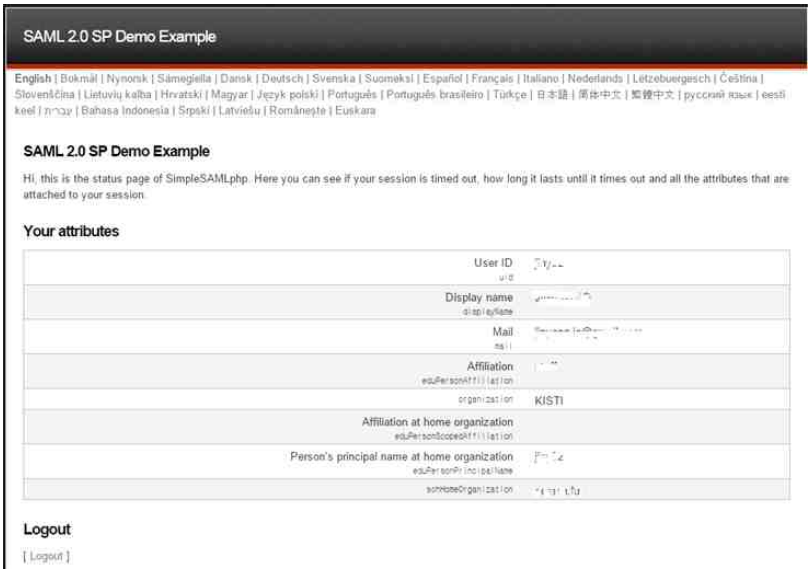
SP 의 메타데이터도 IdP 에 등록되어야 SP-IdP 간 SAML 통신이 가능하다. 아래 그림처럼 http://[SP 서버주소]/simplesaml 에 접근해 Federation 탭을 클릭하면 SP 의 메타데이터 정보를 확인할 수 있다.

SP 의 메타데이터 정보를 복사해서 IdP 에 메타데이터 등록을 요청해야 한다. IdP 의 /var/simplesamlphp/metadata/saml20-sp-remote.php 파일에 SP 의 메타데이터를 등록할 수 있다. simpleSAMLphp 는 평문(flat format) 형태의 메타데이터와 XML 형태의 메타데이터를 함께 제공한다. ID 연계되는 상대 IdP 또는 SP 에 자신의 메타데이터를 등록할 때는 평문 형태의 메타데이터를 이용한다.

SP 에 IdP 의 메타데이터가 등록되었다면 아래 그림과 같이 http://[SP 의 주소]/simplesaml 에 접속해서 Authentication → Test authentication sources 의 'default-sp'를 클릭한다.

아래 그림과 같이 Select your identity provider 에서 saml20-idp-remote.php 의 'name'으로 등록한 이름을 클릭한다. 본 예시에서는 'name'을 'Coreen IdP -guest users'로 설정했다.

IdP 에 등록된 사용자 ID 와 비밀번호를 이용해 로그인하면 아래 그림과 같이 IdP 가 제공하는 사용자 속성정보를 확인할 수 있다.



제 4 장 메타데이터의 설정

4.1 절 oid to name 변환

SAML 2.0 는 Attribute 이름의 oid 표기를 권장한다. ID 제공자가 oid 형태(e.g.,urn:oid:2.5.4.4)의 속성이름을 제공하고 서비스 제공자가 friendly name(e.g., sn) 을 이용해 사용자를 인가한다면 oid 를 friendly name 으로 변경해야 한다. KAFE 는 oid 표기법을 권장하므로 SP 에서 다음과 같이 oid2name 변환을 한다.

```
~# cd    /var/simplesamlphp/config
~# nano   config.php

// 'authproc.sp' => array( 가 포함된 라인을 찾아 다음과 같이 추가한다.
50 => array(
      'class' => 'core:AttributeMap',
      'oid2name',
      // 서비스 제공자에서 사용자 인가를 위해 friendly name 'o' 대신에 friendly name 'organizationName'
         을 이용한다면 다음 라인을 추가한다.
      'o'   => 'organizationName',
),
```

4.2 절 제공 기관 등록

서비스 제공자는 OrganizationName, OrganizationDisplayName, OrganizationURL 을 메타데이터에 등록해야 한다.

```
~# cd    /var/simplesamlphp/config
~# nano   authsources.php

// 'default-sp' => array( 가 포함된 array에 다음을 추가한다.
  'OrganizationName' => array(
      'en' => '[my school]',
  ),
  'OrganziationDisplayName' => array(
      'en' => '[my school]',
  ),
```

```
'OrganizationURL' => array(
    'en' => '[https://my.school.ac.kr/]',
),
```

4.3 절 Metadata Signing

config.php 파일을 열어 다음과 같이 수정한다.

```
~# cd    /var/simplesamlphp/config
~# nano   config.php


'metadata.sign.enable' => true,
```

4.4 절 메타데이터 수집 자동화

Shibboleth 의 경우 SAML 2.0 Metadata XML Format 을 통한 자동화된 메타데이터 관리 설정이 가능하다. OpenConext 의 경우, 페더레이션에 속한 모든 엔터티들의 메타데이터(a SAML 2.0 document)를 다운로드할 수 있는 URL 을 제공한다.

