ILIESI digitale
Memorie

Roberto Vitali

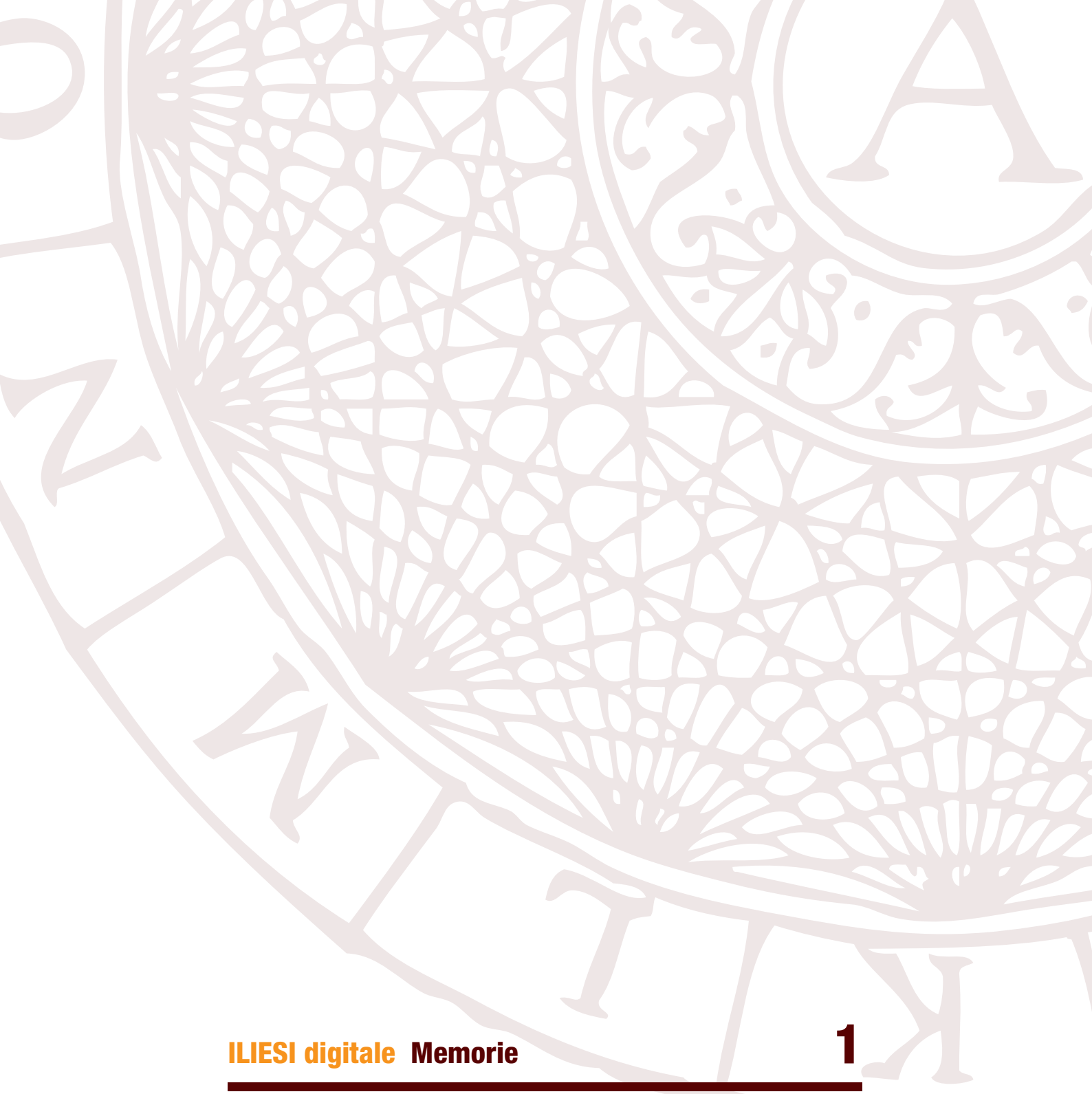
LINEE GUIDA PER LA CONFIGURAZIONE DI UN AMBIENTE OPERATIVO CHE OSPITI UN SOFTWARE LEGACY

Lo studio di un caso:
Ripristino di un sistema con SearchServer™

ILIESI CNR

2014

Istituto per il Lessico Intellettuale Europeo e Storia delle Idee



ILIESI digitale Memorie 1

Secondo le norme dell'ILIESI tutti i volumi pubblicati nella collana sono sottoposti a un processo di *peer review* che ne attesta la validità scientifica

Segreteria di redazione
Maria Cristina Dalfino
Progetto grafico
Silvestro Caligiuri

ISSN
2283-5571

Roberto Vitali

LINEE GUIDA PER LA CONFIGURAZIONE DI UN AMBIENTE OPERATIVO CHE OSPITI UN SOFTWARE LEGACY.

