

# Captive Portal



- System typically used by business centers, airports, hotel lobbies, coffee shops, and other venues which offer Wi-Fi hot spots for Internet users
- It permits to authenticate a client by username and password
- It permits to log the client connections

# Captive Portal



- When an user opens a web page the browser is automatically redirected to a login page on an authentication web server
- On the login page the user inserts his credentials (username and password) and then he can access to Internet
- If the user has not an account, he can register a new account for free or using the payment system (PayPal or Credit Card)



- The Wifidog project is an open source captive portal solution
- <http://www.wifidog.org/>
- It consists of two components:
  - Authenticator (one)
  - Gateway (one or more)
- If there's only one gateway, it can run on the same machine as the authenticator

# Wifidog Gateway



- Designed for and runs on GNU/Linux servers and embedded linux devices (written in C)
- It uses the Linux netfilter/iptables system to filter user traffic
- It implements an embedded web server to redirect to the Authenticator the HTTP connections of the not authenticated user

# Wifidog Gateway



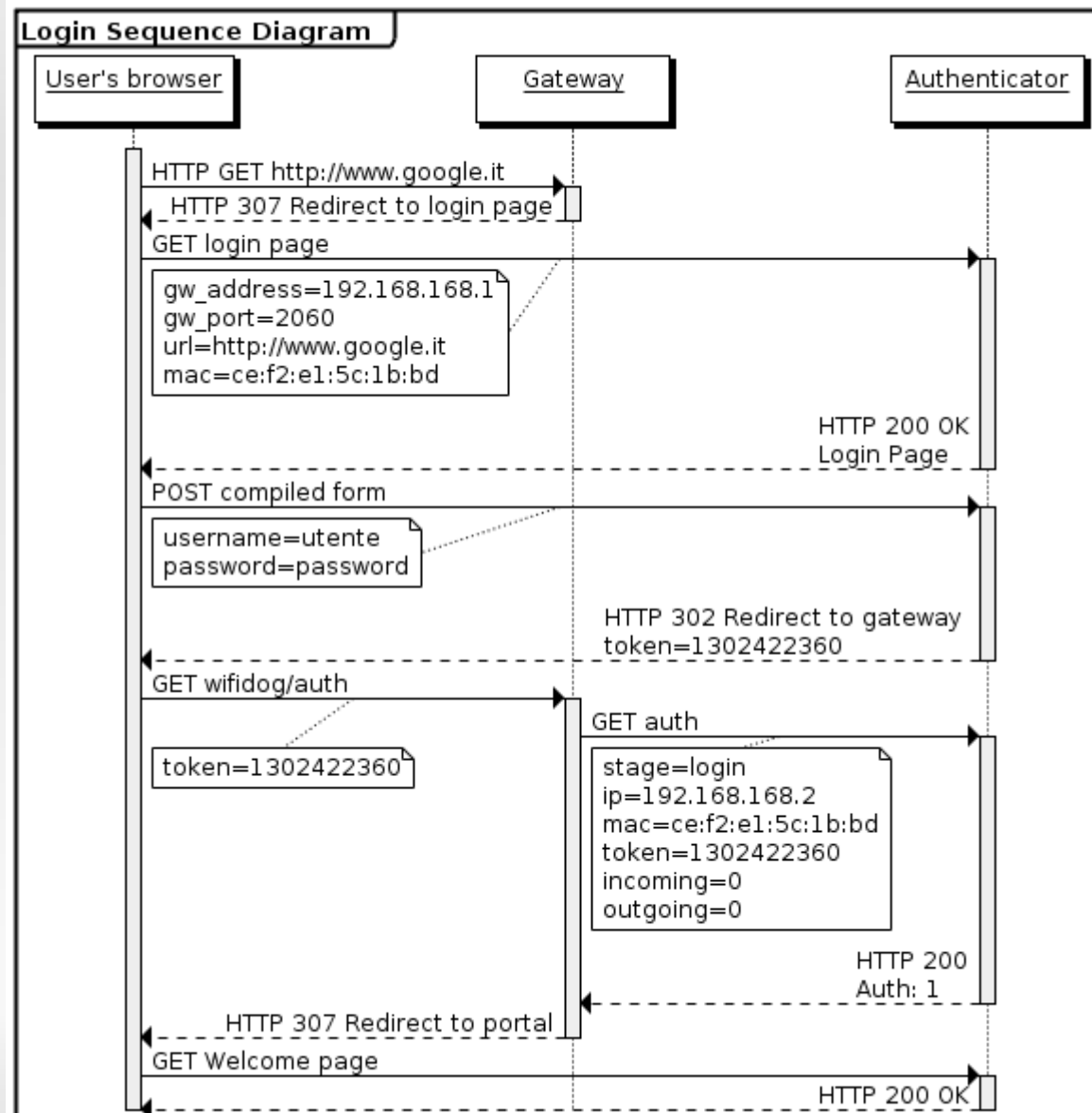
- Each active connection is identified by:
  - Client MAC address
  - Client IP address
  - Token (random id of the session)
- Periodically the gateway:
  - Measures the incoming and outgoing traffic for each client in order to check whether he is still connected
  - For each connected client, asks the authenticator whether the client connection is still valid and then it updates the iptables rules accordingly

