NullSec

YOUR SHIELD IN THE DIGITAL BATTLE

A new horizon to SunToWater.

Project Report to SunToWater co.

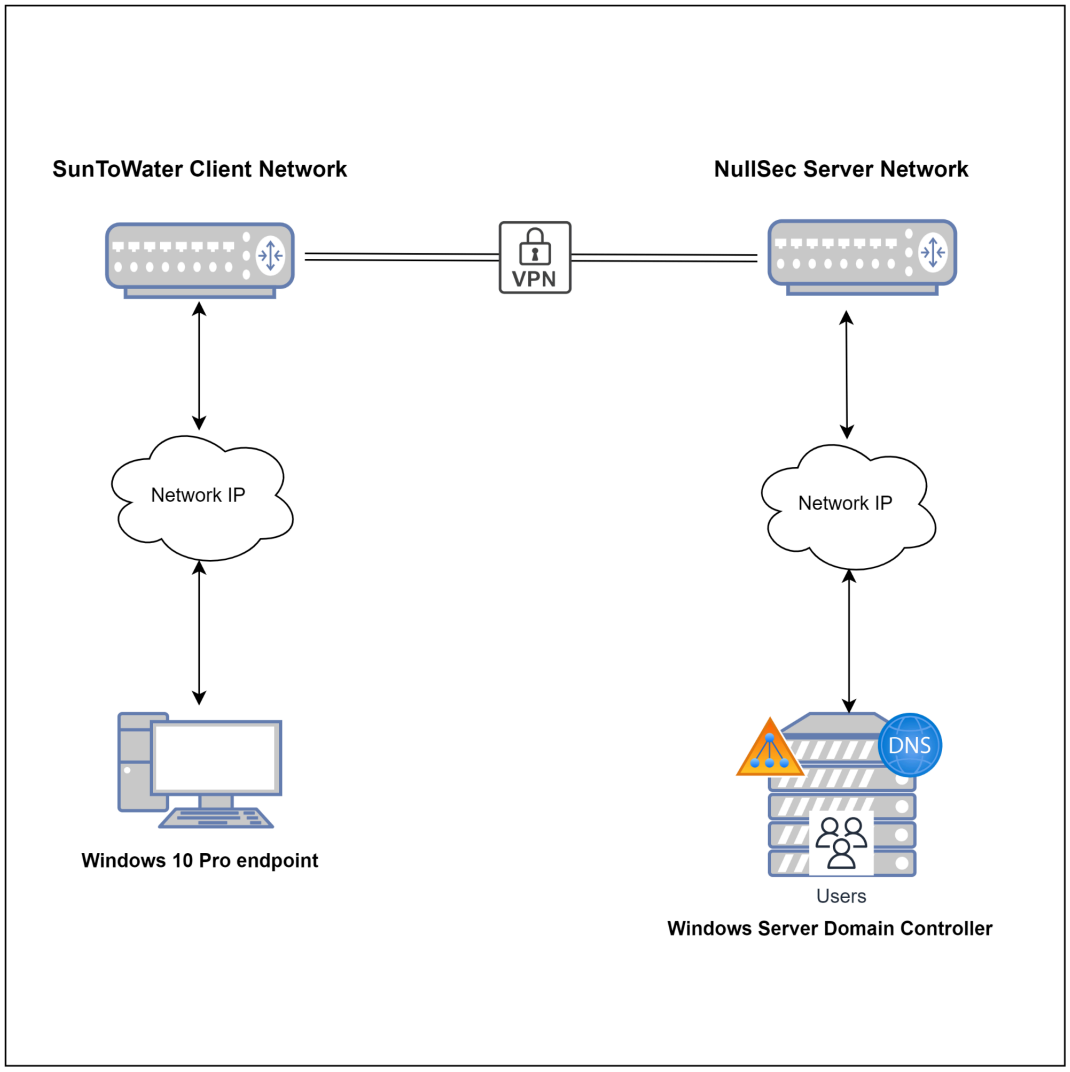
Abstract

In this project, the NullSec team successfully implemented a site-to-site VPN tunnel connecting the SunToWater company network to our own infrastructure. Additionally, we deployed a RADIUS authentication server to enhance network security. Furthermore, we developed a PowerShell script capable of automating the setup process for a new installation of Windows Server, enabling the server to function as a domain controller with all requisite services operational.

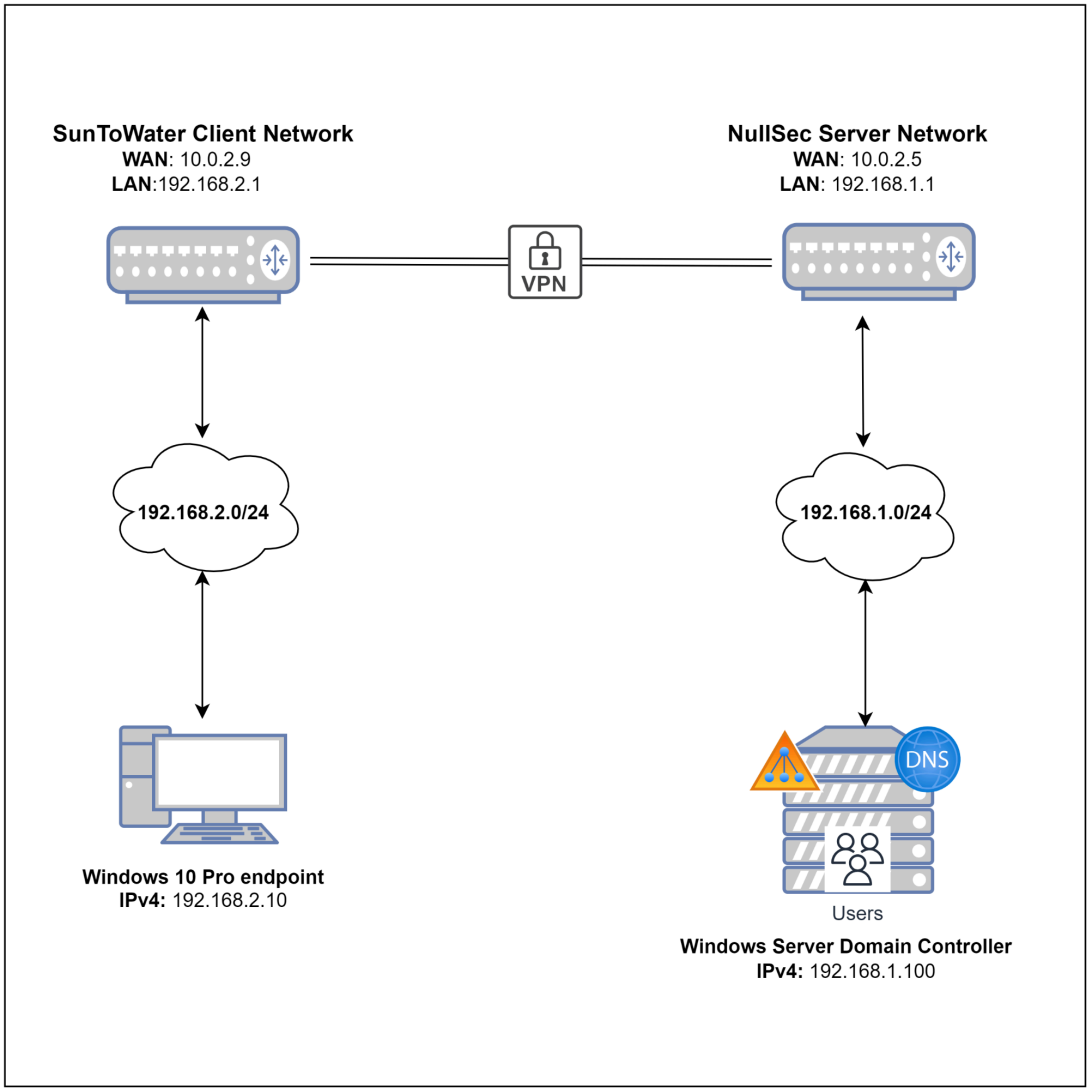
https://www.securew2.com/blog/configure-windows-server-2019

Network Topology diagram

- Before



- After


SunToWater Client Network
WAN: 10.0.2.9
LAN:192.168.2.1
NullSec Server Network
WAN: 10.0.2.5
LAN: 192.168.1.1
VPN
192.168.2.0/24
192.168.1.0/24
DNS
Windows 10 Pro endpoint
IPv4: 192.168.2.10
Users
Windows Server Domain Controller
IPv4: 192.168.1.100

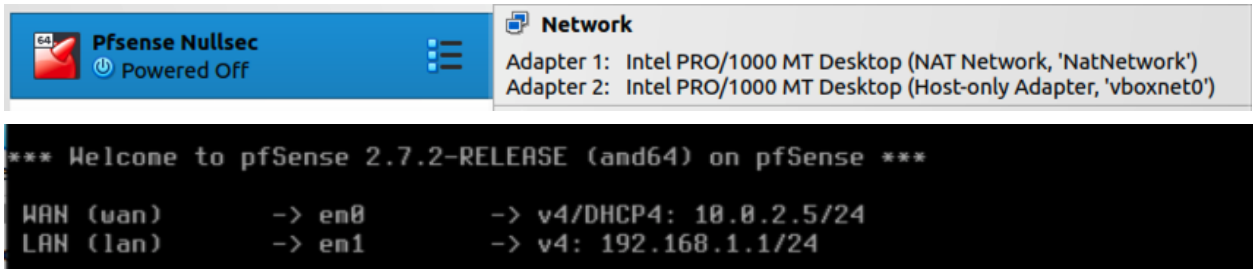
Implementation of Virtual Private Networking (VPN)

We have set up two environments in VirtualBox to simulate the **Nullsec** server and the **Sun To Water client.**

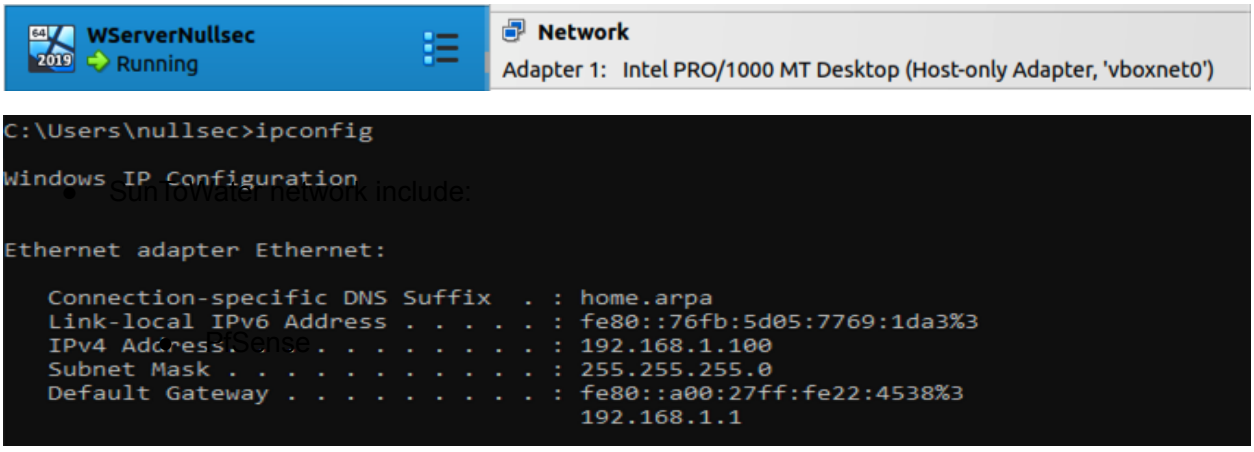
The **Nullsec** environment consists of a Pfsense and a Windows Server, while the **Sun to Water** client contains a PfSense and a Windows 10. We modified the WAN of the PfSense on the client to place them on different networks.

Below, we provide screenshots of each VM and their respective networks and IPs.

- **Nullsec** network include:
    - pfSense

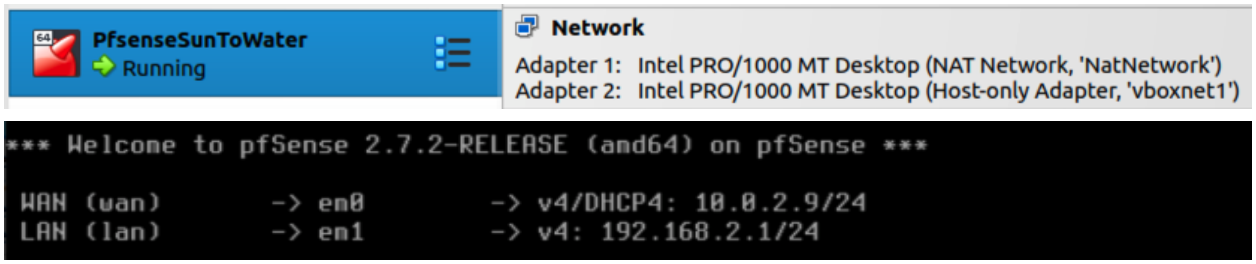


    - Windows Server Domain Controller

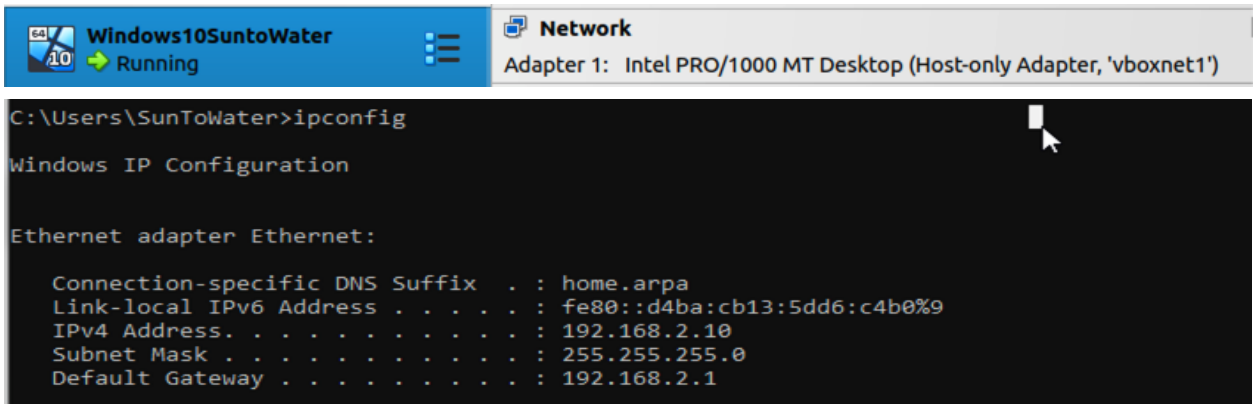


- Sun to Water network include:
    - pfSense

- **Sun To Water** network include:
    - pfSense



    - Windows 10 Endpoint

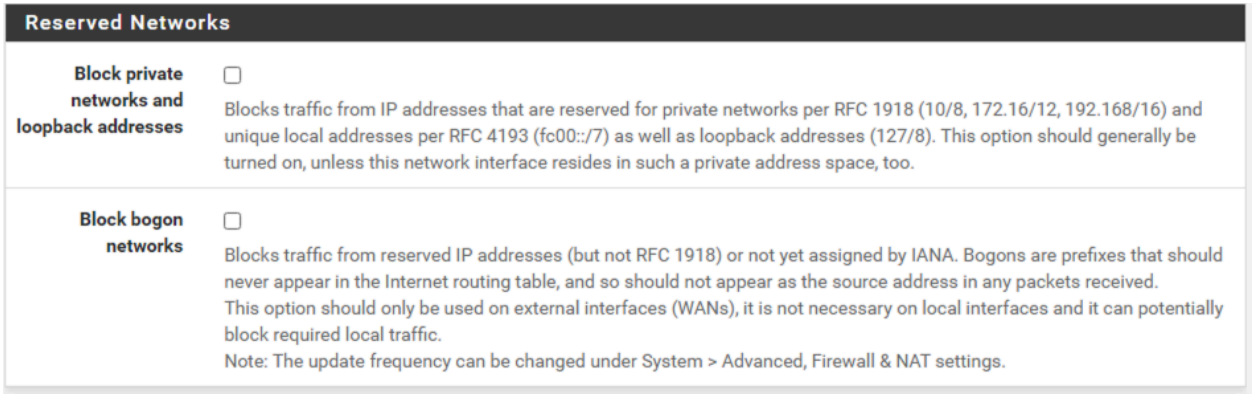


Configuring pfSense

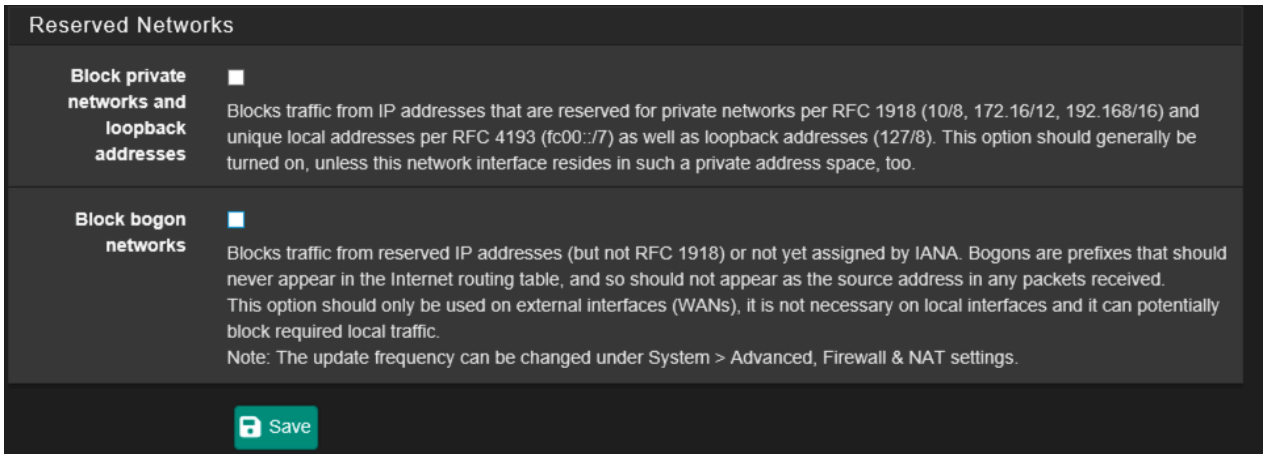
Preliminary note: The Client's pfSense will have a white background, while the **Nullsec** will have a dark background.

