브로커 및 에이전트 기반의 통합 단일 인증 시스템 설계에 관한 연구

(A Study on the design of broker and agent based Single Sign-On system)

최홍민° 손태식 서정택* 채송화 유승화 김동규
아주대학교 정보통신공학과, *국가보안기술연구소

{ partout, tsshon, portula, swyoo, dkkim }@madang.ajou.ac.kr, *seojt@etri.re.kr

요 약

현재의 정보통신 사회에서 널리 사용되는 인터넷 서비스는 기본적으로 ID/PW(PassWord) 기반의 인증을 사용한다. 이때 사용자는 여러 웹 서비스에서 대해서 각각 다양한 ID/PW를 기억해야 한다는 어려움을 가진다. 마찬가지로 웹 서비스 관리자 역시 여러 사용자들의 ID/PW를 관리하는데 많은 비용 및 노력을 소모해야 한다. 따라서 한 번의 안전한 인증 과정을 통해 사용자 및 관리자의 편리를 도모할 수 있는 SSO(Single Sign-On)시스템의 적용이 필수적으로 요구되고 있다. 본 연구에서는 기존의 SSO 시스템을 분석하여 새롭게 모든 인터넷 환경에서 보다 안전하게 사용자에게 서비스를 제공하며, 관리자에게는 편리성을 제공하는 브로커와 에이전트의 기능을 포함한 통합 SSO 시스템을 설계하며, 추후 연구과제로서 실제 브로커 및 에이전트 기반 통합 SSO 시스템 구현 및 적용에 대하여 연구 할 것이다..

1. 서론

현재의 인터넷 서비스는 기본적인 사용자의 인증을 위하여 각각의 패스워드를 사용한다. 많은 패스워드의 사용은 사용자나 관리자 모두에게 큰 문제를 발생시킨다. 사용자들은 여러 패스워드들을 기억해야 하고, 관리자들은 각 서버에 대해 각각의 패스워드 데이터베이스를 유지해야 하는 등의 문제가 발생한다. 이러한 문제를 해결하기 위해 사용자가 하나의 패스워드만을 사용하여 로긴 하였을 때 네트워크를 통한 패스워드의 전송 없이 모든 서버의 인증을 얻는 방법이 필요하다. 이를 SSO(Single Sign-On)시스템이라 하며, 본 연구에서는 SSO 시스템의 소개와 기존의 제안되는 기술 방안에 대해서 알아본 후 브로커 및 에이전트 기반의 SSO 시스템에 대하여 설계하겠다.

2. 기존의 SSO 시스템

일반적인 인증 방법은 기본 인증(Basic Authentication)과 강한 인증(Strong Authentication)으로 나뉜다. 기본 인증은 일반적으로 서버들이 ID/PW를 사용하여 사용자 인증을 하는 것으로서 이러한 형태는 암호화하지 않고 이루어질 수 있고, 또는 암호화된 SSL 연결을 통한 서버 인증으로 이루어질 수 있다. 강한 인증은 인증서를 바탕으로 한 사용자 인증으로 SSL을 사용할 수도 있다. 인증서는 개개인과, 서버 또는 다른 실체를 인증 한다. 사용자를 서버에 인증 하기 위해 클라이언트는 임의로 생성된 데이터를 서명하고 네트워크를 통해 서명된 데이터와 인증서를 보낸다. 어떤 데이터에 관계된 전자 서명은 클라이언트와 서버에 의해 제공된 증거가 된다. 서버는 이러한 증거로 사용자를 인증 한다.

2.1 Common Standard Solutions

우선 SSO를 구현하는데 사용되어지는 일반적인 표준 솔루션들을 살펴본 후에 실질적인 SSO 솔루션에 대하여 알아본다.

2.1.1.GSS-API (Generic Security Service Application Program Interface)

GSS-API는 인증, 무결성, 기밀성 등을 포함하는 강력한 보안 서비스를 제공하기 위한 interface를 의미한다. SSO를 포함하는 서로 다른 보안 서비스나 애플리케이션들을 구성하는데 사용되어진다. 특정 보안 메커니즘을 송가는 interface를 제공하는 GSS-API의 목적은 서로 다른 애플리케이션들의 공동 사용을 가능하게 하는 것이다.

