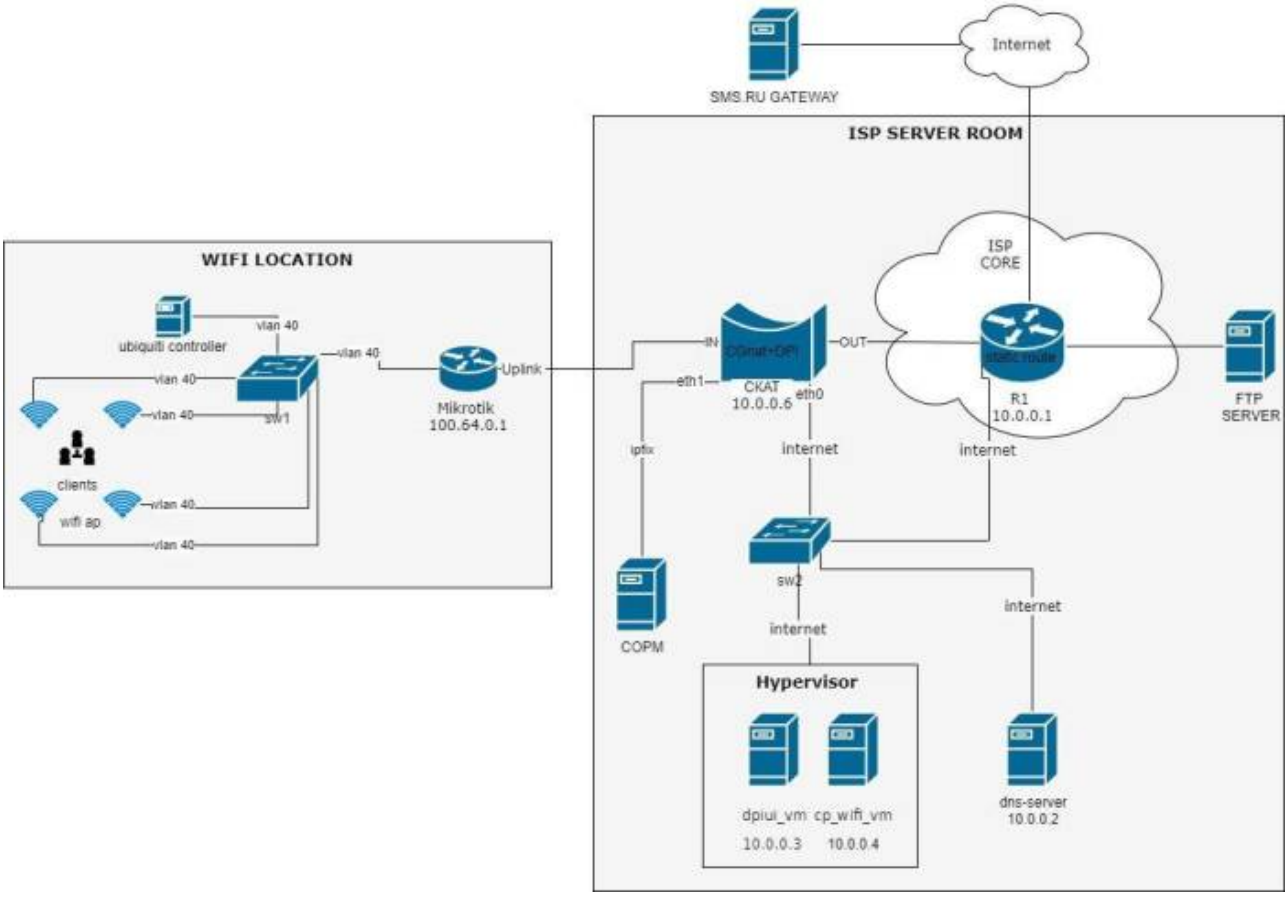
Содержание

Configuring GUI, SSG, and WiFi HotSpot with Session Management Enabled

Network Topology

1. Connect the equipment according to the network topology.



Authorization Sequence

1. The subscriber connects to the WiFi network
2. A welcome page appears informing the subscriber to open a browser and identify themselves
3. The subscriber opens a browser, and upon navigating to any URL, they are redirected to the identification page
4. The subscriber enters their phone number and requests an access code
5. The access code is sent to the phone number via SMS
6. The subscriber enters the received access code
7. Session cookies are recorded on the subscriber's device for a specified period, and the subscriber is redirected to the requested URL