Lo studio di un caso: ripristino di un sistema con SearchServer™

Abstract

The installation of a System that have to host a legacy software and, at the same time, guarantee a certain security and availability degree, is a task that must take care of several aspects. Legacy software, in fact, makes it harder to build a secure system. This is because one of the principal security issues concern the presence of bugs in software. Each software, in general, has some bugs, and the number of bugs increases as the software complexity does. Such errors are discovered time by time during the software utilization, once they are known, they can be corrected by the software developers, who care of producing and distributing software updates. A not-updated software presents errors not corrected, that may produce the system crash. Often, since they may be known, can be exploited by attacker for taking control of the system.

This document wants to propose some guidelines for the installation and configuration of a operating environment that have to host a legacy software. It is used as a case study the configuration of a system for hosting SearchServer™, which release had been distributed ten years earlier.

Keywords: Legacy Software, Security, System Configuration, Linux, Operating System

Parole chiave: Software legacy, Sicurezza, Configurazione di Sistema, Linux, Sistema Operativo

1. PREMESSA

Il presente documento vuole fornire delle linee guida per l'installazione di un sistema aggiornato che deve ospitare un software legacy.[1] Per avvalorare le indicazioni proposte viene esaminato un caso di studio reale, i cui dettagli più specifici sono stati omessi in quanto informazioni confidenziali.
Il caso in esame è il ripristino del software SearchServer™, (data di rilascio 2004) in dotazione all'Istituto per il Lessico Intellettuale Europeo e Storia delle Idee (ILIESI – CNR). Il software era precedentemente installato su una macchina "SUN FIRE V20z Server" con Sistema Operativo Red Hat Enterprise Linux 3.[2] Il sistema, a seguito del mancato mantenimento in aggiornamento e, di conseguenza, presentando un ridotto livello di sicurezza, è stato spento.
Al momento dell'inizio del lavoro, il Sistema Operativo installato sulla macchina presentava una configurazione non ottimale, sia per la gestione delle risorse che come grado di sicurezza. In particolare:

- il sistema presentava due hard-disk non ridondanti, cioè non configurati in RAID[3] (ovvero configurati in RAID0). Uno era altamente sottoutilizzato, presentando unicamente la partizione di boot, nell'altro era installato il resto del sistema;
- il sistema aveva installato componenti software non necessarie: per esempio l'ambiente desktop, nonostante fosse una macchina dedicata al solo utilizzo del software in questione;
- i servizi attivi presentavano le configurazioni di default, con conseguenti ripercussioni sul lato della sicurezza;
- non era presente alcun meccanismo di sicurezza (ad esempio firewall, I.D.S.[4]);
- il sistema non era aggiornato.

[1] I sistemi legacy sono sistemi obsoleti che non possono, per motivi tecnici o di pianificazione aziendale, essere rimpiazzati o sostituiti, e che devono, tuttavia, poter interagire con sistemi più aggiornati.
[2] Red Hat.Com.
[3] Wikipedia.Org.
[4] Gli Intrusion Detection Systems (o I.D.S.) sono dei sistemi di rilevazione delle instrusioni.

Questa configurazione non ottimale non ha garantito alcuna protezione minimale contro i guasti e ha, inoltre, esposto il sistema a possibili attacchi informatici.

Specifiche generali del lavoro in esame

Viene richiesto uno studio sulla configurazione (software) ottimale su cui poter installare SearchServer™, in particolare si auspica un Sistema Operativo che abbia un ciclo di vita tale da permettere una buona continuità del servizio, senza vincoli di licenze. Eventualmente, il sistema installato deve poter essere mantenuto operativo per servizi futuri, anche in caso di dismissione del software SearchServer™.

Alla luce dei risultati di tale studio è richiesto che venga installato il sistema più consono e che venga poi configurato per aumentarne il livello di sicurezza. In caso di incompatibilità del software con nuovi Sistemi Operativi viene richiesta una riconfigurazione del sistema già installato ponendo attenzione agli aspetti di sicurezza e performance.

Lavoro svolto

Nella prima fase del lavoro sono state installate delle macchine virtuali[5] per verificare la compatibilità di SearchServer™ con un Sistema Operativo aggiornato. Nella documentazione del software è indicata esplicitamente la sola compatibilità con Red Hat Enterprise Linux 3, non dando quindi garanzie di piena compatibilità con altri sistemi. Dopo gli esiti positivi della fase di valutazione della compatibilità, viene scelto come Sistema Operativo Debian[6] versione 7 a 64 bit. Debian è una distribuzione[7] GNU/Linux[8] altamente affidabile e stabile, particolarmente adatta a operare come ambiente server. La versione 7 è stata rilasciata il 4 maggio 2013, il che garantisce quindi una buona continuità al sistema. Sono state fatte delle copie di backup del sistema in dismissione, una a livello di file, per permettere il recupero dei dati e la migrazione de-

[5] Una macchina virtuale è un software che emula il comportamento di un computer, grazie all'assegnazione di parte delle risorse a disposizione del computer che ospita la macchina virtuale.