# Wifidog Authenticator



- It's a PHP web application and it stores user credentials and connection logs in a PostgreSQL database
- Support for differing types of hotspots:
  - Splash Only mode: Users are redirected to the portal, but do not have to login in order to use services
  - Normal Mode: Users are unique and must have a valid email address in order to open an account.

# Login Sequence Diagram



# Login Sequence Diagram



- The client browser ask a page on Internet
- An iptables rule on the gateway machine redirects the traffic to the web server embedded on the wifidog gateway
- The wifidog gateway gets the MAC address and the IP address of the client and redirects him to the authenticator



# Login Sequence Diagram



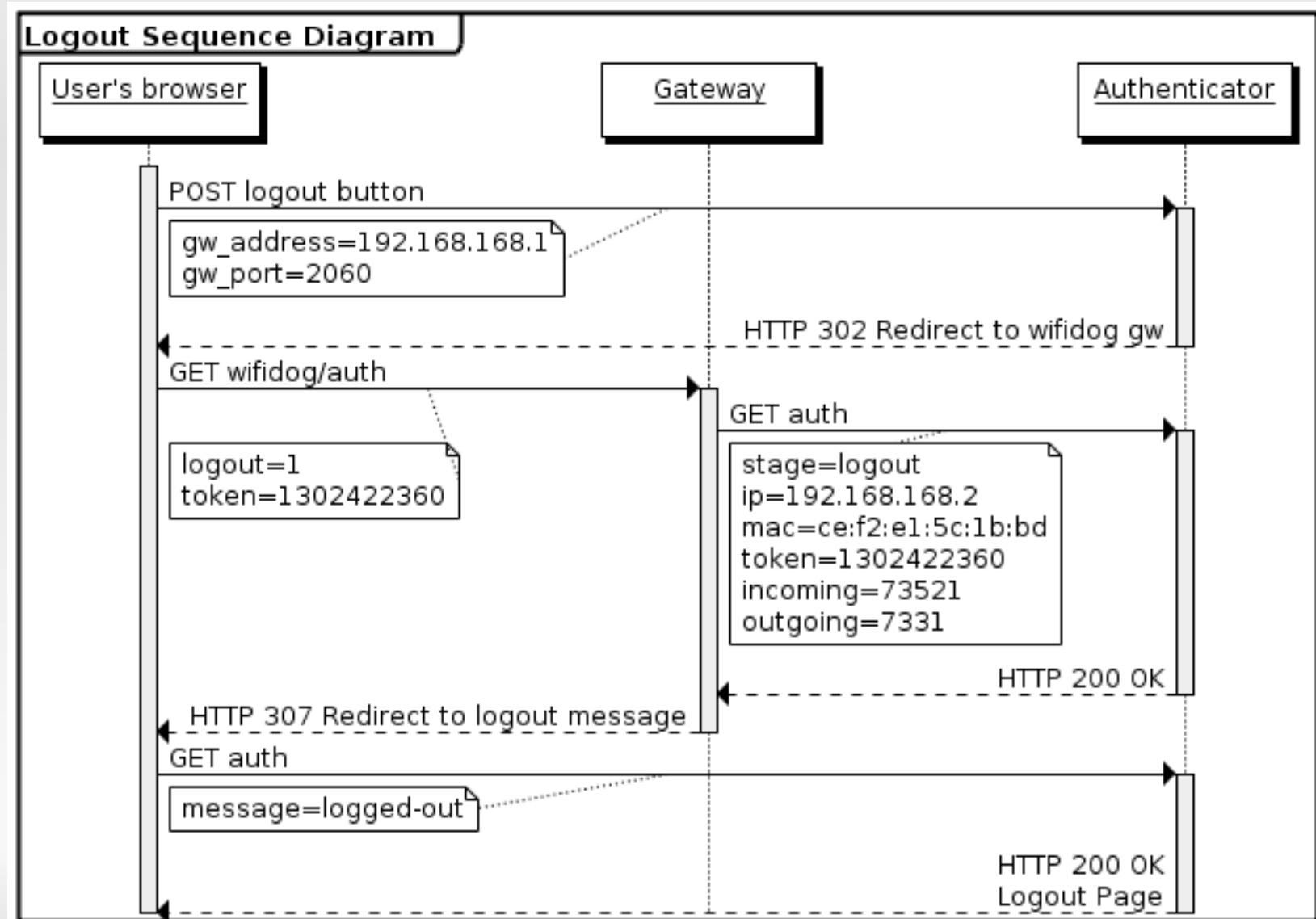
- The authenticator provides the login page
- The client fill the login form and submit it to the authenticator
- The authenticator verifies the credentials and create a new connection identified by MAC,IP,token (only if the provided credentials are valid).
- Then the authenticator redirects the browser to the wifidog gateway

