r t Anno 2009_Numero 102

rapporti tecnici INGV

Nuovo server SSH - INGV Roma. Progettazione e Realizzazione

102

Istituto Nazionale di Geofisica e Vulcanologia

Anno 2009_Numero 102

rapporti tecnici

INGV

NUOVO SERVER SSH – INGV ROMA. PROGETTAZIONE E REALIZZAZIONE

Pietro Ficeli, Melissa Mendicino, Massimiliano Rossi, Emanuele Sammali, Manuela Sbarra, Gianpaolo Sensale, Diego Sorrentino, Francesco Zanolin, Lucio Badiali, Francesca Caprara, Pierluigi Cau

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Centro Nazionale Terremoti – Servizi Informatici e Reti)

102

Sommario

1. Introduzione

Il crescente bisogno di utilizzare le infrastrutture informatiche dell'Ente da postazioni al di fuori della propria sede lavorativa rende necessario realizzare sistemi sicuri, flessibili, affidabili e veloci per poter espletare il proprio lavoro in qualsiasi momento senza però inficiare la sicurezza della rete aziendale.

I servizi più diffusi, adottabili anche in ambito *casalingo*, sono principalmente due:

VPN
: Virtual Private Network;

SSH
: Secure SHell.

La prima é già attiva in Sede ormai da più di un anno, la seconda necessita una completa ristrutturazione per garantire l'accesso ai richiedenti senza esporre la rete interna i rischi dovuti ad accessi non autorizzati (vedi paragrafo 1.1).

1.1. SSH Server

Il servizio *SSH* é un protocollo client/server che permette di stabilire una sessione cifrata con interfaccia a linea di comando tra due host in rete.
Il client *SSH* ha una interfaccia a linea di comando simile a quella di *TELNET* e *RLOGIN*, ma l'intera comunicazione (ovvero sia l'autenticazione che la sessione di lavoro) avviene in maniera cifrata. Per questo motivo *SSH* é diventato uno standard di fatto per l'amministrazione remota di sistemi *Unix* e di *dispositivi di rete*, rendendo obsoleto il protocollo *TELNET*, giudicato troppo pericoloso per la sua mancanza di protezione contro le intercettazioni.

2. Attuale servizio SSH

L'attuale servizio *SSH*, operativo sulla macchina `relay.ingv.it`, permette l'accesso ad alcune macchine di rete **BLU**[1] (nascoste alla *Big Internet* tramite firewall - rete *Militarizzata*-) che, a loro volta, permettono l'accesso alla rete interna dell'Ente. Dopo l'autenticazione con password **cifrata**, uno script di login richiede a quale nodo (*server*) si vuole accedere e, se si dispone di un account sulla macchina richiesta, lo script si preoccupa di reindirizzare automaticamente l'utente sulla macchina richiesta (che si preoccuperà autonomamente dell'autenticazione).

2.1. Vulnerabilità dell'attuale servizio

Verranno di seguito illustrate le vulnerabilità riscontrate nel sistema in uso:

2.1.1. Prima vulnerabilità

Disponendo di uno username e di una password validi si può tentare un attacco allo script di login! Dopo aver tentato un semplice attacco lo script é crashato chiudendo la sessione ma mostrando i programmi utilizzati per effettuare il salto in rete **BLU** (il test di sicurezza effettuato si é fermato qui per non arrecare seri danni al servizio, vedi paragrafo 2.2).
Un veloce controllo sulla *BUGTRACK* del programma crashato permette di conoscere le sue vulnerabilità e realizzare rapidamente un *exploit* per accedere come **super utente** (*root*) alla macchina (`relay.ingv.it`) e prenderne il **TOTALE CONTROLLO**.

2.1.2. Seconda vulnerabilità

La macchina che ospita il servizio *SSH* é condivisa con il server *FTP*, servizio nato per lo scambio dati. In seguito agli ultimi attacchi, in cui ignoti utilizzavano l'accesso in scrittura al server come utente anonimo per scambiare ogni tipo di file -**solitamente illegale**-, si é deciso di restringere l'accesso in scrittura al server ai soli utenti autorizzati e autenticati da password. Purtroppo l'accesso in *FTP*, per come fu concepito, permette l'autenticazione unicamente in *plain-text* ergo username e password dell'utente viaggiano in chiaro sulla rete. Un semplice *sniffing* del traffico di rete permette di conoscere tutti gli username e password che utilizzano il servizio *FTP* e, con molta probabilità, il servizio *SSH*, vanificando l'utilizzo di un servizio con password cifrata!

2.1.3. Terza vulnerabilità

A seguito di una scansione dei servizi di rete (vedi paragrafo 2.2.6) risulta evidente una grave mancanza nell'amministrazione del servizio: l'aggiornamento dei servizi attivi!
Dal test precedente menzionato si evince che la versione del servizio SSH attivo é *OPENSSH 3.6.1P2* mentre il Sistema Operativo[2] in uso é un *GNU/LINUX 2.4.18 - 2.4.20*.
Con una veloce ricerca sulla Rete sui siti di riferimento di tali software possiamo notare che la versione attualmente disponibile per il server SSH é la *OPENSSH 4.7: SEPTEMBER 4, 2007* mentre quella per il kernel é *LINUX 2.4.35.4 - 2.6.23.9*...