2.1.2. PAM (Pluggable Authentication Modules)

PAM은 서로 다른 인증 방법사이에 이동이 쉽도록 해주는 API이다. 많은 운영체제나 응용 소프트웨어가 PW 같은 한 가지 형태의 인증 방법에 의존한다. 만약 더 강력한 인증 방법이 요구되면 그 작업이 엄청나게 된다. PAM의 목적은 다른 인증 방법을 제공함에 있어서 "plug-in" 형태의 간단하고 직접적인 방식을 제공하려는 것이다.

2.2 Broker-Based SSO Solution

이 방법에서는 사용자의 계정을 관리하고, 중앙에서 인증을 처리하기 위한 서버가 존재한다. Broker는 앞서의 접근에서 사용되어진 전자 id를 제공한다.

2.2.1 Kerberos

Kerberos는 MIT에서 고안된 TCP/IP를 위한 인증 프로토콜이며, broker-based 형태의 가장 기본이 된다. Kerberos 서버라 불리는 신뢰할만한 서버가 존재한다. 이는 사용자를 인증하고, 주어진 증명서에 대한 전자 identity를 제공하는 등의 broker의 역할을 수행한다. 제공된 identity는 다른 서비스를 요구할 때 ticket처럼 사용되어진다. 전송되는 모든 data는 암호화되어진다. 일반적으로 Kerberos는 GSS-API를 사용해 구현한다.

2.2.2. SESAME (Secure European System for Applications in a Multi-vendor Environment)

SESAME는 유럽에서 사용되는 Kerberos이다. SESAME V3은 GSS-API의 최상위에 구현되어지고, SSO 서비스와 분산 환경에서의 기밀성을 제공한다. Kerberos를 기반으로 하고 있지만 그대로 가져온 것이 아니라 여러 가지 기능들이 추가되었다. Kerberos에서와 같이 사용자는 인증 서버에서 자신에 대한 인증을 먼저 받는다. 그리고는 인증 서버에서 토큰을 받아서 접근하고자 하는 서버에 있는 애플리케이션을 사용할 때 제공한다. SESAME에서는 인증 서버를 PAS(Privilege Attribute Server)라 하며, 토큰을 PAC(Privilege Attribute Certificate)라

한다.

2.3 Agent-Based SSO Solution

Agent-based solution에서는 서로 다른 애플리케이션들에 대해서 사용자들의 신원을 자동적으로 확인해주는 agent 프로그램이 존재한다. Agent는 다른 방법들의 기능으로 만들어진다. password list나 암호화 키를 보낼 수 있고, 그런 것들을 사용자로부터 인증을 하는 노력을 줄이는데 사용할 수 있다. Agent는 서버 측에서 인증 시스템과 인증 방법 사이의 인터프리터와 같은 역할을 할 수도 있다. Agent-based solution의 예로 SSH가 있다.

SSH는 인터넷상에서 안전한 연결을 만들기 위한 보안 소프트웨어 시스템이다. SSH의 사용자는 다른 인증 방법을 사용해서 인증을 받을 수 있다. 원하는 identity가 agent에 추가되고 나면, 모든 새로운 연결은 agent의 자식으로 시작된다. 그러므로 계속 상속될 수 있다.

2.4 Token-Based SSO Solutions

Security Dynamics SecurID는 시간에 따른 one-time password를 생성하는 물리적 토큰이다. SecureID는 하드웨어 token 상의 동기화된 시간과 ACE 서버라 불리는 네트워크의 서버에 기반을 둔다. 미리 정해진 간격으로 토큰이 유일하고, 주어진 윈도우 시간에만 받아들여지는 새로운 PW를 생성한다. 이 방법은 WebID라 불리는 모듈도 포함한다. WebID에서는 Netscape Suitespot web server에 프로그램이 설치되어서 SecurID에의해 생성된 PW를 받아들인다. 첫 번째 URL에대한 인증이 유지되는 동안 WebID는 암호화된 ticket인 소프트웨어 토큰을 생성한다. 이는 새로운 URL에 접근할 때 인증서로 사용된다.