# Login Sequence Diagram



- The wifidog gateway ask the authenticator whether the token provided by the client is valid.
- In case of success, the wifidog gateway updates the iptables rules in order to avoid that the traffic of the authenticated client will be redirected to its embedded web server.
- Then the wifidog gateway redirects the browser the welcome page. At this moment the client has completed the login process and can start to browse the Internet.

# Logout Sequence Diagram

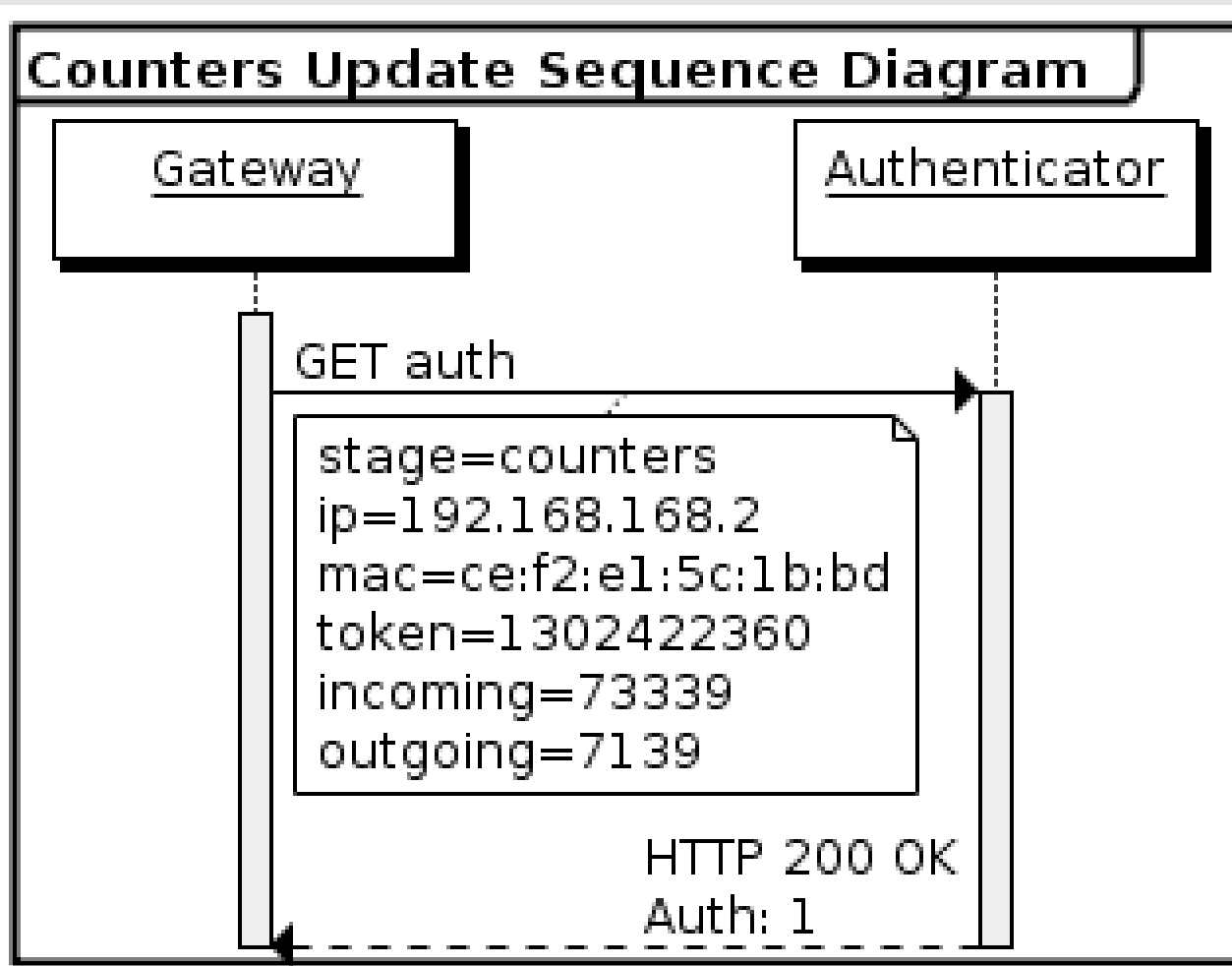


# Logout Sequence Diagram



- The client press the button "Logout" and his browser contacts the authenticator
- The authenticator redirects the browser to the wifidog gateway providing the session token
- The wifidog gateway updates its iptables rules, performs an explicit logout calling the authenticator and at last redirects the browser to the authenticator
- At last, the browser loads the "Successfully Logout" page

# Periodic Validation



# Periodic Validation



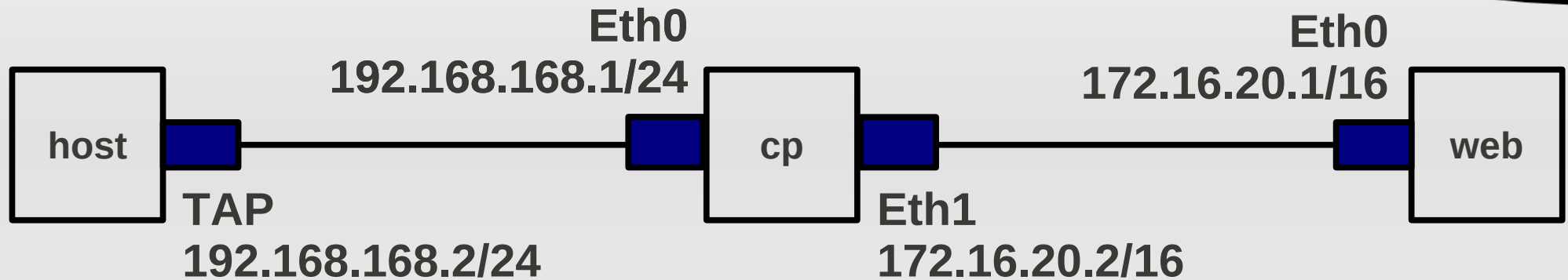
- The wifidog gateway periodically asks the authenticator whether the client is still authorized (sending to the authenticator also the outgoing traffic counter and the incoming traffic counter)
- The authenticator verifies the specific user constraints (traffic, time, period of the day, day, etc..) and replies with “Auth: 0” or with “Auth: 1”

# Session timeout



- The wifidog gateway periodically sends an ICMP Echo Request in order to stimulate traffic activity
- When the gateway receives an IP packet from the client it updates the relative outgoing traffic counter
- Then it checks whether the outgoing traffic counter has been updated since the last check
- After a configurable number of failed checks, the wifidog considers the session timed out and it updates the iptables rules

# Netkit lab (wifidog.tar.gz)



- Run the lab and add a static route for the network 172.16.0.0/16 on the host machine:

```
sudo ip route add 172.16.0.0/16 via 192.168.168.1
```

- Check the connectivity with the web server pointing your browser to <http://172.16.20.1>



# Netkit lab (wifidog.tar.gz)



- Start wifidog on the cp machine:

```
wifidog -c /tmp/cpgateway.conf -d 0
```

- And verify that it's running:

```
# wdctl status
WiFiDog status
Version: 20090925
Uptime: 0d 0h 0m 9s
Has been restarted: no
Internet Connectivity: yes
Auth server reachable: yes
Clients served this session: 0
```

