CONCORSO PUBBLICO, PER TITOLI ED ESAMI, A N. 1 POSTO DI CATEGORIA C, POSIZIONE ECONOMICA C1, AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, CON RAPPORTO DI LAVORO SUBORDINATO A TEMPO INDETERMINATO E PIENO PRESSO L'AREA SISTEMI INFORMATIVI (codice 20PTA052).

Criteri di valutazione per la prova scritta:

La prova scritta sarà valutata per un massimo di 30 punti, secondo i seguenti criteri di valutazione:

- Grado di conoscenza della materia
- Capacità di sintesi
- Chiarezza espositiva e uso di una terminologia adeguata
- Coerenza dell'elaborato con la traccia assegnata.

Criteri di valutazione per la prova orale:

La prova orale sarà valutata per un massimo di 30 punti, secondo i seguenti criteri di valutazione:

- Conoscenze dimostrate dal candidato
- Maturità, professionalità e chiarezza espositiva
- Completezza della risposta
- Conoscenza della lingua inglese

Per l'accertamento della conoscenza della lingua inglese ogni candidato troverà nella busta, contente la prova orale, un brano tratto da "Mastering OpenLDAP" ISBN 978-1-847191-02-1.

Criteri di valutazione dei titoli:

I titoli valutabili, purché attinenti alle attività del posto messo a concorso, sono quelli previsti dall'articolo 5 del bando e sono:

A) anzianità di servizio, calcolata alla data di scadenza del bando, fino ad un massimo di 8 punti:

- A1 - anzianità di servizio prestata a tempo indeterminato o determinato presso le Università per un periodo continuativo almeno pari ad un anno: **2 punti per ogni anno**. Il punteggio è **dimezzato** se il servizio è stato prestato nelle categorie inferiori rispetto alla categoria dei posti messi a concorso. **Fino ad un massimo di 6 punti;**
- A2 - anzianità di servizio prestata sotto forma di co.co.co. presso le Università, per un periodo continuativo almeno pari ad un anno: **1 punto per ogni anno**. **Fino ad un massimo di 3 punti;**
- A3 - anzianità di servizio prestata a tempo indeterminato o determinato o co.co.co. presso altre Pubbliche Amministrazioni, per un periodo continuativo almeno pari ad un anno: **0,5 punti per ogni anno**. **Fino ad un massimo di 4 punti;**

B) incarichi professionali, fino ad un massimo di 3 punti:

- incarichi di responsabilità o funzione specialistica, formalmente attribuiti, di durata continuativa almeno pari ad un anno (0,25 punto per ogni anno di incarico).

C) **altri titoli, fino ad un massimo di 3 punti:**

- C1 1 punto per Dottorato di Ricerca
- C2 1 punto per Master
- C3 1 punto per Diplomi Universitari, Laurea Triennale, Diploma di Laurea v.o., Laurea Magistrale, Laurea Specialistica
- C4 1 punto per pubblicazioni/tutor attività didattiche/relatore o correlatore a convegni e corsi di formazione

D) **precedenti esperienze professionali, fino ad un massimo di 3 punti**:

- precedenti attività lavorative, svolte a qualsiasi titolo di durata continuativa almeno pari ad un anno (0,5 punto per ogni anno di incarico).

Non vengono ricomprese le esperienze pregresse rientranti nella categoria A) di valutazione, anche qualora venga raggiunto il massimale e pertanto non verranno valutate in questa categoria.

E) **formazione, fino ad un massimo di 3 punti**:

- E1 0,10 punti per attestati di corsi di formazione, seminari, convegni senza valutazione finale
- E2 0,75 punti per attestati di corsi di qualificazione, corsi di specializzazione con valutazione finale

Non verranno valutati i titoli presentati dai candidati non attinenti e/o non previsti tra le categorie di titoli valutabili previsti dal bando e la somma dei titoli sopra elencati non potrà superare i 20 punti.

Tracce relative alla prova scritta:

TRACCIA 1

Definizione e differenze di Iaas, Paas, SaaS nell'ambito del Cloud Computing

Il candidato fornisca degli esempi di servizio forniti attraverso tali modalità

TRACCIA 2

Infrastrutture virtuali

Il candidato descriva i vantaggi e gli svantaggi della virtualizzazione delle infrastrutture informatiche

TRACCIA 3

IP, TCP, UDP, HTTP: tutti e quattro i termini si riferiscono a protocolli di rete

Il candidato li descriva anche in riferimento alla pila ISO\OSI

Tracce relative alla prova orale

Prova numero 1

1. Metodi di archiviazione nel cloud e relative caratteristiche
2. Indicare qual è il dominio di primo livello dell'indirizzo mail info.test@si.unimib.it
    a) info
    b) test
    c) si
    d) unimib
    e) it