[1] La *rete Blu* è una sezione della rete pubblica INGV, sotto operatore GARR, dedicata allo scambio dati tra server interni e pubblici della rete INGV. Al suo interno sono contenuti server non direttamente raggiungibili dalla *BigInternet* e possono comunicare esclusivamente con alcune macchine presenti nella rete interna e con alcune macchine della rete pubblica INGV, sotto operatore GARR.
[2] In informatica, un sistema operativo (abbreviato in **SO**, o all'inglese **OS**, **operating system**) é il programma responsabile del diretto controllo e gestione dell'hardware che costituisce un computer e delle operazioni di base. Si occupa dei processi che vengono eseguiti e della gestione degli accessi degli utenti. Compito del sistema operativo é inoltre quello di virtualizzare le risorse hardware e software nei confronti dei programmi applicativi.

Figura 1. OpenSSH attualmente disponibile.

Figura 2. OpenSSH operativo sul server.

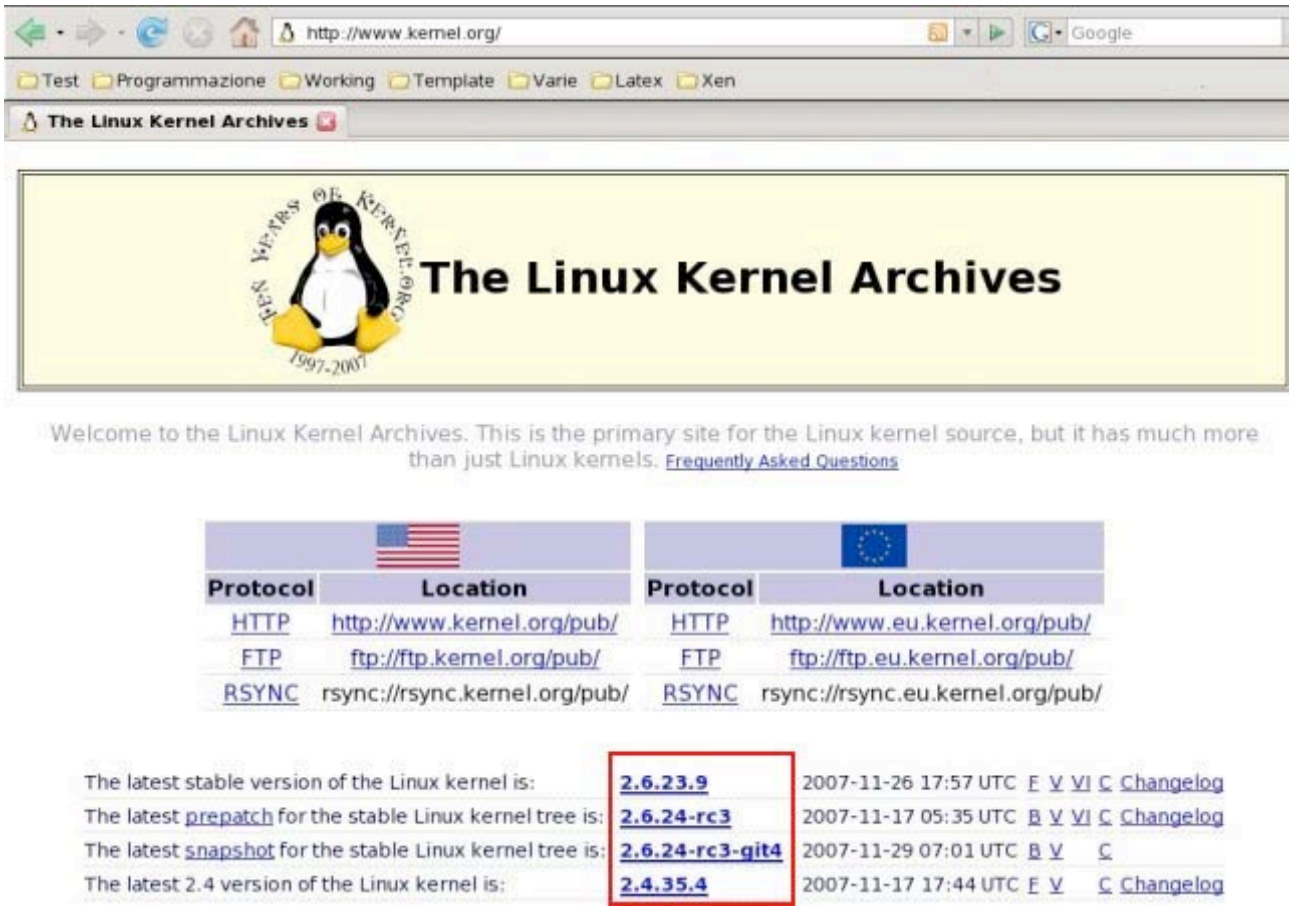

Figura 3. Kernel GNU/Linux attualmente disponibile.