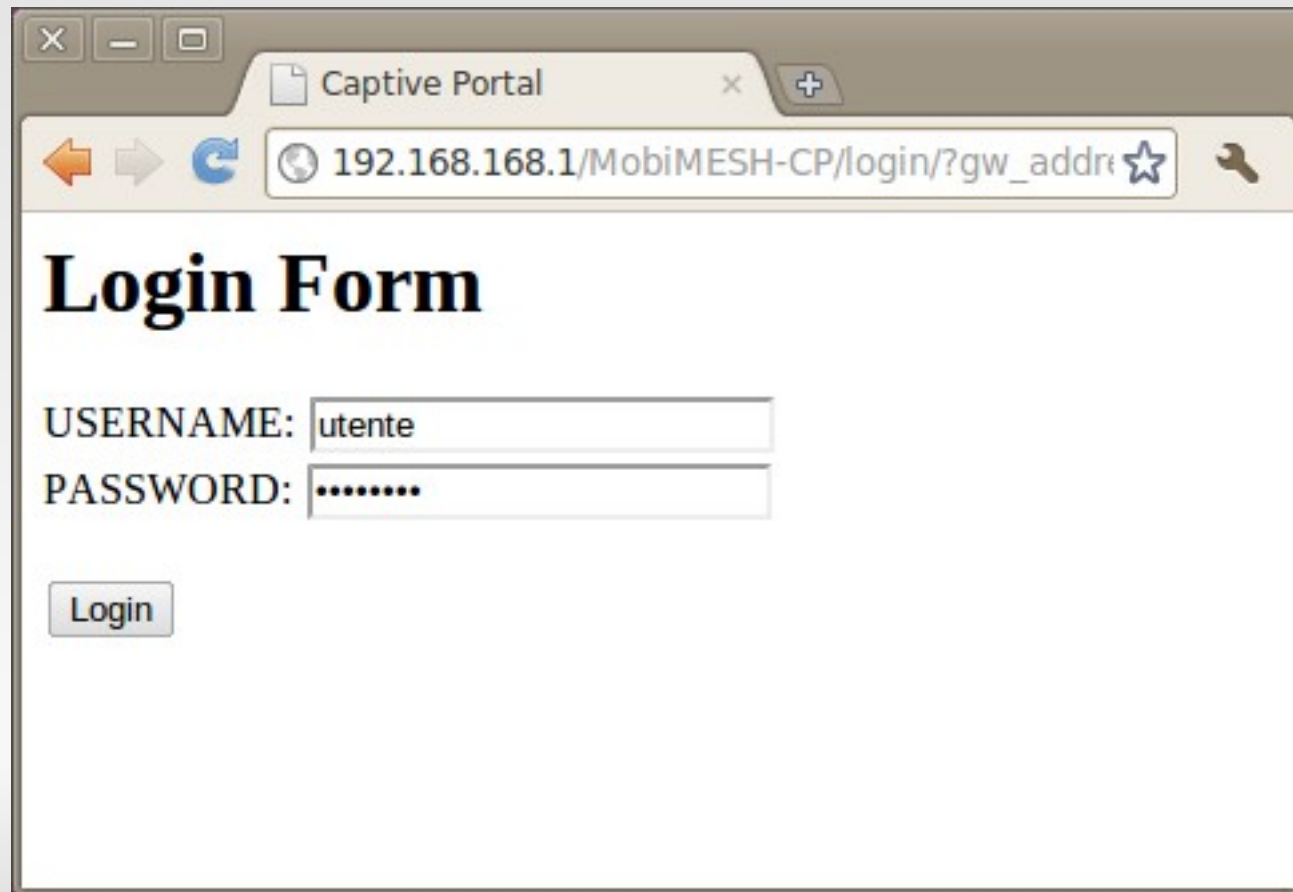
```
0 clients connected.
```

```
Authentication servers:
Host: 192.168.168.1 (192.168.168.1)
```

# Netkit lab (wifidog.tar.gz)



- Point your browser to <http://172.16.20.1> again
- You will be redirected to the login page:

A screenshot of a web browser window. The title bar shows 'Captive Portal'. The address bar contains '192.168.168.1/MobiMESH-CP/login/?gw\_addr'. The main content area displays a 'Login Form' with two input fields: 'USERNAME:' with the value 'utente' and 'PASSWORD:' with masked characters '.....'. Below the fields is a 'Login' button.