Now, let's prepare the two pfSense devices to handle the VPN connection. We are enabling default blocking of private networks and Bogon networks to give the green light to pfSense routers to respond to pings on the WAN.

Pfsense **Sun To Water**



Pfsense **Nullsec**

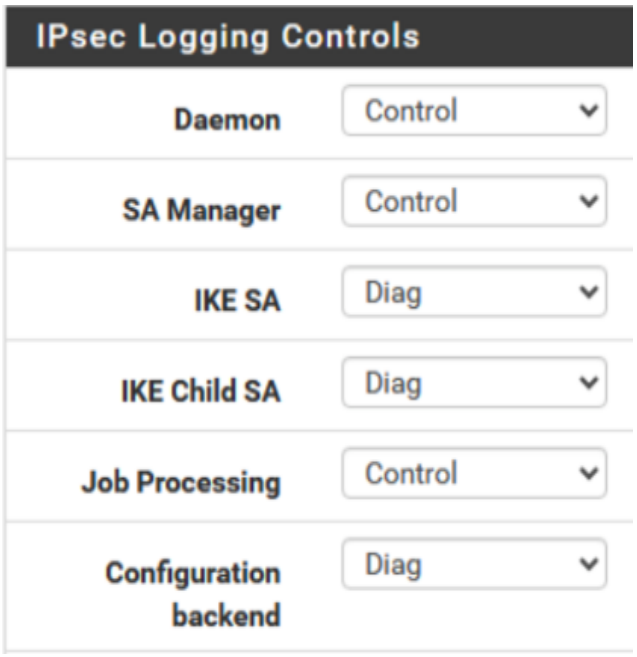


At this stage, we configured the tunnel negotiations on both pfSense devices following these steps:

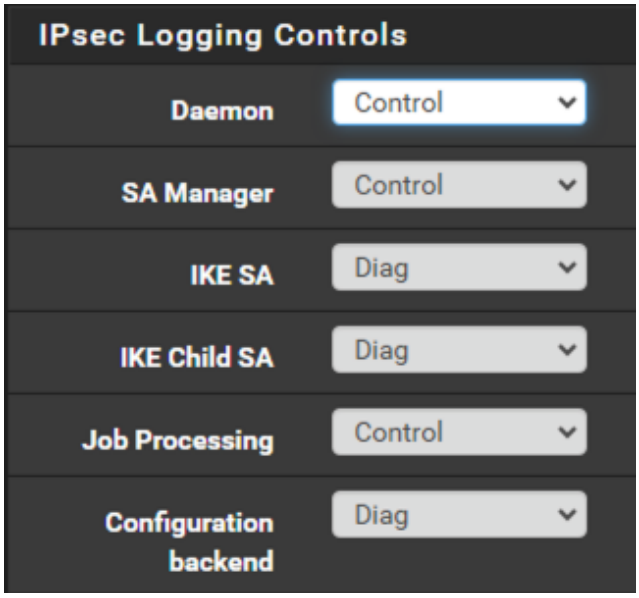
Navigate to VPN > IPsec and select the Advanced Settings tab.

Set IKE SA, IKE Child SA, and Configuration Backend to Diag.

PfSense **Sun To Water**



Pfsense **Nullsec**



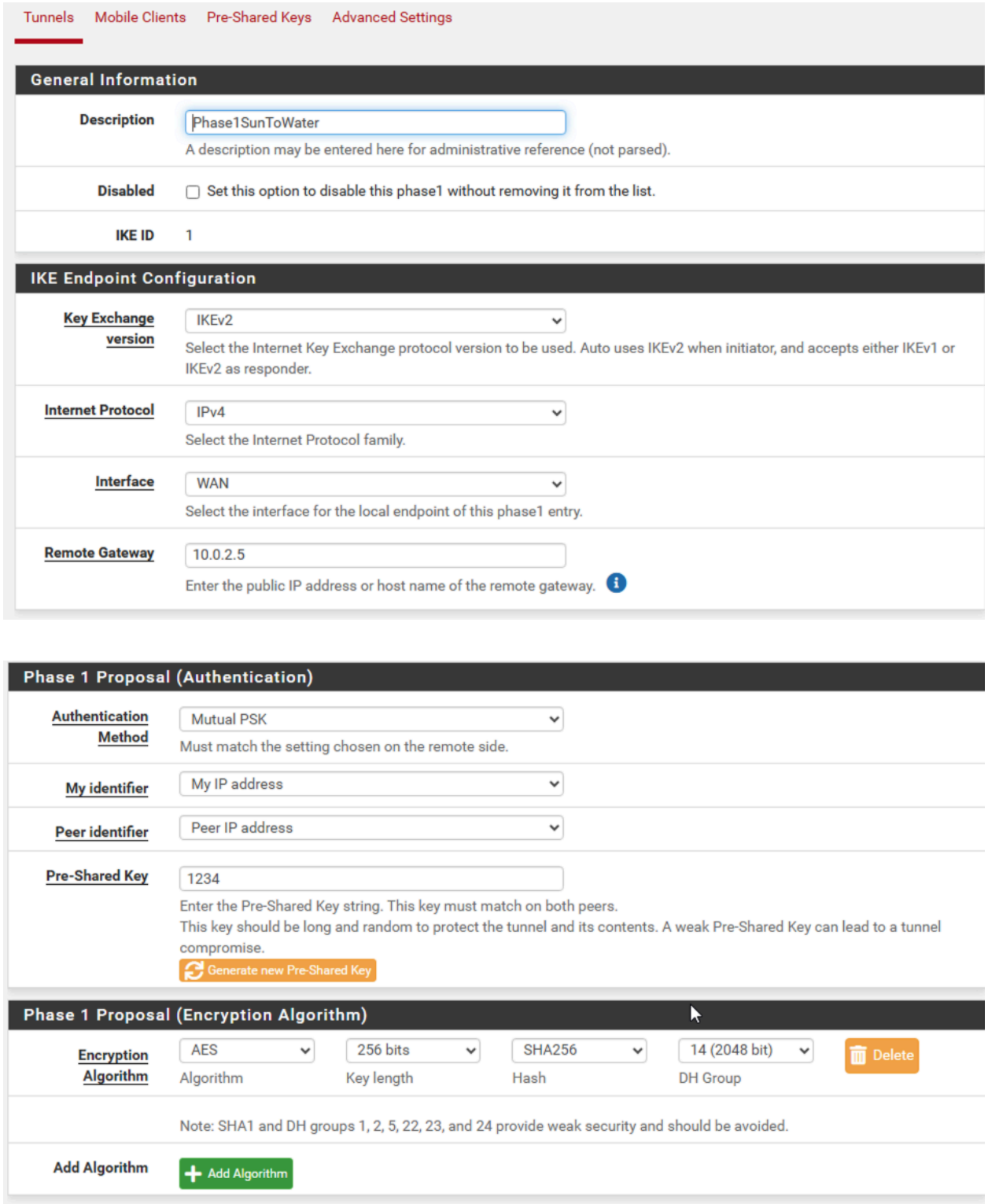
We configured the following specifications on each pfSense for Phase 1

Phase 1

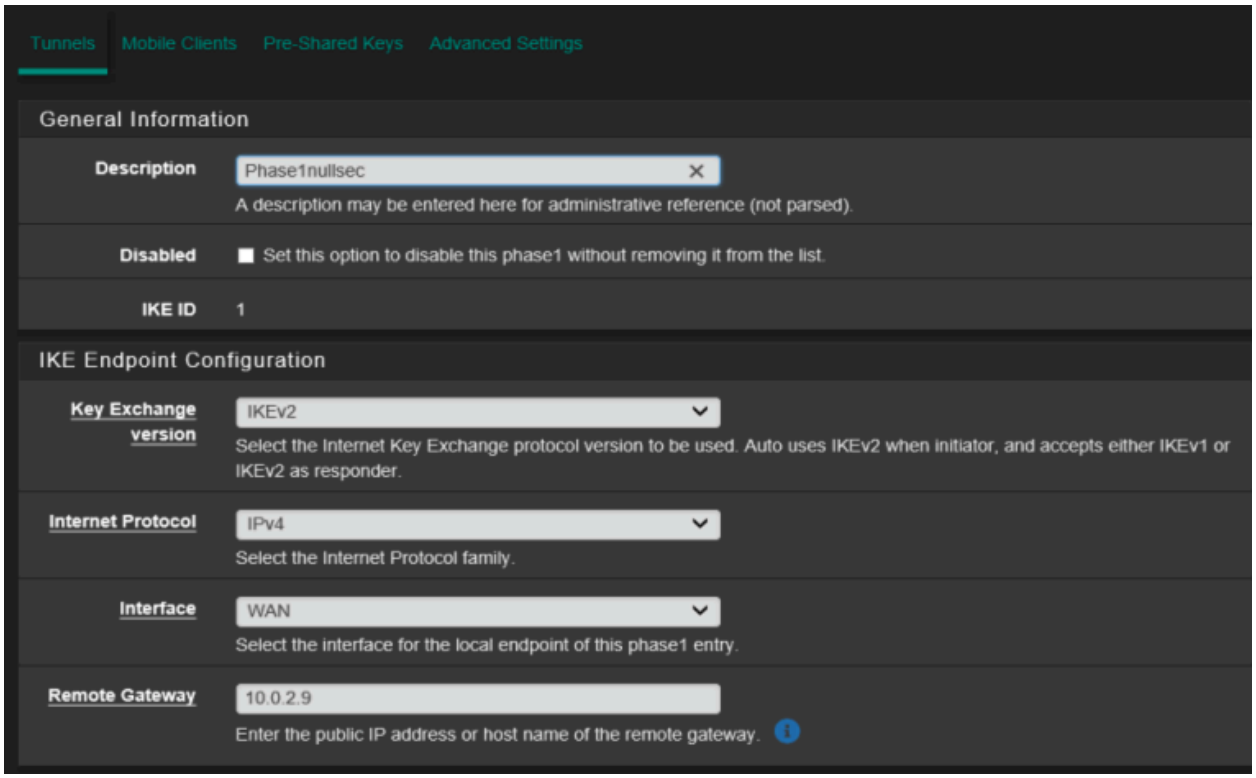
- Protocol: IKEv2
- Interface: WAN
- Remote Gateway: the WAN IPv4 address of the other router
- Method: Mutual PSK (Pre-Shared Key 1234)
- Encryption: AES 256-bit
- Lifetime: 28800 sec

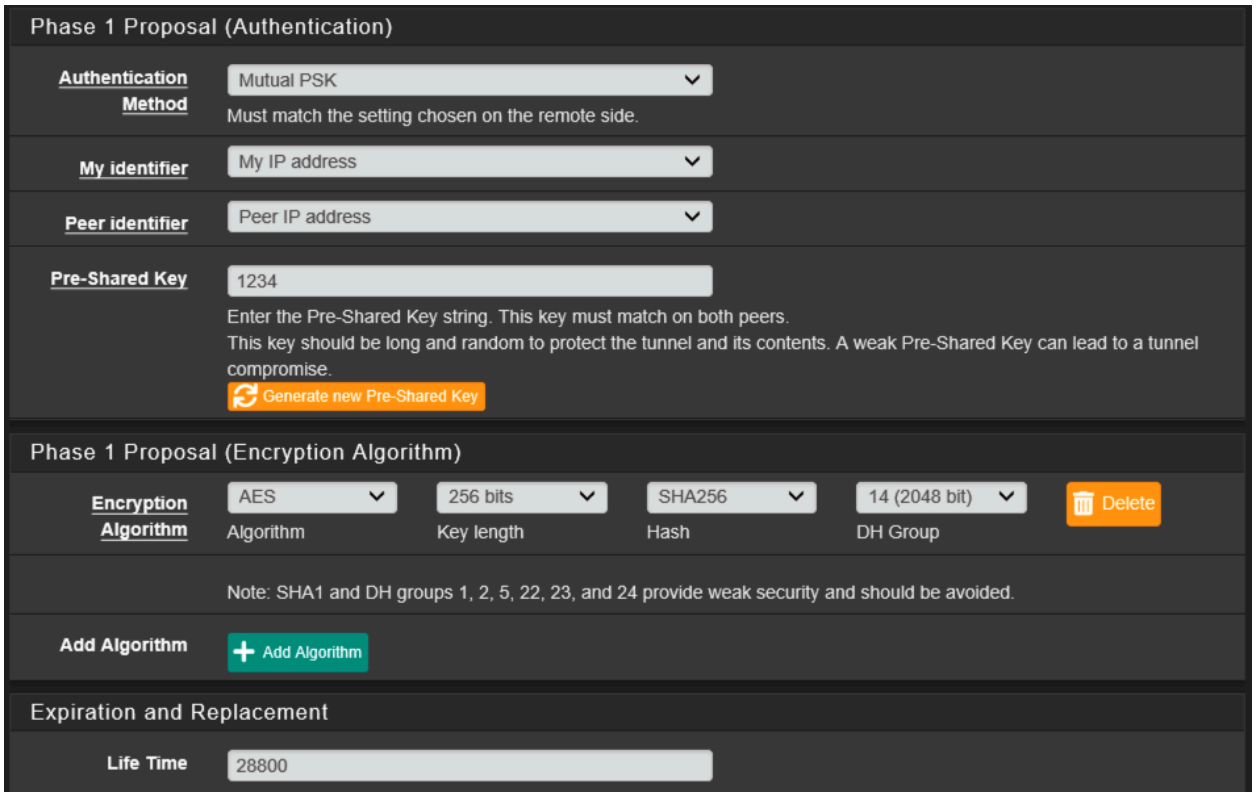
For better visualization, prints on the page below.

Pfsense **Sun to Water** - Phase 1



Pfsense **Nullsec** - Phase 1





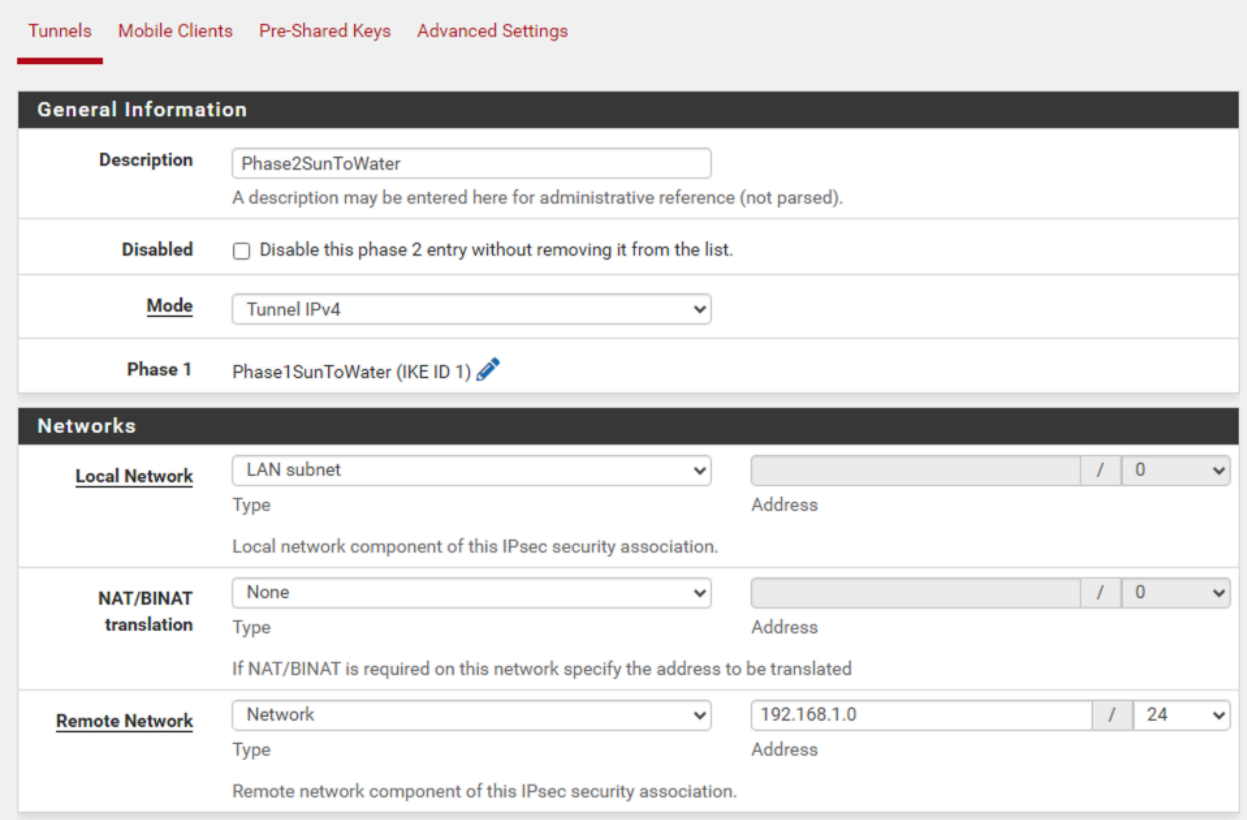
For Phase 2, we followed the following steps on both pfSense devices:

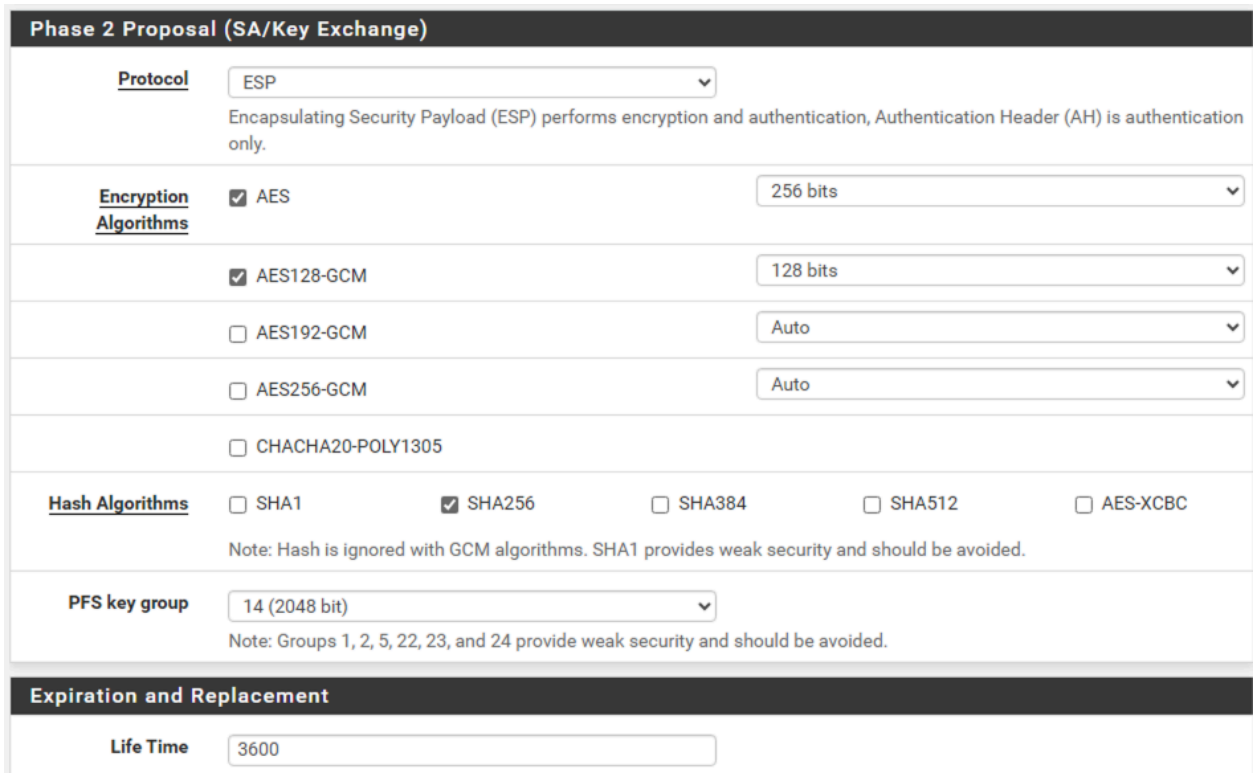
Phase 2

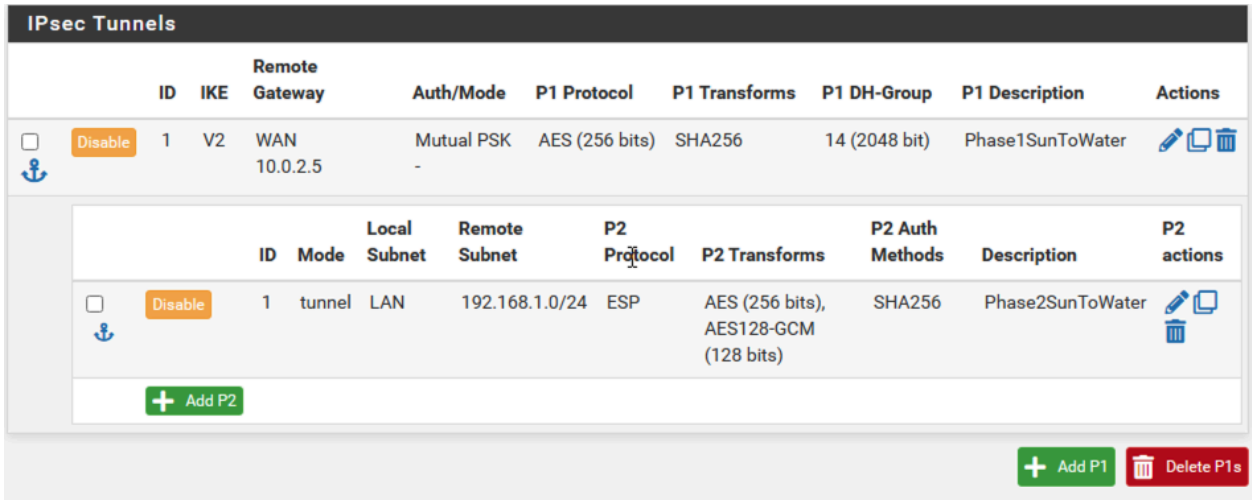
- Mode: Tunnel IPv4
- Local Network: LAN subnet
- Remote Network: the the *subnet address* of the LAN on the other router
  This is a subnet address address, e.g. 10.0.2.0/24, which is different from a gateway address such as 10.0.2.1
- Protocol: ESP
- Encryption: AES 256-bit
- PFS Key Group: 14 (2048 bit)
- Lifetime: 3600 sec

For better visualization, prints on the page below.

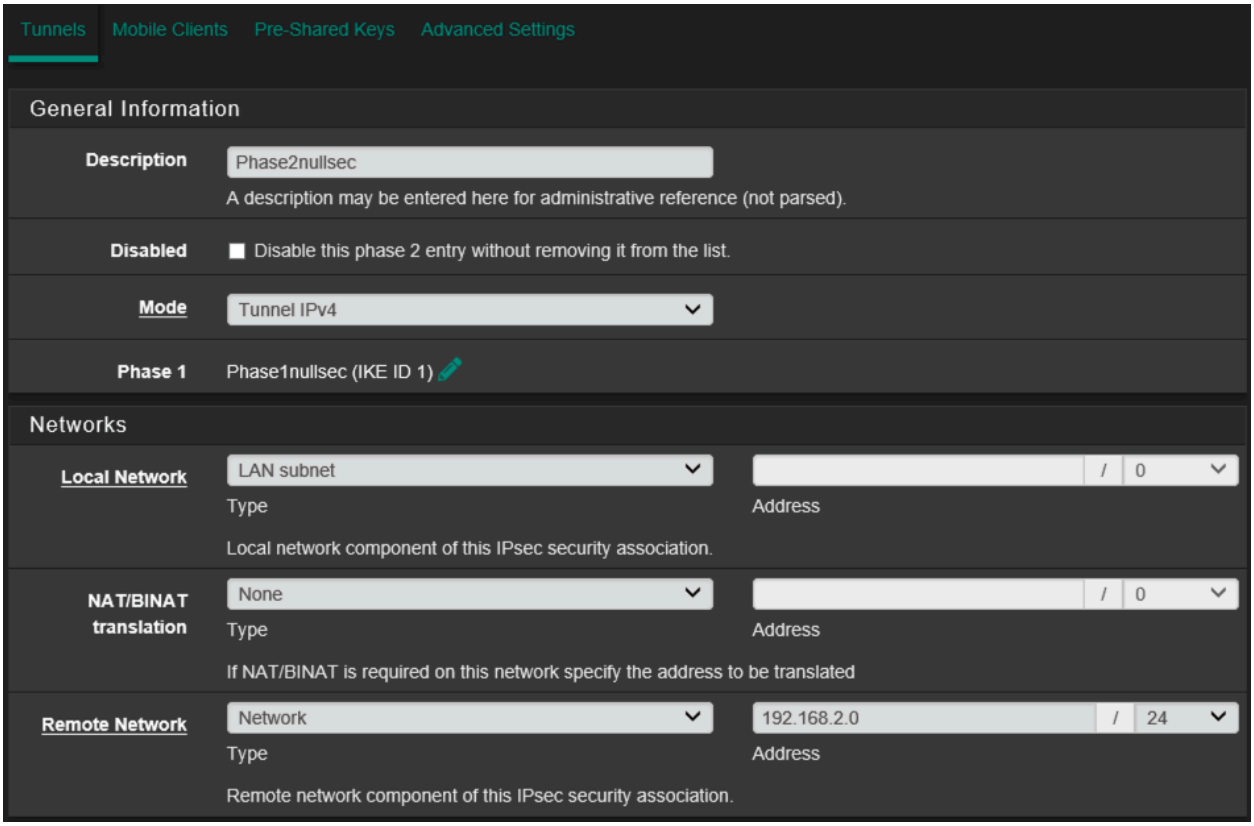
Pfsense **Sun to Water** - Phase 2

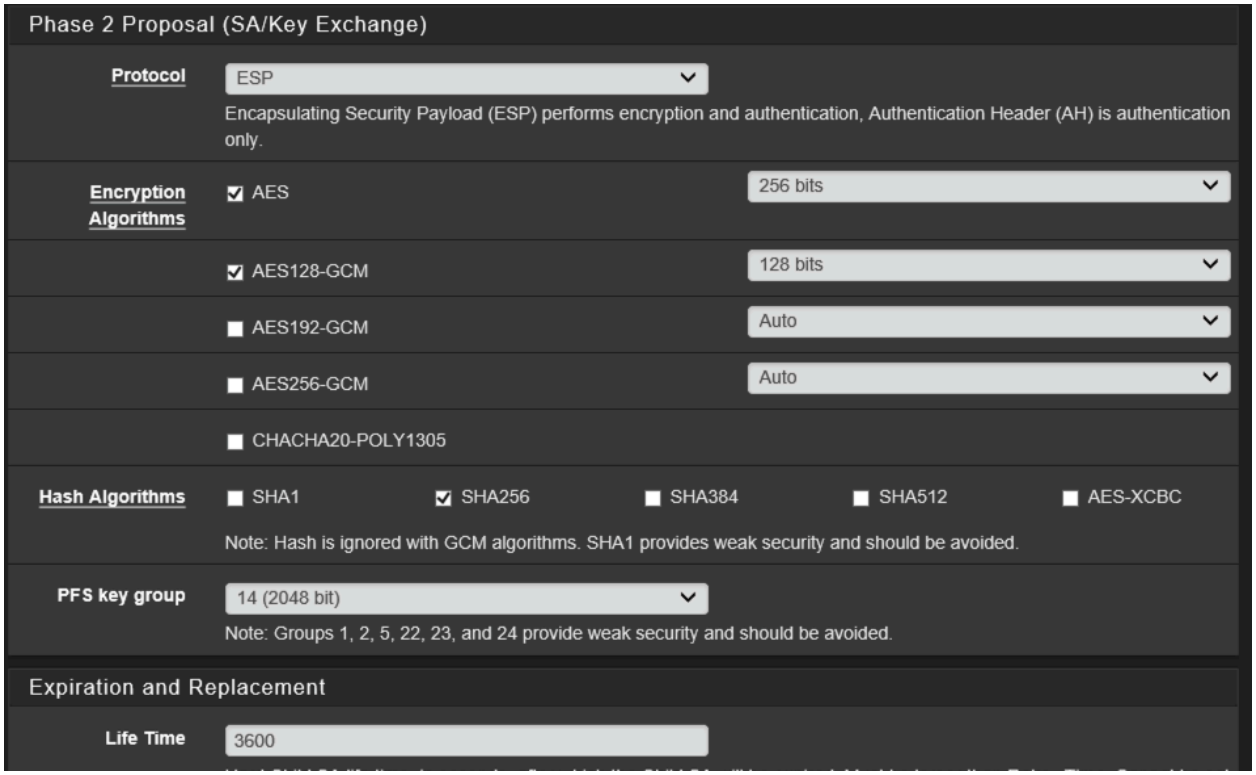


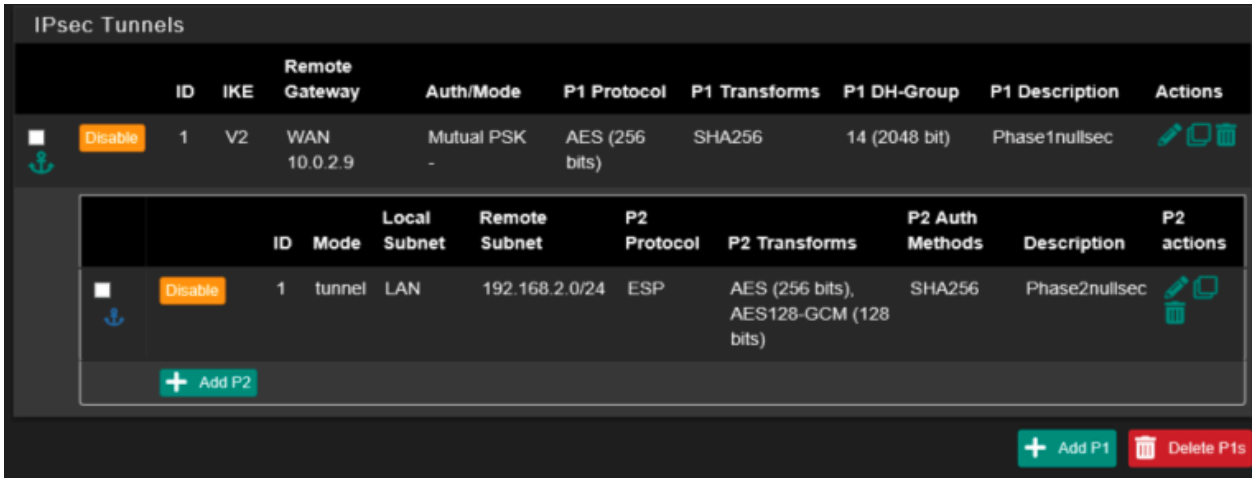



IPsec Tunnels
ID
IKE
Remote Gateway
Auth/Mode
P1 Protocol
P1 Transforms
P1 DH-Group
P1 Description
Actions
Disable
1
V2
WAN 10.0.2.5
Mutual PSK -
AES (256 bits)
SHA256
14 (2048 bit)
Phase1SunToWater
ID
Mode
Local Subnet
Remote Subnet
P2 Protocol
P2 Transforms
P2 Auth Methods
Description
P2 actions
Disable
1
tunnel
LAN
192.168.1.0/24
ESP
AES (256 bits), AES128-GCM (128 bits)
SHA256
Phase2SunToWater
Add P2
Add P1
Delete P1s

Pfsense **Nullsec** - Phase 2





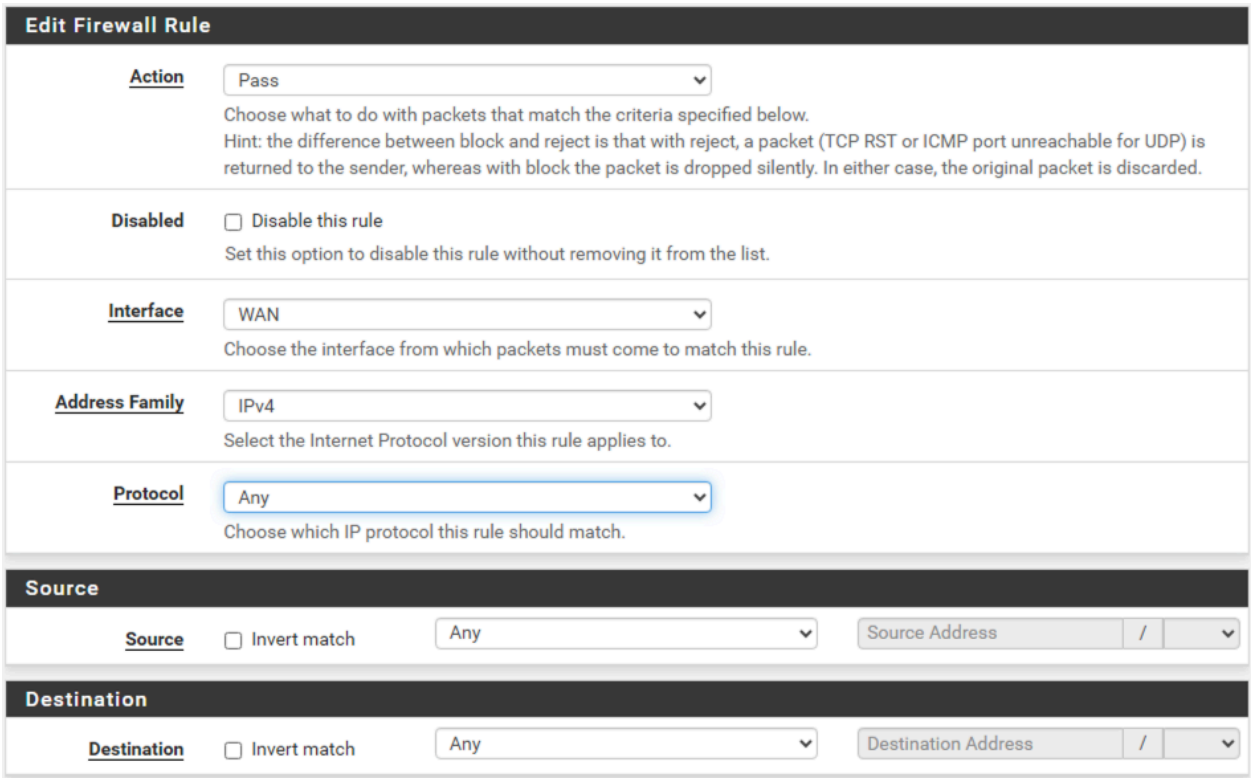

IPsec Tunnels
ID
IKE
Remote Gateway
Auth/Mode
P1 Protocol
P1 Transforms
P1 DH-Group
P1 Description
Actions
Disable
1
V2
WAN 10.0.2.9
Mutual PSK -
AES (256 bits)
SHA256
14 (2048 bit)
Phase1nullsec
ID
Mode
Local Subnet
Remote Subnet
P2 Protocol
P2 Transforms
P2 Auth Methods
Description
P2 actions
Disable
1
tunnel
LAN
192.168.2.0/24
ESP
AES (256 bits), AES128-GCM (128 bits)
SHA256
Phase2nullsec
Add P2
Add P1
Delete P1s

We created matching rules on WAN and IPsec interfaces.