2.2. Test di sicurezza effettuati

Di seguito verranno illustrati solo alcuni dei test effettuati per forzare il sistema. Le operazioni di forzatura sono state precedentemente concordate con il Responsabile del S.I.R. (Servizi Informatici e Reti).
Da precedenti prove si è notato che al momento dell'autenticazione viene eseguito uno script che utilizza il comando *grep* per ricercare il nome della macchina all'interno di una lista di server.
Il comando prevede almeno due argomenti, il testo da ricercare e il nome del file in cui effettuare la ricerca.
Passando come nome file il carattere hyphen `-' si impone al comando *grep* di ricercare il testo all'interno di una stringa passata per standard input (in questo caso la tastiera).
In tutte le prove è stata sfruttata questa potenzialità passando direttamente al comando il testo in cui ricercare il nome del server, facendo così andare a buon fine ogni richiesta e avendo così la possibilità di tentare l'accesso in una macchine della rete militarizzata.

2.2.1. Test 1

In questo test si é provato a sfruttare le chiavi del super utente (se presenti) per accedere a uno dei nodi del sistema. Passando al comando *grep* tutta la stringa SSH, comprensiva di chiave **RSA** (se presente), si è tentata l'autenticazione al server *nettuno.ingv.it.*

```
Accesso dal nodo  in data Mon Nov 26 20:03:04 CET 2007
Account in uso : sorrentino

Inserire il nome del nodo a cui connettersi :'root@nettuno.ingv.it\ \-i\ /root/id_rsa' -
grep: /root/id_rsa': Permission denied
root@nettuno.ingv.it
Il nodo 'root@nettuno.ingv.it -i /root/id_rsa' - non e' abilitato per il remote login
Connection to relay.ingv.it closed.
```

Figura 4. Accesso come root.

2.2.2. Test 2

Come il precedente test, si è provato a scansionare il sistema alla ricerca di chiavi di autenticazione precedentemente inserite dall'amministratore del servizio.

Figura 5. Ricerca di chiavi nel sistema.

2.2.3. Test 3

Provando nuovamente l'autenticazione come nei precedenti test abbiamo notato che la macchina *plutone.ingv.it* permetteva l'autenticazione. Su questa macchina si è potuto tentare un accesso senza disporre di un account.

Figura 6. Tentato accesso come altro utente.

2.2.4. Test 4

Si è provato nuovamente l'attacco precedente utilizzando un utente che dispone anche di un account FTP sulla stessa macchina. Sniffando il traffico FTP si possono conoscere i parametri di connessione dell'utente sfruttato per lanciare l'attacco. In una situazione normale il 90% di questi attacchi va a buon fine. Non è stato effettuato alcuna analisi del traffico l'operazione non era stata concordata con il Responsabile del SIR.

Figura 7. Richiesta di password come utente differente.

2.2.5. Test 5

Dopo alcune prove si desume la struttura del server e si possono lanciare attacchi cercando di sfruttare chiavi private di altri utenti o semplicemente si tenta l'accesso tramite password sniffata.

Figura 8. Tentativo di sfruttamento di chiave altrui.

2.2.6. Test 6

Con un semplice ma potente tool per analizzare i servizi di rete offerti da una macchina, come *NMAP*[3], é possibile conoscere molte informazioni sulle versioni del software in uso e, sempre riferendosi alla *BUGTRACK*, si ha un ottimo punto di partenza per sferrare nuovi attacchi! Inoltre, sempre utilizzando lo stesso software, é possibile risalire alla versione del Sistema Operativo (OS) in uso e, combinando le eventuali falle dell'OS e dei servizi attivi si ha una buona probabilità che un attacco vada a buon fine.

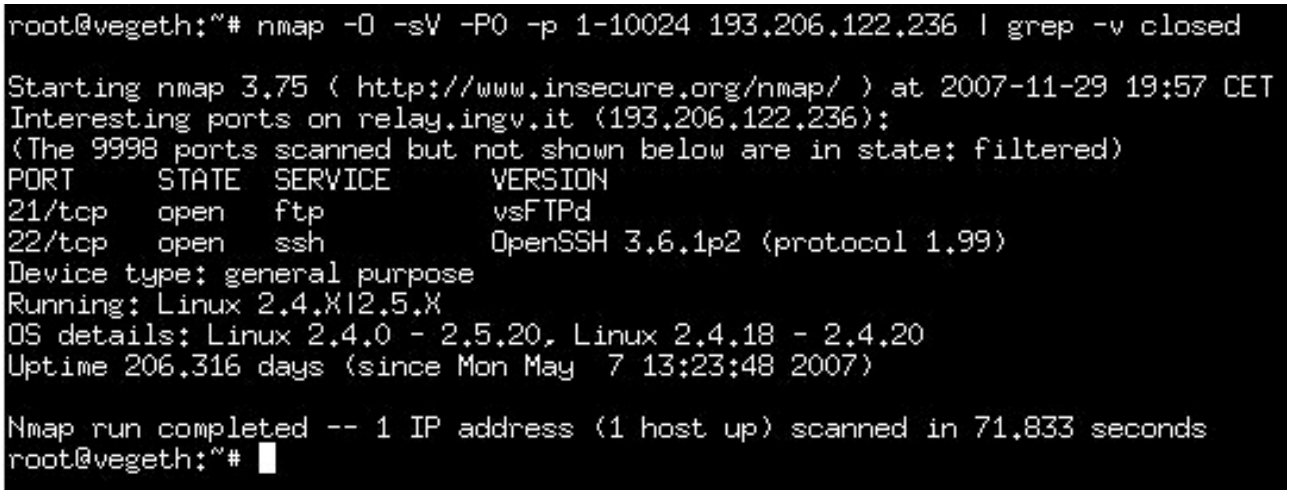

Figura 9. Scansione dei servizi offerti.