Prova di inglese da "Mastering OpenLDAP"

ISBN 978-1-847191-02-1

pag.141 ("Encryption") primi due periodi

Prova numero 2

1. Cos'è e come funziona il DNS?
2. Container e Virtual Machines, quali sono le principali differenze

Prova di inglese da

"Mastering OpenLDAP"

ISBN 978-1-847191-02-1

pag. 140 penultimo paragrafo

Prova numero 3

1. Descrivere le differenze tra cloud pubblico, privato e ibrido
2. Cosa sono TCP e UDP? Differenze?

Prova di inglese da

"Mastering OpenLDAP"

ISBN 978-1-847191-02-1

pag. 43 penultimo paragrafo

Prova numero 4

1. Quali sono le modalità disponibili nel SO Linux per l'accesso alle risorse di Storage condivise
2. Che protocollo usa un host A per scoprire il MAC address di un host B (che sta nella sua stessa sottorete) di cui conosce l'indirizzo IP?

Prova di inglese da

"Mastering OpenLDAP"

ISBN 978-1-847191-02-1

pag. 46 primo paragrafo

Si allegano di seguito i documenti relativi alla prova di inglese delle specifiche tracce.

PROVA N. 1

But it is possible, and often useful, for an organization or individual to simply create a locally used CA, and then use that CA to generate certificates for in-house applications. This is what we will do when we create a certificate for OpenLDAP.

Of course, certificates generated this way may not be considered reliable to users outside of your organization, but hosting an individual or organization-wide CA can be an effective way to add security to your own network, without having to purchase certificates from a commercial vendor.

Not all CAs use the same form of authoritative signing (and not all CAs charge for certificates). Some CAs, such as Cacert.org, use what is called a **web of trust** technique for establishing authenticity. In the web of trust the authenticity of a certificate is established by peers who can play the role of assuring that the certificate is owned by the person or organization that it claims to be owned by. For more information visit `http://www.cacert.org/`.

We have discussed the first role of SSL/TLS, establishing authenticity. Next we will turn to the second role of SSL/TLS; providing encryption services.

Encryption

SSL/TLS provides the features required for sending encrypted messages back and forth between the client and the server. In a nutshell the process goes like this: the server sends the client its certificate, and inside the certificate (among other things) is the server's **public key**. The public key is the first half of a pair of keys. A public key can be used for encrypting a message, but not decrypting it. A second key, the **private key**, is then used for decrypting a message. The server keeps its private key to itself, but gives out its public key to any client that requests it. Clients can then send messages to the server that only the server can decrypt and interpret.

Depending on the configuration the client also sends the server its public key, which the server can use to send messages that only the client can decrypt. At this point, each can transmit encrypted messages to the other.

But there is a drawback to using public/private keys: they are slow and resource-intensive. Rather than trading all information through these public/private key combos, the client and server then negotiate a set of temporary symmetric encryption keys (which use the same key to encrypt and decrypt messages) that they will both use for the duration of the session. All traffic between the two clients is encrypted using these keys. Once the session is complete, both the client and server discard the temporary keys.

PROVA N. 2

Consider the case of online banking. If I use my browser to log on to my bank's website and conduct a few transactions, I want to be sure that the website I am connected to really is my bank's website, and not some other website masquerading as my bank. SSL/TLS provides tools to establish the authenticity of the server using **X.509 certificates**. An X.509 certificate has three important pieces of information:

- Information about the individual or organization that owns the certificate
- A public encryption key (which we will discuss in the next section)
- The **digital signature** of a certificate authority (CA)

A certificate is designed as a sort of assurance that a server is associated with a particular individual or organization. When I contact a server that I believe to be my bank's, I want some assurance that it is, in fact, my bank's server. So one piece of information contained in the certificate is information about who owns the certificate. We can inspect this information ourselves, but since the certificate has a digital signature, it is also possible for software to computationally verify this—in a way much more reliable than reading the certificate and simply trusting that the certificate is accurate.

The digital signature is an encrypted bit of information. It is encrypted with a special "private" key that is owned by a Certificate Authority. The CA can then issue a public key that client software can use to verify that the certificate was in fact signed by the CA. The CA then, plays a very important role in establishing trust. We will discuss public and private keys in the *Encryption* section.

Certificate Authorities are responsible for issuing certificates. Ideally, a CA is a trusted source that can verify the authenticity of the certificate, and provide assurance that the certificate is really owned by the organization or individual that claims to own it.

