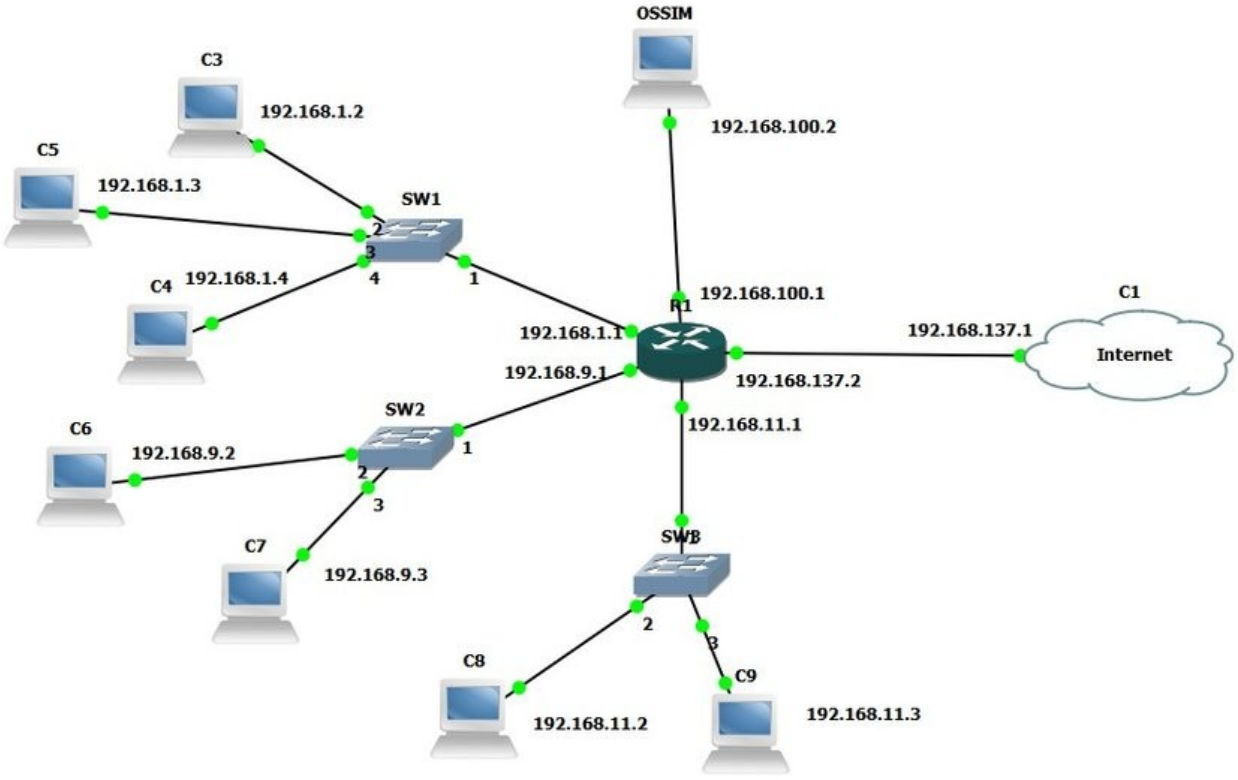
CdLM Ingegneria Informatica
Indirizzo cyber security

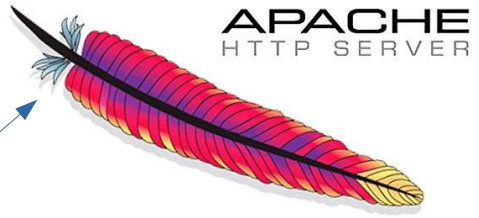
NETWORK SECURITY

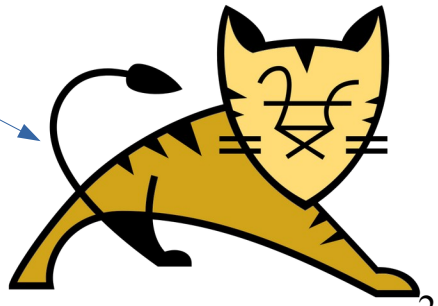


WWW, HTTP e HTTPS

Web Server:

Sicuramente tra le più note applicazioni che utilizzano TCP/IP è WWW (World Wide Web), che utilizza il protocollo HTTP per realizzare la comunicazione.





