УДК 004.4:004.5

М.Карпінський[1], докт. техн. наук; С.Войт[2];
Я.Аляшевич[1], канд. техн. наук

[1]Університет в Бєльску-Бялей (Польща)
[2]Тернопільський державний технічний університет імені Івана Пулюя

АЛГОРИТМИ ТА МОДЕЛІ ОРГАНІЗАЦІЇ ДОСТУПУ ДО ВЕБ-РЕСУРСІВ НА ОСНОВІ СИСТЕМ ОДНОРАЗОВОЇ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

У статті розглянуто алгоритми функціонування, моделі побудови та роботи систем забезпечення одноразової аутентифікації користувачів для доступу до розподілених веб-ресурсів. Здійснено розгляд та аналіз архітектури та принципів функціонування федеративної системи аутентифікації та авторизації користувачів.

M. Karpinskyy, S. Voyt, Y. Alaszewicz

ALGORITHMS AND MODELS OF THE ORGANIZATION OF ACCESS TO THE WEB-RESOURSES ON THE BASIS OF SINGLE SIGN-ON AUTHENTICATION SYSTEMS

Overview algorithms, models and working examples of single sign-on authentication systems of the organization of access to the distributed web-resourses. Overview and analizies of architecture and working principles of the federated authentication and authorization system.

Умовні позначення

LDAP (Lightweight Directory Access Protocol) – протокол доступу до даних каталогу;
SSO-сервіс (Single Sign-On Service) – сервіс одноразової аутентифікації користувачів;
Кука (cookie) – інформація, що зберігається у броузері користувача і призначена для його ідентифікації на веб-сервісі;
Логін (login) – унікальне ім'я корисутвача в системі, яке вказується при аутентифікації;
Тікет (ticket) – спеціальна інформація, що використовується для унікальної ідентифікації користувача, сесії з'єднання або процесу.

1. Вступ

Веб-сервіси (інформаційні портали, форуми, портали дистанційного навчання і бібліотеки) широко використовуються в сучасних комунікаціях та в сфері надання освітніх послуг у навчальних закладах. Загалом, кожен веб-сервіс вимагає аутентифікації користувачів, що зумовлює для кінцевих користувачів необхідність пам'ятати велику кількість логінів та паролів до різноманітних сервісів, кількість яких постійно збільшується, навіть в рамках єдиного навчального закладу.

Використання LDAP-каталогів дозволяє створити єдиний обліковий запис користувача, що може використовуватись як джерело аутентифікації у мережевих веб-сервісах. Однак незважаючи на використання централізованих LDAP-каталогів, що стандартизовані за протоколом доступу та організації даних, залишається декілька проблем у роботі таких систем аутентифікації [1]:

1. Багаторазові аутентифікації – кінцевому користувачу в будь-якому випадку, зокрема на вимогу зі сторони веб-сервісу, необхідно проходити процедуру аутентифікації щоразу при зверненні до захищених ресурсів.
2. Безпека – обліковий запис користувача в області дії єдиного каталогу є унікальним, логін та пароль користувача є критично важливою інформацією. Тому захист цієї інформації є важливою задачею, з огляду на передачу логіна та пароля незахищеними каналами. Крім цього, важливою є здійснення передачі особистих даних користувачів до веб-сервісів з огляду на їхню захищеність.

3. Прозорий доступ до ресурсів для аутентифікованих користувачів – користувачі, які успішно пройшли аутентифікацію, повинні мати доступ як до внутрішніх, так і до зовнішніх ресурсів.
4. Процедура авторизації користувачів – більшість веб-ресурсів вимагає додаткових даних про аутентифікованого користувача, що обумовлює необхідність передачі користувацьких даних з каталогу до сервісу ресурсів з необхідним рівнем захисту.

Виходячи з вищенаведених умов, необхідним є сервіс, що забезпечує єдину сеансову аутентифікацію користувача та надає інформацію про його аутентифікацію різноманітним сервісам, з якими може працювати користувач.

Принцип роботи SSO-сервісів полягає у централізації процесу аутентифікації користувачів, формування спеціальних тікетів при успішній аутентифікації та надання доступу за допомогою тікетів до сервісів, забезпечення перевірки істинності тікетів користувачів та сервісів.

2. Опис архітектури SSO

SSO – це метод доступу користувача до захищених ресурсів, що полягає в одноразовій аутентифікації та на її основі отримання доступу до різноманітних захищених ресурсів. SSO-сервіс – сервіс, що реалізує даний метод доступу.

Виділяють два різні способи, які можна використати для побудови систем SSO, – застосування броузерних куків та сертифікатів X.509 [1,6].

2.1 Методи на основі куків

У цьому випадку користувач проходить аутентифікацію в SSO-сервісі, броузер отримує куку, що містить спеціальну аутентифікаційну інформацію з SSO-сервісу та зберігає у своїй базі даних куків. При доступі до захищених ресурсів, сервіс останніх запитує у броузера аутентифікаційну куку. Якщо вона присутня і коректна, то користувачу надається доступ до ресурсів. В іншому випадку, наприклад, вийшов час існування куки, що встановлюється SSO-сервісом, користувач буде перенаправлений на повторну аутентифікацію в SSO-сервісі.

