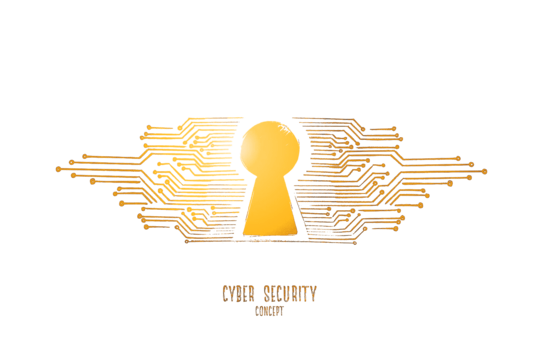
Bastionado de redes 

y sistemas.



# Creación de una PKI en Linux

| Resultado |
| --- |
|  |
|  |

| **Sudo apt-get install openssl** |
| --- |
| **mkdir root/ca** |
| **cd /root/ca** |
| * **certs** → Certificados generados y firmados por la autoridad.   Para el certificado raíz se incluye a sí mismo.  Para los certificados intermedios, incluye los propios certificados y certificados de clientes firmados.   * **crl→** Lista de certificados eliminados antes de la fecha de expiración. * **Newcerts** → Copia de cada certificado firmado, con el número de serie como nombre del archivo. * **Private** → contiene las claves primarias. Usadas para firmar certificados. |
| Sudo su;  apt-get install openssl  mkdir root/ca  cd /root/ca  chmod 700 private  touch index.txt  echo 1000 > serial  echo “configurar openssl, usar [texto adjunto](#_ukunb036wh3n))  vim /root/ca/openssl.cnf # CA raíz openssl genrsa -aes256 -out private/ca.key.pem 4096  chmod 400 private/ca.key.pem  # nos pedirá poner contraseña  openssl req -key private/ca.key.pem -config /root/ca/openssl.cnf -new -x509 -days 7300 -extensions v3\_ca -out certs/ca.cert.pem  chmod 444 certs/ca.cert.pem  openssl x509 -noout -text -in /root/ca/certs/ca.cert.pem  # Algoritmo empleado en el campo “Signature Algorithm”.  # Caducidad, en el campo “Validity”.  # Longitud de la clave pública, “Public-Key”.  # Campo “Issuer”, quien firma el certificado.  # Campo “Subject”, a quien se refiere el certificado.  # CA intermedias  mkdir -p /root/ca/intermedia  cd /root/ca/intermedia  mkdir certs crl csr newcerts private  chmod 700 private  touch index.txt  echo 1000 > serial  echo 1000 > crlnumber  cp /root/ca/openssl.cnf /root/ca/intermedia/  # editar los siguientes valores:  # dir  # private\_key  # certificate  # crl  # policy = policy\_loose  dir = /root/ca/intermedia # Where everything is kept  certs = $dir/certs # Where the issued certs are kept  crl\_dir = $dir/crl # Where the issued crl are kept  database = $dir/index.txt # database index file.  #unique\_subject = no # Set to 'no' to allow creation of  # several certs with same subject.  new\_certs\_dir = $dir/newcerts # default place for new certs.  certificate = $dir/certs/intermedia.cert.pem # The CA certificate  serial = $dir/serial # The current serial number  crlnumber = $dir/crlnumber # the current crl number  # must be commented out to leave a V1 CRL  crl = $dir/intermedia.crl.pem  private\_key = $dir/private/intermedia.key.pem # The private key |
| cd /root/ca  openssl genrsa -aes256 -out intermedia/private/intermedia.key.pem 4096  chmod 400 intermedia/private/intermedia.key.pem  openssl req -config intermedia/openssl.cnf -new -sha256 -key intermedia/private/intermedia.key.pem -out intermedia/csr/intermedia.csr.pem  openssl ca -config openssl.cnf -extensions v3\_ca\_intermediate -days 3650 -notext -md sha256 -in intermedia/csr/intermedia.csr.pem -out intermedia/certs/intermedia.cert.pem  chmod 444 intermedia/certs/intermedia.cert.pem  openssl x509 -noout -text -in intermedia/certs/intermedia.cert.pem  openssl verify -CAfile certs/ca.cert.pem intermedia/certs/intermedia.cert.pem  cat intermedia/certs/intermedia.cert.pem certs/ca.cert.pem > intermedia/certs/ca-cadena.cert.pem  openssl genrsa -out intermedia/private/dominio1.key.pem 2048  openssl req -config intermedia/openssl.cnf -key intermedia/private/dominio1.key.pem -new -sha256 -out intermedia/csr/dominio1.csr.pem  openssl ca -config intermedia/openssl.cnf -extensions server\_cert -days 365 -notext -md sha256 -in intermedia/csr/dominio1.csr.pem -out intermedia/certs/dominio1.cert.pem  cp /root/ca/intermedia/certs/dominio\* /etc/ssl/certs/  cp /root/ca/intermedia/private/dominio\* /etc/ssl/private/  cp /root/ca/intermedia/certs/ca-cadena.cert.pem /etc/ssl/certs/ |