WWW, HTTP e HTTPS

WWW

- Primi anni di Internet:
- trasferimenti dati FTP pari ad un terzo del traffico globale di Internet
- dal 1995, HTTP supera FTP, occupando la maggior parte della larghezza di banda
- dal 2000 HTTP(s) surclassa le altre applicazioni

- Il Web :
  - consiste di molti documenti, le pagine Web
  - accessibili su Internet
  - contenenti elementi di varia natura: testo, audio, video, collegamenti ipertestuali ad altri documenti correlati
  - Elementi usati per implementare il Web :
  - un browser Web, client applicativo che l'utente esegue per accedere e visualizzare le pagine Web
  - un server Web appropriato, per ottenere una copia della pagina specificata

WWW, HTTP e HTTPS

- Il Web:
  - le pagine vengono rappresentate mediante HTML (HyperText Markup Language)
  - semplice linguaggio di formattazione dei documenti
  - utilizzato per preparare le pagine che devono essere visualizzate dai browser Web

- Un documento HTML:
  - costituito da un file contenente del testo e dei comandi (marcatori)
  - forniscono le linee di riferimento per la visualizzazione (racchiuse tra i simboli < e >)
  - Ogni pagina Web ha un nome univoco, detto URL (Uniform Resource Locator)
  - l'URL inizia con la specifica dello schema usato per accedere all'elemento

    http: // hostname [: port] / path [; parameters] [? query]

- le parentesi quadrate indicano un elemento facoltativo;
- `hostname` specifica il nome di dominio o l'indirizzo IP del computer su si trova il server che gestisce la risorsa;
- `:port` è il numero della porta del protocollo, necessario solo nei casi in cui il server non usa la well-known port 80;
- ***path*** è una stringa che identifica un particolare documento sul server;
- ***;parameters*** è una stringa facoltativa che specifica ulteriori parametri forniti dal client
- ***?query*** è una stringa facoltativa utilizzata dal browser per inviare domande

WWW, HTTP e HTTPS

- Differenza tra HTTP e HTTPS
  - differenza tra HTTP e HTTPS: sicurezza della navigazione on-line
  - un sito web con HTTPS garantisce un livello di sicurezza maggiore rispetto ad uno con HTTP

- Protocolli HTTP ed HTTPS:
  - un set di regole che definiscono il modo in cui due elaboratori debbano effettuare lo scambio di un certo tipo di dati
  - HTTP HyperText Transfer Protocol
  - HTTPS HyperText Transfer Protocol over Secure Socket Layer
  - in HTTPS la comunicazione tra client e server avviene secondo le regole del protocollo HTTP all'interno di una connessione criptata (over Secure Socket Layer).
  - i dati da e verso un sito web raggiunto via HTTPS non viaggiano "in chiaro" ma criptati, evitando che malintenzionati li possano catturare e/o manomettere durante il tragitto

WWW, HTTP e HTTPS

- Certificato SSL *TLS Self-Signed*
  - SSL (Secure Sockets Layer) è una tecnologia standard che garantisce la sicurezza di una connessione a Internet e protegge i dati sensibili scambiati fra due sistemi
  - impedendo ai criminali informatici di leggere e modificare le informazioni trasferite, che potrebbero comprendere anche dati personali
  - Impedisce la lettura e l'intercettazione di qualsiasi dato trasferito fra utenti e siti o due sistemi
  - usa algoritmi di crittografia per crittografare i dati in transito
  - impedendisce la lettura agli hacker durante il transito su una connessione digitale.