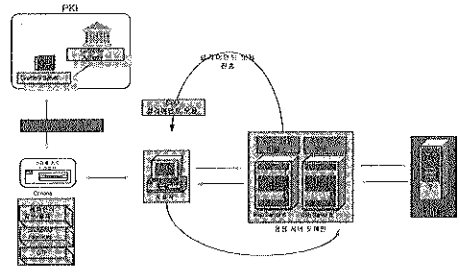
2.5 Gateway-Based SSO Solutions

네트워크 상에 감사건을 두었던 형태의 모델을 제공하는 broker-based 방법에 비해 gateway-based 방법은 신뢰할 만한 네트워크 안쪽에 서비스로 들어가는 문이 존재한다. 이 문은 방화벽일 수도 있고, 암호화를 위한 서버일 수도 있다. 이 방법에서는 모든 서비스는 gateway 뒤에 존재하며, gateway는 서비스 고유의 ip 주소를 통해서 각각의 서비스를 구별한다. 사용자는 gateway를 통해서 모든 서비스에 대한 인증을 하게 된다. gateway는 client들의 identity를 기억하고, 요구되어지는 서비스를 더 이상의 인증과정 없이 제공한다. gateway는 서비스로 향하는 모든 데이타의 흐름에 접근할 수 있기 때문에, 데이타의 흐름을 살펴고, 조절한다. 따라서 서비스를 향해서 가는 인증 정보를 바꿀 수 있고, 애플리케이션의 변환 없이 적당한 접근 제어를 할 수 있다.

3. 브로커 및 에이전트 기반 통합 SSO 설계

아래에서 설계하는 SSO 시스템은 모든 인터넷 환경에서 적용이 가능하며 ID/PW 인증 기술과 스마트카드를 이용한 인증서 기반의 인증 기술을 사용하며, 신임장 포맷으로는 암호화 쿠키를 사용한다.

Agent-Based SSO로서의 기능은 웹 기반에서의 ID/PW 인증 기능, 암호화된 쿠키 형태로 credential 이용 그리고 쿠키 암호화에 필요로 하는 비도에 따라 다양한 알고리즘 사용과 같은 기능을 가지고 있으며 Broker-based SSO로서의 기능은 네트웍을 통한 클라이언트 모듈의 실행으로 스마트카드를 사용한 인증서 기반의 인증(challenge response scheme), 암호화된 쿠키 형태로 credential 이용 가능 그리고 쿠키 암호화에 필요로 하는 비도에 따라 다양한 알고리즘 사용 등이 있다. 특징으로는 모든 인터넷 환경에 적용 가능, 클라이언트 단의 별도 수정 및 부가 작업이 필요 없음, 다단계 보안관리 가능(암호화 알고리즘에 따른 다단계 보안, ID/PW 와 인증서 기반의 다단계 보안 그리고 사용자 ID 기반의 다단계 보안), 공통된 신임장(credential) 폼의 사용으로 인한 서버 및 클라이언트의 부하 감소 그리고 다양한 인증 기법 사용 가능(ID/PW, 인증서, 스마트 카드, OTP, 생체인식 등)등이 있다.



[그림 1] 통합 SSO 구성도 통합 SSO 인증 절차

3-1. ID/PW 기반 인증 절차

인증 전 고려 사항으로는 응용 서버 도메인의 보안 정책에 기반하여 사용자 등록시 ID를 통한 접근 수준 제한, 응용 서버 도메인의 보안 정책에 기반하여 사용하는 신임장의 암호화 수준을 적정하게 고려 그리고 응용 서버 도메인의 보안 정책에 따라 인증 방식이 ID/PW 또는 인증서 기반의 두 가지 중 하나로 결정되어야 한다는 점이다.