2.3. Conclusioni

[3] Nmap é un software libero distribuito con licenza GNU GPL da *INSECURE.ORG* creato per effettuare *PORT SCANNING*, cioè mirato all'individuazione di porte aperte su un computer bersaglio, in modo da determinare quali servizi di rete siano disponibili. É in grado di ipotizzare quale sistema operativo sia utilizzato dal computer bersaglio, tecnica conosciuta come *FINGERPRINTING*. Nmap é divenuto uno degli strumenti praticamente indispensabili della ``cassetta degli attrezzi'' di un amministratore di sistema, ed é usato per test di penetrazione e compiti di sicurezza informatica in generale. Come molti strumenti usati nel campo della sicurezza informatica, Nmap può essere utilizzato sia dagli amministratori di sistema che dai *cracker* o *script kiddies*. Gli amministratori di sistema possono utilizzarlo per verificare la presenza di possibili applicazioni server non autorizzate, così come i *cracker* possono usarlo per analizzare i loro bersagli.

L'unione delle vulnerabilità precedentemente descritte rende il sistema **totalmente insicuro**.
Un attaccante (*cracker*[4]) potrebbe utilizzare una combinazione delle vulnerabilità per poter ottenere il controllo completo di una macchina e poterla utilizzare sia come sistema base per altri attacchi sia per carpire informazioni dalla rete interna.
La prima vulnerabilità é sfruttabile solo potendosi autenticare ma questo restringe solamente il numero di persone che potrebbero tentare l'attacco.
La seconda vulnerabilità invece crea una vera e propria falla al sistema in quanto permette di conoscere proprio i dati per autenticarsi e sfruttare la vulnerabilità precedentemente descritta.
Inoltre accedendo con utente anonimo al servizio *FTP* é possibile ottenere l'elenco completo degli utenti della macchina e da questo é possibile iniziare un attacco al server.

[4] In ambito informatico il termine inglese cracker indica colui che entra abusivamente in sistemi altrui e li manipola allo scopo di danneggiarli (*cracking*), lasciare un segno del proprio passaggio, utilizzarli come teste di ponte per altri attacchi oppure per sfruttare la loro capacità di calcolo o l'ampiezza di banda di rete. I cracker possono essere spinti da varie motivazioni, dal guadagno economico (tipicamente coinvolti in operazioni di spionaggio industriale o in frodi) all'approvazione all'interno di un gruppo di cracker (come tipicamente avviene agli *script kiddie*, che praticano le operazioni di cui sopra senza una piena consapevolezza né delle tecniche né delle conseguenze).

3. Progettazione del nuovo servizio SSH

Il servizio *SSH* che si intende implementare risolverebbe ogni problematica riscontrata con un elevato livello di sicurezza e eliminando la seccatura del *doppio salto*[5] per accedere alla rete interna.

3.1. Accesso al servizio

Il servizio *SSH* permette l'autenticazione tramite scambio di chiavi (**RSA** o **DSA**) senza quindi dover inserire ad ogni accesso la password e eliminando uno dei problemi principali: *l'errore umano*. Spesso, infatti, si trovano le password di servizi critici scritte su *post-it* appese ai monitor, sotto la tastiera o sul proprio taccuino. Si elimina anche il problema di password troppo semplici come, ad esempio, il nome della propria fidanzata, la propria data di nascita, il cognome da nubile della moglie, ecc...

3.2. Hardening del servizio

Lo scambio delle chiavi é un buon inizio per una corretta sicurezza ma non é sufficiente! Il servizio verrà blindato adottando le seguenti misure cautelari:

- Generazione di chiavi **DSA versione 2**, distribuite direttamente dall'amministratore del servizio al richiedente;
- Generazione di chiavi a **4096 bit** di lunghezza, invece di quella standard a **512 bit**, per rendere più complessa la forzatura;
- Inibire l'inserimento di nuove chiavi pubbliche sul server per gli utenti;
- Realizzazione di chiavi con *passphrase*;
- Accesso consentito solo ed unicamente tramite scambio di chiavi;
- Apertura del server solo sulla porta del servizio (porta 22) e eliminando tutti gli altri servizi;
- Blocco dell'accesso via *SSH* al **super utente**;
- Restrizione del personale abilitato a diventare **super utente**;
- Restrizione del personale abilitato a riavviare la macchina;
- Adozione di una shell ristretta per poter eseguire unicamente il client *SSH* per permettere l'accesso alla macchina di rete interna interessata;
- Mail all'indirizzo di servizio dell'utente che si sta loggando ogni volta che si accede al servizio;
- Invio giornaliero dei log del server *SSH*, debitamente filtrati, mostrando unicamente i tentativi di accesso falliti.

Come ulteriore difesa il servizio verrà richiuso in un ambiente *chrootato*[6], con utente non privilegiato, così da rendere nulli attacchi alle classiche *vulnerabilità del Tempo Zero* (bug appena scoperti e che ancora non dispongono di una patch).