- TLS (Transport Layer Security)
  - versione aggiornata e più sicura di SSL.
  - si indicano certificati di sicurezza con la dicitura SSL poiché si tratta di un termine di utilizzo "storico"

- HTTPS://
  - dicitura visualizzata negli URL di un sito Web protetto con un certificato SSL.
  - Facendo clic sul simbolo del lucchetto nella barra del browser, è possibile visualizzare i dettagli del certificato

WWW, HTTP e HTTPS

- *Catene di certificati*
  - Certificato della CA firmato da un intermediario
  - Certificato dell'intermediaro firmato dalla root CA
  - Certificato self signed della root CA (AddTrust External CA Root, questo certificato lo conoscono anche i browser)
  - Il CA Root Certificate (in italiano “Certificato SSL Radice” o “Certificato SSL Intermedio“) è un certificato a chiave pubblica utilizzato per definire la Certification Authority (CA) responsabile per l’emissione dei certificati SSL.

Certificato *Self-Signed*

- totalmente gratuito
- può essere generato facilmente sulle macchine di test.

WWW, HTTP e HTTPS

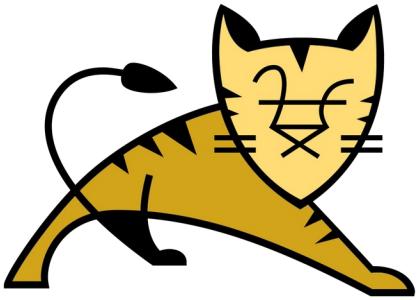
- Apache:
  - web server open-source multipiattaforma
  - il server web più popolare esistente
  - gestito attivamente da Apache Software Foundation
  - usato da aziende di respiro internazionale come Cisco, IBM, Salesforce, General Electric, Adobe, VMware, Xerox, LinkedIn, Facebook, Hewlett-Packard, AT & T, Siemens, eBay.
  - uno dei più vecchi server web, la prima versione risale al 1995.
  - Molti host cPanel oggi utilizzano Apache.
  - controlla ciò che accade dietro le quinte nel servire i file del vostro sito web ai visitatori.

WWW, HTTP e HTTPS

- NGINX

  - software open source che opera come:
  - web server
  - reverse proxy
  - cache
  - media streaming
  - struttura semplice dalle elevate prestazioni
  - funziona su Unix, Linux, macOS, Solaris e Windows
  - fornisce rapidamente contenuti statici senza colpire duramente le risorse di sistema
  - attualmente uno dei software più utilizzati al mondo e in Italia

WWW, HTTP e HTTPS



- Apache Tomcat
  - software open source
  - Web Application Server java oriented, con supporto per pagine JSP e servlet
  - supporto di framework di sviluppo come Struts
  - supporto dei Web Services col framework ApacheAxis
  - supporto di tecnologie Object-relational mapping (Hibernate)
  - standard Web Server, solo per pagine statiche
  - sviluppato dalla Apache Software Foundation

WWW, HTTP e HTTPS

- OpenSSL:
  - realizzazione in forma di software libero dei protocolli SSL/TLS per la certificazione e la comunicazione cifrata
- si compone di:
  - alcune librerie che permettono di incorporare le funzionalità dei protocolli SSL/TLS all'interno di programmi di comunicazione
  - una serie di programmi di utilità per la gestione delle chiavi e dei certificati, arrivando eventualmente anche alla gestione di un'autorità di certificazione.
  - si solito questi programmi sono costituiti da un solo eseguibile monolitico: “openssl”
  - Non esiste una definizione ben precisa di dove devono essere collocati i file che compongono la configurazione e gli strumenti di OpenSSL
  - Solitamente si fa riferimento al file di configurazione “openssl.cnf” e le chiavi sono di solito nei folder “/etc/ssl/” o “/etc/openssl/”

```
openssl <comando> [<opzioni>]
```

Comando	Descrizione
openssl req	Gestione delle richieste di certificazione.
openssl ca	Gestione relativa all'autorità di certificazione.
openssl crl	Gestione del certificato delle revoche.
openssl genrsa	Generazione di parametri RSA.
openssl rsa	Conversione del formato di una chiave privata o di un certificato.
openssl x509	Gestione dei dati dei certificati X.509.