There are a number of commercial CAs that provide certificate generation services for a price. To obtain a certificate through these services, an organization or individual must provide a certain amount of information that can be used to verify that the person or organization signing up for the certificate is legitimate. Once investigation of this material has been done, and the person or organization has paid the requisite fee, the CA issues a digitally-signed certificate.

The certificates of large CAs are included by default in most SSL-aware applications, such as popular web browsers (like Mozilla Firefox) and SSL libraries (like OpenSSL). These certificates include the public keys necessary for verifying digital signatures. Thus, when a client gets an X.509 certificate that is signed by one of these CAs, it has all of the tools it needs to verify the certificate's authenticity.

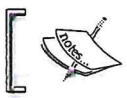
PROVA N. 3


```
line 47 (directory /var/lib/ldap)
line 48 (index objectClass eq)
index objectClass 0x0004
line 49 (index cn eq,sub,pres,approx)
index cn 0x071e
slapd starting
```

This can be one other useful way to ferret out configuration issues. The `-d` flag will take any of the debugging levels specified in the `slapd.conf` man page. I find `acl` useful for debugging access problems, and `filter` is often useful in figuring out trouble with searches.

When `-d` is specified the program will run in the foreground. To stop the server simply hit *CTRL+C*. This will stop the server and return you to a shell prompt.

Other useful command line parameters to use with `slapd` are `-u` and `-g`. Each takes one argument: `-u` takes a username and `-g` takes a groupname. These control the effective UID and GID (user ID and group ID) that SLAPD runs as. Once SLAPD has started and connected to the appropriate ports (which it must do as root), it will switch its UID and GID to the names specified in these parameters.



To get a list of other command line flags that we can use with `slapd`, refer to the man page for `slapd`.

In the next section, we will be using some of the OpenLDAP clients to connect to our directory. This will require that the SLAPD server be running. You can verify that `slapd` is running by checking if `/var/run/slapd/slapd.pid` exists, or by running `pgrep slapd`, which will display the process ID of `slapd` if it's running. If no process ID number is returned, `slapd` is not running.

Configuring the LDAP Clients

In the last couple of sections we have focused exclusively on the SLAPD server. Now that the server is running we need to get the client configuration so that we can make test connections to the server.

Fortunately all of the OpenLDAP client programs share one common configuration file, `ldap.conf`, which is located in Ubuntu at `/etc/ldap/ldap.conf` (if you build from source, according to Appendix A, the default location for this file is `/usr/local/etc/openldap/ldap.conf`).

PROVA N. 9

Size and Time Limits

The next two directives, `SIZELIMIT` and `TIMELIMIT`, indicate the upper limits on the number of records returned (`SIZELIMIT`) and the amount of time the client will wait for the server to respond (`TIMELIMIT`). Here we have set both to 0, a special value for these directives that indicates that there should be no limit.

The way that size and time limits are handled can be a little confusing. On the client side there are two ways of specifying these limits: through the `ldap.conf` configuration file (as we are doing here) and through command-line parameters (as we will see in the next chapter).

However, the `SIZELIMIT` and `TIMELIMIT` directives above are not exactly defaults in the usual sense of the word. They are the absolute upper limit that the client can request. With command-line arguments the client can specify lower time and size limits, and those lower numbers will be used. But if the client attempts to specify larger size or time limits, they will be ignored, and the values of `SIZELIMIT` and `TIMELIMIT` will be used instead.

But the story doesn't end here. The SLAPD server can also define size and time limits (with the `limits`, `sizelimit` and `timelimit` directives in `slapd.conf`). If a client specifies a limit higher than the server's, the server will ignore the client's limit and use its own. We will look more at setting server limits in Chapter 5.

Now we have a functioning `ldap.conf` file that will alleviate the need to specify these parameters on the command line.

The last thing we need to do in this chapter is to use an OpenLDAP client to test out the SLAPD server.

Testing the Server

At this point, we have a SLAPD server configured and running, and we have an `ldap.conf` file that specifies many of the defaults for our tools. Now we are going to query the directory and fetch some information.

We haven't actually put any entries in our database, though. So what will we query? SLAPD does provide directory-based access to certain information, including currently-loaded schemas and subschemas, configuration information, and a special record called the **root DSE**. The root DSE (**DSA-Specific Entry**, where **DSA** stands for **Directory Service Agent**—the technical term for an LDAP server) is a special entry that provides information about the server itself. Like all other entries in an LDAP, the root DSE has a DN. Unlike all other entries, the root DSE's DN is an empty string.

IL PRESIDENTE DELLA COMMISSIONE

Milano, 21/05/2021

Dott. VIRZI ETTORE