Стандартні куки не є безпечними, тому що вони не мають цифрового підпису власника куки. В зв'язку з цим задачею будь-якої SSO-схеми є розробка та впровадження ефективного способу забезпечення цілісності куків.

Прикладом SSO-сервісів, що базуються на використанні куків для аутентифікації користувачів, є CoSign, Pubcookie, CAS.

2.2 Методи на основі сертифікатів X.509

Тоді, функції SSO-сервісу полягають у перевірці цифрових підписів сертифікатів X.509 та аутентифікації користувача, якщо сертифікат правильний.

Прикладом SSO-сервісу на основі X.509 сертифікатів є сервіс KX.509.

2.3 Порівняння можливостей систем на основі куків та X.509 сертифікатів

Куки не розроблялись для аутентифікації, але використовуються як складова частина цього процесу. З точки зору безпеки, є безпосередня можливість перехоплення куків під час їх передачі між користувачем та сервісом. Тому задля безпеки необхідно використання або захищених каналів передачі даних, або захищених транспортних рівнів протоколу TCP/IP, зокрема використання SSL/TLS (Secure Sockets Layer/Transport Layer Security).

З точки зору зручності, куки опрацьовуються усіма найпоширенішими броузерами. Якщо кука була отримана користувачем, то її використання є більш прозорішим для користувача, ніж застосування сертифікатів X.509. Проте у користувачів є можливість заблокувати отримання куків, що робить менш зручним SSO-сервіс на основі кук.

Виходячи з безпеки, сертифікати X.509 є стандартизовані, давно використовуються та тестуються на захищеність. Безпека сертифікатів X.509 пов'язана з безпекою передачі та зберігання персональних ключів у користувачів і з терміном дії самого сертифіката: тобто чим довший термін дії сертифіката, тим більша імовірність його компрометації через втрату персонального ключа користувачем. Тому задля безпеки варто використовувати сертифікати, термін дії яких менший.

Також для користувачів повинно бути встановлене спеціальне програмне забезпечення з метою безпечного отримання сертифікатів.

3. Архітектури, моделі роботи SSO-сервісів

3.1 Сервіс CAS

CAS – SSO-сервіс, що базується на моделі Kerberos обміну куками. Його основні характеристики – це безпека, можливість працювати в режимі проксі-сервера, гнучкість, велика кількість клієнтських бібліотек [2].

Безпека CAS забезпечується наступними шляхами:

1. паролі користувачів передаються від броузера до сервера тільки через захищені канали та протоколи;
2. переаутентифікація є прозорою для користувачів за допомогою використання TGC (Ticket Granting Cookie);
3. веб-сервіси "пізнають" користувачів за допомогою спеціальних ST-тікетів (Service Ticket) – унікальних параметрів, що генеруються CAS-сервером, передаються через броузер користувача до сервісу ресурсів, де ST перевіряється на його правильність через CAS-сервер.

3.1.1 Архітектура CAS

CAS-сервіс складається з таких складових:

- CAS-сервер, що проводить аутентифікацію користувачів, здійснює передачу та перевірку ідентифікаторів аутентифікованих користувачів;
- CAS-клієнт, що забезпечує захист веб-ресурсів;
- Веб-броузер користувача, що повинен підтримувати обмін даними по HTTPS-протоколу та роботу з куками.

3.1.2 Модель аутентифікації користувача

На рис. 1 зображено процес доступу до ресурсів, захищених CAS без попередньої аутентифікації.

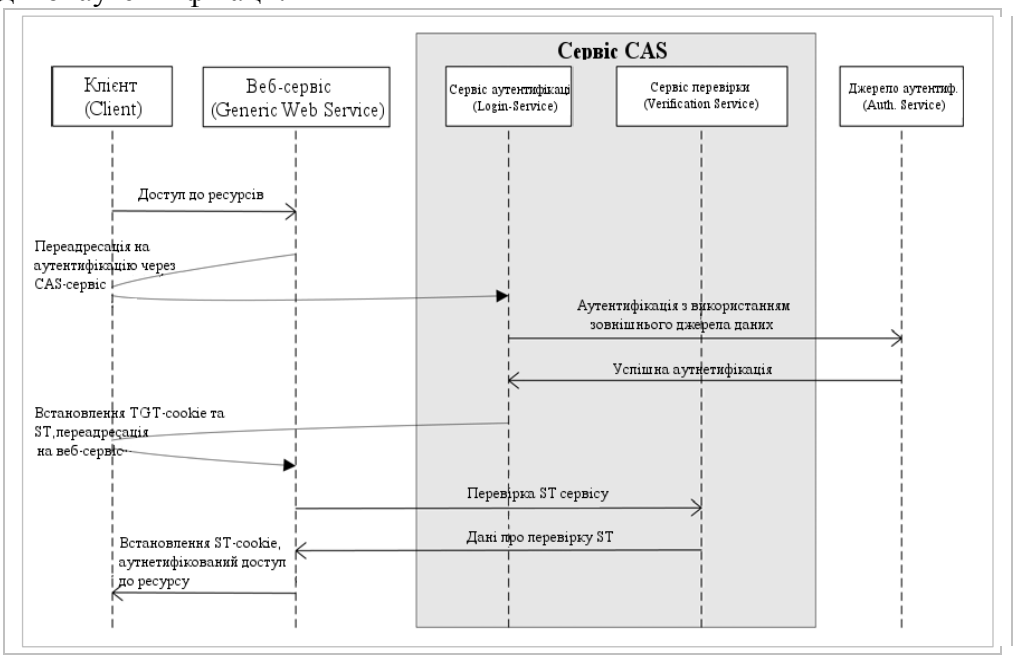

Рисунок 1 – Діаграма процесу доступу до ресурсів, захищених CAS, без попередньої аутентифікації користувача

Зазначений процес складається за наступних етапів:

1. Доступ до захищених ресурсів.
2. Перенаправлення до сервісу аутентифікації користувачів CAS-сервера.

