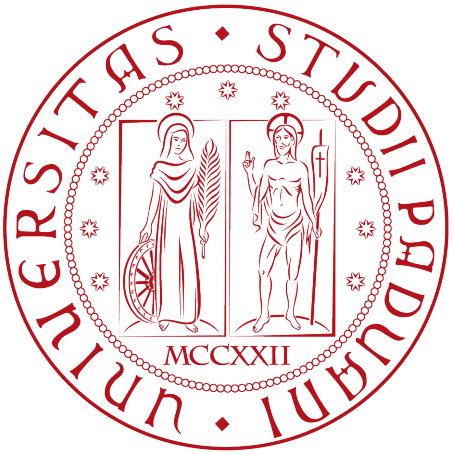
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Integrazione di Single Sign-On in Unix Pluggable Authentication Module (Unix PAM)

Tesi di laurea

Relatore

Prof. Davide Bresolin

Laureando

Ivan Antonino Arena

Anno Accademico 2022-2023

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

— Oscar Wilde

Dedicato a ...

Abstract

Lo scopo della tesi è illustrare il lavoro eseguito con Athesys, system integrator specializzato nello sviluppo di soluzioni di Identity and Access Management (IAM), in termini di innovazione e ricerca applicate all'ambito dell'integrazione di servizi web con sistemi unix-based. Nello specifico, si discute la collaborazione con il team infrastructure ed il team dedicato alla ricerca in materia di identità digitale e la conseguente realizzazione di un componente Single Sign-On compatibile con Unix Pluggable Authentication Module (Unix PAM).

"Life is really simple, but we insist on making it complicated"

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Davide Bresolin, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.
In secondo luogo, vorrei ringraziare di cuore l'azienda ospitante, Athesys Srl, in particolare, il mio tutor esterno Mattia Zago, Roberto Griggio e Leonardo Speranzon, per la disponibilità e l'impegno con cui mi hanno affiancato durante il periodo di stage.

Desidero, inoltre, ringraziare con affetto i miei genitori per avermi appoggiato in ogni mia scelta durante il mio periodo universitario e per avermi fornito il supporto ed i mezzi per portarlo a termine con serenità.

Infine, ci terrei a ringraziare tutte le amicizie più significative che ho stretto a Padova, che mi hanno regalato gioie e sorrisi durante questi tre emozionanti anni.

Padova, Maggio 2023 Ivan Antonino Arena

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

In questo capitolo vengono descritti l'azienda ospitante ed il progetto dell'attività di stage curriculare.

1.1 L'azienda

Athesys Srl è un'azienda di consulenza informatica nata a Padova nel 2010 "dalla sinergia di affermati professionisti del settore IT", specializzata in ambito System Integration, Database Management, Sicurezza applicativa, Governance Cloud Platform, Hyperconvergenza e Sviluppo Software in modalità Agile.

Athesys comprende la start-up Monokee, fondata nel 2017 come soluzione cloud-based per la gestione dell'identità e dell'accesso (IDaaS), la quale offre, come funzionalità principale, un sistema di Single Sign-On basato su diversi tipi di autenticazione, sia passwordless che tramite soluzioni di Self-Sovereign Identity (SSI).

Figura 1.1: Logo di Athesys Srl

Figura 1.2: Logo di Monokee

1.2 Il progetto

L'idea per l'attività di stage nasce proprio dall'esigenza dell'azienda di aumentare la portata di Monokee estendendo il relativo Single Sign-On anche a livello macchina, per poter, successivamente, configurare dei terminali che possano gestire accesso e sessioni degli utenti Monokee già da sistema.

L'obiettivo del progetto del tirocinio era, dunque, era la ricerca e l'eventuale sviluppo di una soluzione che consentisse di accedere a macchine UNIX Debian-like e RHEL-like tramite il proprio account Monokee in modo nativo, sfruttando l'infrastruttura di Single Sign-On fornita dalla start-up.

Il framework da utilizzare era FreeIPA, un gestore delle identità e degli accessi (Identity and Access Management, IAM) gratuito ed open-source che combina tecnologie quali Linux, LDAP, MIT Kerberos, NTP, DNS ed SSSD e consta di un'interfaccia web e di strumenti di amministrazione tramite command-line.

L'attività, dalla durata totale di circa trecento ore, si è sviluppata inizialmente in un fase di ricerca e sperimentazione con l'installazione del software FreeIPA su più macchine virtuali CentOS, RHEL e Ubuntu, messe a disposizione dall'azienda.