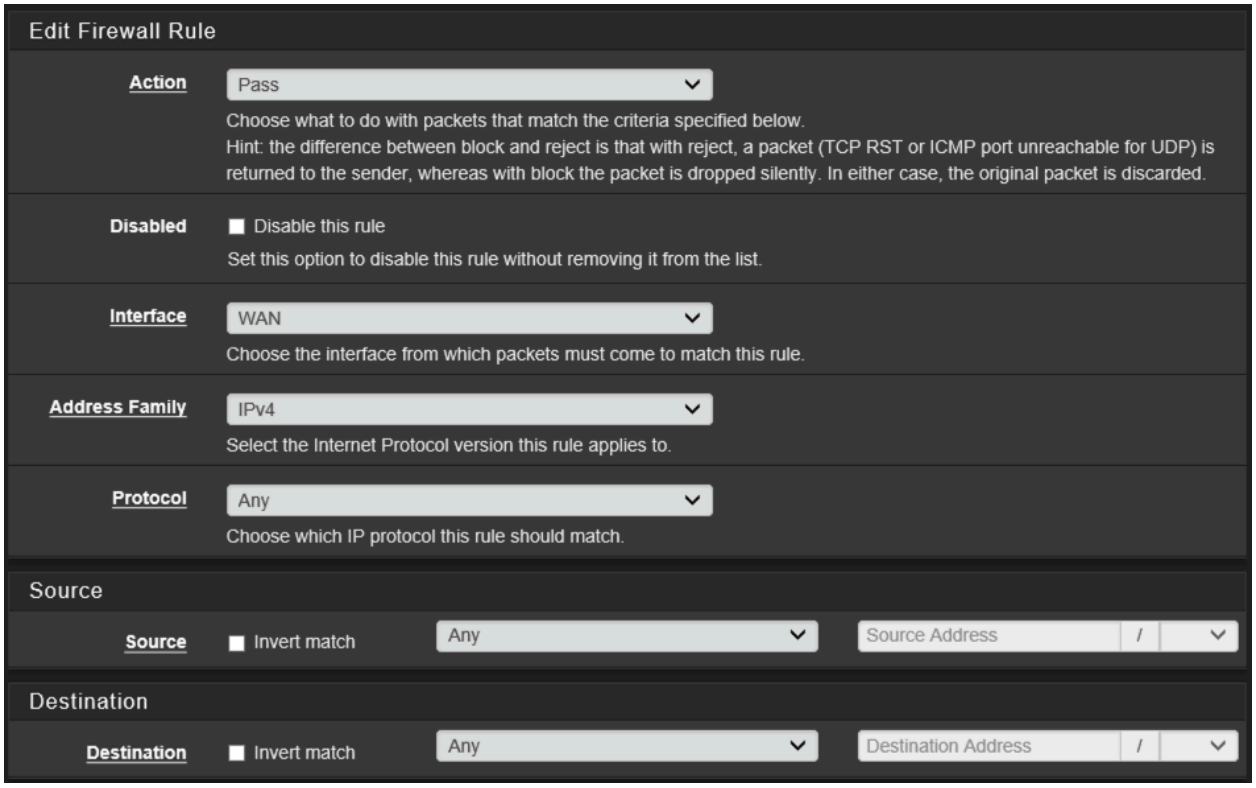
Firewall Rule specification:

- Action: Allow
- Protocol: Any
- Source: Any
- Destination: Any

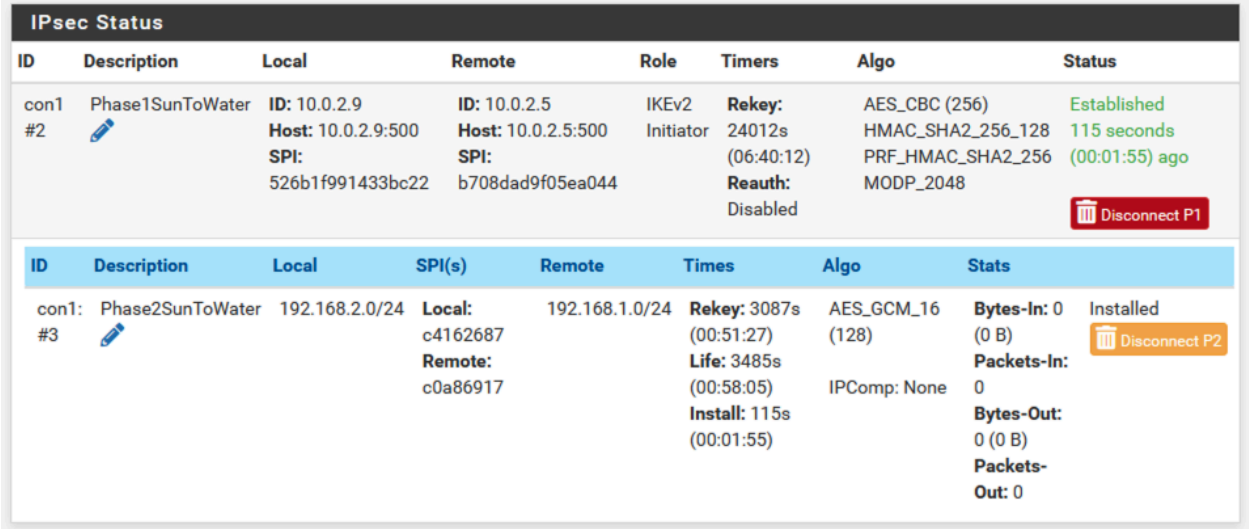
Pfsense **Sun to Water** - Firewall Rules WAN/IPsec



Pfsense **Nullsec** - Firewall Rules WAN/IPsec



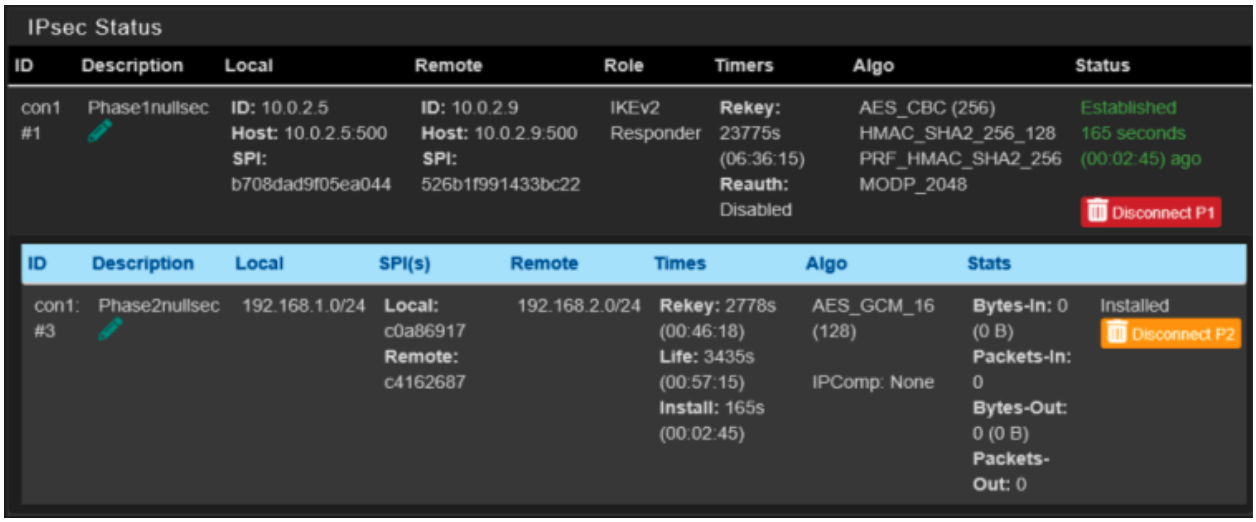
Attempt to verify connectivity



IPsec Status

ID	Description	Local	Remote	Role	Timers	Algo	Status
con1 #2	Phase1SunToWater	**ID:** 10.0.2.9 **Host:** 10.0.2.9:500 **SPI:** 526b1f991433bc22	**ID:** 10.0.2.5 **Host:** 10.0.2.5:500 **SPI:** b708dad9f05ea044	IKEv2 Initiator	**Rekey:** 24012s (06:40:12) **Reauth:** Disabled	AES_CBC (256) HMAC_SHA2_256_128 PRF_HMAC_SHA2_256 MODP_2048	Established 115 seconds (00:01:55) ago Disconnect P1

ID	Description	Local	SPI(s)	Remote	Times	Algo	Stats	
con1: #3	Phase2SunToWater	192.168.2.0/24	**Local:** c4162687 **Remote:** c0a86917	192.168.1.0/24	**Rekey:** 3087s (00:51:27) **Life:** 3485s (00:58:05) **Install:** 115s (00:01:55)	AES_GCM_16 (128) IPComp: None	**Bytes-In:** 0 (0 B) **Packets-In:** 0 **Bytes-Out:** 0 (0 B) **Packets-Out:** 0	Installed Disconnect P2



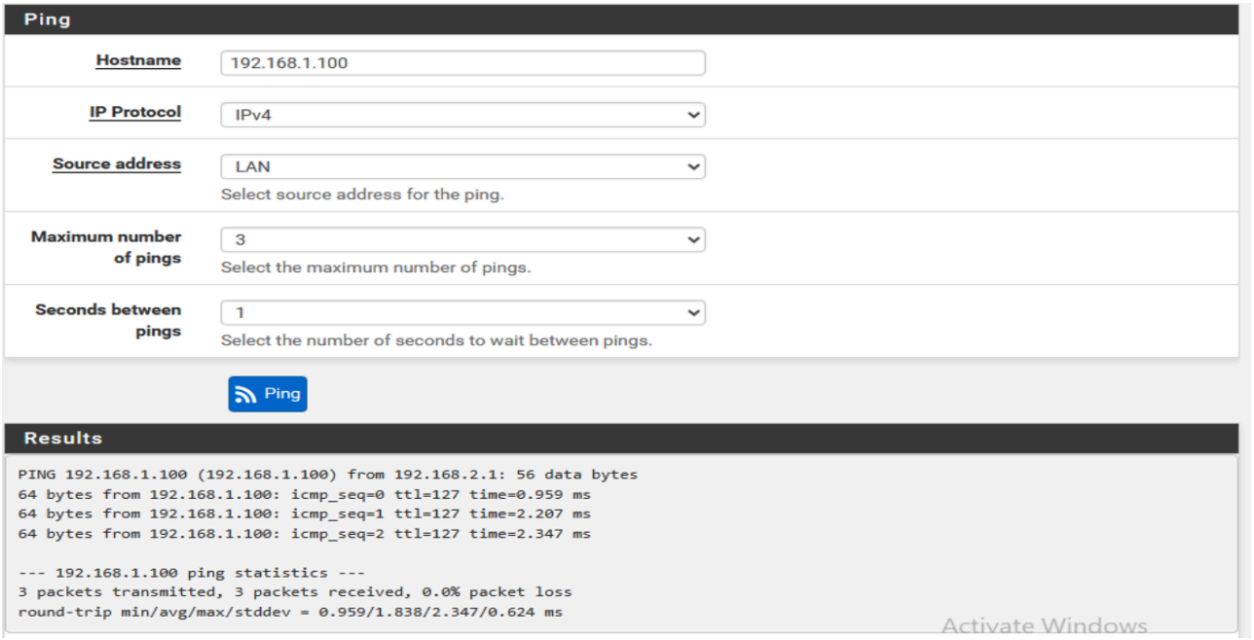
IPsec Status

ID	Description	Local	Remote	Role	Timers	Algo	Status
con1 #1	Phase1nullsec	**ID:** 10.0.2.5 **Host:** 10.0.2.5:500 **SPI:** b708dad9f05ea044	**ID:** 10.0.2.9 **Host:** 10.0.2.9:500 **SPI:** 526b1f991433bc22	IKEv2 Responder	**Rekey:** 23775s (06:36:15) **Reauth:** Disabled	AES_CBC (256) HMAC_SHA2_256_128 PRF_HMAC_SHA2_256 MODP_2048	Established 165 seconds (00:02:45) ago Disconnect P1

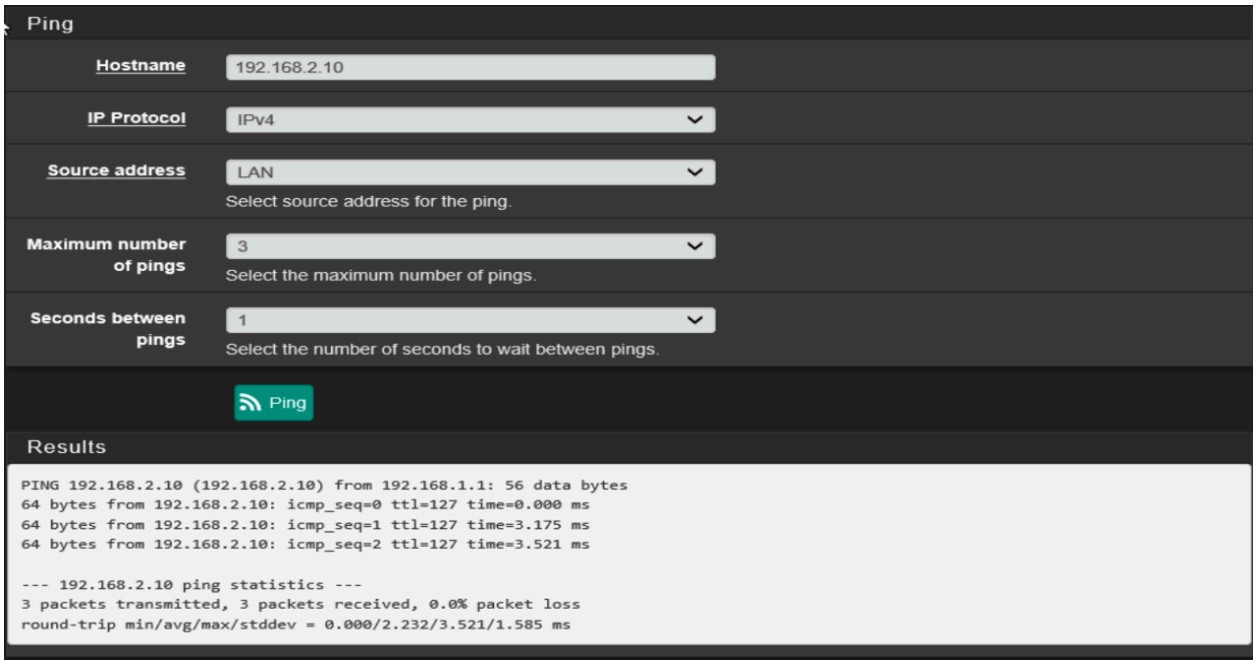
ID	Description	Local	SPI(s)	Remote	Times	Algo	Stats	
con1: #3	Phase2nullsec	192.168.1.0/24	**Local:** c0a86917 **Remote:** c4162687	192.168.2.0/24	**Rekey:** 2778s (00:46:18) **Life:** 3435s (00:57:15) **Install:** 165s (00:02:45)	AES_GCM_16 (128) IPComp: None	**Bytes-In:** 0 (0 B) **Packets-In:** 0 **Bytes-Out:** 0 (0 B) **Packets-Out:** 0	Installed Disconnect P2

Success in establishing the VPN tunnel connection. We will now attempt to ping between the machines.

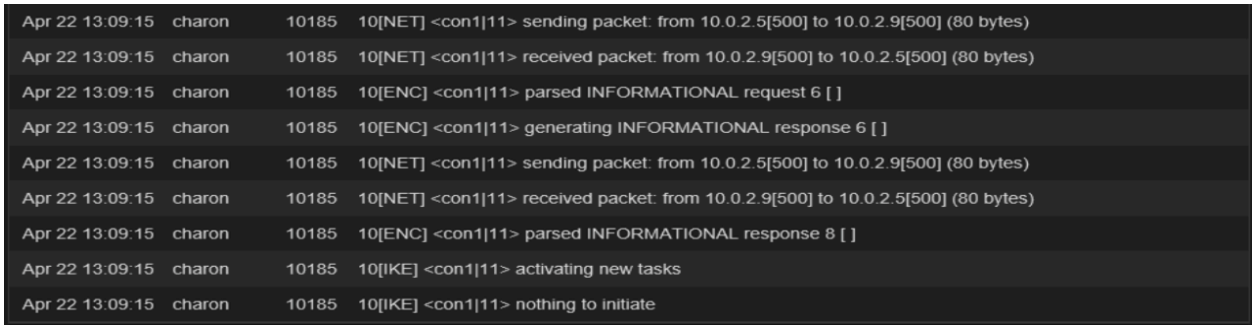
Sun To Water machine to **Nullsec** Server - Testing connectivity (ping)



Nullsec Server to **Sun To Water** Machine - Testing connectivity (ping)



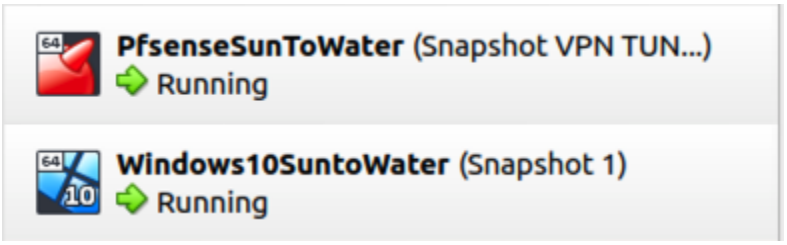
We've confirmed successful VPN tunnel connectivity by analyzing pfSense status indicators and system logs.



Apr 22 13:09:15	charon	10185	10[NET] <con1|11> sending packet: from 10.0.2.5[500] to 10.0.2.9[500] (80 bytes)
Apr 22 13:09:15	charon	10185	10[NET] <con1|11> received packet: from 10.0.2.9[500] to 10.0.2.5[500] (80 bytes)
Apr 22 13:09:15	charon	10185	10[ENC] <con1|11> parsed INFORMATIONAL request 6 [ ]
Apr 22 13:09:15	charon	10185	10[ENC] <con1|11> generating INFORMATIONAL response 6 [ ]
Apr 22 13:09:15	charon	10185	10[NET] <con1|11> sending packet: from 10.0.2.5[500] to 10.0.2.9[500] (80 bytes)
Apr 22 13:09:15	charon	10185	10[NET] <con1|11> received packet: from 10.0.2.9[500] to 10.0.2.5[500] (80 bytes)
Apr 22 13:09:15	charon	10185	10[ENC] <con1|11> parsed INFORMATIONAL response 8 [ ]
Apr 22 13:09:15	charon	10185	10[IKE] <con1|11> activating new tasks
Apr 22 13:09:15	charon	10185	10[IKE] <con1|11> nothing to initiate

Captive Portal configuration

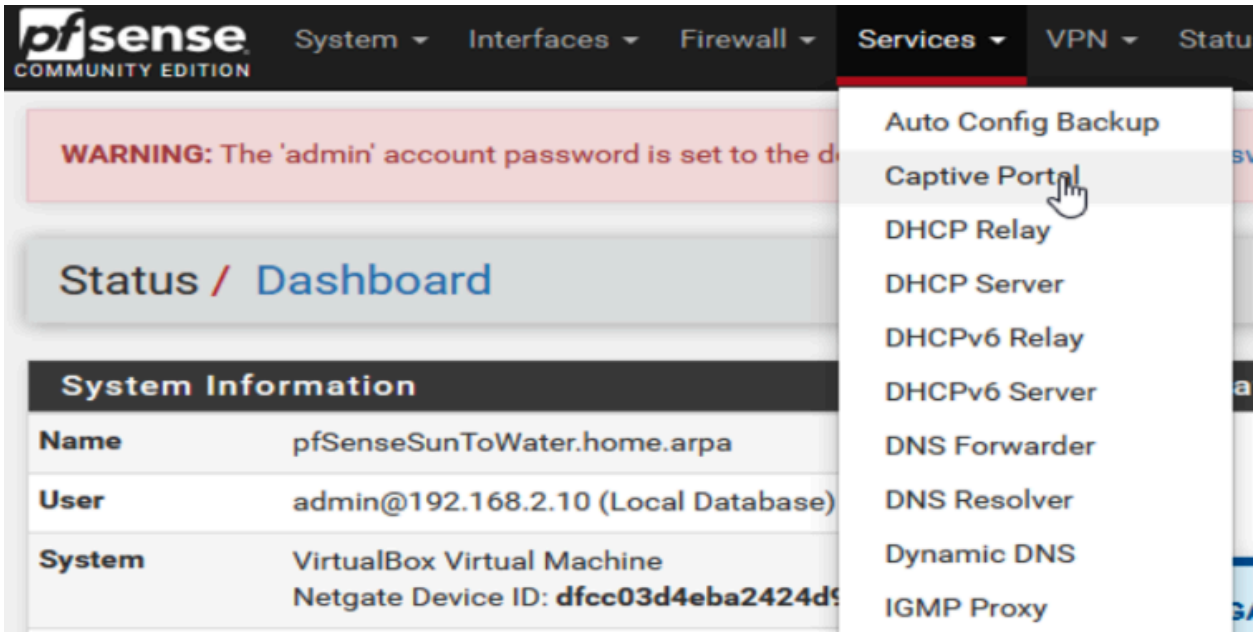
After successfully creating the VPN tunnel, we will use the client's machine to create access to the network with an AAA security configuration level (Authentication, Authorization and Accounting), for this we will implement a RADIUS system that creates a captive portal for network users and authenticates them using AD credentials.