Lo stato di salute della macchina verrà costantemente monitorato grazie al servizio *SNMP*[7]. Il server *SNMP*, di recente acquisto, provvederà a controllare lo stato della macchina ad intervalli regolari e avvisando via Mail o SMS i gestori del servizio.

---

[5] Per poter entrare in rete interna si doveva effettuare l'accesso alla macchina relay.ingv.it in rete pubblica, da questa si doveva effettuare una nuova autenticazione in una macchina collocata in una *rete privata* e da qui finalmente si poteva tentare l'autenticazione in una macchina della rete interna.

[6] Contrazione di *change root*, è un metodo di sicurezza usato per isolare i limiti operativi di una applicazione. Il nome deriva dal termine informatico root che indica la directory principale del sistema operativo in cui sono contenute tutte le altre directory. Normalmente un software può accedere a tutti i dischi e le risorse del sistema operativo, compatibilmente con i permessi; l'operazione di chroot consiste nell'eseguire il programma bloccato dentro una sottodirectory, permettendogli di accedere solo alle risorse di cui ha strettamente bisogno.

In ultimo, per difendere il servizio esposto, l'*INTRUSION PREVENTION SYSTEM*[8], anch'esso di recente acquisto, verrà configurato per monitorare la bontà del traffico in ingresso al server.

3.3. Mantenimento del servizio

Tutti, utenti e amministratori, devono contribuire a mantenere il servizio sicuro e efficiente.

3.3.1. Utenti

Gli utenti interessati a utilizzare il servizio, previa richiesta, riceveranno la loro chiave privata SSH dall'amministratore del servizio e saranno responsabili della propria chiave privata.
In caso l'utente disponga già di una chiave che soddisfi i requisiti menzionati potrà utilizzarla inviando agli amministratori del servizio la propria chiave pubblica.
Essi potranno installare la chiave privata su tutte le macchine che intendono utilizzare, avendo cura di eliminarla dalle macchine che non utilizzano più.
Potranno richiedere in qualsiasi momento la sostituzione della chiave e sono obbligati a farlo nel caso se ne sospetti la duplicazione (un ottimo riscontro é l'arrivo della mail di login quando non si é effettuato alcun accesso).
La chiave privata identifica univocamente ogni utente e per nessun motivo deve essere divulgata o ceduta a terzi!

3.3.2. Amministratore

L'amministratore del servizio ha il dovere di monitorare e controllare il servizio e installare le dovute patch di sicurezza appena sono disponibili.
Deve provvedere alla disabilitazione dell'account di un utente in caso si rilevino attività malevole verso la rete informatica, interna o esterna.

---

[7] SNMP (**S**imple **N**etwork **M**anagement **P**rotocol) appartiene alla suite di protocolli Internet definita dalla *IETF* (**I**nternet **E**ngineering **T**ask **F**orce). Il protocollo opera al livello 7 del modello *OSI*. Esso consente la gestione e la supervisione di apparati collegati in una rete, rispetto a tutti quegli aspetti che richiedono azioni di tipo amministrativo

[8] Gli Intrusion Prevention System sono dei componenti sviluppati per incrementare la sicurezza informatica. Sono stati sviluppati per impedire ad un programma non autorizzato di entrare in esecuzione. La tecnologia ``Intrusion prevention'' spesso viene considerata come un estensione della tecnologia *INTRUSION DETECTION* (**IDS**) sebbene sia più simile ad una lista di controllo degli accessi di un firewall. Intrusion Prevention System evita l'attivazione di programmi potenzialmente malevoli. Questi sistemi hanno un numero molto basso di falsi positivi e possono essere utilizzati in congiunzione con gli **IDS** per evitare la propagazione di virus o worm.

4. Attivazione del servizio

Durante la realizzazione del servizio sono state effettuate alcune modifiche al progetto iniziale.

4.1. Diminuzione del livello di sicurezza

All'atto pratico sono stati disabilitati alcuni aspetti di sicurezza sia per impossibilità di realizzazione che per rendere il servizio più usabile per l'utente medio:

- Sono state abilitate sia le chiavi **RSA** che **DSA**;
- Le chiavi utilizzate sono a **2048 bit** per alleggerire il carico del server;
- L'utente é stato abilitato a caricare personalmente le proprie chiavi sul server così da renderlo indipendente dall'amministratore del servizio;
- Non si é potuta utilizzare la shell ristretta (**RSSH** o **IBSH**) a causa di limitazioni software della stessa;
- Non é stato abilitato l'ambiente *chrootato* per non imporre ulteriori restrizioni di accesso al sistema.

4.2. Configurazione Firewall

Con *IPTABLES* é stato attivato un servizio interno di firewalling per consentire solo determinate operazioni:

- L'accesso alla macchina può provenire solo dall'interfaccia di rete esterna e solo in SSH;
- L'uscita in SSH dalla macchina é concessa solo ed esclusivamente per raggiungere macchine di rete interna;
- L'invio di posta (notifiche di login) é permesso solo utilizzando il mail server dell'Ente come relay di posta;
- Dei vari pacchetti di controllo (ICMP) solo l'*echo-request* e l'*echo-replay* sono abilitati.