La seconda fase è stata dedicata alla ricerca di un metodo che consentisse di effettuare l'autenticazione con il proprio account Monokee su tali macchine; in tal senso, è stata approfondita la parte relativa a Unix PAM (Pluggable Authentication Modules) per studiare la possibilità dello sviluppo di un modulo aggiuntivo.

In seguito a tale ricerca, ho deciso di optare per il sistema di autenticazione tramite Identity Provider esterno messo a disposizione dall'applicativo di FreeIPA e di procedere, dunque, con la configurazione di un'applicazione Monokee OAuth2 e di un provider OpenID Connect (OIDC) che fornissero gli end-point e l'infrastruttura necessaria alla comunicazione con il server di FreeIPA e la successiva implementazione degli stessi su di esso.

Verificato il corretto funzionamento di questo sistema di autenticazione da Command-Line Interface (CLI), ho proseguito cercando di implementare questo sistema anche tramite SSH fino al raggiungimento delle ore previste.

Capitolo 2

Processi e metodologie

In questo capitolo vengono descritte le modalità con cui si è svolto lo stage e le tecnologie utilizzate.

2.1 Organizzazione del lavoro

Il lavoro è stato suddiviso in n periodi.
Nel primo ho fatto xyz

2.2 Tecnologie utilizzate

2.2.1 FreeIPA

FreeIPA è una soluzione open-source gratuita (GNU General Public License) di gestione dell'identità e dell'accesso per ambienti di rete basati su Linux/UNIX, originariamente sviluppato dalla comunità Fedora ed ora supportato da diverse organizzazioni, tra cui Red Hat e la FreeIPA Foundation. Consiste in un insieme di servizi integrati, che consentono di centralizzare l'autenticazione, l'autorizzazione e la gestione degli utenti e delle risorse in un'organizzazione.

FreeIPA è progettato per semplificare la gestione dell'identità e dell'accesso in ambienti di rete complessi, con molti utenti e computer. Consente agli amministratori di gestire facilmente l'accesso degli utenti a risorse e applicazioni, di delegare i privilegi di amministrazione e di definire ed applicare politiche di sicurezza coerenti in tutta la rete, come, ad esempio, limitare l'accesso alle risorse in base al ruolo dell'utente. Per fare ciò, mette a disposizione, oltre che agli strumenti della CLI, un'interfaccia utente web intuitiva per la gestione degli utenti, dei gruppi e delle risorse della rete. Inoltre, FreeIPA è altamente scalabile e può essere distribuito su più server per gestire grandi reti.

Per l'autenticazione degli utenti, FreeIPA utilizza il protocollo Kerberos: gli utenti possono accedere alle risorse della rete utilizzando le loro credenziali Kerberos, senza dover inserire le password ogni volta. Per archiviare e gestire le informazioni sugli utenti, i gruppi e le risorse della rete, invece, utilizza il server di directory open-source 389 Directory Server, il quale offre funzionalità avanzate di ricerca, replica e sincronizzazione.

FreeIPA supporta l'autenticazione SSO tramite il protocollo SAML (Security Assertion Markup Language), ciò significa che gli utenti possono accedere a più applicazioni utilizzando le stesse credenziali di accesso.

2.3 LXC

LXC è l'acronimo di Linux Containers, un sistema di virtualizzazione basato sul kernel Linux che consente di eseguire più sistemi operativi isolati su una singola macchina host. A differenza della virtualizzazione completa, in cui ogni sistema operativo guest ha accesso all'intero hardware dell'host,

Utilizza la virtualizzazione basata sui contenitori, in cui ogni sistema operativo guest condivide le risorse hardware dell'host. La condivisione del kernel fa sì che i container siano molto leggeri e veloci e che abbiano un overhead di risorse molto basso rispetto ad altre tecnologie di virtualizzazione.

LXC fornisce un'interfaccia di riga di comando per la gestione dei container, che consente di creare, avviare, fermare, eliminare e gestire i container in modo semplice ed efficiente. Inoltre, supporta la creazione di immagini di container, che possono essere utilizzate per creare nuovi container in modo rapido e semplice.

Durante l'attività di stage ho utilizzato principalmente container basati su immagini CentOS (Community Enterprise Operating System), una distribuzione Linux basata su Red Hat Enterprise Linux (RHEL) particolarmente adatta all'uso in ambiente server, che offre una vasta gamma di funzionalità e strumenti per gestire un'infrastruttura IT.

2.4 SSH

Secure Shell (SSH) è un protocollo di rete crittografato utilizzato per la gestione sicura di dispositivi di rete e per l'accesso remoto a sistemi informatici. Il protocollo SSH fornisce un canale di comunicazione sicuro tra due dispositivi, garantendo l'integrità, la riservatezza e l'autenticità delle informazioni trasmesse.