Setting Up Virtual Machines (VM)

1. Create two virtual machines with the following minimum specifications:
   - VM `dpiui_vm` – 1 CPU, 2GB RAM, 50GB hard disk, Guest OS CentOS 7, NIC 1
   - VM `cp_wifi_vm` – 1 CPU, 1GB RAM, 30GB hard disk, Guest OS CentOS 7, NIC
2. Install the latest version of CentOS 7 (build-2009 at the time of writing) on both virtual machines. Choose minimal installation during setup.
   After OS installation, open the console and install packages on both VMs: first

   ```
   yum install epel-release
   ```

   and then:

   ```
   yum install nano tcpdump openssh-server openssh-clients
   ```

3. Disable SELinux on both VMs:
   - Edit the file `/etc/sysconfig/selinux`
   - Set the parameter `SELINUX=disabled` and reboot the VM

Installing and Configuring dpiui_vm

1. Install DPIUI on `dpiui_vm` following the instructions
2. Configure the network on both VMs and SSG:

   ```
   BOOTPROTO=static
   ONBOOT=yes
   IPADDR=10.0.0.x
   NETMASK=255.255.255.0
   GATEWAY=10.0.0.1
   DNS1=10.0.0.2
   ```

   IPADDR — specify for each host according to the scheme (or use your own addressing).

3. Log in to the GUI and add both VMs and SSG in the "EQUIPMENT" section, following the instructions:

Installing and Configuring cp_wifi_vm

1. Install the `wifi_hotspot` package on the `cp_wifi_vm` VM following the instructions:
2. Edit the configuration file for Hotspot:

   ```
   nano /var/www/html/wifi_hotspot/backend/.env
   ```

   Change/add only these lines:

   1. **AAA_HOTSPOT_IP – 10.0.0.4**
      NAS server address, IPv4/IPv6, if unknown — 0.0.0.0
   2. **AAA_HOTSPOT_PORT – 0**

NAS server port, number, if unknown — 0

3. **AAA_HOTSPOT_ID – 2**
   Network access point ID, integer between 0 and 1000, must be filled in for public WiFi access points, corresponds to the access point ID in field 1 from the access point export
4. **AAA_EXPORT_ENABLED=1**
   Enable AAA export
5. **AUTH_CODE_LENGTH=4**
   Change the number of characters in the SMS authorization code

If the parameter AUTH_CODE_LENGTH is set, then in the file `/var/www/html/wifi_hotspot/frontend/env.js` set the value:

```
AppEnv.AuthCodePlaceHolder = "0000";
```

Finally, run the command:

```
php /var/www/html/wifi_hotspot/backend/artisan queue:restart
```

Installing and Configuring dhcp-isc on cp_wifi_vm

1. Install the `dhcp-isc` package:

   ```
   yum install dhcp expect
   ```

2. Configure the static ARP scripts and the `dhcpd.conf` configuration file:
   - First, the `dhcpd` configuration file:

     ```
     nano /etc/dhcp/dhcpd.conf
     ```

     Set your values for `option domain-name` and `option ntp-servers`!

     ```
     ddns-update-style none;
     authoritative;
     db-time-format local;
     log-facility local7;

     subnet 100.64.0.0 netmask 255.255.252.0 {
       range 100.64.0.3 100.64.3.254;
       default-lease-time 600;
       max-lease-time 600;
       option subnet-mask 255.255.252.0;
       option broadcast-address 100.64.3.255;
       option routers 100.64.0.1;
       option ntp-servers <ntp-server>;
       option domain-name-servers 10.0.0.2;
       option domain-name "name.local";

       on commit {
           set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
           set ClientMac = concat (
     ```

```
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,1,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,2,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,3,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,4,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,5,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,6,1))),2));
      log(concat("Request: IP: ", ClientIP, " Mac: ", ClientMac));

 execute("/usr/local/etc/dhcpd/clients_add_drop.sh", "add",
ClientIP, ClientMac);}
  on release {
      set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
      set ClientMac = concat (
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,1,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,2,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,3,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,4,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,5,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,6,1))),2));

      log(concat("Release: IP: ", ClientIP, " Mac: ", ClientMac));
      execute("/usr/local/etc/dhcpd/clients_add_drop.sh",
"drop_rls", ClientIP, ClientMac);}
  on expiry {
      set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
      log(concat("Timeout: IP: ", ClientIP));
      execute("/usr/local/etc/dhcpd/clients_add_drop.sh",
"drop_exp", ClientIP);}
}
subnet 10.0.0.0 netmask 255.255.255.0 {
}
```