**Login Form**

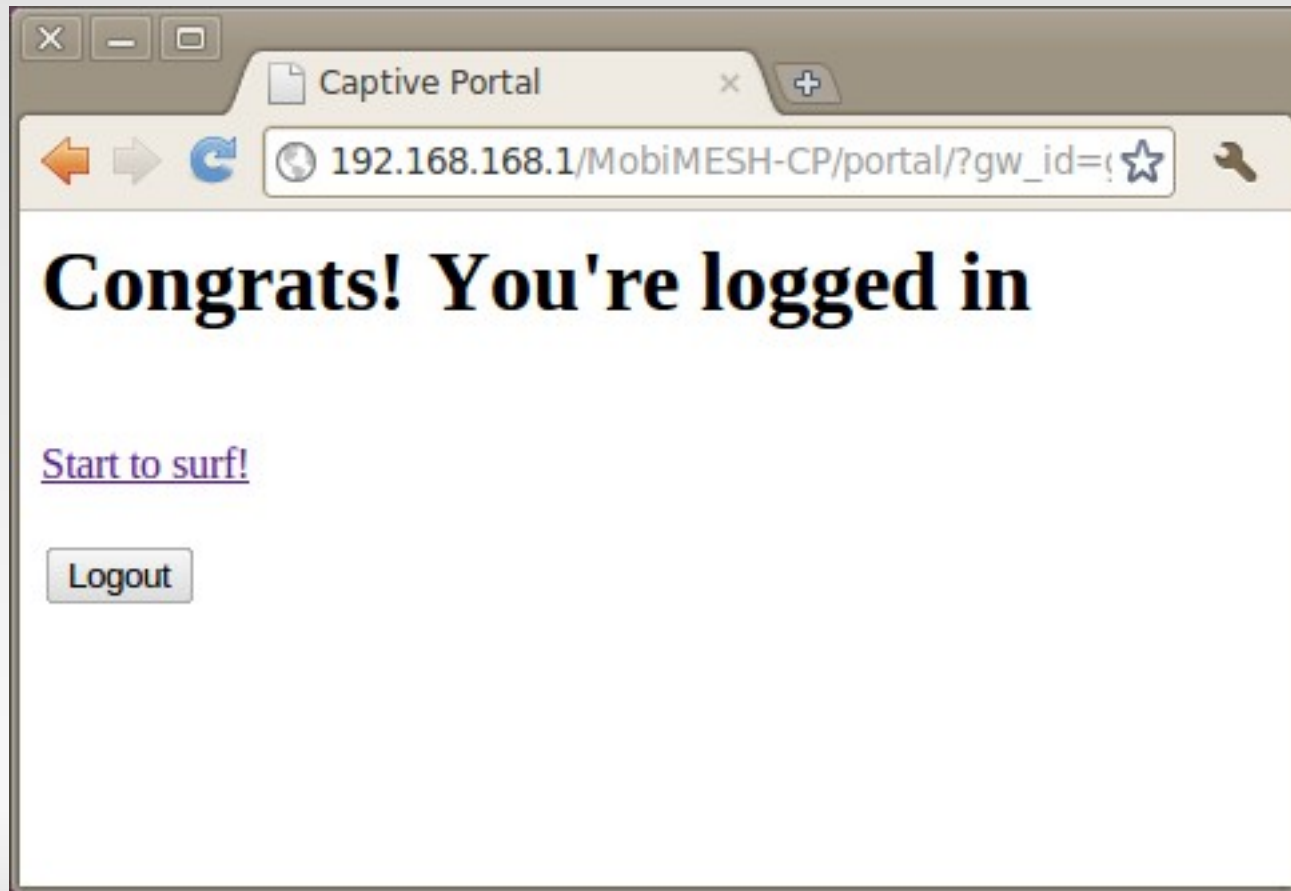
USERNAME:

PASSWORD:

# Netkit lab (wifidog.tar.gz)



- Insert your credentials (utente,password) and press Login:



# Netkit lab (wifidog.tar.gz)



```
# wdctl status
WiFiDog status
Version: 20090925
Uptime: 0d 0h 6m 14s
Has been restarted: no
Internet Connectivity: yes
Auth server reachable: yes
Clients served this session: 2
```

1 clients connected.

Client 0

```
IP: 192.168.168.2 MAC: aa:5a:4a:4e:a0:7a
Token: aa:5a:4a:4e:a0:7a1302602976
Downloaded: 41452
Uploaded: 7788
```

Authentication servers:

```
Host: 192.168.168.1 (192.168.168.1)
```

# Netkit lab (wifidog.tar.gz)



- In order to force the session timeout run the following command on the host machine and wait the timeout period (5 minutes by default):
- To force immediately the session reset, run the following command:

```
iptables -I INPUT -p icmp -s 192.168.168.1 -j DROP
```

```
wdctl reset 192.168.168.2
```