[6] Debian.Org.

[7] Le distribuzioni Linux sono distribuzioni di software che compongono un Sistema Operativo basato sul kernel Linux. Il kernel implementa il *core* di un Sistema Operativo, la distribuzione va a definire anche tutti i software operano al di spora del kernel.

[8] Gnu.Org.

gli stessi e una a livello di byte.[9] Questo permette di installare il sistema su una macchina virtuale e, nel caso di impossibilità nell'installazione di un nuovo sistema, permette il ripristino del sistema precedentemente installato. Una volta verificata la compatibilità del software con la versione scelta del Sistema Operativo, si è proceduto ad all'installazione. Il sistema è stato quindi configurato per aumentare il livello di sicurezza. Tutte le singole fasi della configurazione sono più specificatamente discusse in seguito.

2. ANALISI E VALUTAZIONE DELL'AMBIENTE OPERATIVO PER SUPPORTARE SEARCHSERVER™

La realizzazione di un nuovo sistema impone uno studio preliminare di fattibilità e un'attenta fase di progettazione e validazione della soluzione. Lo studio, per quanto riguarda la parte tecnica, deve dimensionare il sistema e, qualora non si tratti di una soluzione integrata, valutare strategie di integrazione dei diversi sottosistemi che lo andranno a comporre.

Il dimensionamento dell'architettura deve tener conto del carico di lavoro cui sarà soggetto il sistema. Un sistema sottodimensionato, non avendo risorse sufficienti a soddisfare tutte le richieste in arrivo, soffrirà di un basso grado di *disponibilità*, cioè la capacità del sistema di rispondere alle richieste di un utente autorizzato. Al contrario uno eccessivamente sovradimensionato produrrà uno spreco di risorse e, di conseguenza, economico. Nel caso in esame, l'infrastruttura hardware a disposizione dell'istituto è risultata adeguata per il carico di lavoro del servizio da ripristinare. Il servizio offerto era infatti già operativo precedentemente sulla macchina e l'installazione di un nuovo Sistema Operativo, adeguatamente configurato, non avrebbe influenzato negativamente le prestazioni dello stesso.

In questo caso il primo parametro da tenere in considerazione riguarda

[9] Nella copia dei file, questi sono copiati e sono salvati fisicamente in celle di memoria che in genere non rispecchiano l'ordine di partenza. Nella copia a livello dei byte, sono le singole celle di memoria (il loro contenuto) a essere copiate, mantenendo lo stesso ordine. Questa seconda modalità di salvataggio dei dati occupa più spazio in quanto considera anche le celle che non contengono dati significativi.

l'*affidabilità* del sistema. Questa è definita come la capacità del sistema di non mostrare dei malfunzionamenti in un dato intervallo temporale, dato che nessun malfunzionamento esisteva all'inizio dell'intervallo. Un sistema affidabile determina un alto grado di *disponibilità* dello stesso. L'*affidabilità* di un sistema è influenzata da molti fattori legati a criticità software, hardware e di infrastruttura. La progettazione di un sistema altamente affidabile deve tener conto di tutti gli aspetti che potrebbero rendere non disponibile il servizio: dalla continuità dell'alimentazione, al mantenimento dell'integrità dei dati, alla gestione del carico di lavoro fino agli errori nel software e agli accessi non autorizzati che potrebbero compromettere il sistema. Un sistema altamente affidabile deve in ultima istanza prevedere meccanismi di *Disaster Recovery*, ovvero di mantenere l'operatività anche in caso di calamità naturali, quali terremoti, alluvioni e, in generale, eventi non prevedibili. In questo caso è necessaria la replicazione dell'infrastruttura in un luogo abbastanza lontano per cui una medesima catastrofe, con molta probabilità, non inficia entrambi i sistemi.

Per mantenere un buon livello di *affidabilità* bisogna innanzitutto replicare i diversi componenti che compongono il sistema. Ogni componente infatti è soggetto guasti che possono verificarsi nel tempo. A seconda del grado di *affidabilità* che si vuole ottenere si possono creare molteplici repliche e adottare dei meccanismi di decisione che, anche in presenza di più guasti, siano in grado di determinare il corretto funzionamento del sistema.

Le repliche possono essere sia *calde* che *fredde*. Nel primo caso tutte le repliche sono operative contemporaneamente, questo meccanismo permette la non interruzione del sistema anche in presenza di guasti singoli. Nel secondo caso le repliche non sono attive ma mantengono uno stato coerente con le componenti sane. In quest'ultimo caso, nel momento del guasto di un componente attivo, è necessario un certo intervallo di tempo, in cui il servizio risulterà non disponibile, per permettere alle repliche di attivarsi.