L'autenticazione avviene attraverso l'uso di chiavi pubbliche e private: in questo metodo, un'entità che desidera accedere a un sistema remoto genera una coppia di chiavi, una pubblica e una privata; la chiave pubblica viene fornita al sistema remoto, mentre la chiave privata viene conservata dall'entità; quando l'entità si connette al sistema remoto, la chiave privata viene utilizzata per autenticare l'entità.

Inoltre, SSH utilizza la crittografia per proteggere i dati trasferiti tra i dispositivi. In particolare, il protocollo utilizza la crittografia a chiave simmetrica per proteggere i dati durante la trasmissione, e la crittografia a chiave pubblica per autenticare le parti coinvolte.

Capitolo 3

Studio tecnologico

In questo capitolo vengono illustrati nel dettaglio i concetti e le tecnologie utlizzate.

3.1 Single Sign-On

Il Single Sign On (SSO) è una tecnologia che consente agli utenti di accedere a più applicazioni e servizi utilizzando un'unica identità di accesso. In pratica, l'utente inserisce le proprie credenziali di accesso una sola volta e successivamente può accedere a tutte le applicazioni e servizi che supportano l'SSO senza dover inserire nuovamente le credenziali, alternativamente a come accade con l'autenticazione tradizionale. Ciò può risultare particolarmente vantaggioso se l'utente necessita di accedere a molte applicazioni o servizi diversi.

L'SSO funziona attraverso l'utilizzo di un'autorità di autenticazione centralizzata, chiamata Identity Provider (IdP). L'IdP autentica l'utente e fornisce un token di sicurezza che contiene le informazioni sull'utente e sui servizi a cui ha accesso. Questo token può essere utilizzato per accedere a tutti i servizi che supportano tale sistema.

Per utilizzare l'SSO, le applicazioni e i servizi devono supportare uno dei protocolli SSO standard, come SAML (Security Assertion Markup Language) o OpenID Connect. Questi protocolli definiscono il modo in cui le informazioni di autenticazione dell'utente vengono trasmesse tra le diverse applicazioni e servizi.

L'SSO offre, dunque, numerosi vantaggi, tra cui una maggiore comodità per gli utenti, una maggiore sicurezza attraverso l'utilizzo di token di sicurezza a breve termine e una maggiore efficienza nella gestione delle identità e delle autorizzazioni degli utenti. Tuttavia, l'SSO richiede una pianificazione e una configurazione adeguata per garantire la sicurezza e la protezione dei dati degli utenti.

3.2 Self-Sovereign Identity

L'SSI (Self-Sovereign Identity) è un nuovo approccio alla gestione delle identità digitali che consente agli utenti di possedere, controllare e condividere le proprie informazioni di identità in modo sicuro e privato. A differenza dei sistemi di identità tradizionali, in cui le informazioni di identità sono conservate in modo centralizzato da terze parti, l'SSI consente agli utenti di essere i proprietari esclusivi dei propri dati di identità digitali.

L'SSI si basa sulla tecnologia blockchain, che consente di creare registri distribuiti di informazioni sicure e immutabili. In tal modo, le informazioni di identità degli utenti vengono conservate in modo decentralizzato e sicuro, senza la necessità di un'autorità centralizzata di controllo.

Per utilizzare l'SSI, gli utenti creano un'identità digitale che include le informazioni di identità necessarie, come nome, indirizzo e informazioni di contatto. Questa identità digitale viene conservata sulla blockchain e protetta da una chiave privata unica, che solo l'utente possiede.

Gli utenti possono utilizzare la propria identità digitale SSI per accedere a servizi online e condividere le proprie informazioni di identità solo con le parti che desiderano. Questo viene fatto attraverso l'utilizzo di un protocollo di scambio di informazioni sicuro e decentralizzato, chiamato DID (Decentralized Identifier).

L'SSI offre numerosi vantaggi, tra cui un maggiore controllo e privacy per gli utenti rispetto ai sistemi di identità tradizionali, una maggiore sicurezza attraverso l'utilizzo della tecnologia blockchain e una maggiore efficienza nella gestione delle identità digitali. Tuttavia, è ancora una tecnologia emergente e richiede una maggiore adozione e sviluppo per diventare un approccio mainstream alla gestione delle identità digitali.

3.3 OAuth2

OAuth2 è un protocollo di autorizzazione che consente a un'applicazione di accedere alle risorse di un utente senza richiedere le credenziali dell'utente - e, di conseguenza, senza memorizzarle - nato per assicurare l'accesso sicuro e controllato ai dati di un utente da parte di applicazioni di terze parti.