WWW, HTTP e HTTPS

- Installazione Apache2 con SSL/TLS
  - ***apt-get update -y***
  - ***apt-get upgrade -y***
  - ***apt-get install -y apache2 apache2-utils***
  - ***apt-get install -y php7.3 libapache2-mod-php7.3 php7.3-mysql php-common php7.3-cli php7.3-common php7.3-json php7.3-opcache php7.3-readline***
  - ***sudo a2enmod php7.3***
  - ***sudo systemctl restart apache2***
  - ***sudo systemctl restart apache2***
  - ***sudo nano /etc/apache2/ports.conf***
  - ***sudo systemctl restart apache2***

WWW, HTTP e HTTPS

- *cat /etc/apache2/ports.conf*

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8090

<IfModule ssl_module>
Listen 9443
</IfModule>

<IfModule mod_gnutls.c>
Listen 9443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

WWW, HTTP e HTTPS

- Installazione Apache2 con SSL/TLS

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
nano /etc/apache2/conf-available/ssl-params.conf
cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.bak
nano /etc/apache2/sites-available/default-ssl.conf
a2enmod ssl
a2enmod headers
a2ensite default-ssl
a2enconf ssl-params
apache2ctl configtest
systemctl restart apache2
systemctl status apache2
netstat -punta | grep apache2
```

WWW, HTTP e HTTPS

- *cat /etc/apache2/conf-available/ssl-params.conf*

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLHonorCipherOrder On
# Disable preloading HSTS for now.  You can use the commented out header
line that includes
# the "preload" directive if you understand the implications.
# Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
# Requires Apache >= 2.4.11
SSLSessionTickets Off
```

WWW, HTTP e HTTPS

- `cat /etc/apache2/sites-enabled/default-ssl.conf | grep -vE "#"`

```
<IfModule mod_ssl.c>
<VirtualHost _default_:9443>
ServerAdmin webmaster@localhost
ServerName 160.97.52.136
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
SSLEngine on
                SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
                SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
<FilesMatch "\.(cgi|shtml|phtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
SSLOptions +StdEnvVars
</Directory>
</VirtualHost>
</IfModule>
```

WWW, HTTP e HTTPS

- Install Nginx With SSL support

```
apt-get install -y nginx
systemctl status nginx
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
openssl dhparam -out /etc/nginx/dhparam.pem 4096
nano /etc/nginx/snippets/self-signed.conf
nano /etc/nginx/snippets/ssl-params.conf
cd /etc/nginx/sites-available/
cp default default-ssl
nano default-ssl
service nginx restart
nginx -t
cd ../sites-enabled/
ln -s /etc/nginx/sites-available/default-ssl
nginx -t
systemctl restart nginx
systemctl status nginx
netstat -tanup | grep nginx
```

WWW, HTTP e HTTPS

- ***cat /etc/nginx/snippets/self-signed.conf***

  ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;

  ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;

WWW, HTTP e HTTPS

- *cat /etc/nginx/snippets/ssl-params.conf*

```
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/nginx/dhparam.pem;
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;
ssl_ecdh_curve secp384r1; # Requires nginx >= 1.1.0
ssl_session_timeout  10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off; # Requires nginx >= 1.5.9
ssl_stapling on; # Requires nginx >= 1.3.7
ssl_stapling_verify on; # Requires nginx => 1.3.7
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
# Disable strict transport security for now. You can uncomment the following
# line if you understand the implications.
# add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
```