The Snapshots you can see in the image were created after the VPN Tunnel was completed for security reasons so that if something goes wrong during the RADIUS implementation, we can go back.

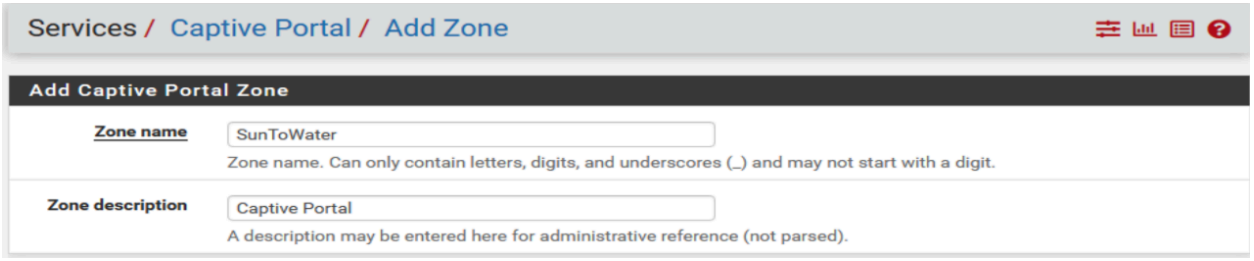
Configuration chosen

It will condition navigation and traffic transmitted on the user's local network, until the user authenticates the portal's login page.

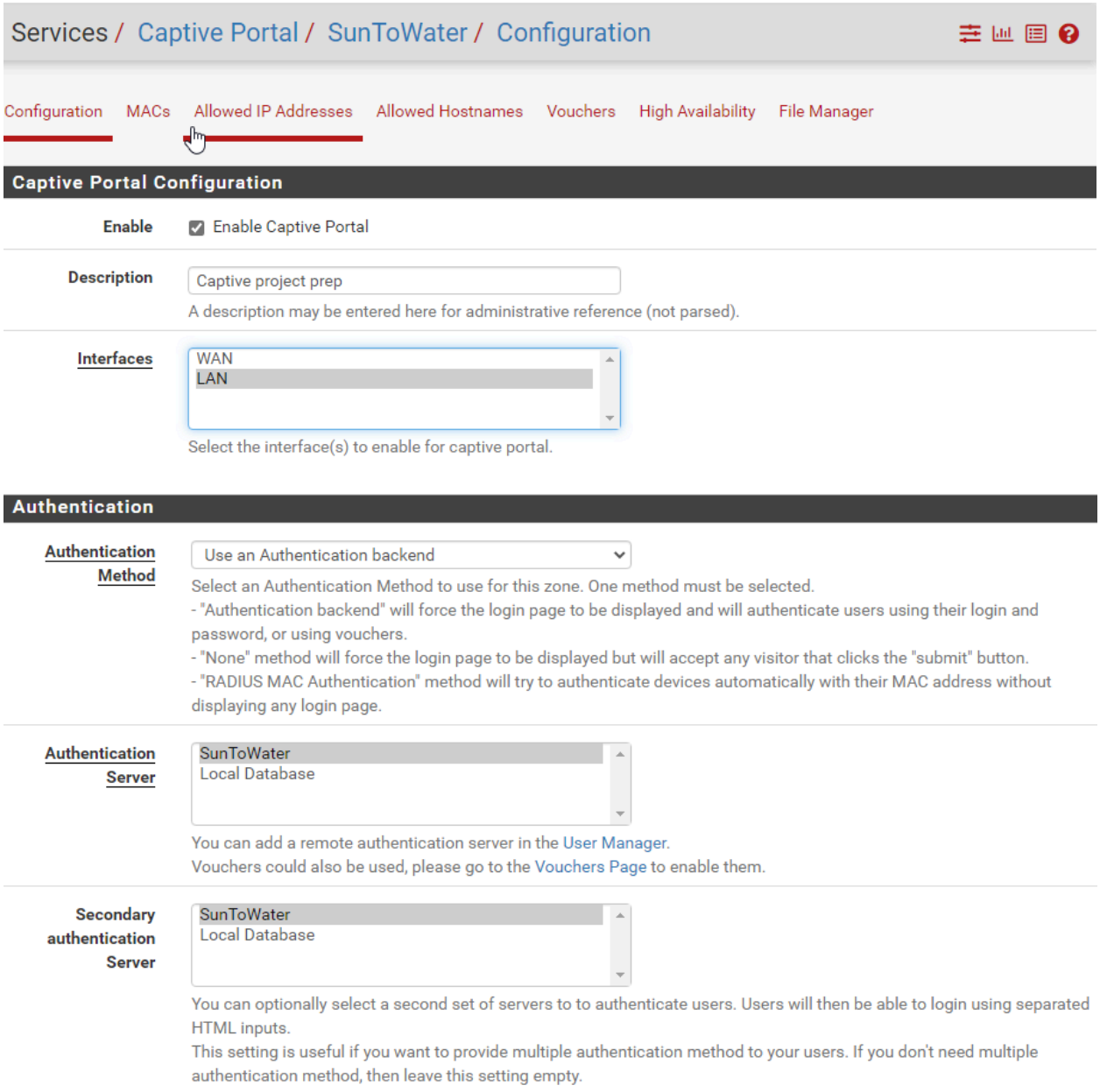


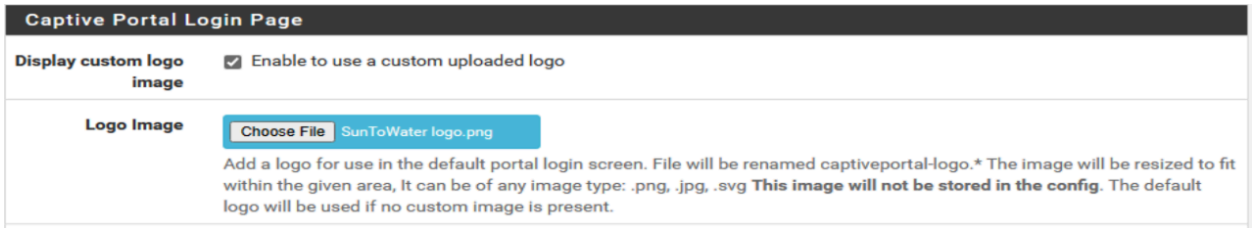
Defining zone name.

This feature will allow us to divide the network used into different areas with different access levels.



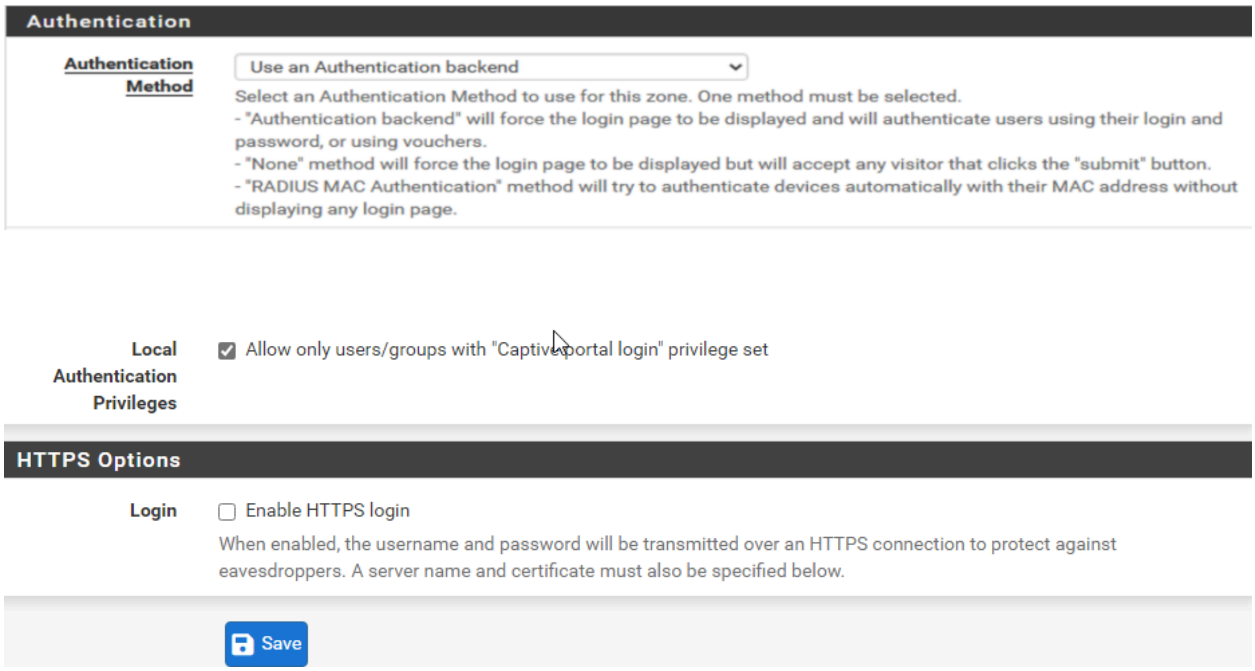
Configuring Captive Portal to do Authentication Server from SunToWater Data





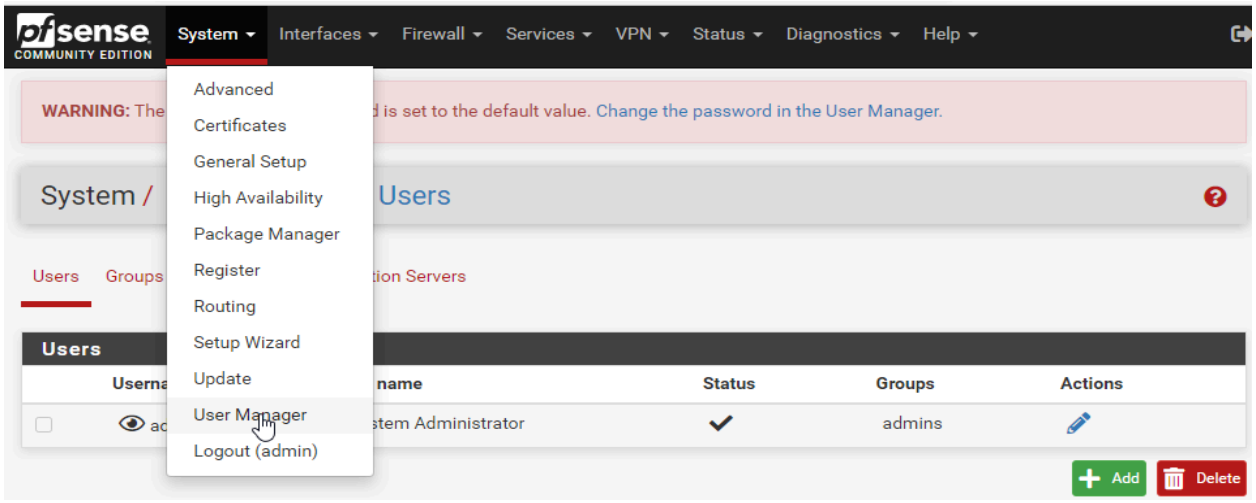
Authentication to anyone who want to access to network

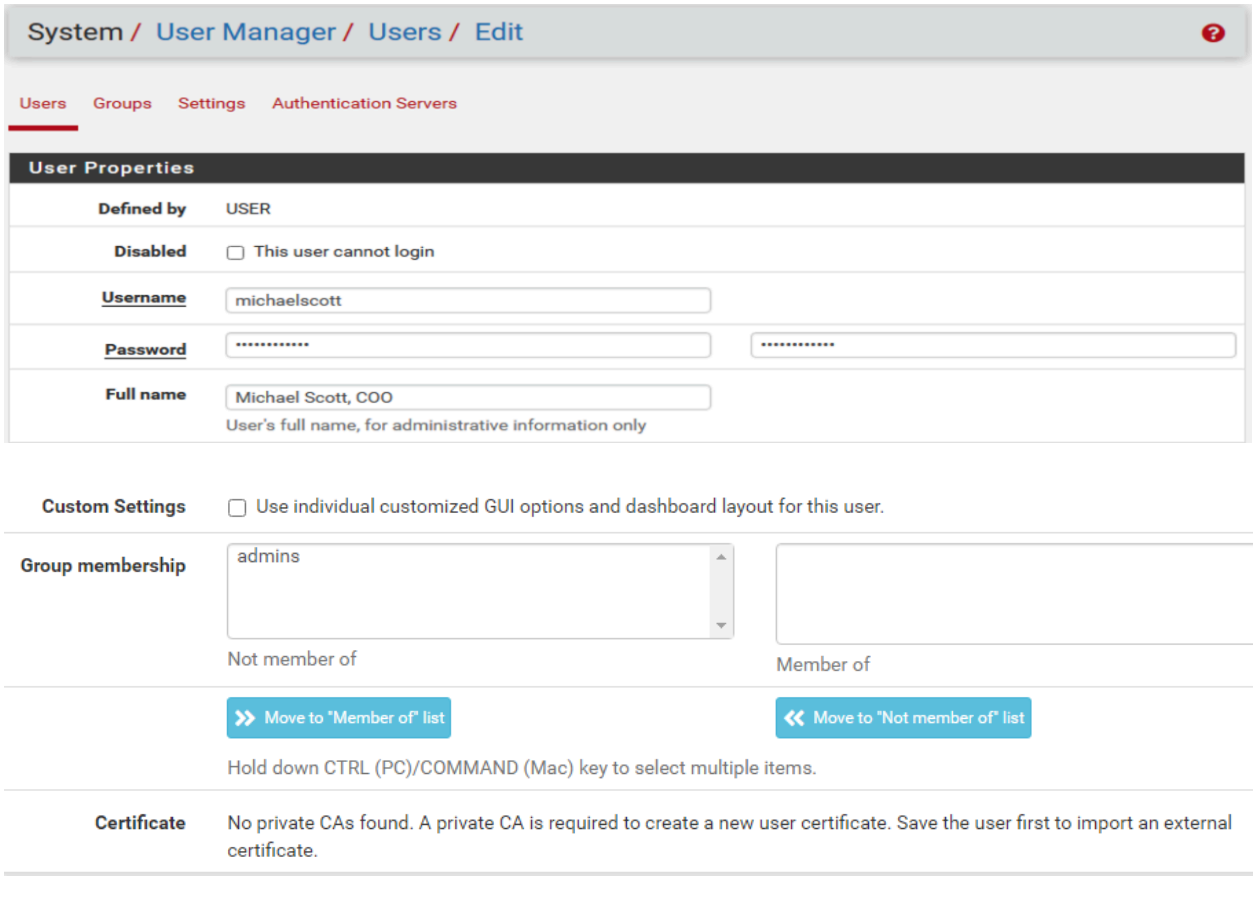
This type of authentication will force users to use a login and a password.

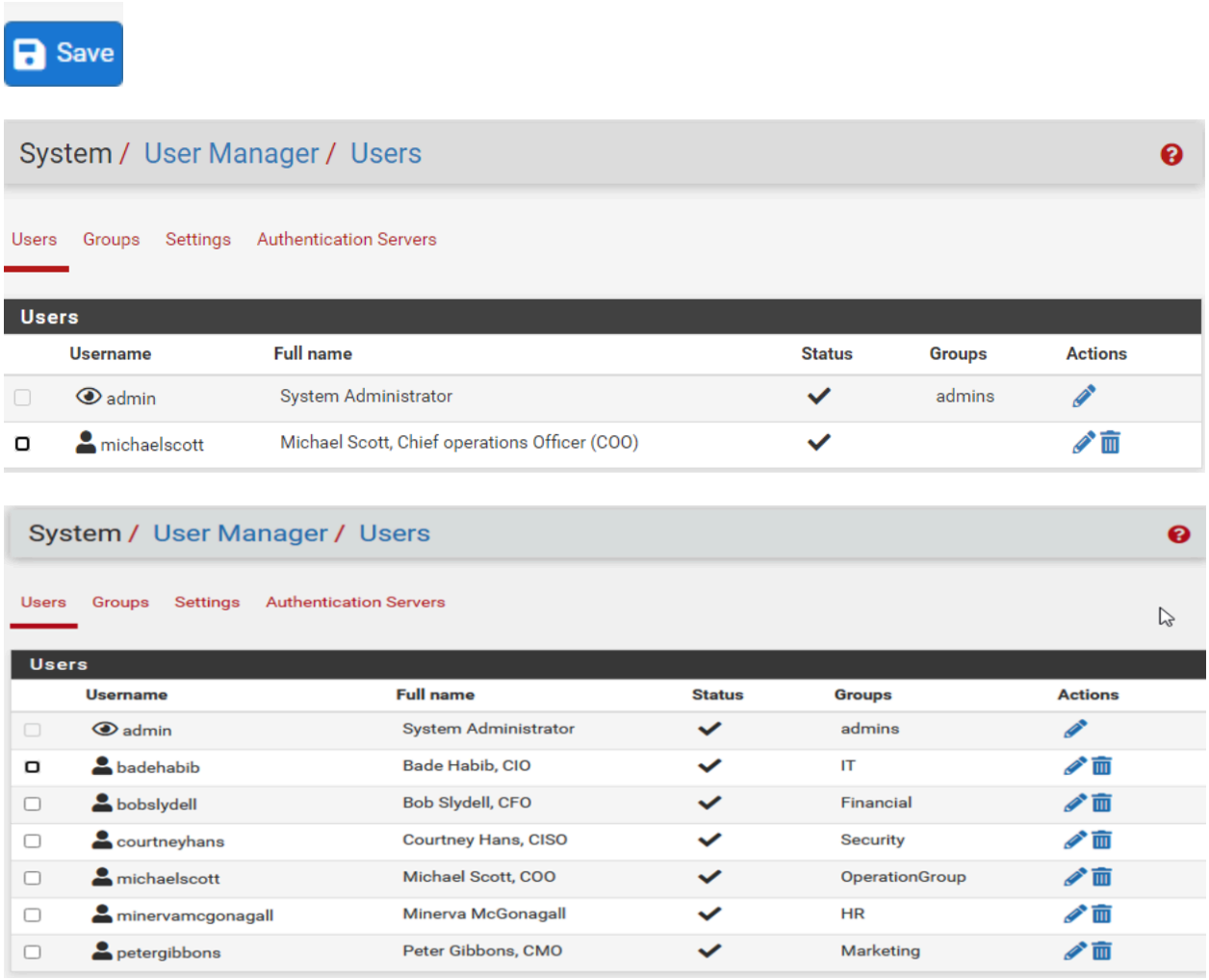


Users add, that should be access.