SimpleSAMLphp 를 이용해 특정 URL 에서 메타데이터 문서를 다운로드하고 파싱하는 방법은 다음과 같다.

```
~# cd /var/simplesamlphp
~# touch modules/cron/enable
~# cp modules/cron/config-templates/*.php config/
~# touch modules/metarefresh/enable
~# cp modules/metarefresh/config-templates/*.php config/
```

https://fedinfo.kreonet.net/metadata/federation/KAFE-testfed/metadata.xml 에서 KAFE Test Fededration 메타데이터를 제공한다. 콘솔에서 다음 명령을 통해 metafresh 모듈을 검증할 수 있다. -s 는 콘솔에 결과를 출력하기 위한 옵션이다. 오류가 발생하는지 확인한다.

```
~# cd /var/simplesamlphp/modules/metarefresh/bin
./metarefresh.php -s https://fedinfo.kreonet.net/metadata/federation/KAFE-testfed/metadata.xml
```

cron 모듈을 환경설정한다.

```
~# cd /var/simplesamlphp/config
~# nano module_cron.php
```

```
$config = array (
    'key' => '충분히 길게 secret key를 설정',
    'allowed_tags' => array('daily', 'hourly', 'frequent'),
    'debug_message' => TRUE,
    'sendemail' => TRUE,
);
```

웹 브라우저로 http(s)://[Base URL]/simplesaml/module.php/cron/croninfo.php 에 접속한다. 화면에 출력되는 정보를 이용해 crontab 을 설정한다.


```
~# crontab -e
// 위 'Cron result page'의 # Run cron: [frequent] 앞 부분까지 총 4라인을 복사해 넣는다.
```

metarefresh 모듈을 설정한다.

```
~# cd /var/simplesamlphp/config
~# nano config-metarefresh.php
  $config = array('sets' => array(
      'kafe' => array(
          'cron' => array('hourly'),
```

```
        'sources' => array(
            array(
              'src' => 'https://fedinfo.kreonet.net/signedmetadata/federation/KAFE-testfed/metadata.xml',
            ),
          ), // 'sources'
          'expireAfter' => 60*60*24*4, // maximum 4 days cache time
          'outputDir' => 'metadata/metadata-kafe-test/',
          'outputFormat' => 'flatfile',
      ), // 'kafe'
    ));
```

메타데이터의 저장을 위해 디렉토리를 생성하고 권한을 설정한다.

```
~# cd /var/simplesamlphp/metadata
~# mkdir metadata-kafe-test

// Ubuntu에서는 아래와 같이
~# chown www-data metadata-kafe-test
// CentOS에서는 아래와 같이
~# chown apache.apache metadata-kafe-test
```

수집한 메타데이터를 이용할 수 있도록 simpleSAMLphp 를 환경설정 한다.

```
~# cd /var/simplesamlphp/config
~# nano config.php

 'metadata.sources' => array(
            array('type' => 'flatfile', 'directory' => 'metadata/metadata-kafe-test'),
            array('type' => 'flatfile'),
 ),
```

http(s)://[Base URL]/simplesaml/로 접속해서 'Federation' 탭을 선택하고 Tools 의 'Metarefresh: fetch metadata 를 클릭한다. 오류없이 다운로드 되는지 확인한다(metadata/metadata-kafe-test 디렉토리에 idp 와 sp 들이 포함되어 있는지 확인한다).


Shib 1.3 IdP Metadata (Trusted)
Tools
Delete my choices of IdP in the IdP discovery services
XML to simpleSAMLphp metadata converter
Metarefresh: fetch metadata
Lookup metadata
Look up metadata for entity:
SAML 2.0 IdP Metadata
Lookup

제 5 장 웹 응용과 SAML SP 의 연동

지금부터는 웹 응용과 SAML SP 를 연동하는 방법에 대해서 기술한다.

5.1 절 연동 시 고려 사항

웹 응용은 simpleSAMLphp 가 제공하는 API(Application Programming Interface)를 이용해 사용자를 인가(Authorization)해야 한다. KAFE(Korean Access Federation)에서는 아래 표와 같은 속성들의 이용을 권장하고 있다.

제공 속성	설명	개인정보 가능성
uid	사용자 ID(시험 서비스 기간에 한시적 적용)	O
eduPersonTargetedID	서비스 제공자 별 암호화된 사용자 고유번호	
sn	성	O
givenName	이름	O
displayName	사용자의 화면표시 이름	O
mail	사용자 이메일 주소	O
eduPersonAffilation	사용자의 기관내 직무정보	
organizationName	사용자의 소속기관명	
schacHomeOrganization	사용자 소속기관의 최상위 도메인 이름	
eduPersonPrincipalName	도메인 내 사용자 ID 정보	O
eduPersonScopedAffilation	도메인 내 사용자 직무 정보	

IdP 가 제공하는 속성 정보는 위 표에 명시된 속성들 보다 확장될 수 있다. 웹 응용이 SAML SP 연동 시 고려해야 할 사항은 다음과 같다.

- 사용자를 구분하는 방법; SP 는 사용자를 구분할 수 있는(또는 사용자 충돌을 피할 수 있는) 방법을 준비해야 한다. 다수의 IdP 에 동일한 사용자 식별자값(예; uid 등)이 존재할 가능성이 있다. SP 는 다수의 사용자 속성 또는 메타데이터를 통해 얻은 값들을 이용해 사용자를 구분할 수 있어야 한다.

5.2 절 사용자 인증 및 인가 관련 웹 응용 코드

simpleSAMLphp 가 /var/simplesamlphp/에 설치되어 있고 SP 로 동작한다고 가정한다. 또한 웹 응용의 root 디렉토리가 /var/www/html 이며 php 구동을 위한 Apache 환경설정이 완료되어 있다고 가정한다. IdP 에 등록된 계정은 student/student1234, faculty/faculty1234 이다.