Користувач проходить аутентифікацію за допомогою власного логіна та пароля, що відсилаються по HTTPS протоколу. Сервіс перевірки аутентифікації користувачів проводить перевірку даних користувача з використанням зовнішніх джерел даних користувачів, наприклад, каталогу LDAP, із застосуванням механізму Kerberos.

3. Якщо користувач успішно проходить аутентифікацію, то броузеру користувача відсилається TGC, броузер перенаправляється на ресурс з додатковим параметром в URL. Додатковий параметр – це ST, який формується на основі TGC, має період існування кілька секунд та є унікальним в межах всього CAS-сервісу.
4. Веб -ресурс за допомогою CAS-клієнта проводить перевірку ST на CAS-сервері, тобто ST з URL через захищені канали передається на CAS-сервіс, де проводиться ідентифікація ST. Якщо такий ST наявний на сервісі, то CAS-клієнту відсилається підтвердження про аутентифікацію користувача, ST знищується на сервері та клієнті CAS. Формується STC (Service Ticket Cookie), яке відправляється броузеру користувача, та надаються відповідні права доступу до ресурсу.

На рис. 2 зображено процес доступу до ресурсів, захищених CAS з попередньою аутентифікацією користувача на CAS-сервері.

Наведений процес складається з наступних етапів:

1. Користувач передає аутентифікаційну інформацію на CAS-сервер, який перевіряє дані і при їх правильності генерує TGC-куку, яка зберігається у броузері користувача.
2. Користувач звертається до ресурсу, що захищається CAS-сервером.
3. Якщо броузер користувача надає ресурсу TGC, то останній перенаправляється на CAS-сервер, де здійснюється перевірка TGC. Якщо TGC-кука існує та дійсна, то формується ST, який є параметром перенаправлення до ресурсу.
4. CAS-клієнт ресурсу відсилає отриманий ST до CAS-сервера, де здійснюється його перевірка. Якщо ST дійсне і правильне, то формується STC, яка є кукою сесії між броузером користувача та ресурсом.

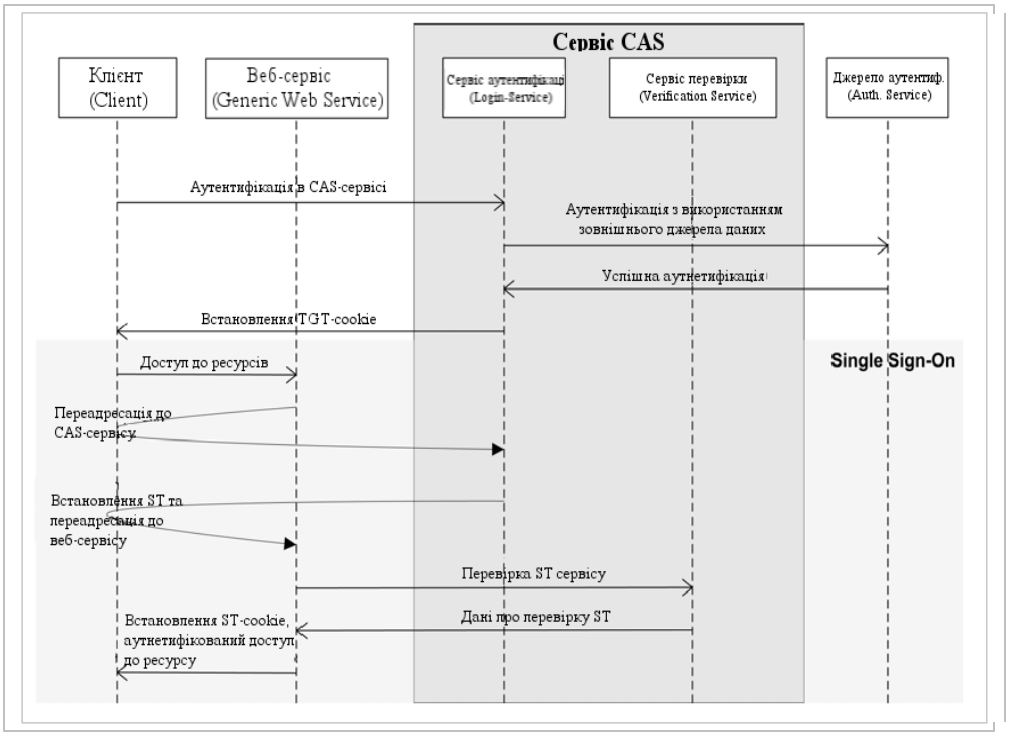

Рисунок 2 – Діаграма процесу доступу до ресурсів, захищених CAS, з попередньою аутентифікацією користувача

На рис. 3 зображено процес доступу до ресурсів, захищених CAS після аутентифікації користувача на CAS-сервері.