| apt-get install apache2  # configurar dominios apache. [Ejemplo](#_sgpq6p1rq905)  vim /etc/apache2/sites-available/dominios-ssl.conf  a2enmod ssl  cd /var/www/html  mkdir dominio1  Chown www-data:www-data dominio\*  cp index.html dominio1  a2ensite dominios-ssl.conf  service apache2 restart  apache2ctl configtest |
|  |

| Configuración de dominios |
| --- |
|
| <IfModule mod\_ssl.c>  <VirtualHost \_default\_:443>  ServerAdmin webmaster@localhost  ServerName dominio1.com  ServerAlias www.dominio1.com  DocumentRoot /var/www/html/dominio1  #  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  # SSL Engine Switch:  # Enable/Disable SSL for this virtual host.  SSLEngine on  SSLCertificateFile /etc/ssl/certs/dominio1.cert.pem  SSLCertificateKeyFile /etc/ssl/private/dominio1.key.pem  SSLCertificateChainFile /etc/ssl/certs/ca-cadena.cert.pem  <FilesMatch "\.(cgi|shtml|phtml|php)$">  SSLOptions +StdEnvVars  </FilesMatch>  <Directory /usr/lib/cgi-bin>  SSLOptions +StdEnvVars  </Directory>  </VirtualHost>  <VirtualHost \_default\_:443>  ServerAdmin webmaster@localhost  ServerName dominio2.com  ServerAlias www.dominio2.com  DocumentRoot /var/www/html/dominio2  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  # SSL Engine Switch:  # Enable/Disable SSL for this virtual host.  SSLEngine on  SSLCertificateFile /etc/ssl/certs/dominio2.cert.pem  SSLCertificateKeyFile /etc/ssl/private/dominio2.key.pem  SSLCertificateChainFile /etc/ssl/certs/ca-cadena.cert.pem  <FilesMatch "\.(cgi|shtml|phtml|php)$">  SSLOptions +StdEnvVars  </FilesMatch>  <Directory /usr/lib/cgi-bin>  SSLOptions +StdEnvVars  </Directory>  </VirtualHost>  </IfModule> |
| Archivo de configuración de openssl RAÍZ( openssl.cnf ) |
| #  # OpenSSL example configuration file.  # This is mostly being used for generation of certificate requests.  #  # Note that you can include other files from the main configuration  # file using the .include directive.  #.include filename  # This definition stops the following lines choking if HOME isn't  # defined.  HOME = .  # Extra OBJECT IDENTIFIER info:  #oid\_file = $ENV::HOME/.oid  oid\_section = new\_oids  # To use this configuration file with the "-extfile" option of the  # "openssl x509" utility, name here the section containing the  # X.509v3 extensions to use:  # extensions =  # (Alternatively, use a configuration file that has only  # X.509v3 extensions in its main [= default] section.)  [ new\_oids ]  # We can add new OIDs in here for use by 'ca', 'req' and 'ts'.  # Add a simple OID like this:  # testoid1=1.2.3.4  # Or use config file substitution like this:  # testoid2=${testoid1}.5.6  # Policies used by the TSA examples.  tsa\_policy1 = 1.2.3.4.1  tsa\_policy2 = 1.2.3.4.5.6  tsa\_policy3 = 1.2.3.4.5.7  ####################################################################  #Archivo de configuración para CA  #Sección obligatoria. Indicamos a openssl que utilice las opciones de configuración desde la seccion [ ca\_default ]  [ ca ]  default\_ca = CA\_default # The default ca section  ####################################################################  #Directorios donde almacenar archivos  [ CA\_default ]  dir = /root/ca # Where everything is kept  certs = $dir/certs # Where the issued certs are kept  crl\_dir = $dir/crl # Where the issued crl are kept  database = $dir/index.txt # database index file.  #unique\_subject = no # Set to 'no' to allow creation of  # several certs with same subject.  new\_certs\_dir = $dir/newcerts # default place for new certs.  certificate = $dir/certs/ca.cert.pem # The CA certificate  serial = $dir/serial # The current serial number  crlnumber = $dir/crlnumber # the current crl number  # must be commented out to leave a V1 CRL  crl = $dir/ca.crl.pem # The current CRL  private\_key = $dir/private/ca.key.pem# The private key  x509\_extensions = usr\_cert # The extensions to add to the cert  # Comment out the following two lines for the "traditional"  # (and highly broken) format.  