It is necessary to create users so that we can later define who can authenticate and have authorization to access the network and with what type of permissions.
password test: *Benfica2024!*

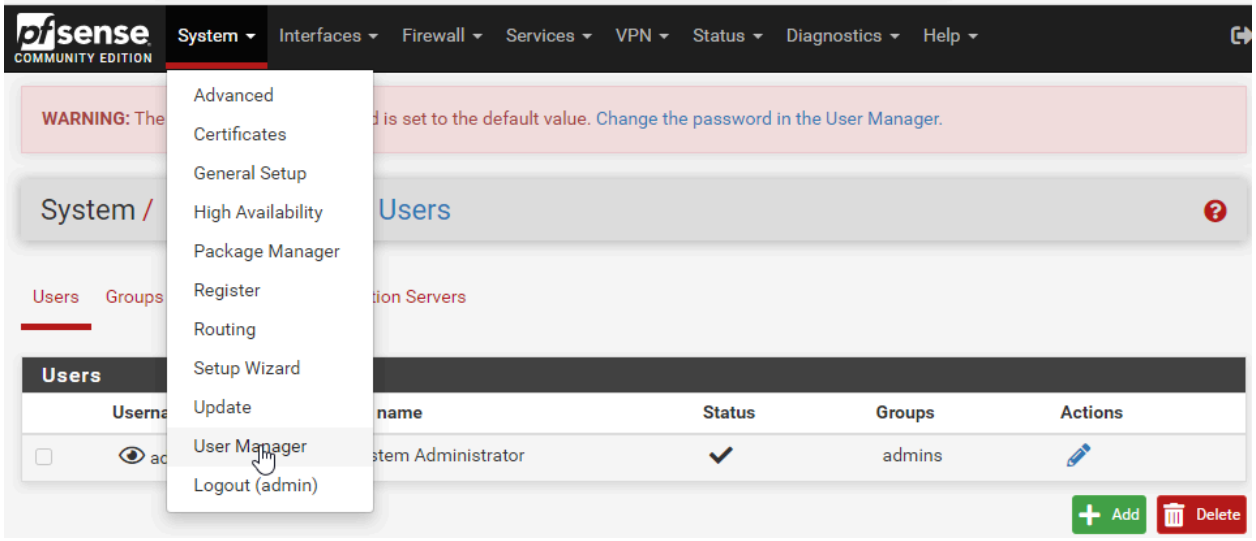


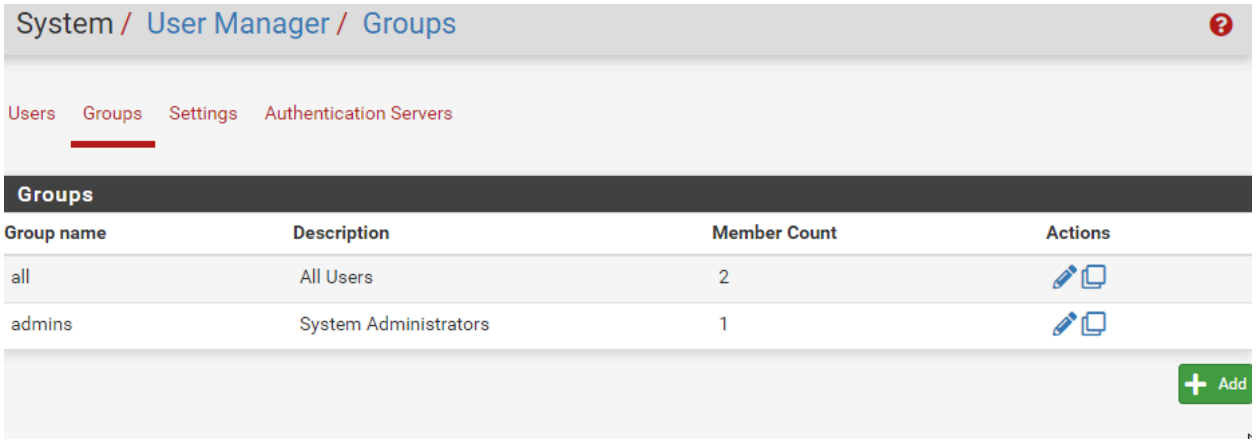


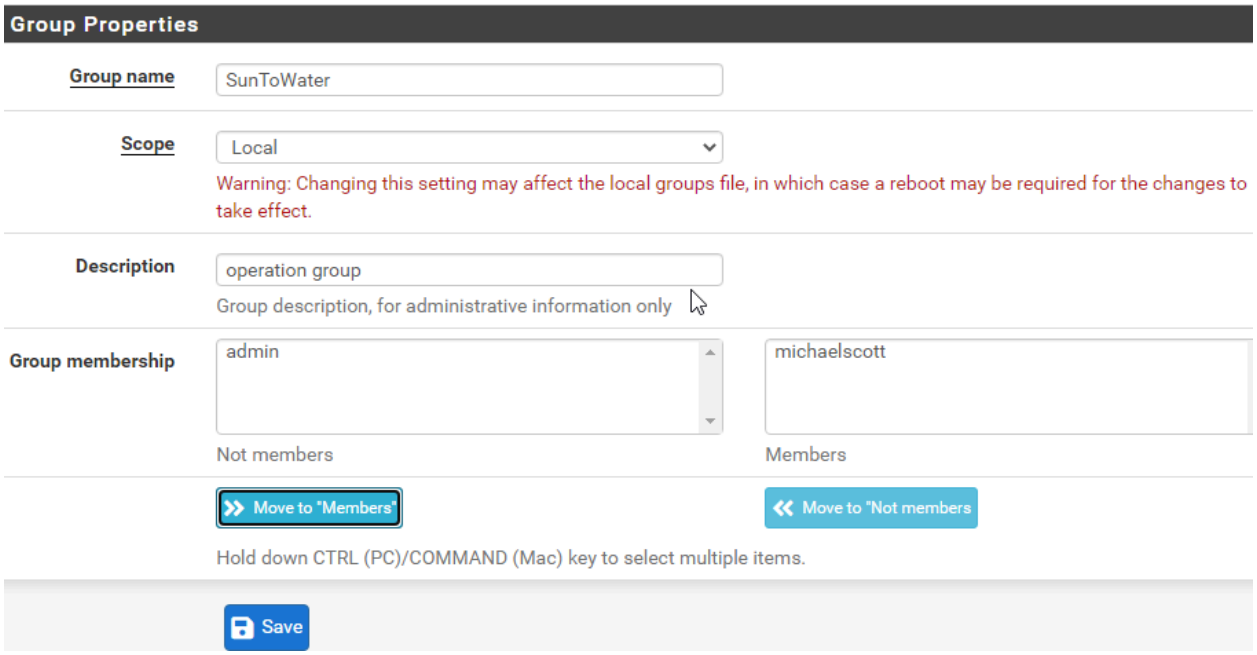


Creating a group of user’s

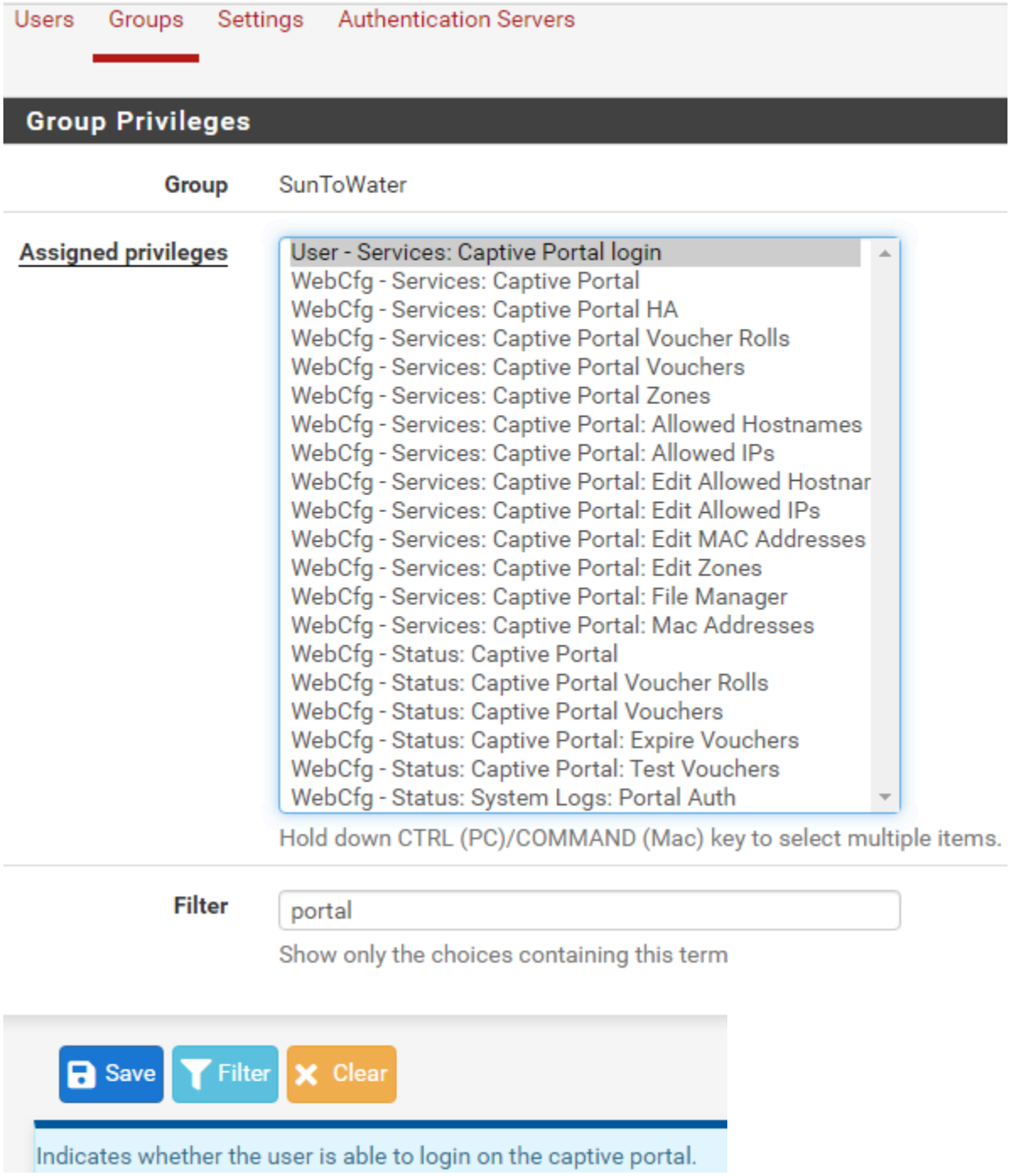
Creating groups allows you to define the structure and framework of the company's various departments to define the corresponding users.

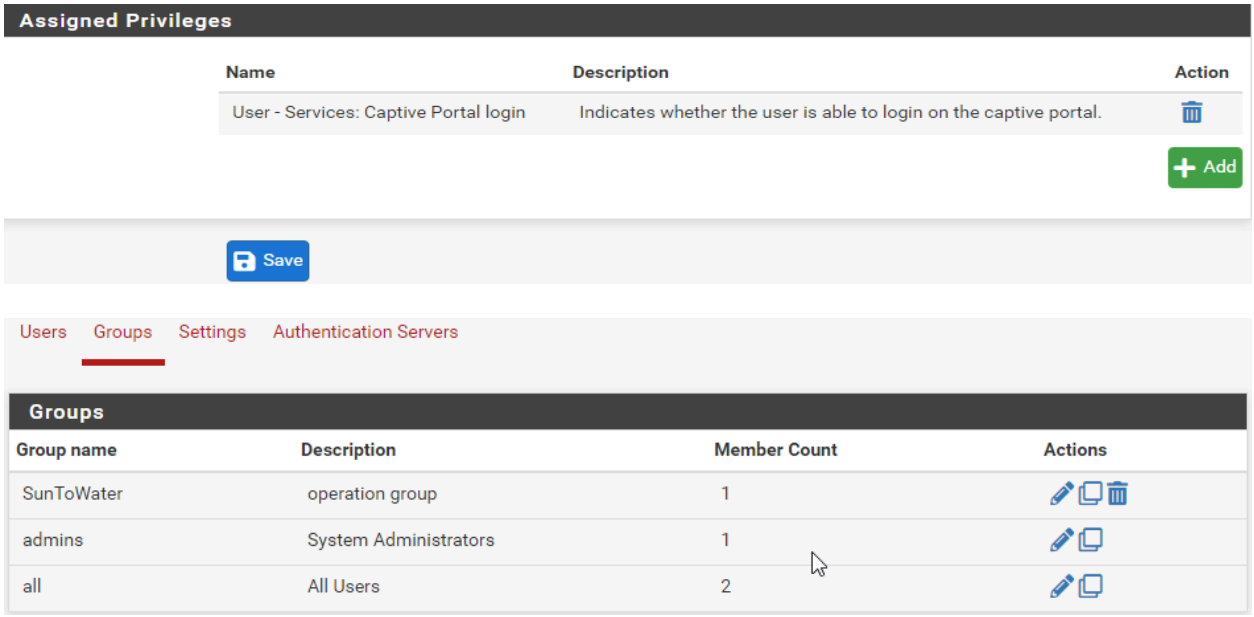

pfsense
COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
Advanced
Certificates
General Setup
High Availability
Package Manager
Register
Routing
Setup Wizard
Update
User Manager
Logout (admin)
WARNING: The
is set to the default value. Change the password in the User Manager.
System /
Users
Users
Groups
Servers
Users
Username
name
Status
Groups
Actions
admin
System Administrator
admins
Add
Delete


System / User Manager / Groups
Users
Groups
Settings
Authentication Servers
Groups
Group name
Description
Member Count
Actions
all
All Users
2
admins
System Administrators
1
Add

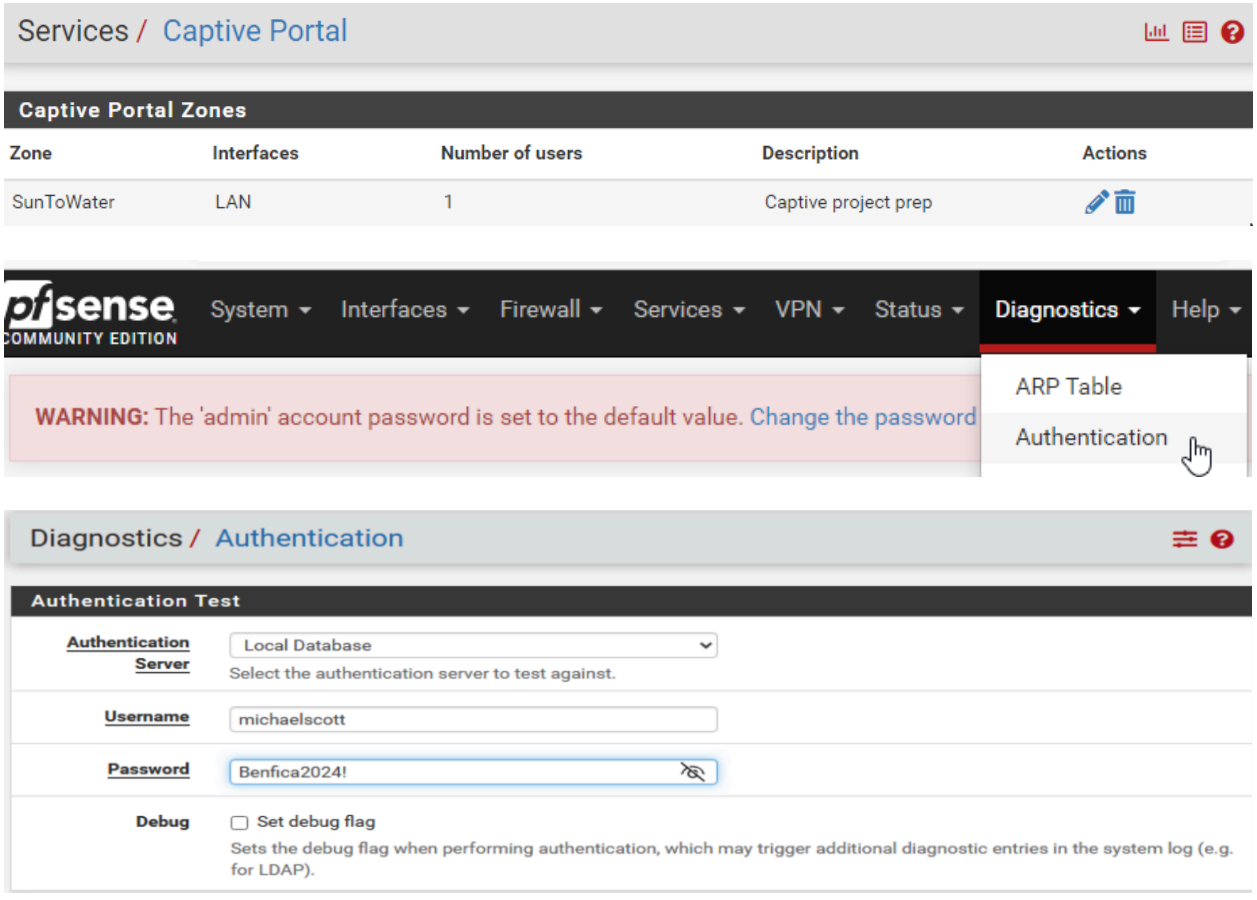

Group Properties
Group name
SunToWater
Scope
Local
Warning: Changing this setting may affect the local groups file, in which case a reboot may be required for the changes to take effect.
Description
operation group
Group description, for administrative information only
Group membership
admin
Not members
michaelscott
Members
Move to "Members"
Move to "Not members"
Hold down CTRL (PC)/COMMAND (Mac) key to select multiple items.
Save

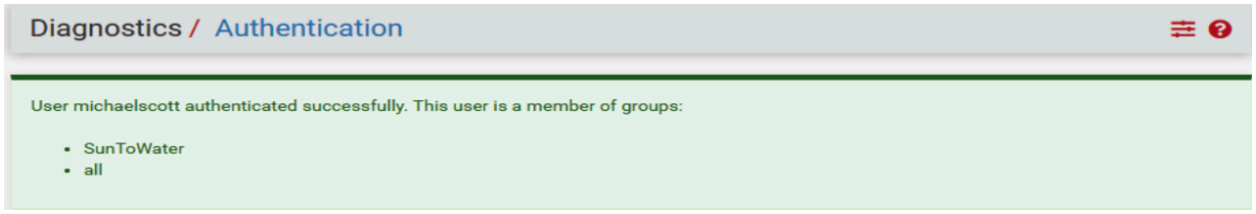
In this field when the user tries to access the internet for the first time after connecting to the network, he is automatically redirected to the captive portal login page.