다음은 웹 응용의 skeleton code 이다. 로그인한 사용자의 속성정보 및 사용자 인증을 수행한 IdP 의 정보를 배열의 형태로 얻게 된다.

```
~# cd /var/www/html
~# nano index.php

// 다음과 같이 추가
<?php
        include_once('/var/simplesamlphp/lib/_autoload.php');
        $as = new SimpleSAML_Auth_Simple('default-sp');
        $as->requireAuth();

        $attributes = $as->getAttributes();
        print_r($attributes);

        $idp = $as->getAuthData('saml:sp:IdP');
        print_r($idp);
?>
```

SP 가 IdP 와 연계되어 있고 각각의 메타데이터가 교차 등록되었을 경우, http://[SP 의 주소]/로 접속하면 아래와 같이 ID 제공자를 선택하는 화면이 나타난다.

사용자가 로그인에 성공할 경우 아래와 같이 사용자 속성이름과 속성값들이 배열형태로 리턴 된다.

```
// 로그인에 성공한 후 결과값 출력 예시
Array( [uid] => Array( [0] => student ) [displayName] => Array( [0] => my name) ..... )
https://[IdP의 주소]/idp/simplesamlphp
```

다음은 권한부여(또는 인가)를 위한 예제 코드이다. ID 제공자가 전달한 속성 정보 중 eduPersonAffilation 이 'faculty', organizationName 이 'DGIST', eduPersonPrincipalName 이 'faculty@coreen.kr'일 경우에 관리자 권한을 갖는 예제이다. 사용자 인가에 사용할 속성은 서비스 상황에 맞게 선택할 수 있다.

```
~# cd /var/www/html
~# nano index.php
```

```
// 다음과 같이 변경
<?php
        include_once('/var/simplesamlphp/lib/_autoload.php');
        $as = new SimpleSAML_Auth_Simple('default-sp');
        $as->requireAuth();

        $attributes = $as->getAttributes();
        $idp = $as->getAuthData('saml:sp:IdP');

        $uid = $attributes['uid'][0];
        $displayName = $attributes['displayName'][0];
        $mail = $attributes['mail'][0];
        $eduPersonAffilation = $attributes['eduPersonAffiliation'][0];
        $organizationName = $attributes['organizationName'][0];
        $schacHomeOrganization = $attributes['schacHomeOrganization'][0];
        $eduPersonPrincipalName = $attributes['eduPersonPrincipalName'][0];
        $eduPersonTargetedID = $attributers['eduPersonTargetedID'][0];

        //authorization
        if ($eduPersonAffilation === 'faculty' && $organizationName === 'DGIST' &&
$eduPersonPrincipalName === 'faculty@coreen.kr'){
                $isAdmin = 1;
        }else{
                $isAdmin = 0;
        }

        if($isAdmin){
                echo "Welcome Prof. ".$displayName."!!<br>";
                echo "You are allowed to access IT resource 1, 2, and 3.";
        }else{
                echo "Welcome Student ".$displayName."!!<br>";
```

```
                echo "You are allowed to access IT resource 1 only.";
        }
?>
```