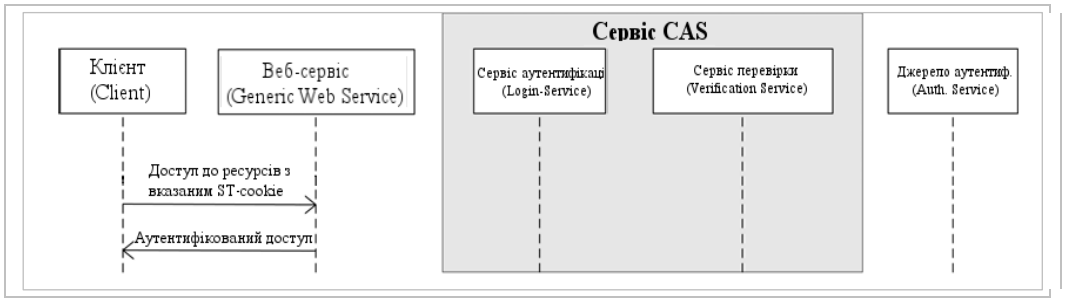

Рисунок 3 – Діаграма процесу доступу до ресурсів, захищених CAS, після аутентифікації користувача на CAS-сервері

Наведений процес складається з наступних етапів:

1. Користувач через броузер надсилає запит до ресурсу, що захищений CAS-сервером. Броузер користувача містить STC.
2. CAS-клієнт ресурсу перевіряє наявність і дійсність STC. Якщо перевірка пройдена, користувачу надається аутентифікований доступ до ресурсу.

3.2 Сервіс Pubcookie

Модель аутентифікації користувачів за допомогою SSO-сервісу Pubcookie базується на наступних компонентах [3]:

1. Pubcookie login server – централізований сервер аутентифікації користувачів, що безпосередньо взаємодіє з користувачами, перевіряє їх логіни та паролі через сервіс зовнішньої аутентифікації, керування куками для позначення аутентифікованих користувачів та надання інформації для сервера ресурсів.
2. Pubcookie application server – сервер ресурсів, здійснює перевірку аутентифікаційної інформації про користувачів, що надходить з сервера

аутентифікації, виконує переадресацію неаутентифікованих користувачів до сервера аутнетифікації, керує куками сесій між користувачами та ресурсами.
3. External authentication service – сервіс зовнішньої аутентифікаіцї користувачів, що здійснює перевірку аутентифікаційної інформації користувачів, надісланої з сервера аутентифікації.
4. User-agent – броузер користувача.

Процес аутентифікації користувача складається з наступних етапів (рис.4) [3,6]:
1. Користувач надсилає запит до сервера ресурсів для отримання доступу до захищених ресурсів.
2. Запит опрацьовується сервером та перевіряється, чи не належить запит до існуючої аутентифікованої сесії з ресурсами та чи не містить в собі інформацію з сервера аутентифікації, необхідної для встановлення нової сесії. Сервер ресурсів генерує відповідь клієнту, що містить посилання на сервер аутентифікації та дві куки: PSC (pre-session cookie) та GRC (granting request cookie), кожна з яких, окрім специфічної інформації, містить також випадкове число, що генерується сервером ресурсів.
3. За допомогою переадресації броузер користувача робить запит на аутентифікацію від сервера аутентифікації. Цей запит містить інформацію про сервер ресурсів, оригінальний ресурс, тобто ресурс, до якого здійснювався перший запит, тип аутентифікації.
4. Сервер аутентифікації опрацьовує GRC та генерує і відправляє броузеру користувача форму для логіна та пароля.
5. Користувач заповнює форму та передає назад на сервер аутентифікації.
6. Сервер аутентифікації отримує дані та відправляє їх на сервіс зовнішньої аутентифікації користувачів на перевірку.
7. Сервер аутентифікації отримує відповідь про перевірку логіна та пароля.
8. Якщо аутентифікація успішна, то сервер аутентифікації генерує відповідь, яка містить переадресацію та дві нові куки. Одна з них – це GC (granting cookie), що містить інформацію про користувача (логін) та деяку іншу інформацію, зокрема випадкове число, що було отримане в GRC. GC підписується цифровим підписом з використанням приватного ключа сервера та шифрується з використанням алгоритму симетричного шифрування за допомогою ключа, що відомий серверам аутентифікації та ресурсів. Друга кука, що називається LC (login cookie), містить інформацію про користувача і застосовується при наступних зверненнях до сервера аутентифікації.
9. Броузер користувача переадресовується на початковий URL сервера ресурсів. Запит броузера містить куку GC, встановлену сервером аутентифікації та pre-session cookie, встановлену раніше сервером ресурсів.
10. Сервер ресурсів опрацьовує запит, знаходить куки GC та PSC. GC розшифровується за допомогою симетричного ключа та перевіряється цифровий підпис за допомогою відкритого ключа сервера аутентифікації. Порівнюється випадкове число з GC з випадковим числом, що зберігається в PSC. Якщо вони однакові, то користувач аутентифіковується в сервері ресурсів, надається відповідний доступ до ресурсів та встановлюється кука сесії між броузером та сервером ресурсів.