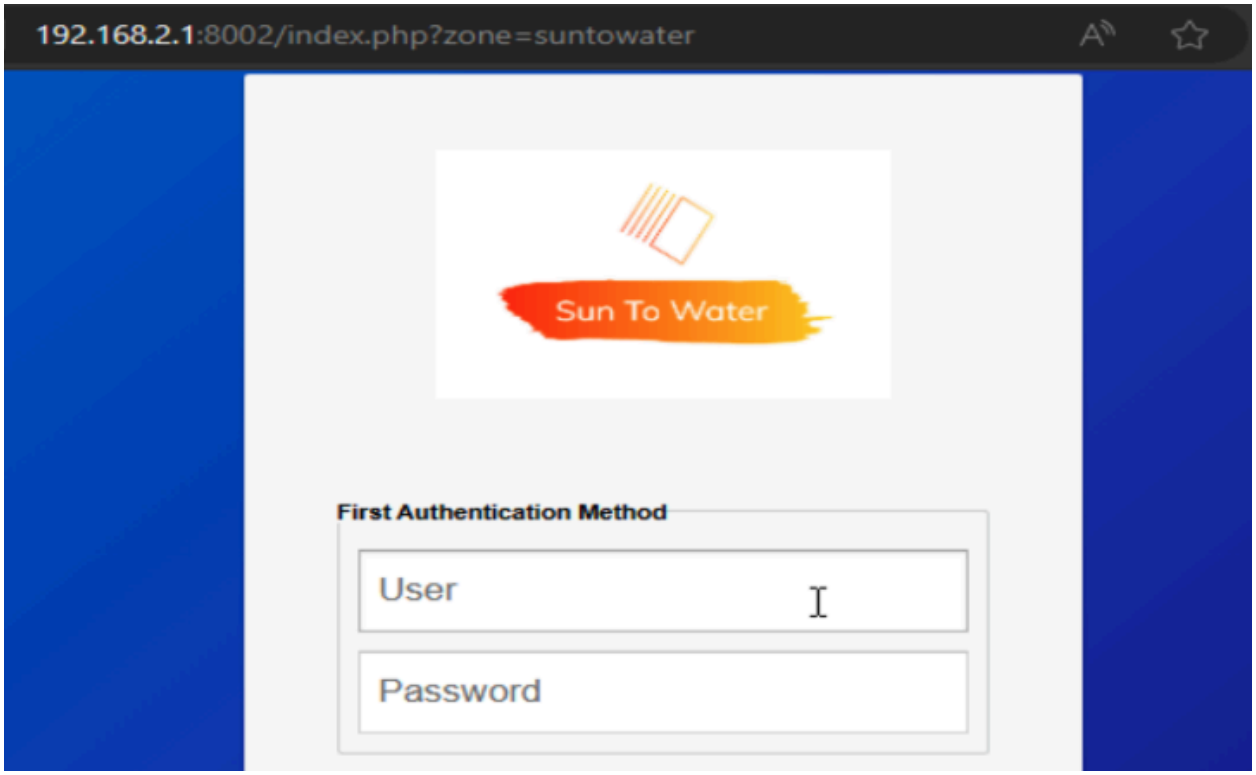


Verification of authentication.





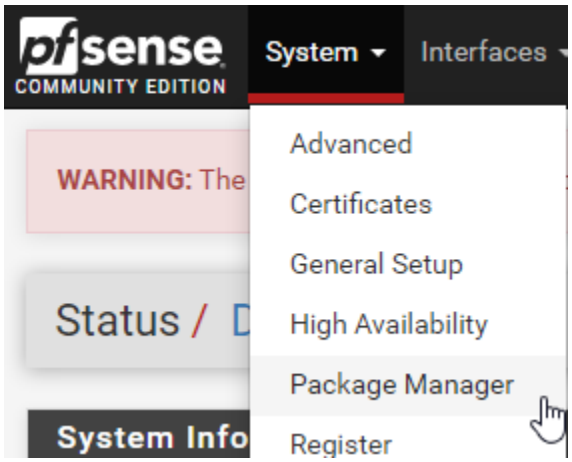
Requesting username and password access to authenticate the login page of the portal you intend to access.

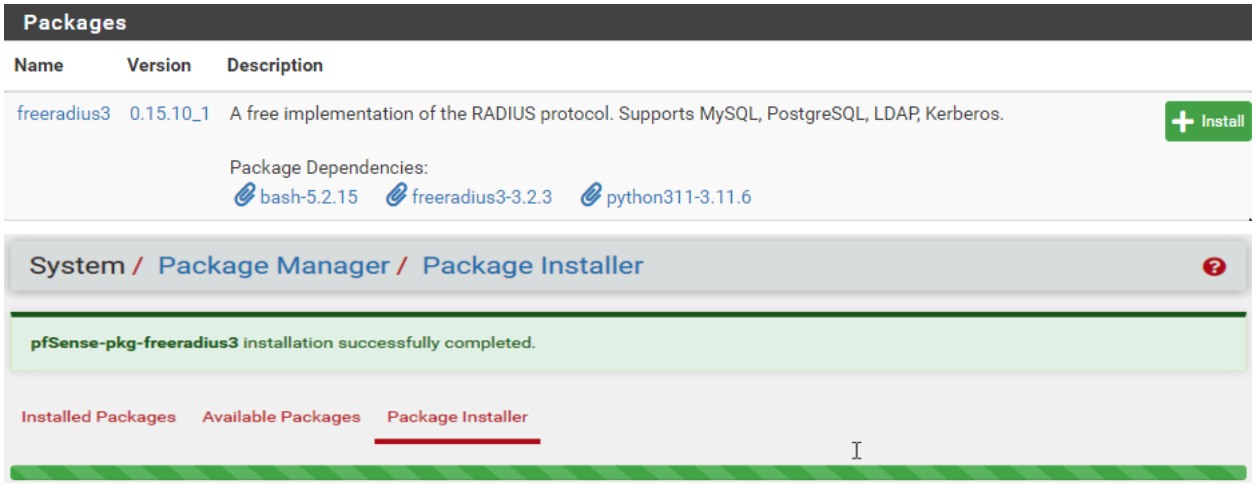


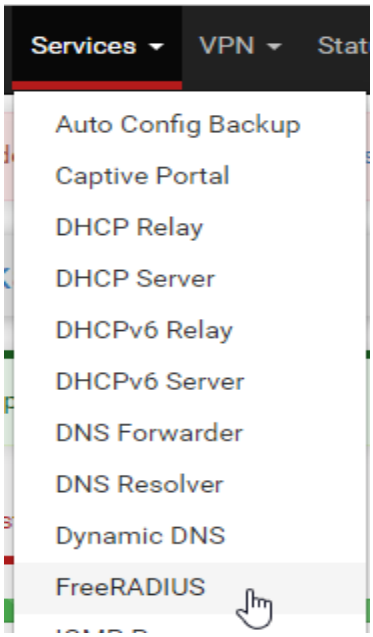
Deployment of RADIUS Authentication Server

- Deploying a RADIUS system that raises a captive portal for new network users and authenticates them using Active Directory (AD) credentials, thus centralizing management to control who can access the network infrastructure.

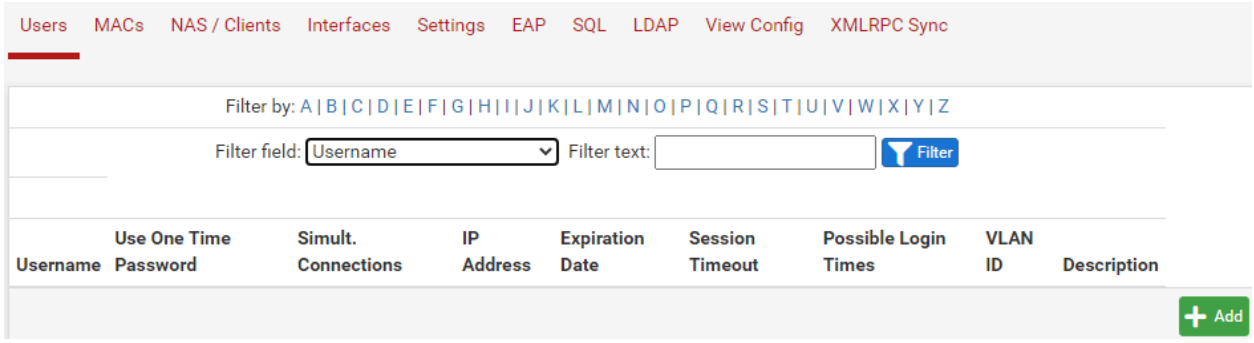
Installing and configuring Radius (It is a client-server protocol and software that enables remote access servers to communicate with a central server to authenticate and authorize their access to the request system or service).

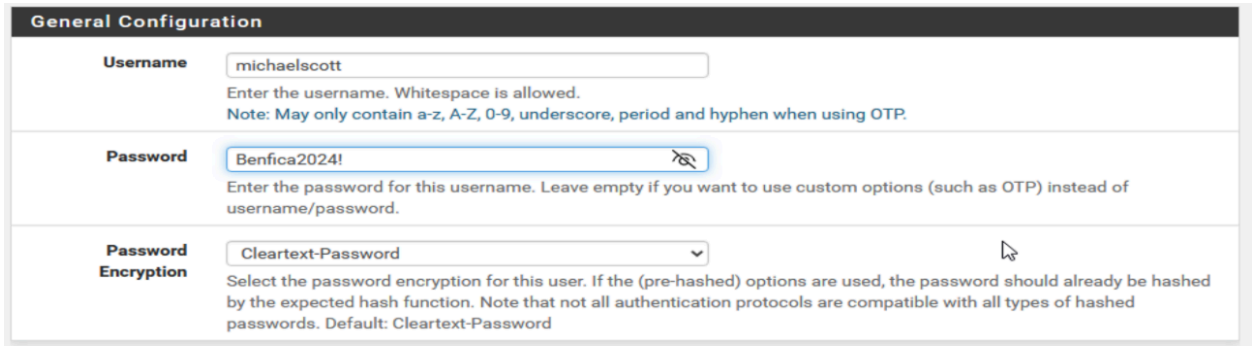


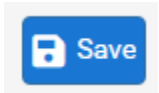


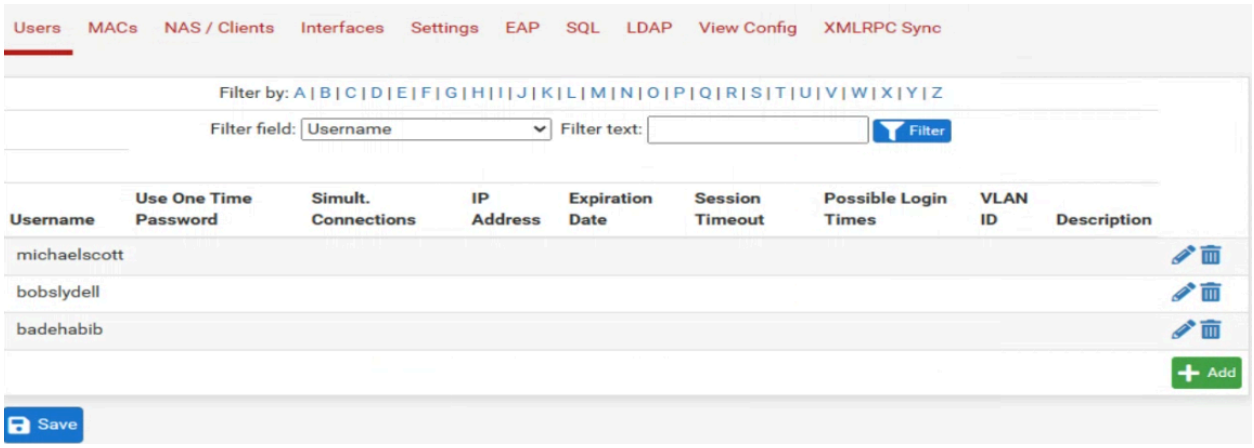


Creating a user.