Al momento sulla macchina non é ancora stato attivato un completo controllo via *SNMP*, difatti é attivo solo il controllo via *echo-request*.

4.3. Configurazione SSH

Il servizio **SSHd** é stato configurato in modo da rifiutare categoricamente:

- L'accesso ROOT via SSH;
- L'accesso con chiavi **RSA** o **DSA versione 1**;
- L'accesso dalla rete interna (*10.x.y.z*). Questa restrizione é ridondante in quanto già il firewall inibisce questo tipo di accesso;
- La realizzazione di *TUNNEL SSH*;

Dietro richiesta di alcuni utenti sono state invece abilitate le seguenti funzionalità:

- Accesso con password (vedi paragrafo 5.3.2);
- Forwarding dell'ambiente grafico;
- L'utilizzo di sotto sistemi, ad esempio: *SECURE-FTP SERVER* per lo scambio dati.

Per mantenere un buon livello di sicurezza abilitando l'accesso con password sono state introdotte le seguenti regole:

- Password a scadenza mensile;
- Controllo real-time della qualità della nuova password inserita.

4.4. Creazione di nuovi utenti

É stato realizzato uno script wrapper[9] per la creazione di un nuovo utente.
Lo script, posizionato nella *home* dell'utente ROOT, richiede alcune informazioni per poter creare il nuovo account:

- Username;
- Nome;
- Cognome;
- N° stanza;
- N° telefono;
- Indirizzo email;
- É amministratore.

Questi dati permettono di creare una home per ogni utente e avere più di un contatto per rintracciarlo in caso di necessità.
Al momento dell'esecuzione il wrapper controlla che lo username non sia già presente (in caso lo script si interrompe avvertendo l'amministratore) e procede con la creazione dell'account inserendo nei campi *GECOS*[10] le informazioni personali dell'utente; di fondamentale importanza é il campo **indirizzo email** in quanto, ad ogni accesso, il sistema informa via mail l'utente dell'avvenuta connessione.
Per ogni utente si setta il periodo di validità della propria password, con la possibilità di reimpostarla ogni volta che lo desidera.
Viene in seguito aggiunto l'utente all'interno del gruppo **sshusers** per poter limitare le risorse di sistema utilizzabili (quantità di processi eseguibili contemporaneamente, massima grandezza file, numero massimo di login simultanei, ecc...).
Il flag "*É amministratore*" impone allo script di inserire l'account nel gruppo di utenti che hanno diritto di diventare amministratori (inserimento nel gruppo **wheel**). Per come é stato ristrutturato il sistema l'inserimento della prima chiave pubblica non é più da farsi tramite gli amministratori del sistema in quanto é stato abilitata l'autenticazione interattiva.

[9] Un programma o uno script che 'prepara il terreno' per un programma importante che viene eseguito successivamente.
[10] Informazioni generali sull'utente memorizzate nel file contenente la lista di utenti riconosciuti dal sistema.

5. Dettagli tecnici

Il server é stato installato con il software strettamente necessario per attivare il servizio e le restrizioni precedentemente illustrate. La macchina monterà quindi:

Debian 4.0
: come sistema operativo ospite;

OpenSSH Server
: server per garantire l'accesso SSH;

Nail
: software per l'invio di mail da CLI;

SNMPd
: daemon per il monitoraggio dello stato della macchina.

5.1. Pacchetti supplementari installati

SSH
: che permette di stabilire una sessione remota cifrata ad interfaccia a linea di comando con un altro host;

Nail
: potente client di posta *CLI*, per l'inoltro degli alert;

XAuth
: pacchetto utilizzato per permettere l'esportazione della finestra grafica (*X Forwarding*).

5.2. Configurazioni

Verranno di seguito riportati i file di configurazione presenti nel sistema che sono stati modificati per la realizzazione del servizio.

5.2.1. SSH rules

É stato creato il file di regole generali da eseguire all'accesso di un utente nel server.
In particolare é stato settato per inviare una mail all'utente ad ogni login effettuato.

```
#!/bin/sh
if read proto cookie && [ -n "$DISPLAY" ]; then
       if [ `echo $DISPLAY | cut -c1-10` = 'localhost:' ]; then
              # X11UseLocalhost=yes
              echo add unix:`echo $DISPLAY |
              cut -c11-` $proto $cookie
       else
              # X11UseLocalhost=no
              echo add $DISPLAY $proto $cookie
       fi | xauth -q -
fi



set NAIL_EXTRA_RC=/etc/nailrc
/usr/sbin/alert_mail `id -un`
```