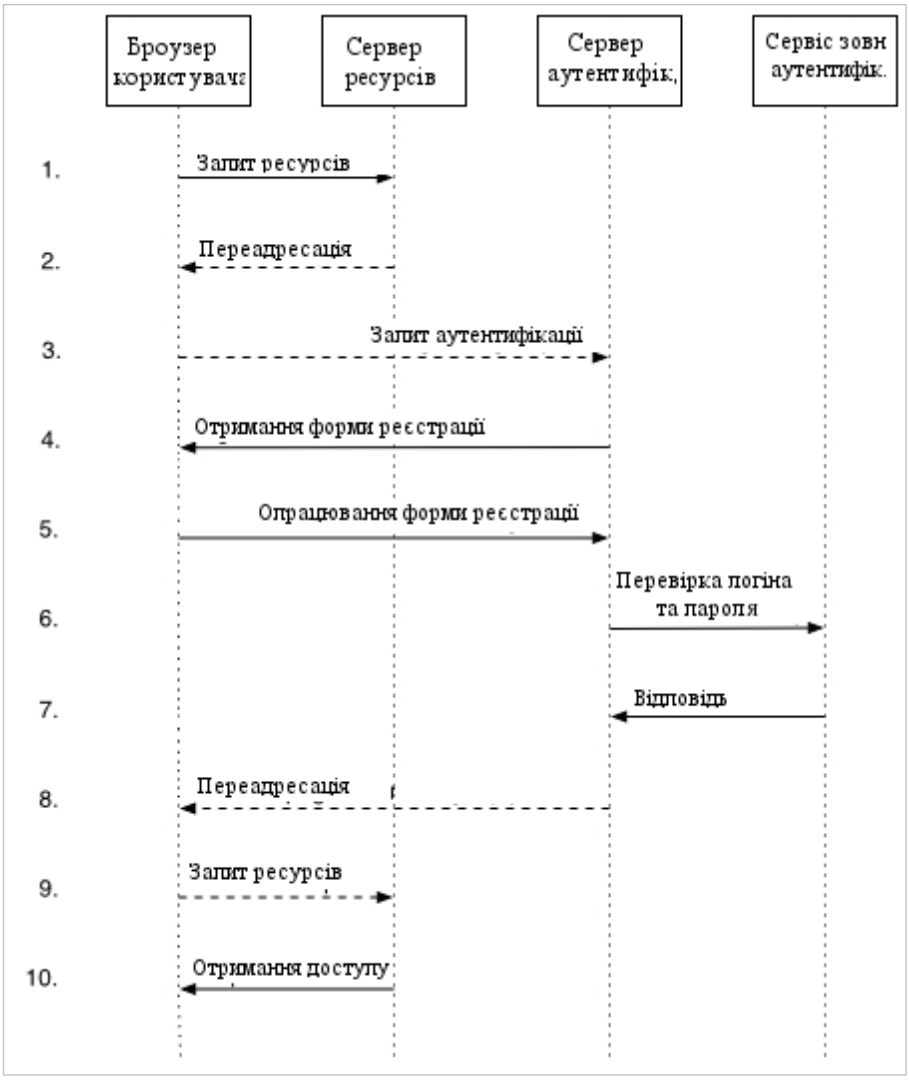

Рисунок 4 – Діаграма процесу доступу до ресурсів, захищених Pubcookie

3.3 Сервіс CoSign

CoSign – SSO-сервіс, що базується на використанні куків для надання доступу аутентифікованих користувачів до ресурсів [4]. Складається з наступних компонентів:

1. CoSign CGI – компонент, що відповідає за аутентифікацію користувачів на центральному CoSign-сервері, завершення користувацьких сесій, відстежування використання користувачами захищених сервісів і стан центральних аутентифікаційних куків.
2. CoSign Daemon – компонент, що відповідає за статус усіх користувацьких сесій з CoSign-сервером, також відстежує статус усіх куків сервісів, що презентують усі захищені ресурси, до яких має доступ користувач. CoSign Daemon має можливість проводити реплікацію бази даних куків на інші хости мережі, що унеможливлює збій всієї системи внаслідок збою центрального CoSign-сервера.
3. CoSign Filter – призначений для використання на прикладних серверах (веб-серверах), для визначення ресурсів, які потрібно захистити. Якщо користувач намагається отримати доступ до захищених ресурсів, то CoSign Filter перевіряє, чи користувач пройшов аутентифікацію та чи дійсні його куки.

Модель роботи, яка показана на рис. 5, базується на взаємодії броузера користувача, ресурсу, що захищений за допомогою CoSign Filter, та, власне, CoSign CGI та Cosign Daemon [5,6]. При доступі користувача до захищеного ресурсу за допомогою броузера по протоколу HTTPS, CoSign Filter генерує ST (Service Ticket),

який передається в якості параметра GET запиту, броузер переадресовується на центральний сервер аутентифікації CoSign CGI. CoSign CGI генерує куку користувача LC (Login Cookie) і передає її разом із формою для аутентифікації назад до користувача. Користувач, ввівши логін та пароль, відсилає форму назад на CoSign CGI (на сервер також передаються куки сервісу і користувача), на сервері проводиться процедура аутентифікації. Якщо процедура проходить успішно, то ST асоціюється з LC та встановлюється її час існування. ST передається як кука броузеру користувача, який переадресовується на захищений ресурс. На захищеному ресурсі CoSign Filter проводить перевірку ST на CoSign CGI, і якщо перевірка проходить успішно, то користувачу надаються права доступу до ресурсу.