5.3 절 로그인 및 로그아웃

다음은 SURFnet 에서 제공하는 simpleSAMLphp 용 SP 의 예제 코드이다. ID 연계를 위한 SAML 메시지 중개시스템인 OpenConext 와 연동해 보다 강력한 사용자 인증을 수행하기 위한 코드이다. SURFnet 의 ID 연계구조는 Hub&spoke 이지만 KREONET 의 KAFE 는 Full mesh 구조를 갖기 때문에 메시지 중개시스템을 이용하지 않는다. 아래 코드의 LOA(Level Of Assurance) 관련 부분은 KAFE 에서 사용되지 않는다.

```
//source code provided by SURFnet
//https://wiki.surfnet.nl/display/SUAAS/Configuring+a+simpleSAMLphp+SP+for+step-up+authentication


<?php
// Include SimpleSAMLphp. Assume this script is placed in the <simplesaml>/www dir.
require_once('../lib/_autoload.php');

// Name of session variable for storing the min required LOA(Level of Assurance) for a login
define( 'SSP_SESSION_MIN_LOA', 'RequestedMinLOA' );

// Build return URL. This is where ask simplesamlPHP to direct the browser to after login or logout
// Point to this script, but without any request parameters so we won't trigger an login again (and again, and again, and ...)
$returnURL = ($_SERVER['HTTPS'] == 'on') ? 'https://' : 'http://';
$returnURL .= $_SERVER['HTTP_HOST'];
$returnURL .= $_SERVER['SCRIPT_NAME'];

// Map integer level of assurance level to identifier used by the gateway
$gLOAmap = array(
    1 => 'http://suaas.example.com/assurance/loa1',
    2 => 'http://suaas.example.com/assurance/loa2',
    3 => 'http://suaas.example.com/assurance/loa3',
);
```

```
try {
    // Init SP instance
    // Assumes you have setup a SP named "default-sp" in <simplesaml>/config/authsources.php
    // See: https://simplesamlphp.org/docs/stable/simplesamlphp-sp
    $as = new SimpleSAML_Auth_Simple('default-sp');      // Init SP instance

    /** @var $session SimpleSAML_Session */
    $session = SimpleSAML_Session::getInstance();

    // Process login action. Assumes the login function of your SP uses ...?action=login
    if (isset($_REQUEST['action']) && $_REQUEST['action'] == 'login' ) {
        // We use the SSP session to keep track of the LOA we want.
        // Unset any existing RequiredAuthnContextClassRef
        $session->deleteData('string', SSP_SESSION_MIN_LOA);

        // login
        $requiredLOA = 2; // The LOA we want.

        // Store the requested LOA in the session so we can verify it later
        $session->setData('string', SSP_SESSION_MIN_LOA, $requiredLOA);

        $as->login( array(
            'ReturnTo' => $returnURL,
            'ForceAuthn' => false,
            'saml:AuthnContextClassRef' => $gLOAmap[$requiredLOA]   // Specify LOA
        ) );
        exit;    // Never reached. Added for clarity
    }
    // Process logout action
    if( isset($_REQUEST['action']) && $_REQUEST['action'] == 'logout' ) {
        $as->logout( array (
```

```
            'ReturnTo' => $returnURL,
        ) );   // Process logout


        exit;    // Never reached. Added for clarity
    }


    // Display HTML page
    echo <<<head
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
        <head>
            <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
            <style type="text/css">
                table,th,td {border: 1px solid black;}
                th,td {padding 1px}
            </style>
            <title>simpleSAMLphp Demo</title>
        </head>
        <body>
            <h1>SimpleSAMLphp LOA Demo</h1>
head;


    // Show some info when authenticated
    if ( $as->isAuthenticated() ) {
        $attributes = $as->getAttributes();
        $requestedLoA = $session->getData('string', SSP_SESSION_MIN_LOA);    // What we requested
during login
        $authState = $session->getAuthState();
        $authnConext = $authState['saml:sp:AuthnContext'];
        $nameID = $session->getNameID();
        $authnInstant = gmdate('r', $authState['AuthnInstant'] );
```

```
$expire = gmdate('r', $authState['Expire'] );

echo "<h2>You are logged in</h2>";

echo "<h3>SimpleSAMLphp Session</h3>";
echo "<p>SimpleSAMLphp session start: <b>{$authnInstant}</b></br />";
echo "SimpleSAMLphp session expire: <b>{$expire}</b></p>";

echo "<h3>LOA</h3>";
echo "<p>Received authnConext: <b>{$authnConext}</b></p>";

// Map LoA identifier back to integer LoA level
$actualLoA = array_search($authnConext, $gLOAmap);
if (false ==! $actualLoA)
    echo "<p>Actual LoA is: <b>{$actualLoA}</b></p>";
else
    $actualLoA = -1;

if (NULL !== $requestedLoA) {
    echo "<p>Requested LoA was: <b>{$requestedLoA}</b></p>";
    if ($actualLoA >= $requestedLoA)
        echo '<p><b>You were authenticated at or above the minimally required LoA</b></p>';
    else
        echo '<p><b>You were NOT authenticated at the required LoA</b></p>';
}

echo <<<html
<h3>NameID</h3>
<table>
    <tr><th>Value</th><td>{$nameID['Value']}</td></tr>
    <tr><th>Format</th><td>{$nameID['Format']}</td></tr>
</table>
```

```
html;

        echo <<<html
        <h3>SAML Attributes</h3>
        <table>
            <tr><th>Attribute</th><th>Value(s)</th></tr>
html;
        foreach ($attributes as $attrName => $attrVal) {
            echo "<tr><td>{$attrName}</td><td>";
            if (is_array($attrVal))
                echo implode('<br />', $attrVal);
            else
                echo $attrVal;
            echo "</td>";
        }
        echo <<<html
        </table>

        <h3>Logout</h3>
        <p>
            <form name="logout" action="{$returnURL}" method="get">
               <input type="hidden" name="action" value="logout"/>
               <input type="submit" value="Logout" />
            </form>
        </p>
html;
    } else {
        echo <<<html
            <h2>Your are not logged in</h2>
html;
    }
```

```
    echo <<<html
            <h3>Login (again)</h3>
            <p>
                <form name="login" action="{$returnURL}" method="get">
                    <input type="hidden" name="action" value="login"/>
                    <input type="submit" value="Login" />
                </form>
            </p>
html;
    echo <<<html
        </body>
    </html>
html;
}
catch (Exception $e)
{
    echo $e->getFile().':'.$e->getLine().' : '.$e->getMessage();
}
```

제 6 장 보안 및 개인정보

6.1 절 페이지 접근제어

simplesamlphp 를 설치하면 일반사용자가 웹 브라우저를 통해 IdP 또는 SP 의 metadata, phpinfo 등 보안정보에 접근할 가능성이 있다. simplesamlphp 를 설치한 후 해당 정보들을 은닉하기 위해 config.php 파일을 수정해야 한다. default themes 의 userloginpass.php 파일을 수정해 사용자 인터페이스를 변경할 수 있다.

```
~# clear
~# cd /var/simplesamlphp/config/config.php
//아래와 같이 protectindexpage와 protectmetadata 값을 true로 변경한다.
 'admin.protectindexpage' => true,
 'admin.protectmetadata' => true,
```