인증 과정

1. 사용자는 응용 서버에 접속하며, 응용 서버는 인증 모듈에서 인증 과정을 시작한다.
2. 인증 모듈에서 사용자 미리 정의된 사용자 정보(ID, PW, 접근 수준, 신임장 암호화 알고리즘)에 기반 하여 사용자를 인증 한다.
3. 인증이 성공하면 신임장 발급 서버에 신임장을 요청한다. 신임장 발급 서버는 응용 서버 도메인에 접근이 가능한 적정 수준의 신임장을 응용 서버와 신임장 서버간의 세션키로 암호화하여 응용 서버에게 건네준다.
4. 응용 서버는 건네 받은 신임장을 쿠키 폼으로 암호화하여 사용자에게 건네준다.
5. 사용자는 응용 서버의 인증 모듈로부터 발급 받은 신임장을 가지고 응용 서버 도메인내의 다른 응용 서비스를 부가적인 인증 없이 제공받을 수 있다.
6. 응용 서버 도메인 내의 다른 서버들은 신임장을 가지고 있는 사용자의 접근시에 신임장을 응용 서버와 신임장 서버간의 세션키로 복호화하여 인증 사항을 확인하여 적절한 응용 서비스를 제공한다.

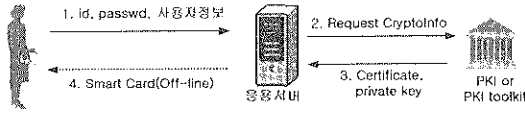
3-2 스마트카드 사용의 인증서 기반 인증 절차

인증 전 고려 사항으로는 인증서 기반 인증을 위하여 사용자 등록시에 사용자의 인증서, 패스워드 그리고 개인키가 저장된 스마트카드를 제공해야 하는 것과 응용 서버 도메인의 보안 정책에 기반하여 사용자 등록시 ID를 통한 접근 수준 제한, 응용 서버 도메인의 보안 정책에 기반하여 사용하는 신임장의 암호화 수준을 적정하게 고려 그리고 응용 서버 도메인의 보안 정책에 따라 인증 방식이 ID/PW 또는 인증서 기반의 두 가지 중 하나로 결정되어야 한다는 점이다.

인증 과정

1. 사용자는 응용 서버에 접속한다.
2. 응용 서버는 인증 모듈에서 인증 과정을 시작한다.
3. 인증 모듈에서 스마트카드에 저장된 인증서를 이용하기 위하여 클라이언트 모듈과 인증에 사용되는 랜덤 값을 함께 사용자에게 전송한다.
4. 사용자측에서 전송받은 클라이언트 모듈이 실행된다.
5. 클라이언트 모듈은 스마트 카드에서 사용자정보(인증서, 개인키, 사용자 ID)를 얻어오기 위하여 사용자의 비밀번호를 입력받는다.
6. 사용자의 비밀번호가 승인되면 스마트카드에서 사용자 정보(인증서, 개인키, 사용자 ID)를 얻어온다.
7. 얻어온 사용자정보(인증서, 개인키, 사용자 ID)중에서 사용자의 개인키를 통하여 인증 모듈로부터 받은 랜덤 값을 암호화한다.
8. 암호화된 랜덤 값과 함께 인증서 및 사용자 ID를 서버 측에 보낸다.
9. 서버 측의 인증 모듈에서 사용자의 인증서에서 공개키를 얻어낸다.
10. 얻어낸 공개키를 통하여 사용자의 개인키로 암호화된 랜덤 값을 복호화해내어 전송전의 랜덤 값과 비교하여 사용자를 인증한다.
11. 인증이 성공하면 신임장 발급 서버에 신임장을 요청한다.
12. 신임장 발급 서버는 응용 서버 도메인에 접근이 가능한 적정 수준의 신임장을 응용 서버와 신임장 서버간의 세션키로 암호화하여 응용 서버에게 건네준다.
13. 응용 서버는 건네 받은 신임장을 쿠키 품으로 암호화하여 사용자에게 건네준다.
14. 사용자는 응용 서버의 인증 모듈로부터 발급 받은 신임장을 가지고 응용 서버 도메인내의 다른 응용 서비스를 부가적인 인증 없이 제공받을 수 있다.
15. 응용 서버 도메인 내의 다른 서버들은 신임장을 가지고 있는 사용자의 접근시에 신임장을 응용 서버와 신임장 서버간의 세션키로 복호화하여 인증 사항을 확인하여 적절한 응용 서비스를 제공한다.