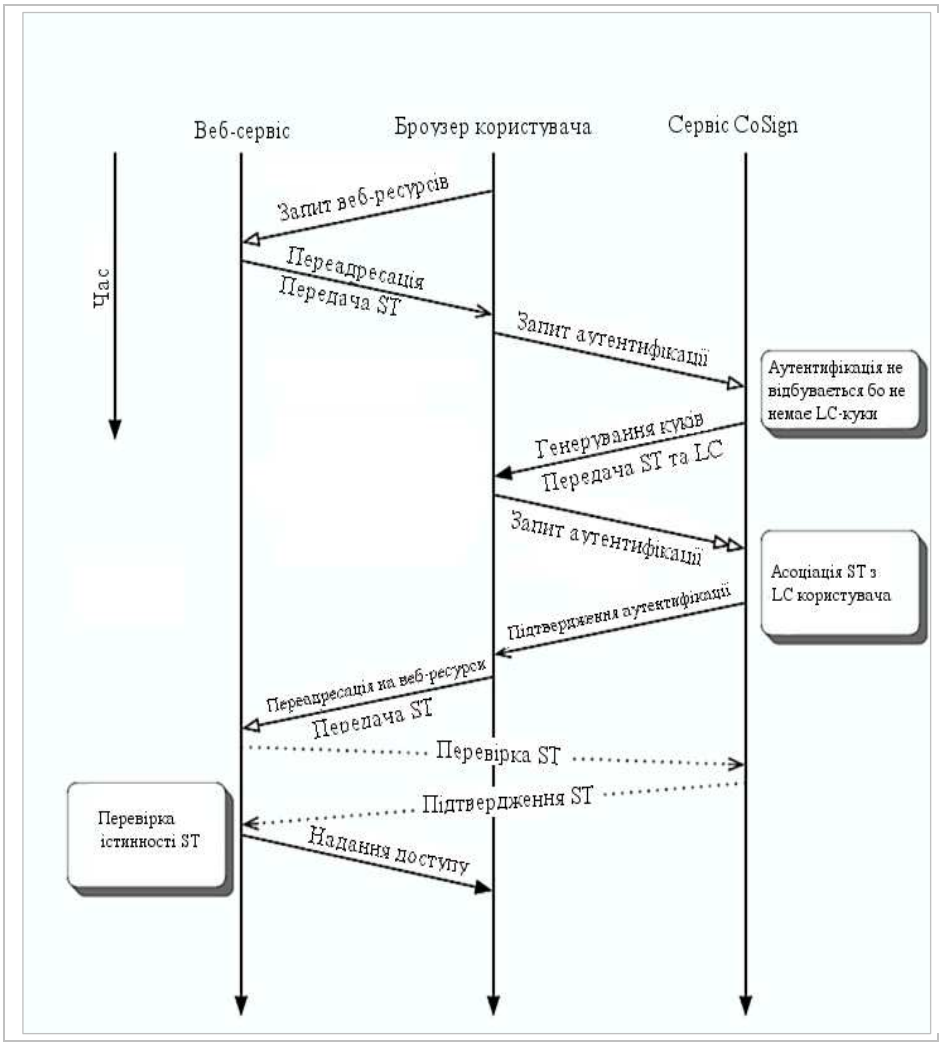

Рисунок 5 - Процес доступу до захищених ресурсів CoSign

4. Федеративна система аутентифікації та авторизації користувачів

Суть федеративної системи аутентифікації та авторизації користувачів полягає у делегуванні інформації про користувачів з одного домену безпеки в інший в межах федерації. Домен безпеки (в даному контексті) – це частина федерації, в якій присутня, загалом, єдина база даних користувачів, діють єдині правила безпеки та доступу до ресурсів. Доменом безпеки, наприклад, може бути університетська мережа з єдиною базою користувачів, переліком ресурсів та системою SSO доступу до цих ресурсів. Федерація – об'єднання доменів безпеки на правах довіри один до одного (trusted parts), що обмінюються даними на основі попередньо визначеного протоколу обміну [7,8].

Використання федеративної системи аутентифікації та авторизації користувачів надає такі переваги:

- єдиний логін та пароль для користувача в межах всієї федерації;
- SSO-аутентифікація до будь-якого ресурсу федерації;
- легкий та “прозорий”(transparent) міждоменний доступ до ресурсів.

Таким чином, для користувачів одного домену безпеки в межах федерації успішна SSO-аутентифікація надає можливість доступу не тільки до ресурсів одного домену, але і до всіх ресурсів федерації (рис.6) [7].