SSP 의 관리자페이지 노출 취약점을 해결하기 위해 apache 설정을 변경해 줘야 한다. CentOS 6.5 기준으로 SP 가 SSL 이 적용되어 있을 때 다음과 같이 설정한다.

```
~# clear
~# cd /etc/httpd/conf.d
~# nano ssl.conf
//</VirtualHost> 앞에 다음과 같이 추가한다. 설치환경에 맞게 적절히 수정되어야 한다.
<Location /simplesaml/module.php/core/loginuserpass.php>
    Order Deny,Allow
    Deny from all
    # 192.168.0.*만 접속 가능
    Allow from 192.168.0.0/24
</Location>
```

6.2 절 Privacy Policy

SP 의 메타데이터에 'privacypolicy'가 설정되어 있으면 consent 에서 해당 privacypolicy 를 링크한다. Consent 화면에는 privacypolicy 의 URL 이 %SPENTITYID%로 변경되어 표시된다.
※ 서비스 제공자는 ID 제공자에게 privacypolicy URL 정보를 전달하고, ID 제공자가 privacypolicy 정보를 설정해야 한다.

```
~# clear
~# nano /var/simplesamlphp/metadata/saml20-sp-remote.php
//특정 SP의 metadata 내에
```

```
'privacypolicy' => 'URL',   // 예; 'privacypolicy' => 'https://yourdomain.com/privacypolicy',
// 이 설정되어 있으면 consent 시 해당 URL이 화면 출력됨
```

6.3 절 SSP 의 보안 강화 사항

showerrors 항목을 false 로 해서 오류가 발생했을 때 노출되지 않아야 할 오류정보(stacktrace 는 시스템 정보를 노출함)가 사이트에 노출되는 것을 방지한다. 또한, admin 비밀번호를 설정해 SSP 의 정보가 노출되지 않도록 한다.

```
~# nano /var/simplesamlphp/config/config.php
//5.1과 동일
 'admin.protectindexpage' => true,
//추가 또는 수정
 'showerrors' => false,
```

쿠키 보안을 위해, 평문 연결(plain text connection, Non-TLS)일 때 쿠키 정보가 전송되는 것을 막고 자바스크립트가 쿠키에 접근하는 것을 막아야 한다. TLS(https connection)를 반드시 이용해야 한다. 쿠키 보안 설정을 하지 않으면 Cross Site Scripting 공격에 취약할 수 있다.

```
~# nano /var/simplesamlphp/config/config.php
// 다음과 같이 수정
 'session.cookie.secure' => true,
 'session.phpsession.httponly' => true,
```

SSP 가 'redirect'를 할 도메인 이름을 설정한다. 다음과 같이 empty array 로 설정하면 SSP 가 자동으로 신뢰하는 도메인으로만 redirect 한다.

```
~# nano /var/simplesamlphp/config/config.php
// 다음과 같이 수정
 'trusted.url.domains' => array(),
```

SHA1(보안취약) 대신 SHA-256 으로 이용한다. ID 제공자는 saml20-idp-hosted.php 를, SP 제공자는 authsources.php 를 수정한다. SSP 1.12 이상의 버전에서는 config 파일에 반영되어 있으므로 주석표시만 제거한다.

```
// ID 제공자일 경우에 해당
~# nano /var/simplesamlphp/metadata/saml20-idp-hosted.php
// 다음과 같이 주석제거
'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
```

[최근 갱신: 2015-11-6 -draft v0.14]

KAFE

제공자 기관 내 설명자료

– Korean Access Federation (KAFE) –

한국과학기술정보연구원

목 차

ID 제공자 기관 내 설명 자료

2015. 08. 20. – 초안

제 1 장 소개

KAFE 는 웹 응용서비스 등 정보자원의 제공자인 서비스 제공자와 사용자 인증 및 인가정보의 제공자인 교육·연구기관으로 구성된 연합체입니다.

KAFE 에 가입하면 교육·연구기관 간, 또는 교육·연구기관과 서비스 제공업체 간에 사용자 인증기능이 연계됩니다. 사용자 인증의 연계를 통해 교육·연구기관 내에서 웹 응용서비스 이용 시 싱글사인온(SSO, Single Sign On)이 실현됨과 동시에 다른 기관이나 상용서비스도 소속기관에 등록된 사용자 ID 와 비밀번호를 이용해 접근할 수 있습니다. 예를 들어, 다른 대학의 무선 LAN 을 자신의 소속기관에서 사용하는 사용자 ID 와 비밀번호를 이용해 접근할 수 있습니다.

ID 페더레이션에 있어서 제공자 간 신뢰 관계(Trusted Relationship)의 확보는 필수적입니다. 서비스 제공자는 각 기관에 등록된 사용자들이 제대로 관리되고 있는지 신뢰할 수 있어야 합니다. 악의적인 사용자가 서비스 제공자의 정보자원을 보안 침해할 수 있기 때문입니다. ID 제공자도 서비스 제공자에게 보낸 사용자 속성 정보가 악용되지 않는다는 것을 신뢰할 수 있어야 합니다. 개인정보 유출의 위험이 있기 때문입니다. KAFE 는 ID 페더레이션의 관리주체(3rd-party Trusted Authority) 역할을 수행할 예정입니다. KAFE 에서는 참여 기관의 신뢰관계 구축을 위해 정책 수립과 정책 준수 여부를 정기적으로 조사할 예정입니다.

KAFE 는 2015 년 9 월부터 ID 페더레이션의 시험서비스를 실시하고 있습니다. 참여 기관의 수는 2015 년 9 월 현재 ID 제공자(Identity Provider: 교육 및 연구기관 등)가 2 개 기관, 서비스 제공자(Service Provider: 서비스 제공 측)로 7 개 서비스가 참여하고 있습니다. ID 제공자와 서비스 제공자는 계속해서 추가될 예정입니다.