name\_opt = ca\_default # Subject Name options  cert\_opt = ca\_default # Certificate field options  # Extension copying option: use with caution.  # copy\_extensions = copy  # Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs  # so this is commented out by default to leave a V1 CRL.  # crlnumber must also be commented out to leave a V1 CRL.  # crl\_extensions = crl\_ext  crl\_extensions = crl\_ext  default\_days = 365 # how long to certify for  default\_crl\_days= 30 # how long before next CRL  default\_md = sha256 # use public key default MD  preserve = no # keep passed DN ordering  # A few difference way of specifying how similar the request should look  # For type CA, the listed attributes must be the same, and the optional  # and supplied fields are just that :-)  policy = policy\_strict  # For the CA policy  [ policy\_match ]  countryName = match  stateOrProvinceName = match  organizationName = match  organizationalUnitName = optional  commonName = supplied  emailAddress = optional  # For the 'anything' policy  # At this point in time, you must list all acceptable 'object'  # types.  [ policy\_anything ]  countryName = optional  stateOrProvinceName = optional  localityName = optional  organizationName = optional  organizationalUnitName = optional  commonName = supplied  emailAddress = optional  [ policy\_strict ]  #Opciones para firmar certificados raiz.  #Se aplica a las firmas del certificado raíz. La CA raíz sólo se emplea para crear CA subordinadas o intermedias.  countryName = match  stateOrProvinceName = match  organizationName = match  organizationalUnitName = optional  commonName = supplied  emailAddress = optional  [ policy\_loose ]  #Opciones para las CA intermedias. Se aplica a todas las firmas de las CA intermedias.  countryName = optional  stateOrProvinceName = optional  localityName = optional  organizationName = optional  organizationalUnitName = optional  commonName = supplied  emailAddress = optional  ####################################################################  #Opciones para crear certificados o peticiones de certificados  [ req ]  default\_bits = 2048  default\_keyfile = privkey.pem  distinguished\_name = req\_distinguished\_name  attributes = req\_attributes  x509\_extensions = v3\_ca # The extensions to add to the self signed cert  default\_md = sha256  # Passwords for private keys if not present they will be prompted for  # input\_password = secret  # output\_password = secret  # This sets a mask for permitted string types. There are several options.  # default: PrintableString, T61String, BMPString.  # pkix : PrintableString, BMPString (PKIX recommendation before 2004)  # utf8only: only UTF8Strings (PKIX recommendation after 2004).  # nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).  # MASK:XXXX a literal mask value.  # WARNING: ancient versions of Netscape crash on BMPStrings or UTF8Strings.  string\_mask = utf8only  # req\_extensions = v3\_req # The extensions to add to a certificate request  [ req\_distinguished\_name ]  #Especifica qué información vamos a requerir al solicitar un certificado.  countryName = Country Name (2 letter code)  countryName\_default = ES  countryName\_min = 2  countryName\_max = 2  stateOrProvinceName = State or Province Name (full name)  stateOrProvinceName\_default = Spain  localityName = Locality Name (eg, city)  0.organizationName = Organization Name (eg, company)  0.organizationName\_default = Mi empresa, S.A.  # we can do this but it is not needed normally :-)  #1.organizationName = Second Organization Name (eg, company)  #1.organizationName\_default = World Wide Web Pty Ltd  organizationalUnitName = Organizational Unit Name (eg, section)  #organizationalUnitName\_default =  commonName = Common Name (e.g. server FQDN or YOUR name)  commonName\_max = 64  emailAddress = Email Address  emailAddress\_max = 64  # SET-ex3 = SET extension number 3  [ req\_attributes ]  challengePassword = A challenge password  challengePassword\_min = 4  challengePassword\_max = 20  unstructuredName = An optional company name  [ usr\_cert ]  #Se aplica al firmar peticiones de certificados para un cliente.  # These extensions are added when 'ca' signs a request.  # This goes against PKIX guidelines but some CAs do it and some software  # requires this to avoid interpreting an end user certificate as a CA.  basicConstraints = CA:FALSE  # Here are some examples of the usage of nsCertType. If it is omitted  # the certificate can be used for anything \*except\* object signing.  # This is OK for an SSL server.  # nsCertType = server  # For an object signing certificate this would be used.  # nsCertType = objsign  # For normal client use this is typical  nsCertType = client, email  # and for everything including object signing:  # nsCertType = client, email, objsign  # This is typical in keyUsage for a client certificate.  # keyUsage = nonRepudiation, digitalSignature, keyEncipherment  # This will be displayed in Netscape's comment listbox.  nsComment = "Certificado cliente generado por la CA"  # PKIX recommendations harmless if included in all certificates.  subjectKeyIdentifier = hash  authorityKeyIdentifier = keyid,issuer  # This stuff is for subjectAltName and issuerAltname.  # Import the email address.  # subjectAltName=email:copy  # An alternative to produce certificates that aren't  # deprecated according to PKIX.  # subjectAltName=email:move  # Copy subject details  # issuerAltName=issuer:copy  #nsCaRevocationUrl = http://www.domain.dom/ca-crl.pem  #nsBaseUrl  #nsRevocationUrl  #nsRenewalUrl  #nsCaPolicyUrl  #nsSslServerName  # This is required for TSA certificates.  # extendedKeyUsage = critical,timeStamping  keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment  extendedKeyUsage = clientAuth, emailProtection  [server\_cert ]  #Para certificados de servidores, por ejemplo web  basicConstraints = CA:FALSE  nsCertType = server  nsComment = "Certificado para servidores generado por la CA"  subjectKeyIdentifier = hash  authorityKeyIdentifier = keyid,issuer:always  keyUsage = critical, digitalSignature, keyEncipherment  extendedKeyUsage = serverAuth  [ v3\_req ]  # Extensions to add to a certificate request  basicConstraints = CA:FALSE  keyUsage = nonRepudiation, digitalSignature, keyEncipherment  [ v3\_ca ]  #Se aplica cuando creamos el certificado raíz.  # Extensions for a typical CA  # PKIX recommendation.  subjectKeyIdentifier = hash  authorityKeyIdentifier = keyid:always,issuer  basicConstraints = critical,CA:true  # Key usage: this is typical for a CA certificate. However since it will  # prevent it being used as an test self-signed certificate it is best  # left out by default.  # keyUsage = cRLSign, keyCertSign  keyUsage = critical, digitalSignature, cRLSign, keyCertSign  # Some might want this also  # nsCertType = sslCA, emailCA  # Include email address in subject alt name: another PKIX recommendation  # subjectAltName=email:copy  # Copy issuer details  # issuerAltName=issuer:copy  # DER hex encoding of an extension: beware experts only!  # obj=DER:02:03  # Where 'obj' is a standard or added object  # You can even override a supported extension:  # basicConstraints= critical, DER:30:03:01:01:FF  [ v3\_ca\_intermediate]  #Para CA intermedias o subordinadas  #Se aplica cuando creamos certificados intermedios, es parámetro pathlen:0 indica que no hay más entidades subordinadas detrás de la intermedia.  