Le operazioni da intraprendere a livello software per aumentare l'*affidabilità* riguardano l'integrità dei dati e una configurazione software che prevenga il manifestarsi di guasti, o ripristini il sistema appena questi si verifichino. Altri meccanismi a livello software necessari per ottenere un elevato livello di *affidabilità* riguardano l'aumento del livello di sicurezza

che sarà argomento della sezione 4.
I meccanismi per l'integrità dei dati si attuano principalmente tramite l'uso di File System Journaled e della tecnologia RAID. Se i dati gestiti sono più complessi, come ad esempio un DataBase, l'integrità dei dati viene mantenuta anche attraverso apposite configurazioni.
Vista la pianificazione dell'Istituto, il lavoro in esame non ha previsto investimenti infrastrutturali e hardware, quindi gli sforzi per aumentare l'affidabilità del sistema sono rivolti alla componente software.
Cruciale nella scelta della soluzione da adottare è il risultato dello studio sull'integrazione di SearchServer™ con un nuovo Sistema Operativo. I potenziali problemi di compatibilità sono molteplici:

- l'implementazione del software potrebbe essere specifica per una determinata distribuzione (o un insieme di distribuzioni) Linux;
- il software potrebbe usare dei servizi di sistema diventati ormai obsoleti e non più supportati da Sistemi Operativi aggiornati;
- l'incompatibilità con un Sistema Operativo a 64 bit.

Gestione di Sistemi Legacy

I sistemi legacy possono presentare un'interfaccia non pienamente compatibile con sistemi più moderni, sia per l'uso di differenti tecnologie, che per l'esigenza di funzionalità non presenti, creando problemi per la loro interazione che vanno considerati fin dalla progettazione della nuova architettura di cui saranno parte.
La soluzione che spesso viene usata per mantenere operativo un sistema legacy in un contesto aggiornato è quella di implementare dei *wrapper* (fig. 2.1), ovvero sistemi che si interpongono tra il sistema legacy e quello moderno. Questo tipo di soluzione viene efficacemente usata in contesti dove l'architettura è multi-tier, ovvero strutturata su più livelli. Spesso infatti un sistema informatico può essere diviso in tre livelli, formando quello che si chiama un sistema 3-tier (fig. 2.2): il livello più basso è dato dal livello *dati*, che gestisce le informazioni e la loro memorizzazione; il livello intermedio è quello *logico*, dove viene implementata l'elaborazione dei dati ovvero dove risiede la logica del sistema; il livello superiore, chiamato di *presentazione*, implementa la presentazione dei dati all'utente finale.

Una classica situazione in cui si vuole mantenere in vita un sistema legacy è quando il livello della logica è molto stabile e la sua ri-progettazione o re-implementazione con una tecnologia più moderna richiede un grande sforzo in termini di tempo e costi. In tal caso si può sviluppare un nuovo livello di presentazione che sia conforme con gli eventuali nuovi requisiti dell'applicazione e compatibile con i client che lo devono interrogare. Questo livello di presentazione dovrà interagire con il sistema legacy.

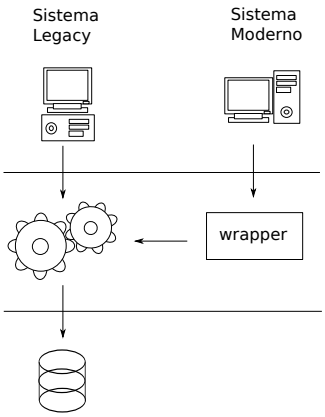


Figura 2.1: Architettura con livelli dati e logica legacy

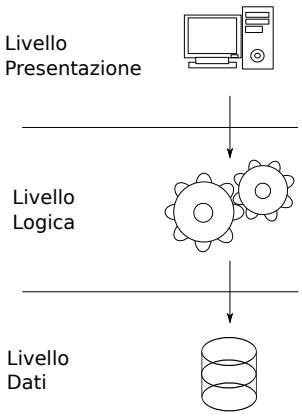


Figura 2.2: Architettura 3-tier

Nel caso in esame si voleva ripristinare l'intera architettura del sistema legacy. SearchServer™ presenta un'architettura 3-tier: il livello dei *dati* è costituito da una serie di file; la *logica* del sistema è costituita dal motore di indicizzazione e ricerca, implementato in Java;[1] come livello

[1] Java.com.

di *presentazione* sono implementate alcune JPS.[2]

L'uso di JSP[3] garantisce tuttora una perfetta compatibilità nell'esportazione dei suoi servizi anche verso i sistemi più aggiornati. Il problema di compatibilità più importante che si è presentato consiste nella presenza di software Java compilato a 32 bit.