제 2 장 연구·교육 기관이 KAFE 에 참여하는 이유

2.1 절 ID 제공기관이 얻는 혜택

ID 페더레이션은 국제표준인 SAML(Security Assertion Markup Language) 프로토콜을 이용해 실현됩니다. 연구·교육기관에서 유지 중인 각종 웹 응용서비스에 SAML 기술을 적용시키면 개별 웹응용서비스가 사용자 인증 DB 를 가질 필요가 없어집니다. 사용자 인증 DB 를 제거하면 사용자 개인정보의 관리 업무의 부담이 경감됩니다. 또한 하나의 사용자인증시스템만 관리하면 되므로 개인정보의 유출 위험성이 줄어들며 보안관리 업무를 효율적으로 추진할 수 있습니다. 기관 내부에 유지 중인 비필수 웹 응용서비스들을 외부 업체에 아웃소싱할 수 있기 때문에 해당 서비스들의 구축 및 유지관리를 위해 필요한 금전적·인적 비용을 줄일 수 있습니다. ID 페더레이션을 통해 외부 기관과 R&D 정보자원을 손쉽게 공동 활용할 수 있는 환경이 조성되기 때문에 연구·교육기관 간 파트너쉽 강화에도 효과적입니다.

2.2 절 ID 제공기관의 구성원이 얻는 혜택

화상회의, 파일전송, 전자저널, 콘텐츠 관리 등 연구협업에 필요한 웹 응용서비스들을 소속기관에서 이용하는 사용자 ID 와 비밀번호를 이용해 손쉽게 접근할 수 있습니다. 하나의 사용자 ID 와 비밀번호만 기억하면 됩니다. 개별 서비스에 추가적으로 사용자 등록을 할 필요가 없기 때문에 개인정보의 노출 위험을 줄일 수 있습니다. 또한 싱글사인온이 지원되므로 하나의 서비스에 로그인하면 다른 서비스에는 로그인하지 않고도 서비스를 이용할 수 있게 됩니다. 마지막으로 ID 연계된 다양한 서비스를 이용할 수 있으므로 동료들 간 협업 강화 및 연구생산성 향상에 도움이 됩니다.

웹 응용서비스 이외에도 eduroam-AND 서비스를 이용하여 eduroam 의 사용을 위한 임시 계정을 생성할 수 있습니다. eduroam 은 글로벌 무선인터넷 접속 서비스입니다. 소속 기관에 등록된 사용자 ID 와 비밀번호를 이용해 eduroam 서비스를 제공하는 국내·외 외부기관에서 무선인터넷을 무료로 활용할 수 있습니다.

제 3 장 KAFE 에서 제공하는 서비스 목록

2015 년 9 월 현재 싱글사인온을 통해 이용할 수 있는 온라인 서비스의 목록은 다음과 같습니다. 제공되는 서비스는 계속 확장될 예정입니다.

서비스 제공자	서비스 명	서비스 내용
국가과학기술연구망	Filesender	대용량 파일전송
국가과학기술연구망	WebCache	웹 파일다운로드 가속
국가과학기술연구망	Webinar	세미나에 최적화된 온라인 화상회의
국가과학기술연구망	Webmeet	회의에 최적화된 온라인 화상회의
국가과학기술연구망	Foodle	미팅시간 결정
국가과학기술연구망	eduroam-AND	글로벌 무선로밍 서비스의 사용자 대리인증
국가과학기술연구망	Shopfront	개인화된 포털
국가과학기술연구망	vidyo	상용 화상회의 시스템

추가적으로 싱글 사인온은 지원되지 않지만 온라인 연구협업을 위해 제공되는 정보자원은 다음과 같습니다. 제공되는 서비스의 일부는 추후에 싱글 사인온 기능을 제공할 예정입니다.

서비스 제공자	서비스 명	서비스 내용
국가과학기술연구망	Communicator	인스턴트 메신저 서비스
국가과학기술연구망	FileStore	대용량 파일 스토리지
국가과학기술연구망	RealLab	가상머신 기반의 컴퓨팅 연구자원

국가과학기술연구망	Emulab	네트워크 R&D를 위한 테스트베드

제 4 장 KAFE 참가 요건

시험 서비스 기간 동안 다음에 기술된 하나 이상의 조항에 해당되는 기관이 ID 제공자로써 KAFE 에 참가할 수 있습니다. 정식 서비스 개시 후 참가 조건이 변경될 수 있습니다.

1. 국가과학기술연구망(KREONET) 회원 가입기관
국가과학기술연구망의 회원기관으로써 교육, 연구, 의료 및 공공분야 해당 기관

2. 한국과학기술정보연구원(KISTI) 수행사업의 파트너 기관
한국과학기술정보연구원에서 수행중인 사업에 참여하고 있거나 제공 중인 서비스를 이용하고 있는 국내 교육, 연구, 의료 및 공공분야 해당 기관

3. 기타 기관
위 2 개 분류에 포함되지 않는 국내 교육, 연구, 의료 및 공공분야 해당 기관

서비스 제공자로써 KAFE 에 참여 가능한 기관은 아래의 한 조항 이상을 만족해야 합니다.