WWW, HTTP e HTTPS

- `cat /etc/nginx/sites-available/default-ssl| grep -vE "#"`

```
server {
listen 443 ssl;
listen [::]:443 ssl;
        include snippets/self-signed.conf;
include snippets/ssl-params.conf;
root /var/www/html;
index index.html index.htm index.nginx-debian.html;
server_name _;
location / {
try_files $uri $uri/ =404;
}
}
```

WWW, HTTP e HTTPS

- Installazione Tomcat 9

```
apt install openjdk-11-jre
apt-get install tomcat9-*
nano /etc/profile.d/tomcat9.sh
source /etc/profile.d/tomcat9.sh
mkdir logs
touch logs/catalina.out
cd /usr/share/tomcat9
cp -Rv skel/conf/ .
cd conf/
nano tomcat-users.xml
nano context.xml
cd ../webapps
```

WWW, HTTP e HTTPS

```
mkdir -p /usr/share/tomcat9/temp/
nano /usr/share/tomcat9-admin/host-manager/META-INF/context.xml
nano /usr/share/tomcat9-admin/manager/META-INF/context.xml
nano /etc/systemd/system/tomcat.service
useradd -s /bin/false -g tomcat9 -d /usr/share/tomcat9 tomcat9
chown -R tomcat9:tomcat9 /usr/share/tomcat9*
chown -R 775 /usr/share/tomcat9*
nano /etc/systemd/system/tomcat.service
systemctl enable tomcat
systemctl start tomcat
systemctl status tomcat
```

WWW, HTTP e HTTPS

```
cd /usr/share/tomcat9/webapps/
ln -s /usr/share/tomcat9-admin/host-manager/ .
ln -s /usr/share/tomcat9-admin/manager/
chown -R tomcat9:tomcat9 /usr/share/tomcat9*
chown -R 775 /usr/share/tomcat9*
cd $JAVA_HOME/bin
keytool -genkey -alias tomcat -keyalg RSA
cd /usr/share/tomcat9
mkdir keystore
keytool -list
cd
cp .keystore tomcat9
mv tomcat9 /usr/share/tomcat9/keystore/
cd /usr/share/tomcat9/keystore/
chown tomcat9:tomcat9 tomcat9
nano ../conf/server.xml
service tomcat stop
service tomcat start
service tomcat status
netstat -punta | grep java
```

WWW, HTTP e HTTPS

- ```
  cat /usr/share/tomcat9/conf/tomcat-users.xml
  <?xml version="1.0" encoding="UTF-8"?>
  <tomcat-users xmlns="http://tomcat.apache.org/xml"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
                version="1.0">
  <!-- user manager can access only manager section -->
  <role rolename="manager-gui" />
  <user username="manager" password="firewall" roles="manager-gui" />

  <!-- user admin can access manager and admin section both -->
  <role rolename="admin-gui" />
  <user username="admin" password="firewall" roles="manager-gui,admin-gui" />
  </tomcat-users>
  ```
  ```

WWW, HTTP e HTTPS

- `cat /usr/share/tomcat9/conf/context.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- The contents of this file will be loaded for each web application -->
<Context>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
</Context>
```