Nonostante i problemi di compatibilità emersi, la scelta è stata quella di puntare su un sistema a 64 bit. Tale scelta è stata dettata, oltre che dai vantaggi che comporta l'adozione di un sistema a 64 bit,[4] dall'essere l'architettura di riferimento già da ora e per il prossimo futuro, garantendo quindi un maggior supporto. L'istituto, infatti, ha chiesto una soluzione che potesse avere un buon ciclo di vita, con una configurazione ottimale e predisposta a ospitare servizi in futuro.

Sono state analizzate diverse soluzioni.

Una prima soluzione percorribile sarebbe stata quella di installare su un nuovo Sistema Operativo una macchina virtuale con la copia esatta della vecchia installazione. Questo avrebbe permesso di avere un sistema di base aggiornato e con un lungo ciclo di vita, e, allo stesso tempo, il sistema da ripristinare completamente configurato e funzionante. Tuttavia questa soluzione non avrebbe risolto i problemi di sicurezza legati al sistema installato nella macchina virtuale, essendo lo stesso non più aggiornabile.

Una seconda soluzione sarebbe stata quella di creare comunque un ambiente virtuale dove ospitare un nuovo sistema con installato il software SearchServer™. In questo caso si sarebbe potuto far convivere più servizi in un regime di isolamento, garantito dalla virtualizzazione. Tuttavia per motivi di risorse limitate, come lo spazio dei dischi, si è preferito non creare una macchina virtuale unicamente per ospitare il software.

Si è scelto quindi di installare SearchServer™ in un ambiente operativo nuovo, essendo attualmente la macchina dedicata al suo uso esclusivo.

---

[2] Oracle.

[3] Le JSP (o Java Server Pages) sono una tecnologia, basata su Java, che genera dinamicamente, sul server dove risiedono, delle pagine web. Le JSP rappresentano una delle possibili implementazioni del livello di *presentazione* in applicazioni web-based.

[4] AMD Inc.; Intel.com.

Installazione di un ambiente di emulazione

La fase di studio sulla configurazione ottimale del sistema di base su cui installare SearchServer™ ha previsto l'installazione di quattro diverse macchine virtuali per verificare la compatibilità del software con altrettanti ambienti operativi. I diversi ambienti sono stati scelti per permettere uno studio che esaminasse diverse configurazioni, con un approccio incrementale, partendo dalla versione inizialmente installata presso l'Istituto, fino ad arrivare a quella ipotizzata come definitiva. In particolare sono stati installati i seguenti ambienti:

- RedHat Enterprise Linux 3 a 32 bit;
- Fedora 19[5] a 64 bit;
- Debian 7 a 32 bit;
- Debian 7 a 64 bit.

La prima macchina è stata installata con una copia esatta delle partizioni del sistema precedentemente installato. Questo per permettere in ogni momento l'accesso al vecchio sistema, sia per studiare la configurazione precedente, che per recuperare eventuali dati non considerati durante il backup dei singoli file.
La seconda macchina ha avuto lo scopo di testare l'installazione su un ambiente che fosse il più simile possibile alla distribuzione indicata dal software,[6] ma presentasse una licenza che permettesse l'aggiornamento del sistema senza vincoli. Questo avrebbe permesso, nonostante il vincolo sulla scelta della distribuzione, un Sistema Operativo aggiornato e aggiornabile.
La terza macchina è stata usata per verificare la compatibilità con un sistema Debian a 32 bit. Questo sistema si presenta con la distribuzione preferibile e la stessa architetture (32 bit) del sistema di partenza. Nelle prime tre macchine virtuali si è scelto quindi di cambiare distribuzione Linux, senza modificare l'architettura (32 bit contro 64 bit). Il cambio di architettura avrebbe potuto infatti presentare maggiori problemi di compatibilità rispetto al cambio della distribuzione.

[5] FedoraProject.Org.

[6] I rilasci di software RedHat Enterprise Linux infatti derivano da quelli di Fedora, una volta che hanno passato una intensa fase di testing e sono considerati sufficientemente stabili.

La quarta macchina rappresenta la configurazione su cui si è puntato per il nuovo sistema: una distribuzione stabile e affidabile che sfruttasse l'architettura a 64 bit del processore.

3. INSTALLAZIONE DEL SISTEMA

Creazione dei backup

L'installazione di un nuovo sistema su una macchina dove era già presente un sistema con dei dati sensibili richiede come primo passo un backup dei dati.

Sono stati eseguiti due diversi tipi di backup. Il primo backup è stato a livello di singoli file, per permettere il recupero dei dati sensibili e la migrazione degli stessi sull'eventuale nuovo sistema. Il secondo è stato a livello di byte, copiando interamente le partizioni per permettere l'installazione del sistema su una macchina virtuale e, eventualmente, per studiare nuove soluzioni.