Create directories and change their permissions:

```
mkdir /usr/local/etc/dhcpd/ && chown dhcpd:dhcpd
/usr/local/etc/dhcpd/

touch /usr/local/etc/dhcpd/clients_add_drop_mysql.sh && touch
/usr/local/etc/dhcpd/clients_add_drop.sh
```

```
&& chown dpiacc:dpiacc /usr/local/etc/dhcpd/*

chmod 755 /usr/local/etc/dhcpd/

chmod 755 /usr/local/etc/dhcpd/*
```

Then copy the following script to `/usr/local/etc/dhcpd/clients_add_drop.sh`:

```
#!/usr/bin/expect -f

set METHOD [lindex $argv 0]
set IP_ADDR [lindex $argv 1]
set MAC_ADDR [lindex $argv 2]
set MAC_ADDR [string toupper $MAC_ADDR]
#Client interface on MikroTik:
set INT_CLIENT "vWifi"
set status 0

#Record dhcp-lease (start and end) in the Hotspot database
spawn /usr/local/etc/dhcpd/./clients_add_drop_mysql.sh "$METHOD"
"$IP_ADDR" "$MAC_ADDR"

expect "end_mysql";

#Connecting to the router
spawn ssh -i /usr/local/etc/dhcpd/.ssh/id_rsa admin+t@100.64.0.1 -
oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev

/null

#Adding a static ARP record to the router
expect {
   "*$ " {
      set timeout 15
      if { $METHOD == "add" } {
         send "ip arp add interface=$INT_CLIENT address=$IP_ADDR
mac-address=$MAC_ADDR\r"
         expect {
            "*failure*" { set status 2 }
            "*dynamic*" { set status 2 }
            "*duplicate*" { set status 2 }
            "*invalid*" { set status 2 }
            "*success*" { set status 0 }
            "*already*" { set status 0 }
            "*input does not match*" { set status 2 }
         }
      }
      if { $METHOD == "drop_rls" } {
         send "ip arp remove [find address~\"$IP_ADDR\"]\r"
         expect {
            "*failure*" { set status 2 }
```

```
            "*invalid*" { set status 2 }
            "*not such item*" { set status 2 }
            "*success*" { set status 0 }
         }
      }
      if { $METHOD == "drop_exp" } {
         send "ip arp remove [find address~\"$IP_ADDR\"]\r"
         expect {
            "*failure*" { set status 2 }
            "*invalid*" { set status 2 }
            "*not such item*" { set status 2 }
            "*success*" { set status 0 }
         }
      }
   }
}
exit $status
```

and /usr/local/etc/dhcpd/clients_add_drop_mysql.sh:

```
#!/bin/bash
#Expecting external IP
EXTIP=10.0.0.3
if [ $1 == 'add' ]; then
   echo "Connecting to Hotspot database"
   mysql -u hotspot -ppassword -h $EXTIP -D hotspot -e "SELECT *
FROM radcheck WHERE attribute = 'Calling-Station-Id' AND
value='$3'" | grep $3
   if [ $? -eq 1 ]; then
      mysql -u hotspot -ppassword -h $EXTIP -D hotspot -e "INSERT
INTO radcheck (username, attribute, op, value) VALUES ('$3',
'Calling-Station-Id', '==', '$3')"
      echo "Inserted $3 MAC address to Hotspot"
   else
      mysql -u hotspot -ppassword -h $EXTIP -D hotspot -e "UPDATE
radcheck SET username='$3',attribute='Calling-Station-
Id',op='==',value='$3' WHERE attribute='Calling-Station-Id' AND
value='$3'"
      echo "Updated $3 MAC address in Hotspot"
   fi
else
   echo "Connecting to Hotspot database"
   mysql -u hotspot -ppassword -h $EXTIP -D hotspot -e "DELETE
FROM radcheck WHERE attribute = 'Calling-Station-Id' AND
value='$3'"
   echo "Removed $3 MAC address from Hotspot"
fi
echo "end_mysql"
```

Enable DHCP at startup:

```
systemctl enable dhcpd.service
```

Finally, generate the key on the `cp_wifi_vm` host and copy it to the router:

```
ssh-keygen -q -t rsa -N '' -f /usr/local/etc/dhcpd/.ssh/id_rsa && ssh-copy-id -i /usr/local/etc/dhcpd/.ssh/id_rsa.pub admin+t@100.64.0.1
```