5.2.2. Nail

Il client testuale di posta é stato impostato per inoltrare ogni richiesta di nuova mail direttamente al mail server di istituto, delegato ad esso ogni controllo e l'onere dell'inoltro dell'alert.

```
# This is the configuration file for Heirloom mailx (formerly
# known under the name "nail".
# See mailx(1) for further options.
# This file is not overwritten when 'make install' is run in
# the mailx build process again.

# Sccsid @(#)nail.rc      2.10 (gritter) 3/4/06

# Do not forward to mbox by default since this is likely to be
# irritating for most users today.
set hold

# Append rather than prepend when writing to mbox automatically.
# This has no effect unless 'hold' is unset again.
set append

# Ask for a message subject.
set ask

# Assume a CRT-like terminal and invoke a pager.
set crt

# Messages may be terminated by a dot.
set dot

# Do not remove empty mail folders in the spool directory.
# This may be relevant for privacy since other users could
# otherwise create them with different permissions.
set keep

# Do not remove empty private mail folders.
set emptybox

# Quote the original message in replies by "> " as usual on the Internet.
set indentprefix="> "

# Automatically quote the text of the message that is responded to.
set quote

# Outgoing messages are sent in ISO-8859-1 if all their characters are
# representable in it, otherwise in UTF-8.
set sendcharsets=iso-8859-1,utf-8

# Display sender's real names in header summaries.
set showname

# Display the recipients of messages sent by the user himself in
# header summaries.
set showto

# Automatically check for new messages at each prompt, but avoid polling
# of IMAP servers or maildir folders.
set newmail=nopoll

# If threaded mode is activated, automatically collapse thread.
set autocollapse

# Hide some header fields which are uninteresting for most human readers.
ignore received in-reply-to message-id references
ignore mime-version content-transfer-encoding

# Only include selected header fields when forwarding messages.
fwdretain subject date from to
```

```
# Custom vars
set smtp=ultra.ingv.it
set from="sorrentino@ingv.it"
```

5.2.3. Security

É stato limitato l'accesso ad alcune risorse utilizzabili dagli utenti per ridurre la possibilità che alcuni scripts non correttamente funzionanti blocchino la macchina.

```
# /etc/security/limits.conf
#
#Each line describes a limit for a user in the form:
#<domain>        <type>  <item>  <value>
#Where:
#<domain> can be:
#        - an user name
#        - a group name, with @group syntax
#        - the wildcard *, for default entry
#        - the wildcard %, can be also used with %group syntax,
#                 for maxlogin limit
#
#<type> can have the two values:
#        - "soft" for enforcing the soft limits
#        - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#        - core - limits the core file size (KB)
#        - data - max data size (KB)
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open files
#        - rss - max resident set size (KB)
#        - stack - max stack size (KB)
#        - cpu - max CPU time (MIN)
#        - nproc - max number of processes
#        - as - address space limit
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to
#        - rtprio - max realtime priority
#
#<domain>      <type>  <item>         <value>
@sshusers               soft    nproc           20
@sshusers               hard    nproc           50
@sshusers               hard    fsize           100000
@sshusers               hard    locks           10
@sshusers               hard    maxsyslogins    4
# End of file
```

5.2.4. SSH

Configurazione del daemon SSH per restringere l'accesso al server.

```
# What ports, IPs and protocols we listen for
Port 22
Protocol 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#Banner /etc/issue.net
# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server
UsePAM no
```

5.2.5. Network

Nei settaggi di rete é stata omessa l'impostazione del *default gateway* per non permettere l'utilizzo della rete esterna e creare possibili connessioni loop.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
       address 10.100.70.50
       netmask 255.128.0.0
       network 10.0.0.0
       broadcast 10.127.255.255
       # dns-* options are implemented by the resolvconf package, if installed
       dns-nameservers 10.10.0.4
       dns-search int.ingv.it

# The primary network interface
auto eth1
iface eth1 inet static
       address 93.63.40.78
       netmask 255.255.255.0
       gateway      93.63.40.1
```

5.3. Script

Verranno riportati gli scripts realizzati per la gestione del servizio.

5.3.1. Nuovo utente

Il wrapper crea un'interfaccia comune per la creazione di nuovi utenti, preoccupandosi di impostare correttamente tutti i valori necessari per mantenere un adeguato livello di sicurezza del sistema.

```
#!/bin/sh

USERNAME=sorrentino
NAME=Diego
SURNAME=Sorrentino
ROOM=142
CELL=2345
EMAIL=diego.sorrentino@ingv.it
# WARNING!!! Set 'y' only if the user can become root (or other users)!!!
CAN_BE_SUPERUSER=n

##### DON'T EDIT THIS PART!!! #####
# See man passwd(8)
MINDAYS=0
MAXDAYS=30
SSH_USERS=sshusers
ADMIN=sir@ingv.it

echo "Checking username existence"
exists=`egrep ^${USERNAME} /etc/passwd`
if [ -n "${exists}" ];
then
       echo -e "\nUsername ${USERNAME} ALREADY EXISTS!!! Use another one!\n";
       exit;
fi

echo "Creating new user == A password will be prompted =="
HOME=/home/${USERNAME}
/usr/sbin/adduser --home ${HOME} --shell /bin/bash \
       --gecos "${NAME} ${SURNAME},${ROOM},${CELL},${EMAIL}" ${USERNAME}

echo "Setting password expiration values"
/usr/bin/passwd --mindays ${MINDAYS} --maxdays ${MAXDAYS} ${USERNAME}

echo "Locking resources for ssh-users"
/usr/sbin/adduser ${USERNAME} ${SSH_USERS}

echo "Creating .ssh directory and useful files"
PERSONAL_SSH=/home/${USERNAME}/.ssh
mkdir ${PERSONAL_SSH}
touch ${PERSONAL_SSH}/known_hosts ${PERSONAL_SSH}/authorized_keys
chown -R ${USERNAME}:${USERNAME} ${PERSONAL_SSH}