Installazione del Sistema Operativo

Il Sistema Operativo è stato installato tramite CD-Rom. È stata effettuata una installazione minimale per aumentare l'*affidabilità* del sistema (vedi 4.2). Il sistema viene installato con la sola interfaccia a riga di comando (nessuna ambiente desktop), dovendo essere usato come server.

Partizionamento

La macchina presenta due hard-disk (stesso modello) da 73 GB, il sistema è stato installato mettendo i dischi in RAID1 (modalità MIRROR), per garantire una maggior *affidabilità* dei dati. La modalità RAID1 infatti prevede che ogni scrittura su disco sia replicata su tutti i dischi, così da avere tutti i dischi sempre coerenti. Questo permette di non perdere le informazioni nel caso in cui uno dei due dischi si guasti.

Sono state create diverse partizioni, per permettere una configurazione specifica per ognuna di esse e incrementare la sicurezza del sistema (vedi 4.3). Le partizioni sono state configurate con File System[1]

[1] Il File System è la struttura logica dei file.

ext4[2] che, essendo un File System Journaled, permette il recupero di informazioni e meta-informazioni in alcuni casi di danneggiamento dello stesso.

Gestione dei pacchetti a 32 bit

La più grande rivoluzione a livello di architettura hardware degli ultimi anni è molto probabilmente l'estensione dei processori a 64 bit. Per decenni l'architettura usata nei processori dei computer è stata a 32 bit, dove il numero di bit si riferisce alla dimensione delle celle di memoria all'interno dei processori e dell'indirizzamento a memoria. Questi sistemi potevano indirizzare fino a 4 GB di locazioni di memoria. Con l'abbassamento del costo della memoria si sono costruiti sistemi equipaggiati con memoria sempre maggiore. L'architettura dei processori ha dovuto seguire questa tendenza per poter sfruttare pienamente le potenzialità dei computer. La scelta di adottare questa architettura hardware ha avuto degli effetti sulla compatibilità di alcuni software.

I test di compatibilità del software SearchServer™ con un ambiente a 64 bit, hanno dato esito positivo. Tuttavia l'installazione del software ha richiesto una particolare configurazione del Sistema Operativo. Infatti, il funzionamento di SearchServer™ e dei servizi ad esso associati richiede obbligatoriamente un ambiente a 32 bit. In particolare il codice Java precompilato richiedeva delle librerie a 32 bit. Debian permette l'installazione di software a 32 bit anche su un sistema a 64 bit, tramite la configurazione del suo gestore di pacchetti (dpkg).

Tale gestore, infatti, tramite il comando `dpkg --add-architecture i386` permette di specificare ulteriori architetture (nel caso specifico i386, ovvero a 32 bit), oltre a quella di default, per cui possano essere installati i pacchetti.

Gestione degli Utenti

Sul sistema sono stati creati due utenti, *utente1* e *utente2*.

- *utente1* è l'utente standard con cui accedere e gestire il sistema;
- *utente2* è l'utente tramite cui mandare in esecuzione SearchServer™ e i suoi servizi.

[2] Wikipedia.Org.

L'uso di un utente, con dei permessi minimi, unicamente per i servizi di SearchServer™, permette di limitare possibili conseguenze dovute a errori (intenzionali o no) nell'esecuzione del software. La versione del software infatti è stata sviluppata nel 2003 e potrebbe quindi presentare degli errori, anche noti, che non sono stati corretti.

L'accesso al sistema dell'utente *root* da remoto è stato disabilitato.

Avvio Automatico

Il sistema è stato configurato per permettere l'esecuzione automatica di SearchServer™ all'avvio. Questo è stato fatto tramite un apposito script che viene eseguito dal sistema in fase di avvio in quanto SearchServer™ non viene installato come servizio. Questo meccanismo garantisce che SearchServer™ andrà in esecuzione in automatico dopo ogni riavvio del sistema, senza la necessità di alcun intervento manuale.

4. CONFIGURAZIONE DEL SISTEMA PER LA MESSA IN SICUREZZA

La sicurezza di un Sistema Informatico

La sicurezza di un Sistema Informatico è un fattore chiave sia per le aziende che per le amministrazioni pubbliche. Specialmente in un contesto dove vengono esposti dei servizi.

Un Sistema Informatico con un basso livello di sicurezza mette a rischio la *disponibilità* dei servizi che offre. Un sistema che offre servizi, specialmente se di pubblico dominio, è maggiormente esposto ad attacchi informatici poiché tale sistema rimane raggiungibile costantemente ed è facilmente indirizzabile.