subjectKeyIdentifier = hash  authorityKeyIdentifier = keyid:always,issuer  basicConstraints = critical, CA:true, pathlen:0  keyUsage = critical, digitalSignature, cRLSign, keyCertSign  [ crl\_ext ]  #Se aplica al crear listas de revocación (certificados revocados)  # CRL extensions.  # Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.  # issuerAltName=issuer:copy  authorityKeyIdentifier = keyid:always  [ ocsp ]  #Configuración para OCSP (Online Certificate Status Protocol). Opciones para la gestión en línea de certificados.  basicConstraints = CA:FALSE  subjectKeyIdentifier = hash  authorityKeyIdentifier = keyid,issuer  keyUsage = critical, digitalSignature  extendedKeyUsage = critial, OCSPSigning  [ proxy\_cert\_ext ]  # These extensions should be added when creating a proxy certificate  # This goes against PKIX guidelines but some CAs do it and some software  # requires this to avoid interpreting an end user certificate as a CA.  basicConstraints = CA:FALSE  # Here are some examples of the usage of nsCertType. If it is omitted  # the certificate can be used for anything \*except\* object signing.  # This is OK for an SSL server.  # nsCertType = server  # For an object signing certificate this would be used.  # nsCertType = objsign  # For normal client use this is typical  # nsCertType = client, email  # and for everything including object signing:  # nsCertType = client, email, objsign  # This is typical in keyUsage for a client certificate.  # keyUsage = nonRepudiation, digitalSignature, keyEncipherment  # This will be displayed in Netscape's comment listbox.  nsComment = "OpenSSL Generated Certificate"  # PKIX recommendations harmless if included in all certificates.  subjectKeyIdentifier = hash  authorityKeyIdentifier = keyid,issuer  # This stuff is for subjectAltName and issuerAltname.  # Import the email address.  # subjectAltName=email:copy  # An alternative to produce certificates that aren't  # deprecated according to PKIX.  # subjectAltName=email:move  # Copy subject details  # issuerAltName=issuer:copy  #nsCaRevocationUrl = http://www.domain.dom/ca-crl.pem  #nsBaseUrl  #nsRevocationUrl  #nsRenewalUrl  #nsCaPolicyUrl  #nsSslServerName  # This really needs to be in place for it to be a proxy certificate.  proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo  ####################################################################  [ tsa ]  default\_tsa = tsa\_config1 # the default TSA section  [ tsa\_config1 ]  # These are used by the TSA reply generation only.  dir = ./demoCA # TSA root directory  serial = $dir/tsaserial # The current serial number (mandatory)  crypto\_device = builtin # OpenSSL engine to use for signing  signer\_cert = $dir/tsacert.pem # The TSA signing certificate  # (optional)  certs = $dir/cacert.pem # Certificate chain to include in reply  # (optional)  signer\_key = $dir/private/tsakey.pem # The TSA private key (optional)  signer\_digest = sha256 # Signing digest to use. (Optional)  default\_policy = tsa\_policy1 # Policy if request did not specify it  # (optional)  other\_policies = tsa\_policy2, tsa\_policy3 # acceptable policies (optional)  digests = sha1, sha256, sha384, sha512 # Acceptable message digests (mandatory)  accuracy = secs:1, millisecs:500, microsecs:100 # (optional)  clock\_precision\_digits = 0 # number of digits after dot. (optional)  ordering = yes # Is ordering defined for timestamps?  # (optional, default: no)  tsa\_name = yes # Must the TSA name be included in the reply?  # (optional, default: no)  ess\_cert\_id\_chain = no # Must the ESS cert id chain be included?  # (optional, default: no)  ess\_cert\_id\_alg = sha1 # algorithm to compute certificate  # identifier (optional, default: sha1) |

Bibliografía:

1. https://www.golinuxcloud.com/openssl-create-certificate-chain-linux/#Step\_7\_Create\_Ope SSL\_Intermediate\_CA\_directory\_structure