1. KAFE 의 ID 제공자 자격을 갖춘 기관

2. 클라우드, 웹 응용 등 온라인 연구협업에 필요한 응용의 개발 및 서비스 제공 업체

3. 온라인 연구협업 응용을 개발 또는 서비스 중인 개인 및 유관 커뮤니티

KAFE 회원에게 가입비 및 연회비가 발생하지 않습니다.

제 5 장 제공기관에서 준비해야 하는 사항들

ID 제공자(Identity Provider) 기관은 사용자 인증연계용 서버 1 식이 필요합니다. 사용자 인증연계용 서버는 사용자 인가(Authorization)에 필요한 정보를 서비스 제공자(Service Provider)에게 제공하기 위한 시스템입니다. 예를 들어, 외부 웹 응용서비스가 사용자 인가를 위해 사용자 ID 와 소속기관명에 대한 정보를 필요로 한다면, 인증연계용 서버는 로그인이 성공한 사용자의 사용자 정보를 추출한 후 사용자 ID 와 소속기관명을 외부 웹 응용서비스에게 전달합니다. 서비스 제공자는 전달된 정보를 이용해 사용자 인가를 수행합니다.

5.1 절 운용 위원회

ID 제공자 기관에서는 ID 연계 운용위원회의 구성을 기관 내부 규정으로 명문화할 수 있습니다. 운용위원회는 기관외부 서비스에 대한 심의, 개인정보의 전송 등과 관련된 관리운용규정을 수립합니다. 기관 내부에서 운용 위원회의 설치가 필요하다면 다음 예시를 참조하십시오.

ID 제공자의 운용 및 관리는 OO 부서에서 실시한다. 서비스 제공자의 추가 등 ID 제공자의 운용에 관하여 심의가 필요한 사항은 담당 부서 OO에 연락한 후, △△ 위원회에서 심의한다.

- KAFE 참가 신청자: 홍길동 (기관장)
- 운용 책임자: 김철수 (OO 부서 장)
- 운용 담당자: 차순희 (OO 부서 전문 직원)
- 심의위원회: △△ 위원회

제 6 장 ID 제공자의 구축 비용

국내 여건 상 ID 제공자의 구축이 가능한 외부 업체는 없는 것으로 파악됩니다. 하지만, 사용자 인증과 웹 프로그래밍에 대한 간단한 지식만 있다면 기관 구성원들이 어렵지 않게 ID 제공자를 구축할 수 있습니다. 기관 내부에서 운용 중인 사용자 인증 DB 를 이용하는 가정 하에 예산 투입이 요구되는 항목은 다음과 같습니다.

1. 인증연계용 서버 1 식 구입비용(가상머신 이용 가능): 300~500 만원

2. 공인인증서(SSL) 구입비용: 약 50 만원(연간)

3. 참여기관 내 (ID 연계 관련) 시스템 관리운영 비용

제 7 장 서비스 시작까지의 로드맵

개별 기관에서는 다음과 같은 절차로 ID 페더레이션 환경을 구축·서비스 할 수 있습니다. 기관 상황에 맞춰 추진하십시오.

1. ID 페더레이션 기술과 구축 방법을 습득(기술지원 가능, 국가과학기술연구망)

2. KAFE 참여에 대한 기관 내부 설명 및 예산의 확보

3. ID 제공자 시스템의 구축(기술지원 가능, 국가과학기술연구망)

4. KAFE 시험 페더레이션에 참가 신청

5. ID 제공자의 동작 확인(기술지원 가능, 국가과학기술연구망)

6. 기관 내 공지 및 홍보 실시

제 8 장 개인정보의 취급

사용자가 KAFE 를 통해 연계된 웹응용 서비스를 이용할 경우, 사용자 속성정보가 서비스 제공자에게 전달됩니다. 요구되는 속성 정보는 서비스 제공자마다 다르지만 속성 정보에는 개인정보가 포함될 수

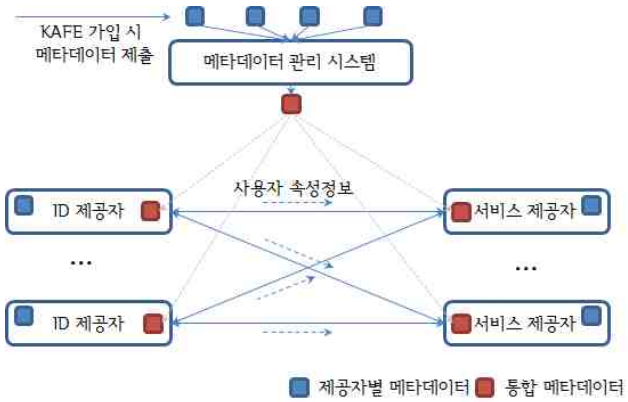
있습니다. 대한민국 개인정보보호법에 의하면 개인정보를 제 3 자에 제공할 때에는 사용자 동의를 얻어야 하는 조항이 있습니다. 관련 법령 준수를 위해, KAFE 에서는 사용자 동의 시스템을 적용해 사용자가 서비스에 로그인할 때마다 속성 정보의 전달에 관한 사용자 동의를 얻고 있습니다. 아래 표는 KAFE 에서 정의한 핵심 및 권고 속성으로써 서비스 제공자에게 전달될 수 있는 사용자 속성 정보입니다.

ID 제공자 기관에서는 기관 내부의 개인정보보호 규정 등에 맞춰 아래 정의된 속성정보 중 일부를 제공할 수 있습니다.