WWW, HTTP e HTTPS

- `cat /usr/share/tomcat9-admin/host-manager/META-INF/context.xml`

```
<?xml version="1.0" encoding="UTF-8"?>

<Context antiResourceLocking="false" privileged="true" >

<!--  <Valve
className="org.apache.catalina.valves.RemoteAddrValve"

         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|160\.\
d+\.\d+\.\d+" />-->

  <Manager sessionAttributeValueClassNameFilter="java\.lang\.
(?:Boolean|Integer|Long|Number|String)|
org\.apache\.catalina\.filters\.CsrfPreventionFilter\
$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>

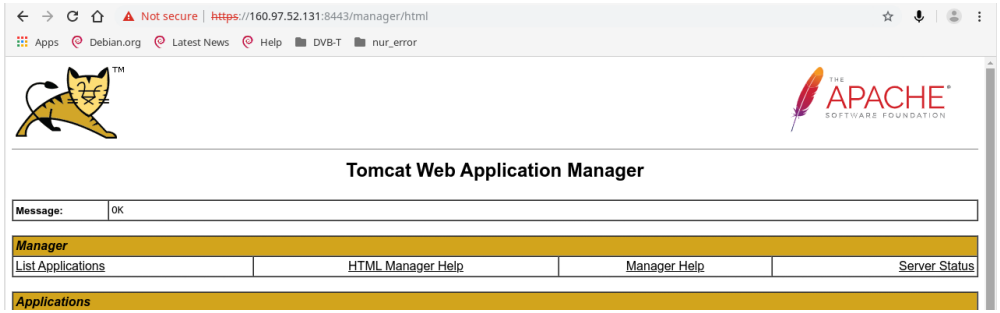
</Context>
```