3-3 스마트카드와 사용자 사이의 오프라인 절차



[그림 2] 스마트카드와 사용자 사이의 오프라인 절차

1.사용자는 응용서버(웹에플리케이션)에 원하는 ID/PW와 사용자 정보를 가지고 온라인으로 가입 신청을 한다.
2. 응용서버는 이 사용자가 사용할 Certificate과 private key쌍을 생성하기 위해 애플리케이션 도메인 내의 PKI에 요청한다.
3. 생성된 Certificate, private key쌍을 PKI로부터 얻어와서 스마트카드에 Certificate과 사용자의 PW로 암호화된 사용자의 private key를 저장한다.
4. 위의 과정에서 만들어진 Smart Card를 우편 등으로 사용자에게 전달한다.

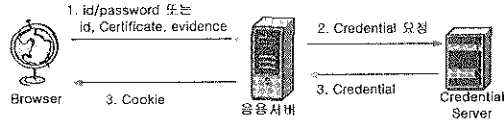
3-4 신임장(Credential)의 사용 절차

~ 신임장의 내용

Credential = E_{Ksso}{
String sso_id; (SSO 시스템에서 식별하기 위한 대표 id)
List<String> id_list; (각 사이트에 등록된 id 리스트)
Long timestamp; (타임스탬프)
Long valid_time; (유효기간)
}

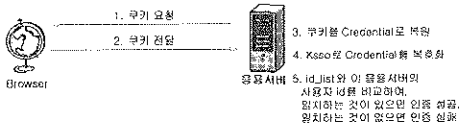
Ksso : Credential 서버가 Credential을 생성하기 위해서 사용하는 키로 Credential을 검증하기 위해 응용서버들 간에 공유된다.

~ 신임장 사용 절차



[그림 3] Credential의 저장

사용자가 id/password 또는 id/Certificate/evidence를 응용서버에 보내 인증을 요청하면 응용서버는 이를 검증하고 인증이 성공한 경우 Credential Server에게 요청한다. Credential 서버는 자신의 DB를 검색해 그 사용자에 해당하는 id 리스트를 얻어서 Credential를 생성해 이것을 응용서버에 전달하고 응용서버는 이것을 Cookie로 변환해서 사용자의 브라우저에 저장한다.



[그림 4] 브라우저에 저장된 쿠키의 사용

위의 순서대로 브라우저에 저장되었던 쿠키가 인증에 이용되고 인증이 성공하면 사용자는 별도의 로그인 절차없이 응용서버의 서비스를 이용할 수 있다. 인증이 실패한 경우 응용서버는 사용자에게 로그인을 요구한다.

4. 결론

기존의 에이전트 기반 SSO와 브로커 기반의 SSO 시스템의 특징을 바탕으로 현재의 인터넷 환경에 적합한 SSO 시스템을 설계하였다. 본 연구에서 설계된 SSO 시스템은 인증 기법에 있어 ID/PW 및 스마트 카드를 통한 인증서를 사용하여 다단계 보안을 지원하며, 웹 서비스에 적합한 쿠키 품의 신임장을 사용하여 편이성을 더 하였다. 향후에는 인증서 기반 인증에 있어 스마트 카드 보다 간편한 인증 도구의 사용이나 온라인 인증 기법에 대한 연구 및 실제 시스템 적용에 대한 연구가 이루어져야 할 것이다.

참고문헌

[1] J.Hursti, "Single Sign-On", in Proceeding of Helsinke Univ of Technology, Seminar on Netwokr Security, 1997
[2] T.Tervoi, "Single Sign-On Solutions in a Mixed Computing Environment", in Proceeding of Helsinke Univ of Technology, Seminar on Netwokr Security, 1998
[3] Camilloi, "Unified Single Sign-On", in Proceeding of Helsinke Univ of Technology, Seminar on Netwokr Security, 1998
[4] 김동규외, 분산통신망 환경 통합 정보보호 소프트웨어기술, 3 차년도 보고서, 정보통신부, 1999.01
[5] William Stallings, " Cryptography and Network Security : Principles and Practice "