Funziona attraverso una serie di flussi di autorizzazione, in cui l'utente concede l'autorizzazione all'applicazione per accedere alle sue risorse. L'applicazione, a sua volta, ottiene un token di accesso che può essere utilizzato per accedere alle risorse dell'utente.

Il protocollo OAuth2 è utilizzato da molte grandi piattaforme online come Google, Facebook e Twitter ed è diventato, di fatto, uno standard nei servizi cloud, nei social network, nei servizi di pagamento online e in molti altri contesti.

3.4 OpenID Connect

OpenID Connect (OIDC) è un protocollo di autenticazione basato su OAuth2, utilizzato per l'autenticazione degli utenti in applicazioni web e mobile. Progettato per risolvere il problema dell'autenticazione sicura e decentralizzata in applicazioni di terze parti, consente agli utenti di utilizzare l'SSO per accedere a diverse applicazioni, senza, quindi, dover creare un nuovo account per ogni applicazione, bensì delegando l'autenticazione ad un provider esterno (OpenID Provider).

OIDC fornisce un framework standard per l'autenticazione basata su JSON Web Tokens (JWT), in cui l'utente viene autenticato una sola volta e poi viene rilasciato un token di accesso contenente le informazioni di base dell'utente, come l'identificatore univoco, il nome e l'email, che può essere utilizzato per accedere alle risorse protette.

Questo protocollo è stato adottato da molte grandi piattaforme online, tra cui Google, Microsoft e Amazon ed è anche supportato da molte librerie di sviluppo, caratteristica che lo rende semplice da implementare per gli sviluppatori.

3.5 Linux PAM

Linux PAM (Pluggable Authentication Modules) è un framework di autenticazione per i sistemi operativi Linux e UNIX che consente di configurare diversi metodi di autenticazione, come quella tramite password, a due fattori, basata su token, biometrica, ecc.

Utilizzato in una vasta gamma di applicazioni e servizi, tra cui il sistema di login del sistema operativo ed il server SSH, il framework di PAM è composto da una serie di moduli, ognuno dei quali implementa una particolare funzionalità di autenticazione. I moduli PAM sono progettati per essere "pluggable", ovvero possono essere facilmente sostituiti o aggiunti senza dover modificare il codice sorgente del sistema operativo.

L'architettura modulare di PAM consente di creare una catena di moduli, in cui ciascuno dei quali può verificare una parte dell'identità dell'utente. Ad esempio, un modulo può verificare la password dell'utente, mentre un altro può verificare il certificato del client. Se uno qualsiasi dei moduli nella catena fallisce, l'intero processo di autenticazione viene interrotto. Inoltre, è possibile sviluppare dei moduli personalizzati ed integrarli nelle diverse funzioni che richiedono PAM.

Capitolo 4

Configurazione dei sistemi

In questo capitolo vengono descritti i procedimenti attuati per configurare i sistemi utilizzati.

4.1 Creazione ambienti virtuali

4.2 Configurazione FreeIPA

4.3 Configurazione Monokee

Capitolo 5

Ricerca e sperimentazione

In questo capitolo viene descritto il processo di ricerca e sperimentazione di una soluzione efficace per l'implementazione del SSO nativo

5.1 Unix PAM

La prima idea che ho avuto per integrare il Single Sign-On di Monokee via SSH sul container CentOS che ho predisposto è stata quella di creare un nuovo modulo PAM. Ciò perché, studiando i file presenti al percorso `/etc/pam.d/` ho trovato il file `sshd`, che stabilisce i moduli da utilizzare per autenticazione, autorizzazione, sessione e gestione della password. In un primo momento mi sono concentrato sulla parte di autenticazione, controllando il file di configurazione `common-auth`, incluso in `/etc/pam.d/sshd`, che rappresenta l'autenticazione predefinita di UNIX (con password memorizzata localmente).

Tuttavia, l'installazione di FreeIPA sovrascrive il parametro `UsePam yes` del file `/etc/ssh/sshd_config` anteponendo dei parametri relativi a Kerberos, in modo da potersi autenticare con la password dell'utente FreeIPA specificato nel prefisso della macchina nel comando di SSH.

La mia idea era, dunque, quella di rimuovere questi parametri e tornare all'autenticazione via PAM, sostituendo però il modulo predefinito con uno creato appositamente per l'SSO di Monokee.