Le tecniche per attaccare un Sistema Informatico sono molteplici e si sono evolute nel tempo. Molti attacchi sfruttano i bug (errori) presenti nei software installati, e, tramite questi bug, sono in grado di effettuare accessi non consentiti al sistema stesso. Altre tipologie di attacco sfruttano delle cattive configurazione dei software installati. I software, infatti, in genere sono distribuiti con delle configurazioni che non prediligono l'aspetto della sicurezza informatica. Questo perché i software

potrebbero essere installati su sistemi con diverse esigenze in termini di sicurezza, e una configurazione ''troppo'' sicura potrebbe in alcuni andare a discapito dell'usabilità dello stesso, soprattutto in presenza di utenti poco esperti. Alcuni attacchi mirano a esaurire le risorse a disposizione del sistema generando dei comportamenti non conformi nei software. Altri attacchi sono basati sulla falsificazione dell'identità, permettendo l'accesso al sistema attaccato oppure ottenendo informazioni confidenziali. Una tale varietà di tipologie di attacco impone uno scrupoloso lavoro per la messa in sicurezza di un sistema.
I problemi che derivano da un attacco informatico sono molteplici.
Il primo problema riguarda l'*integrità* e la *confidenzialità* dei dati. Se infatti avviene un accesso fraudolento al sistema, c'è la possibilità che i dati presenti siano letti (problema di *confidenzialità*), o modificati (problema di *integrità*) da un utente non autorizzato. Questa problematica ha un impatto tanto più importante quanto più i dati gestiti dal sistema siano *dati sensibili*.
Il secondo problema che può sorgere quando si subisce un attacco è legato alla non *disponibilità* del servizio. Il disservizio può causare dei disagi, tanto maggiori se tale servizio è pubblico, con una relativa caduta di immagine e quindi economica, soprattutto nel caso di aziende.
Il terzo problema consiste nel controllo parziale o totale di un sistema per commettere attacchi informatici verso altri sistemi. In questo caso chi ha attaccato il sistema può:

- installare un bot, ovvero un software che esegue in automatico dei compiti. I bot in genere sono usati per effettuare degli attacchi di tipo DDoS;[1]

- transitare attraverso il sistema compromesso per sferrare un attacco a un terzo sistema. Questa procedura ha il vantaggio di rendere meno tracciabile l'attaccante. A volte questa tecnica è usata in quanto il sistema attaccato può garantire un accesso facilitato all'obiettivo finale, ad esempio qualora per l'obiettivo finale esso sia un sistema fidato.

[1] Gli attacchi di tipo DDoS (Distributed Denial of Service), consistono in un invio distribuito, ovvero da più sorgenti, di richieste a un obiettivo con il fine di saturare le risorse e interrompere la disponibilità del servizio.

Nel caso in cui il sistema sia stato usato per eseguire attacchi informatici o azioni fraudolente, rischia di essere etichettato come nocivo. Un chiaro esempio è quando un sistema attaccato viene usato per effettuare *spamming*.[2] In questo caso il sistema rischia di essere inserito in delle *blacklist*,[3] con la possibile conseguenza di non poter più effettuare determinati servizi.
Le principali contromisure ai tipi di attacco più comuni includono:

- un aggiornamento continuo dei software presenti nel sistema, per non avere software con bug noti presenti;
- un'attenta configurazione dei software presenti nel sistema e del sistema stesso;
- l'installazione di software o meccanismi per incrementare il livello di sicurezza.

Di seguito sono riportate le contromisure apportate al sistema in esame per aumentarne il livello di sicurezza.

Disattivazione di servizi non necessari

La sicurezza di un sistema si basa anche sulla disattivazione di servizi non necessari. Maggiore è il numero di servizi attivi, maggiore è la possibilità che qualcuno di essi presenti dei bug o sia mal configurato, permettendo un accesso non autorizzato al sistema o il crash del sistema. Inoltre attivare un servizio che non è necessario sottrae, anche se in maniera contenuta, risorse al sistema.

Partizionamento del Sistema

Il partizionamento dei dispositivi di memorizzazione permette la protezione del sistema contro diversi tipi di attacco. Le diverse partizioni possono infatti essere montate con parametri diversi, abilitando per ogni partizione solo le caratteristiche di cui necessita.
Il partizionamento effettuato ha limitato la possibilità di montare partizioni ed eseguire codice (con privilegi propri o altrui).

[2] Lo spamming è l'invio di messaggi, generalmente pubblicitari, senza il consenso del ricevente. In molti paese è considerato un reato.

[3] Le blacklist sono delle liste contenenti un elenco di Indirizzi Internet da cui sono partite delle azioni fraudolente e quindi sono considerati non fidati.

Il partizionamento ha lo scopo anche di limitare l'azione di attacchi mirati contro le partizioni, come ad esempio per esaurimento dello spazio disco.[4] Questo tipo di attacco non sfrutta una cattiva configurazione della partizione, ma la possibilità di scrivere dati arbitrariamente grandi sul dispositivo di memorizzazione. La saturazione dello spazio sul dispositivo causa l'impossibilità del sistema di operare. Partizionando il disco, la saturazione di alcune partizioni non critiche non compromette l'intero sistema.