제공 속성	설명	개인정보 가능성
uid	사용자 ID(시험 서비스 기간에 한시적 적용)	O
eduPersonTargetedID	서비스 제공자 별 암호화된 사용자 고유번호	
displayName	화면표시 이름	O
mail	사용자 이메일 주소	O
eduPersonAffilation	사용자의 기관내 직무정보	
organizationName	사용자의 소속기관명	
schacHomeOrganization	사용자 소속기관의 최상위 도메인 이름	
eduPersonPrincipalName	도메인 내 사용자 ID 정보	O
eduPersonScopedAffilation	도메인 내 사용자 직무 정보	

제 9 장 ID 연계 시스템의 구조

KAFE 는 full mesh 형태의 ID 연계 구조를 갖습니다. KAFE 가 ID 연계를 위해 제공하는 어떤 서비스도 서비스 제공자 또는 ID 제공자로부터 사용자 정보를 전송 받거나 요구하지 않습니다. 또한 KAFE 는 가입 기관별 관리자 이름과 이메일 주소를 제외한 어떠한 개인정보도 저장하지 않습니다. ID 제공자에서 사용자 인증이 완료되고, 사용자가 속성정보의 제공에 동의할 경우에만 ID 제공자가 서비스 제공자에게 사용자 속성정보를 직접 전송됩니다.



위 그림은 KAFE 의 ID 연계 구조를 보여줍니다. KAFE 가입이 완료되면, ID 제공자나 서비스 제공자는 메타데이터 파일을 KAFE 에 제출해야 합니다. 메타데이터 파일에는 제공자 시스템에 대한 정보, 공개키 등이 포함됩니다. 서비스 제공자와 ID 제공자가 서로 통신하기 위해서는 상대방의 메타데이터 정보를 반드시 가지고 있어야 합니다.

KAFE 가 운용하는 메타데이터 관리시스템은 서비스 제공자와 ID 제공자가 제출한 개별 메타데이터를 통합해 하나의 페더레이션 메타데이터 파일을 생성합니다. KAFE 에서 페더레이션 메타데이터를 제공함으로써 ID 제공자 및 서비스 제공자는 메타데이터 관리에 소요되는 시간적·인적 부담을 덜 수 있습니다. 서비스 제공자와 ID 제공자는 KAFE 의 메타데이터 관리시스템이 생성한 페더레이션 메타데이터 파일을 주기적·자동으로 다운로드받아 제공자 시스템에 적용하게 됩니다.

ID 제공자와 서비스 제공자는 페더레이션 메타데이터를 다운로드받는 것 이외에 어떤 경우에도 메타데이터 관리시스템과 통신하지 않습니다. 사용자 속성 정보 등의 메시지 교환은 전적으로 ID 제공자와 서비스 제공자 간에 이루어집니다.

[최근 갱신: 2015-9-25 -draft v0.14]

연구교육인증위원회 운영 규정

한국과학기술정보연구원

연구교육인증위원회 운영규정

2015 년 10 월 xx 일 제정

(설치)

제 1 조 연구교육인증연합(이하 “인증연합”이라 한다)은 산하에 연구교육인증위원회(이하 "위원회"라 한다)를 둔다.

(목적)

제 2 조 위원회는 대한민국의 연구 · 교육 · 학술단체의 인증 연계 추진을 위하여 ID 페더레이션 및 관련 사항을 기획 · 입안하고 운영하는 것을 목적으로 한다.

(조직)

제 3 조 가. 위원회는 다음 각 호에 해당하는 인원들로 조직한다.

- 한국과학기술정보연구원(이하 “연구원”이라 한다)의 관련 직원 약간 명
- 인증연합 참여 기관의 직원 약간 명
- 기타 연구원장이 필요하다고 인정한 자

나. 위원은 원장이 위촉한다.

(임기)

제 4 조 위원의 임기는 2 년 이하로, 위촉할 때 결정하며 연임할 수 있다.

(위원장)

제 5 조 가. 위원회에 위원장을 두며 연구원장이 지명한다.

나. 위원장의 유고시에는, 위원장이 사전에 지명한 위원이 그 직무를 대행한다.

(회의 개최)

제 6 조 회의는 위원장이 소집하고 위원장이 의장이 된다.

(의사)

제 7 조 가. 회의는 위원 과반수가 출석하여야 개최할 수 있다.

나. 회의의 의사는 출석 한 위원의 과반수로 결정되며, 가부동수일 경우에는 위원장이 결정한다.

(워킹 그룹)

제 8 조 가. 위원회는 인증연합과 관련된 사항의 구체적인 조사, 검토 및 작업 등을 실시(이하 "작업 등"이라 한다)하기 위한 워킹 그룹을 둘 수 있다.

나. 워킹 그룹은 위원회의 요구에 따라 작업 등을 실시하고 그 결과를 위원회에 보고하거나 심의사항을 제안하여야 한다.

다. 워킹 그룹에 간사를 두고 간사가 워킹 그룹을 총괄하며 위원회의 위원 중에서 선출한다.

라. 워킹 그룹 회원은 위원회가 지명한다.

마. 위원회에 관하여 필요한 사항은 위원회가 따로 정한다.

(비밀 유지)

제 9 조 제 3 조의 위원은 소정의 기밀 유지에 관한 각서를 체결한다.

(일반적 사무)

제 10 조 위원회의 일반적인 사무는 운영기관에서 처리한다.

(기타 사항)

제 11 조 이 규정에 정한 것 이외에, 위원회의 운영 등에 관하여 필요한 사항은 위원회에서 따로 정한다.