if [ 'Y' == ${CAN_BE_SUPERUSER} ] || [ 'y' == ${CAN_BE_SUPERUSER} ];
then
       echo "Adding user in wheel group"
       /usr/sbin/adduser ${USERNAME} wheel
fi

echo "Sending mail to $EMAIL"
nail -r ${ADMIN} -c ${ADMIN} -s "Attivazione account SSH" ${EMAIL} <<-BODY
       Ciao ${NAME},
       il tuo account SSH e' stato creato, ti riporto di seguito le tue
       credenziali di accesso:

       server: ssh-server.rm.ingv.it
       username: ${USERNAME}
```

```
      password: [ti verra' consegnata]

      questa macchina la userai come ponte per entrare in rete interna o
      trasferimento dati, e' abilitata anche l'esportazione del desktop, nel
      caso ne avessi necessita'.

      Ricorda
      ========
      - La password scade ogni ${MAXDAYS} giorni e, al primo login dopo la
      scadenza ti verra' chiesto di cambiarla.

      - Ti verra' inviata una mail di notifica ad ogni connessione al server,
      con le informazioni sulla connessione instaurata.

      Consigli
      =========
      - copia la tua chiave ssh pubblica sul server, almeno eviti ogni volta di
      digitare la password. [1]

      - creati uno shortcut per collegarti al server (aggiungi una sezione nel
      tuo .ssh/config) per non scrivere ogni volta la stringa di connessione.[2]

      Contattaci per ogni problema

      [1]
      Per procedere allo scambio di chiavi basta semplicemente avere la tua
      coppia di chiavi sulla macchina che usi per collegarti:
      ~/.ssh/id_rsa.pub (o id_dsa.pub)
      (se non hai la coppia di chiavi puoi crearle con:
      $ ssh-keygen -b 2048 -t rsa
      ...puoi anche lasciare vuota la passphrase)
      e lanciare il seguente comando (con chiave RSA):
      $ cat ~/.ssh/id_rsa.pub | ssh ssh-server "cat - >> .ssh/authorized_keys"
      Ti chiedera' la pwd per poter caricare la chiave e da qui in poi, da
      quella macchina, non ti verra' piu' richiesto l'inserimento della pwd!

      [2]
      Se poi vuoi evitarti ogni volta di scrivere tutta la riga di connessione:
      ssh utente@ssh-server.rm.ingv.it  oppure
      scp file utente@ssh-server.rm.ingv.it:file

      basta mettere nel file ~/.ssh/config (se non c'e' va creato)
      questo blocco di codice:
      ===========
      Host ssh-server
      User il-tuo-utente
      Hostname ssh-server.rm.ingv.it
      ForwardAgent yes
      ForwardX11 yes
      ForwardX11Trusted yes
      ===========

      e per collegarti bastera' dare:
      ssh ssh-server
      ovviamente puoi anche usarlo per copiare files:
      scp file ssh-server:file

      --
      SIR - Servizi Informatici e Reti
      INGV - Istituto nazionale di Geofisica e Vulcanologia
      Via di Vigna Murata 605
      00143 Roma
BODY


##### DON'T EDIT THIS PART!!! #####
```

5.3.2. Login alert

Lo script viene eseguito ad ogni login utente. Viene estratto dai campi *GECOS* l'indirizzo di posta da lui comunicato e invia una mail con le informazioni necessarie per tracciare la connessione (data e IP).

```
#!/usr/bin/perl

$username=$ARGV[0];

$admin_email = 'diego.sorrentino@ingv.it';
$conn_string = `date +"giorno %F alle ore %H:%M.%S UTC"`;
$conn_time = `date +"%F %H:%M.%S UTC"`;
$client_ip = $ENV{'SSH_CLIENT'};
$client_ip =~ s/ .*//;

chomp($conn_string);
chomp($conn_time);

open(INPUT, '/etc/passwd');

while(<INPUT>){
       if(/^$username/){
($name, $surname, $room, $cell, $email) = /:(\w+) (\w+),(.*?),(.*?),(.+?):/;

              if($email =~ m/^$/){
                     $email = $admin_email;
              }
       }
}

close(INPUT);

$body = <<END_BODY;
Ciao $name $surname,
ti sei connesso a ssh-server.rm.ingv.it il $conn_string
dall'IP $client_ip.

Se questa informazione non risultasse esatta informa immediatamente
il SIR (sir\@ingv.it) che provvedera' ad effettuare i dovuti controlli.

Buona giornata

SIR

--
SIR - Servizi Informatici e Reti
INGV - Istituto Nazionale di Geofisica e Vulcanologia
Via di Vigna Murata 605
00143 Roma
END_BODY

$command = "/usr/bin/nail -r $admin_email -s '[SSH-Server] Utente $username connesso il
$conn_time' $email";

system ("echo \"$body\" | $command");
```

Coordinamento editoriale e impaginazione
Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale
Laboratorio Grafica e Immagini | INGV Roma

Via di Vigna Murata, 605
00143 Roma
Tel. +39 06518601 Fax +39 065041181

http://www.ingv.it

Istituto Nazionale di Geofisica e Vulcanologia