Configurazione di parametri di rete

Essendo il sistema un ambiente server, quindi accessibile dalla rete, sono stati configurati anche i parametri di rete per aumentare la sicurezza del sistema tramite la modifica del file *sysctl.conf*. La configurazione di default dei parametri di rete, come degli altri parametri del sistema, si basa sul presupposto che il Sistema Operativo (Debian in questo caso) è di uso generico. Questo infatti può essere usato in molti contesti diversi, ad esempio come un ambiente desktop.

Sicurezza della rete: Firewall

Il firewall è un componente (hardware o software) che funziona da filtro tra la rete e l'host, analizzando e operando sui pacchetti[5] in transito. I firewall possono essere *stateless*, interpretando unicamente i singoli pacchetti, o *statefull*, interpretando anche lo stato delle connessione.

La configurazione è stata effettuata in modo tale da permettere l'accesso solo ai servizi che si vogliono esporre, usando il classico paradigma per cui viene negato tutto, ad eccezione dei servizi che si espongono esplicitamente. La configurazione posta in essere ha considerato le necessità per il corretto funzionamento del sistema di base, garantendo gli accessi ai soli servizi necessari.

Per aumentare ulteriormente la sicurezza del sistema nel firewall sono stati impostati meccanismi per contrastare attacchi basati su numero e stato delle connessioni, come gli attacchi di tipo floodying.[6]

[4] Cert.org.

[5] Il pacchetto è la più piccola unità di dato che viene scambiata in una *rete a pacchetto*, che rappresenta la rete classica dove operano i computer.

[6] Cert.org.

Selinux

Per incrementare ulteriormente la sicurezza del sistema è stato installato e configurato Selinux.[7] Selinux è un sistema di sicurezza implementato a livello del kernel Linux per supportare un controllo degli accessi di tipo *mandatorio*. Gli accessi alle risorse vengono controllati a livello kernel e le politiche per discriminare gli accessi consentiti sono caricate nel sistema e non possono essere modificate da utenti o programmi non abilitati. Le politiche standard di sicurezza di Linux, invece, implementano meccanismi di sicurezza di tipo *discrezionale*. In un tale ambiente gli utenti possono, tramite dei comandi o delle applicazioni, cambiare i permessi sulle risorse che gli appartengono e quindi accedervi indiscriminatamente. L'utente *root* può operare su tutte le risorse, anche quelle di cui non è il proprietario, cambiare i permessi e avere accesso quindi a tutte le risorse.

Un'altro grande vantaggio in termini di sicurezza di un sistema con Selinux attivo risiede nel maggior grado di dettaglio con cui è possibile identificare le risorse e i permessi su queste risorse. In questo modo si riesce a limitare ogni accesso garantendo solo le operazioni ammissibili.

Inoltre Selinux abilita l'accesso alle risorse sulla base del processo in esecuzione, quindi due diversi processi in esecuzione di uno stesso utente in genere, non hanno lo stesso accesso alle diverse risorse e nemmeno alla stessa risorsa.

Selinux, tramite le policy di default, riconosce e protegge i servizi standard attribuendo ai relativi processi solo i permessi loro indispensabili. Selinux è stato quindi ulteriormente configurato secondo le esigenze particolari dell'Istituto.

[7] SelinuxProject.

Bibliografia

AMD Inc.= *AMD64 Architecture Programmer's Manual Volume 2: System Programming*. http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2012/10/24593_APM_v21.pdf

Cert.org= *Debian*. http://www.cert.org/advisories/CA-1996-21.html

Cert.org= *Denial of Service*. http://www.cert.org

Debian.Org= *Debian*. http://www.debian.org/

FedoraProject.Org= *Fedora*. http://fedoraproject.org/

Gnu.Org= *Gnu/Linux*. https://www.gnu.org/gnu/linux-and-gnu.html

Intel.com= *Take advantage of the memory features of 64 bit computing*. http://software.intel.com/en-us/articles/take-advantage-of-the-memory-features-of-64-bit-computing

Java.com= *Java*. http://www.java.com

Oracle= *Java Server Pages*. http://www.oracle.com/technetwork/java/jsp-138432.html

Red Hat.Com= *Red Hat Linux Enterprise*. http://www.redhat.com/products/enterprise-linux/

SelinuxProject= *Selinux*. http://selinuxproject.org

Wikipedia.Org= *Filesystem EXT4*. http://en.wikipedia.org/wiki/Ext4

Wikipedia.Org= *RAID*. http://en.wikipedia.org/wiki/RAID

Istituto per il Lessico Intellettuale Europeo e Storia della Idee CNR

www.iliesi.cnr.it

ILIESI digitale **Memorie**

2014

Istituto per il Lessico Intellettuale Europeo e Storia delle Idee