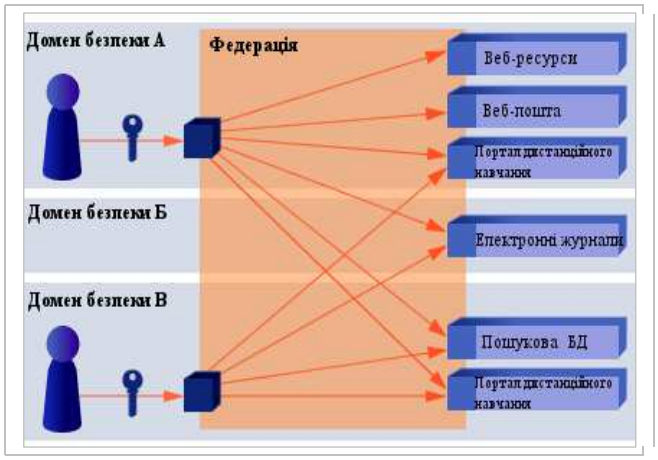

Рисунок 6 – Приклад федеративної системи аутентифікації та авторизації користувачів

4.1 Сервіс Shibboleth

Shibboleth є проектом Internet2 Middleware Initiative, що спрямований на створення архітектури та реалізації федеративної системи аутентифікації та авторизації користувачів на основі SAML. Дана система забезпечує крос-доменну SSO-аутентифікацію, можливість використання провайдерів ідентифікаторів користувачів для надання інформації та сервіс провайдерів, що на основі інформації про користувачів надають доступ до захищених чи закритих ресурсів [9].

4.2 Архітектура Shibboleth

Архітектура Shibboleth базується на наступних стандартах:

- Hypertext Transfer Protocol (HTTP);
- Extensible Markup Language (XML);
- XML Schema;
- XML Signature;
- SOAP;
- Security Assertion Markup Language (SAML).

Функціональні компонети, що забезпечують реалізацію Shibboleth, містять провайдер ідентифікаторів користувачів IdP (Identity Provider), провайдер сервісів SP (Service Provider), сервіс WAYF (“Where are you from” Service) [8].

4.2.1 Провайдер ідентифікаторів користувачів IdP

IdP здійснює керування користувацькими аутентифікаційними даними та їх атрибутами, зокрема підтверджує аутентифікацію користувачів і достовірність їх атрибутів для зовнішніх джерел, зокрема SP. Субкомпоненти IdP показані на рис. 7.

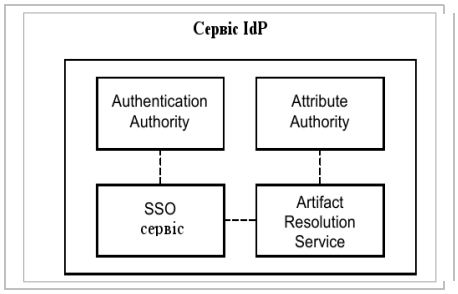

Рисунок 7 – Компонентна будова IdP

Authentication Authority – субкомпонент, що надає дані про аутентифікацію користувачів іншим субкомпонентам. В загальному Authentication Authority інтегрований з сервісом аутентифікації IdP.

Artifact Resolution Service – сервіс, що забезпечує передавання з IdP до SP підтверджень про аутентифікацію на вимогу останнього, з використанням додаткових каналів обміну інформацією між IdP та SP без застосування броузера користувача.

Attribute Authority – опрацьовує запити на отримання атрибутів, завжди вимагає аутентифікації та авторизації запитів.

4.2.2 Провайдер сервісів SP

SP здійснює керування захищеними ресурсами. Доступ користувачів до захищених ресурсів базується на даних, отриманих SP від IdP. Компоненти SP показані на рис. 8.

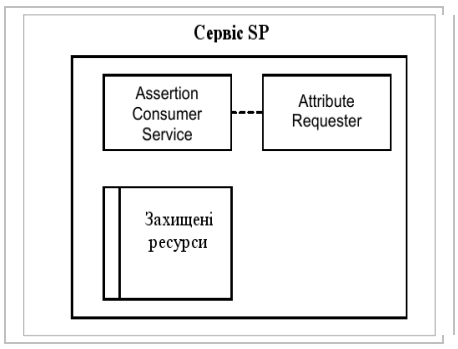

Рисунок 8 – Компонентна будова SP

Assertation Consumer Service (також називається SHIRE) опрацьовує дані аутентифікації, що надаються SSO-сервісом, визначає необхідні додаткові атрибути, встановлює тікети сесій та переадресовує користувачів до ресурсів.

Attribute Requester (також називається SHAR) – сервіс, що забезпечує формування запитів до Attribute Authority IdP на отримання додаткових атрибутів без використання броузера користувача та опрацювання відповідей.

4.2.3 Сервіс WAYF

WAYF-сервіс виконується незалежно від SP та IdP. WAYF-сервіс використовується SP для визначення користувацького IdP. в Загальному випадку WAYF-сервіс виступає переадресатором запитів аутентифікації з SP до SSO-сервісу IdP.

4.3 Модель отримання доступу до ресурсів в Shibboleth

Модель отримання доступу користувача до ресурсів, захищених Shibboleth, показана на рис. 9 [8].