Il problema restava quello del riconoscimento dell'utente ma avevo già pensato a diversi modi in cui poterlo risolvere, così ho deciso di proseguire e sperimentare con lo sviluppo di un modulo PAM di test, per verificare la fattibilità della mia intuizione.

INSERIRE MODULO

5.2 FreeIPA IdP

Dato che la soluzione con il modulo PAM si è rivelata essere più impegnativa del previsto, ho deciso di provare a configurare l'SSO con Monokee da FreeIPA.

Navigando nell'interfaccia web del software, infatti, ho notato che nella sezione *Authentication > Identity Provider Servers* era possibile definire un Identity Provider che utilizzasse OAuth 2.0 come protocollo di autenticazione.

A questo punto, con l'aiuto del team, ed in particolare del CTO di Athesys Srl, mi sono spostato sull'infrastruttura di testing di Monokee per configurare un'applicazione OAuth2 da poter utilizzare come Identity Provider per FreeIPA.

All'interno dell'ambiente di test, tramite l'interfaccia web,

Capitolo 6

Implementazione e documentazione

In questo capitolo viene illustrato il processo di implementazione della soluzione trovata e la stesura della documentazione relativa.

6.1 Configurazione FreeIPA

6.2 Configurazione Monokee

6.3 Testing

6.4 Documentazione

L'azienda ha richiesto la redazione di una guida che illustrare il processo di configurazione del server FreeIPA e dei sistemi di Monokee per l'integrazione del Single Sign-On sulle macchine UNIX, per fornire una base documentativa per facilitare le future progressioni e sperimentazioni relative. Ho steso tale documentazione in formato Markdown, versionando il codice sul repository GitHub aziendale fornito da Athesys Srl.

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

L'attività è stata svolta quasi totalmente in linea con la pianificazione prevista: non ci sono stati ritardi di alcun tipo ed il primo periodo, quello di studio delle tecnologie, ha richiesto meno tempo di quanto preventivato, consentendomi, così, di approfondire ulteriormente lo sviluppo delle soluzioni trovate nelle fasi successive. Ho completato tutti gli obiettivi richiesti con successo, ad inclusione di quelli desiderabili e opzionali.

7.3 Conoscenze acquisite

Grazie allo stage con Athesys Srlmi sono addentrato in un campo dell'informatica che poco conoscevo, quello della sicurezza. Ho avuto modo di conoscere i concetti e le tecnologie più significative del momento presente in ambito di identità digitale, come la Self-Sovereign Identity, il Single-Sign On ed alcuni dei protocolli di autenticazione ed autorizzazione più diffusi, apprendendo, anzitutto, cosa significa creare e gestire un'identità digitale e quali sono i rischi di sicurezza legati ad essa. Oltre ad una già ampia formazione teorica, ho avuto anche la possibilità di migliorare le mie competenze in ambito di sistemi UNIX, entrando a contatto con parti di codice che cambiano direttamente il comportamento del sistema operativo, come i moduli PAM e l'SSH. Inoltre, ho imparato a creare dei container LXC dalle immagini messe a disposizione e, successivamente, a configurare un server di Identity and Access Management, quale FreeIPA, modificando anche, in alcuni casi, manualmente dei file di sistema. Infine, ho messo in atto le conoscenze acquisite nella prima fase dell'attività, in particolare quelle riguardanti il funzionamento di OAuth2 ed OIDC, per implementare il PoC richiesto.

7.4 Valutazione personale

Dopo circa trecento ore passate al fianco del team di Athesys Srlsono convinto di aver acquisito delle conoscenze e delle competenze, non strettamente tecniche, fondamentali per il mio ingresso prossimo nell'industria: questa esperienza, che costituisce la mia prima nell'ambito del percorso che mi appartiene, quello dell'informatica, mi ha

permesso di affrontare personalmente e toccare con mano le sfide, i problemi, le metodologie ed i traguardi propri della realtà delle aziende informatiche. A partire dalla comunicazione, dall'organizzazione e dalla gestione del tempo e delle risorse, arrivando poi agli effettivi processi risolutivi e di sviluppo, sento di aver ricevuto un contributo signifcativo e di essermi messo alla prova, applicandomi al meglio in un ambiente a me quasi del tutto sconosciuto, al di fuori della mia zona di comfort. Ora, al momento della stesura di questo documento, ripercorrendo ciò che ho fatto durante questa attività di stage, mi rendo conto ancora meglio del valore che essa ha avuto ed ha per me e per la mia carriera.

Acronimi e abbreviazioni

CLI Command-Line Interface. 2, 17

Bibliografia

Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Siti web consultati

Manifesto Agile. URL: http://agilemanifesto.org/iso/it/.