WWW, HTTP e HTTPS

- ```
  cat /etc/systemd/system/tomcat.service
  [Unit]
  Description=Apache Tomcat Web Application Container
  After=network.target

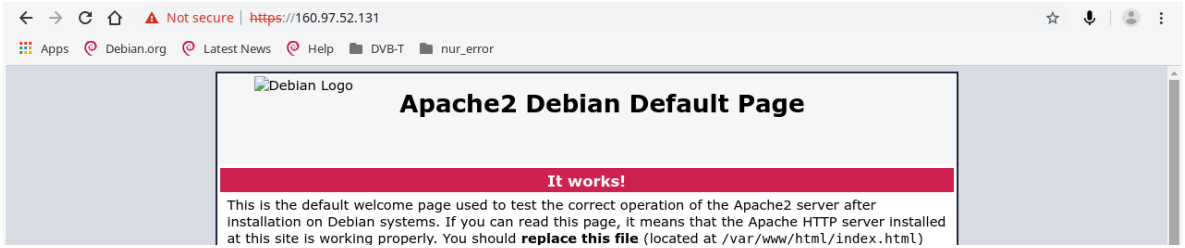
  [Service]
  Type=forking
  Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
  Environment=CATALINA_PID=/usr/share/tomcat9/temp/tomcat.pid
  Environment=CATALINA_HOME=/usr/share/tomcat9
  Environment=CATALINA_BASE=/usr/share/tomcat9
  Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
  Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'
  ExecStart=/usr/share/tomcat9/bin/startup.sh
  ExecStop=/usr/share/tomcat9/bin/shutdown.sh
  User=tomcat9
  Group=tomcat9
  UMask=0007
  RestartSec=10
  Restart=always

  [Install]
  WantedBy=multi-user.target
  ```
  ```

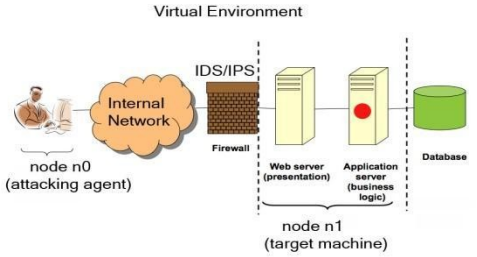
WWW, HTTP e HTTPS

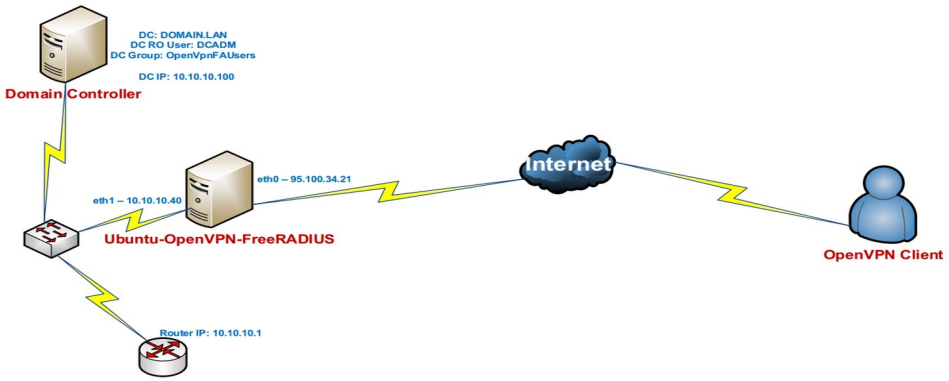
- Webserver SSL Pages (Self signed Certs)

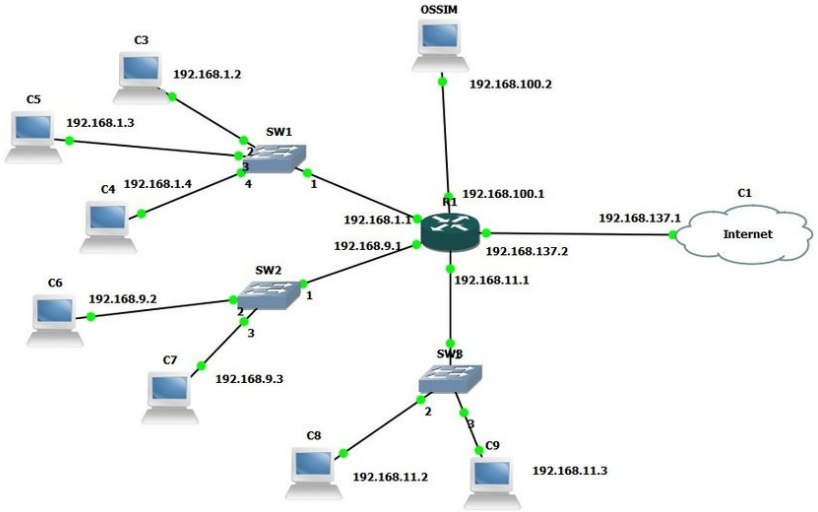


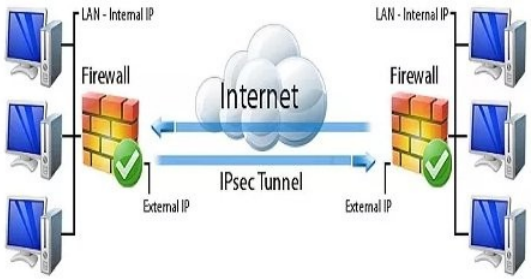



DATA PROTECTION
COMPUTER
CYBER SECURITY
NETWORK SECURITY
DEVICES


Virtual Environment
IDS/IPS
Internal Network
Firewall
Database
Web server (presentation)
Application server (business logic)
node n0
(attacking agent)
node n1
(target machine)


DC: DOMAIN.LAN
DC RO User: DCADM
DC Group: OpenVpnFAUsers
DC IP: 10.10.10.100
Domain Controller
eth1 – 10.10.10.40
eth0 – 95.100.34.21
Ubuntu-OpenVPN-FreeRADIUS
Internet
OpenVPN Client
Router IP: 10.10.10.1


OSSIM
192.168.100.2
C3
192.168.1.2
C5
192.168.1.3
SW1
C4
192.168.1.4
R1
192.168.100.1
192.168.1.1
192.168.137.1
C1
Internet
192.168.9.1
192.168.137.2
SW2
C6
192.168.9.2
192.168.11.1
C7
192.168.9.3
SW3
C8
192.168.11.2
C9
192.168.11.3


LAN - Internal IP
LAN - Internal IP
Firewall
Internet
Firewall
IPsec Tunnel
External IP
External IP


NETWORK SECURITY

La saggezza vi darà la morale,
La conoscenza vi darà la verità,
La verità vi darà libertà,
La conoscenza libera vi darà la saggezza ….

GRAZIE PER L'ATTENZIONE