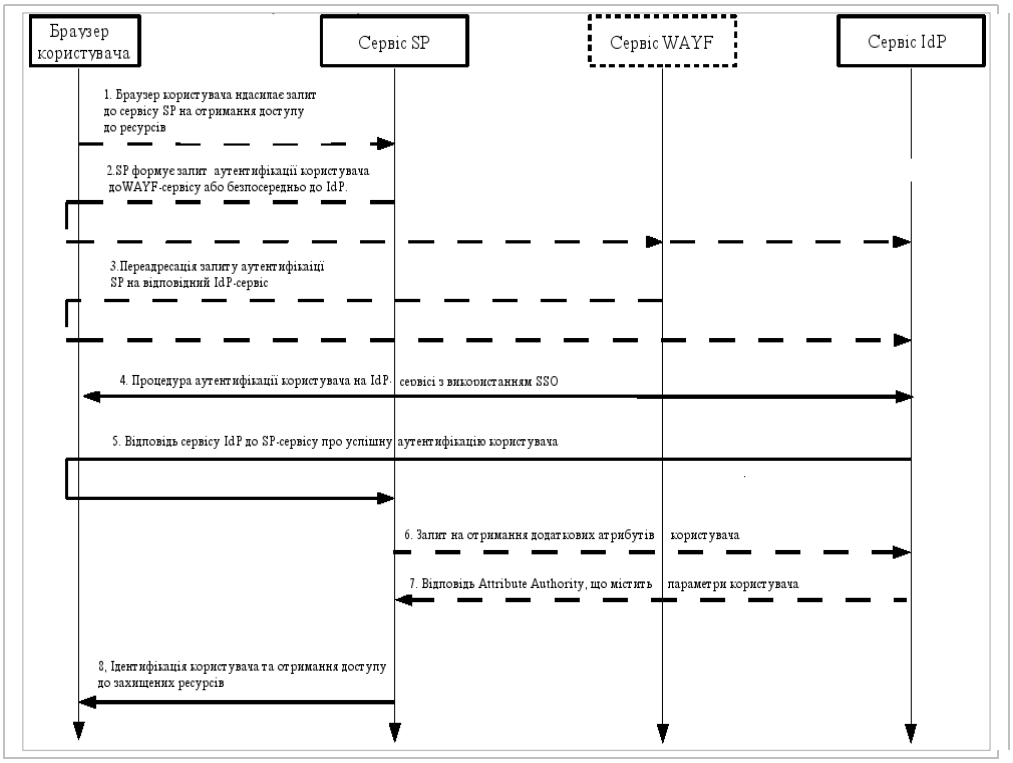

Рисунок 9 – Діаграма отримання доступу на основі Shibboleth

Процес отримання доступу на основі Shibboleth складається з наступних етапів:

1. Користувач, застосовуючи броузер, по HTTP-протоколу надсилає запит на отримання доступу до захищених ресурсів на SP.
2. SP формує запит аутентифікації у вигляді додаткового параметру URL і переадресовує броузер користувача до WAYF-сервісу або безпосередньо до IdP.
3. Якщо використовувався WAYF-сервіс, то він взаємодіє з броузером користувача для вибору IdP та переадресовує на відповідний IdP із запитом аутентифікації від SP.
4. IdP аутентифіковує користувача з використанням SSO-сервісу.
5. Якщо аутентифікація успішна, то IdP використовує спеціальне повідомлення на SAML *<samlp:Response>,* яке через броузер користувача надсилається до SP і містить інформацію про аутентифікацію користувача та про параметри встановлення безпосереднього прямого з'єднання між SP та IdP для отримання додаткових даних.
6. На основі даних, отриманих у попередньому пункті, SP формує запит (якщо це потрібно) *<samlp:AttributeQuery>* до AttributeAuthority IdP.
7. AttributeAuthority IdP на підставі запиту від SP формує повідомлення формату SAML, що може містити один або декілька атрибутів користувача.
8. SP на основі даних, отриманих від IdP, формує відповідь броузеру користувача, що може містити або відмову у доступі до захищених ресурсів, або формування куки сесій та надання доступу до ресурсів.

Висновки

Проведений аналіз концепції одноразової сеансової аутентифікації користувачів для доступу до веб-ресурсів, огляд архітектури та моделей існуючих SSO-сервісів може бути використаний при побудові розподілених веб-сервісів з єдиним аутентифікаційним сервісом SSO. Здійснено огляд концепції федеративної системи аутентифікації та авторизації користувачів, що базується на основі доменів безпеки. Можна застосувати для побудови безпечної розподіленої системи обміну аутентифікаційною інформацією доступу до веб-ресурсів.

Література

1. Authentication World - The Bussines of Authentication. - 2006. - Режим доступу: http://www.authenticationworld.com/. - Заголовок з екрану.
2. Central Authentication Service (CAS) - by JASIG. - 2006. - Режим доступу: http://www.ja-sig.org/products/cas/.- Заголовок з екрану.
3. Pubcookie:open-source software for intra-institutional web authentication. - 2007. - Режим доступу: http://www.pubcookie.org/. - Заголовок з екрану.
4. CoSign: Secure, Intra-Institutional Web Authentication. - 2004. - Режим доступу: http://www.umich.edu/~umweb/software/cosign/. - Заголовок з екрану.
5. The University Login: Authentication for Web Applications – Implementation Comparison University of Auckland, ITSS, 2004.
6. Brian Gilmore, Keith Farvis, John Maddock: Core Middleware and Shared Services Studies. CMSS:Gilmore, 2004.
7. The Swiss Education and Research Network, Authentication and Authorization Infrastructure. - 2007. - Режим доступу: http://www.switch.ch/aai/. - Заголовок з екрану.
8. Scott Cantor.Mace shibboleth arch conformance. University of Washington, NCSA, 2005.
9. Shibboleth Project — Internet2 Middleware. - 2007. - Режим доступу: http://shibboleth.internet2.edu. - Заголовок з екрану.

Одержано 20.09.2007 р.