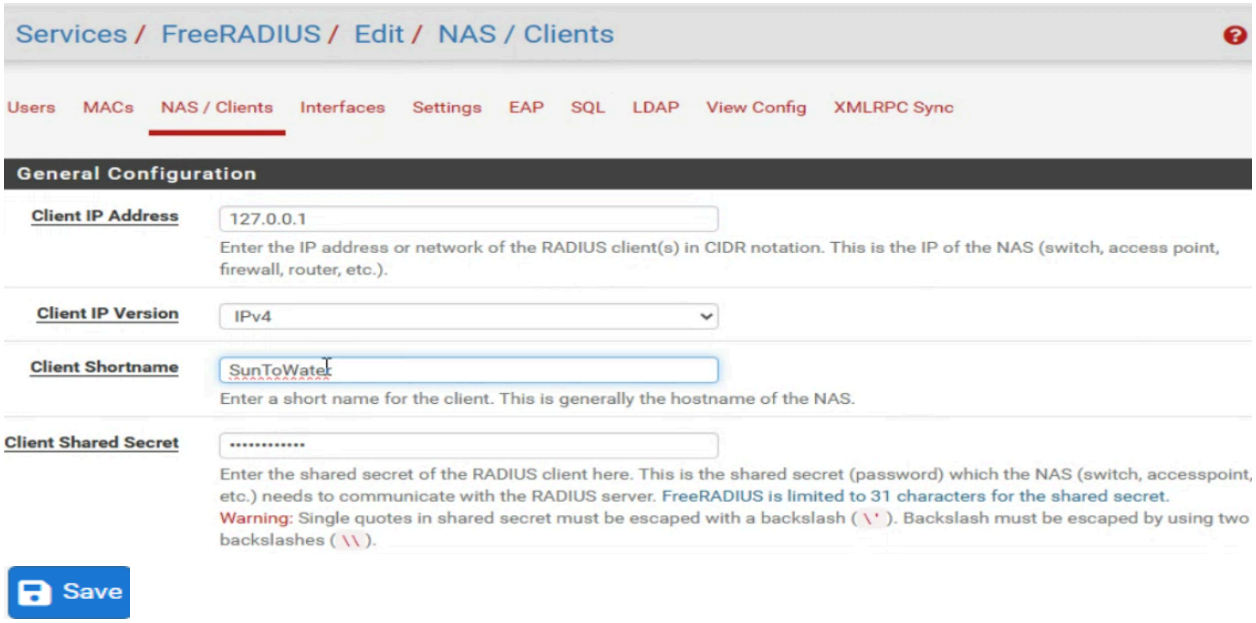


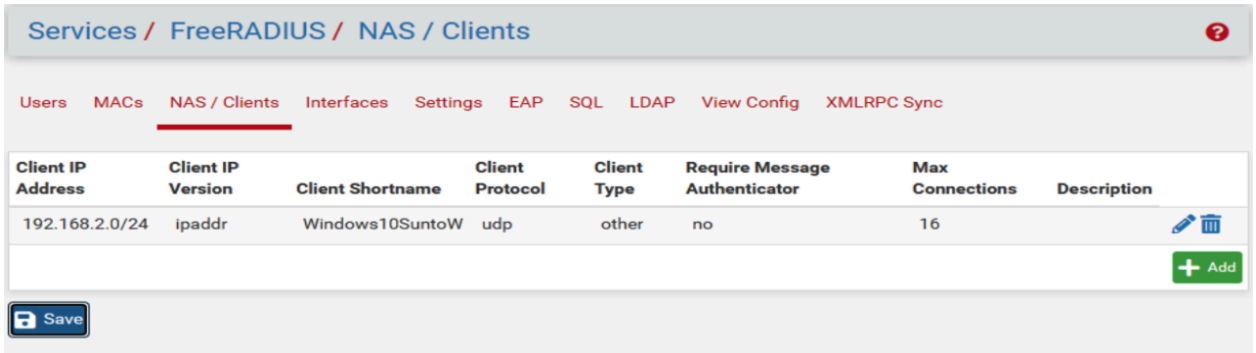




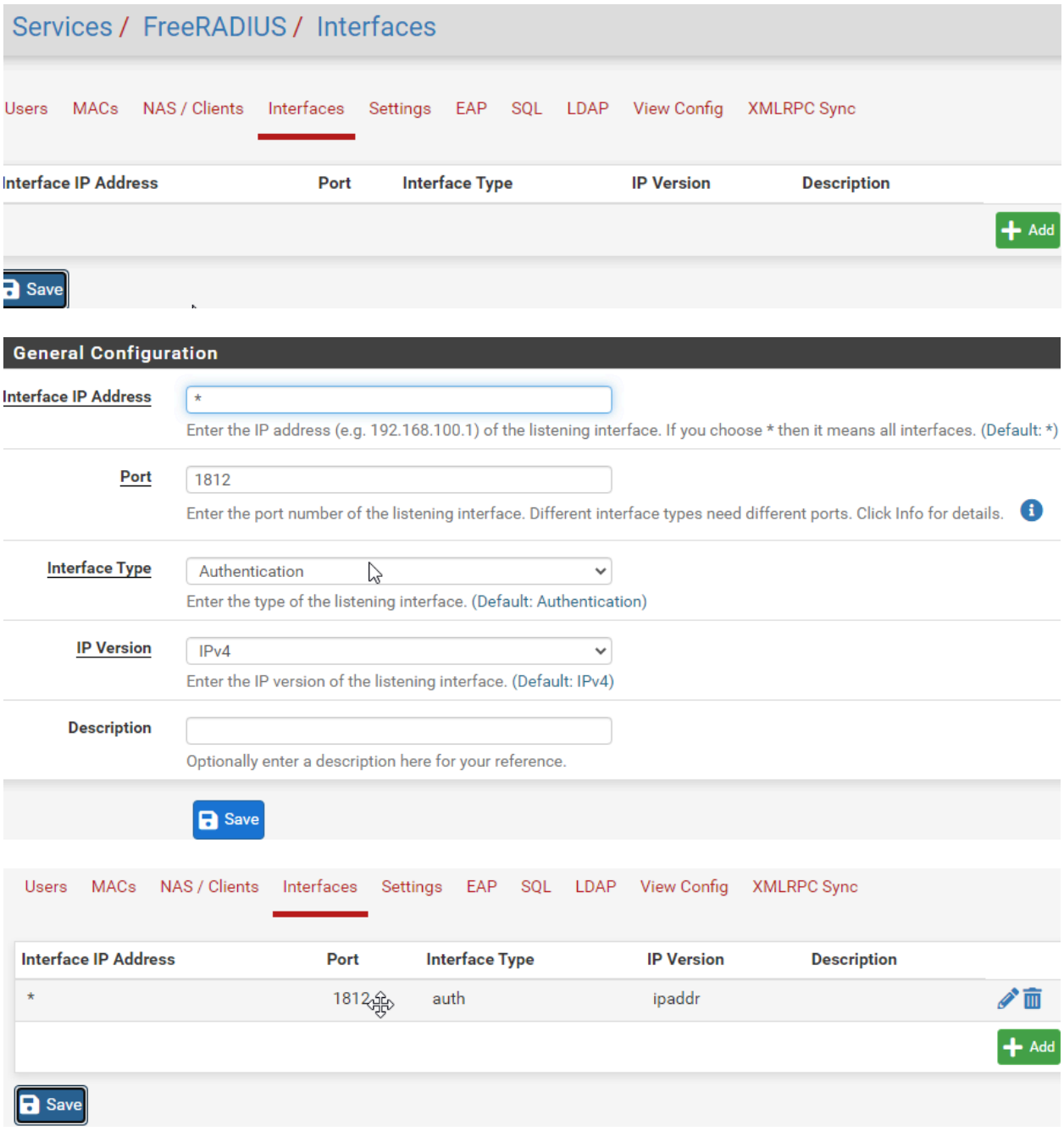
Configure the NAS / client of radius. IP 127.0.0.1/24, password: Benfica2024!

NAS (Network Attached Storage)
Define the ipV4 of the client from which the RADIUS server should accept packets.

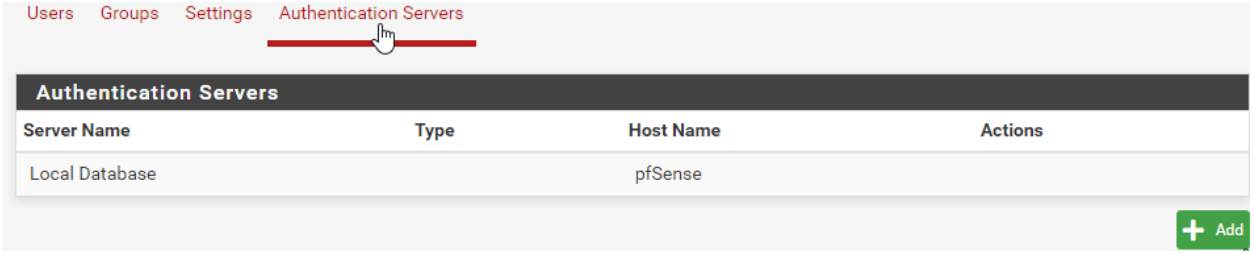


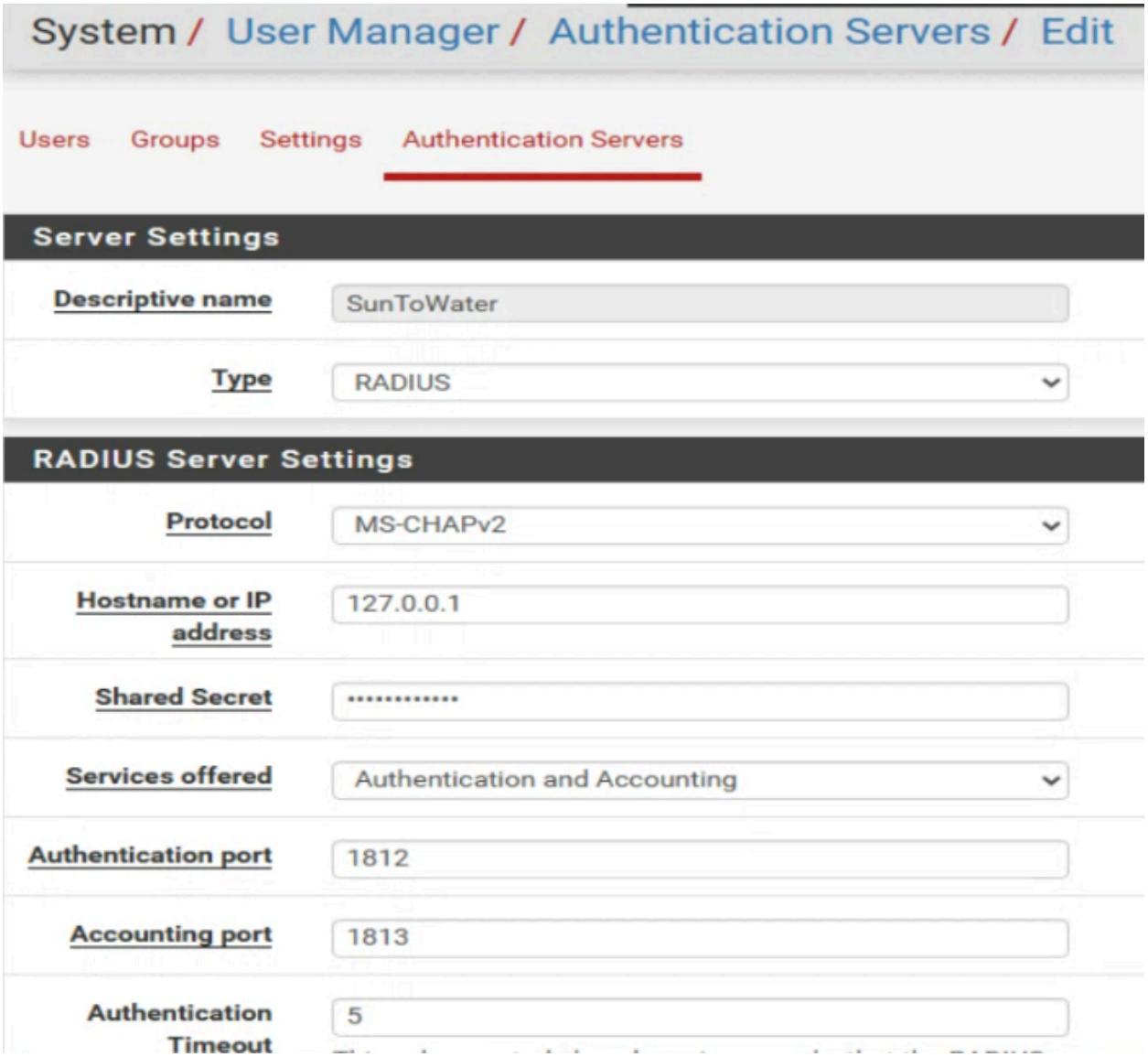


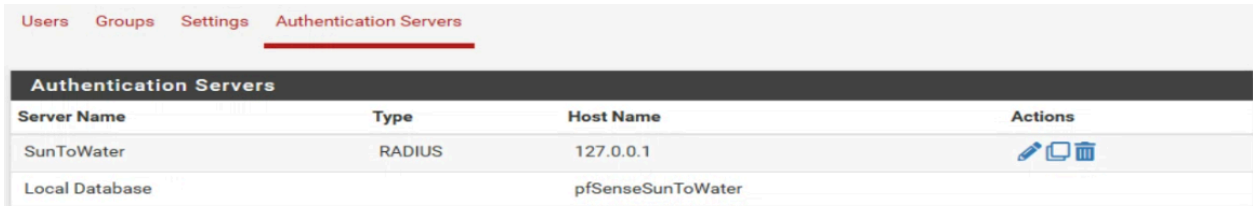
Configure listen interfaces, the ipV4 in this case authentication. We choose (*) because it means all interfaces, and use Port 1812.



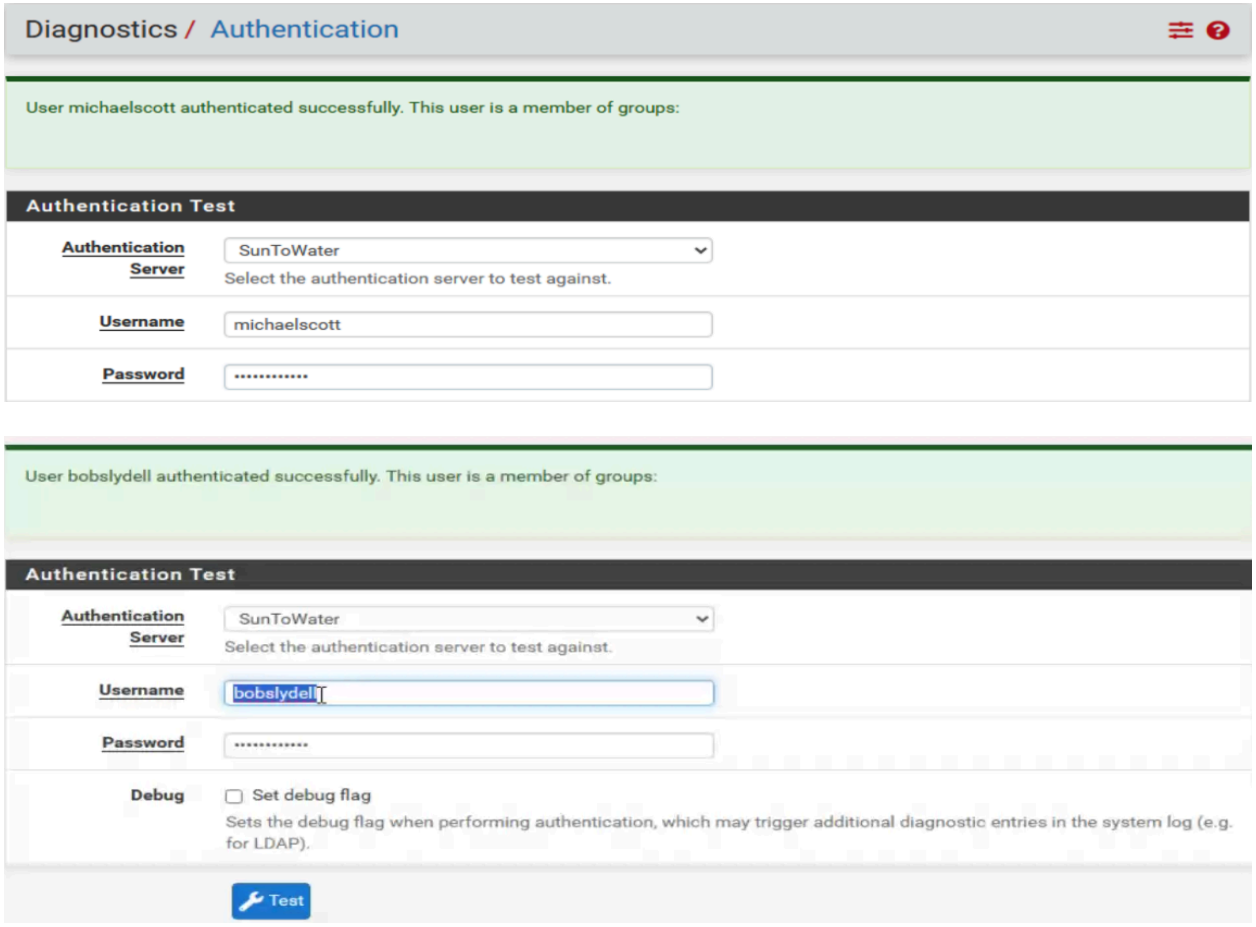
Add radius database. IP server 127.0.0.1, Wan 10.0.2.14, password: benfica

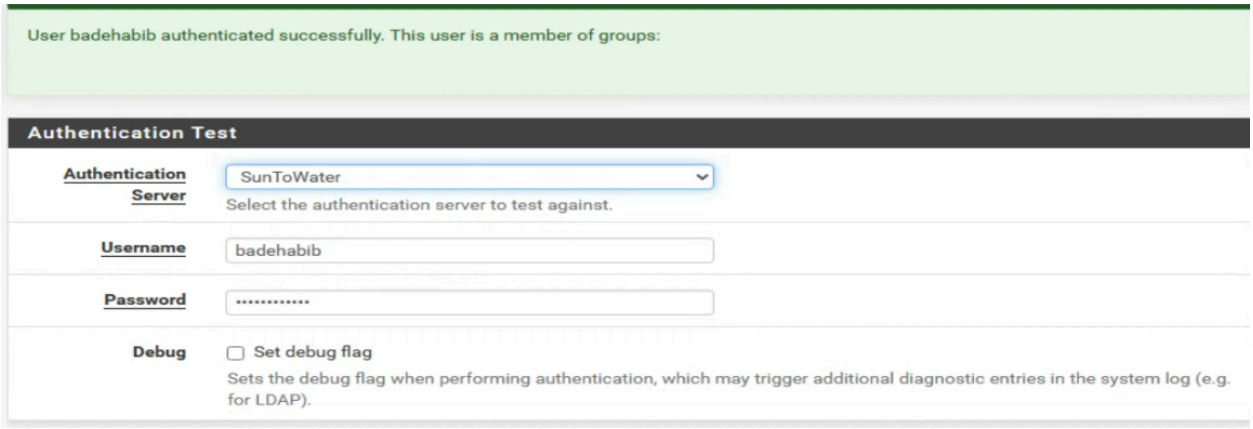






Testing Authentication user’s





Logs authentication report.

- Status / System Logs / Authentication / Captive Portal Auth

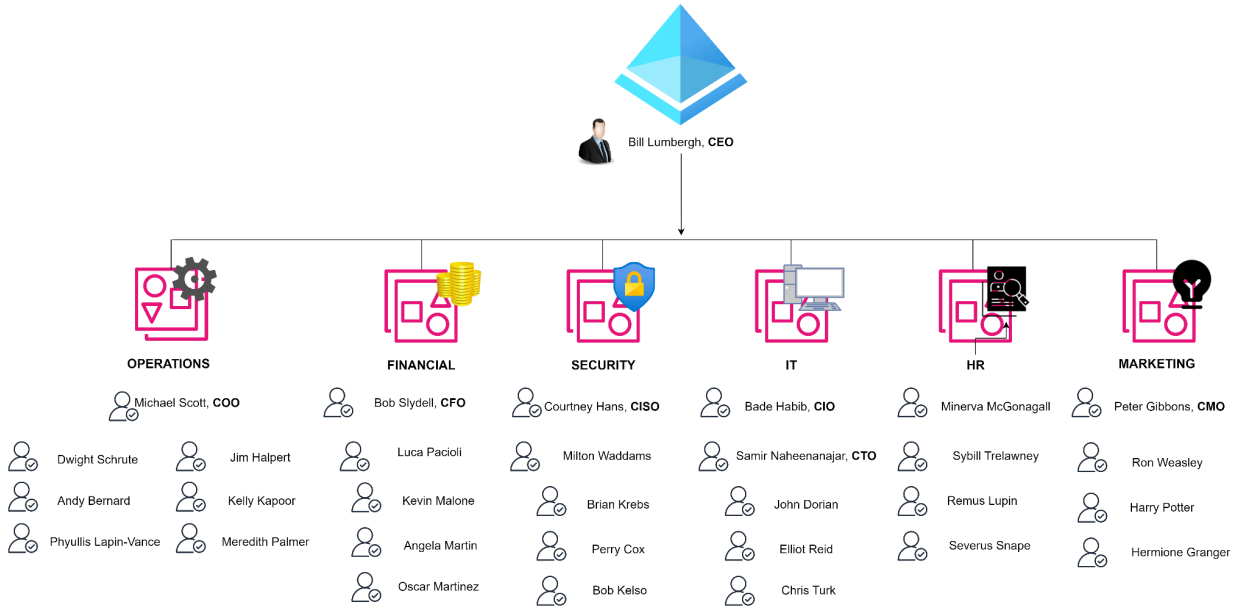
General | Captive Portal Auth | PPPoE Logins | L2TP Logins | OS User Events | OS Account Changes

Last 10 Captive Portal Auth Log Entries. (Maximum 500)

Time	Process	PID	Message
Apr 19 08:32:12	logportalauth	397	Zone: suntowater - Reconfiguring captive portal(SunToWater).
Apr 19 08:35:00	logportalauth	396	Zone: suntowater - ACCEPT: unauthenticated, 08:00:27:28:17:23, 192.168.1.100
Apr 19 09:06:32	logportalauth	396	Zone: suntowater - Reconfiguring captive portal(SunToWater).
Apr 19 09:14:12	logportalauth	397	Zone: suntowater - DISCONNECT: unauthenticated, 08:00:27:28:17:23, 192.168.1.100
Apr 19 09:21:44	logportalauth	397	Zone: suntowater - Reconfiguring captive portal(SunToWater).
Apr 19 09:33:50	logportalauth	397	Zone: suntowater - ACCEPT: michaelscott, 08:00:27:28:17:23, 192.168.1.100
Apr 19 10:47:22	logportalauth	1111	Zone: suntowater - Reconfiguring captive portal(SunToWater).
Apr 19 11:48:36	logportalauth	1111	Zone: suntowater - DISCONNECT: michaelscott, 08:00:27:28:17:23, 192.168.1.100
Apr 19 11:52:46	logportalauth	396	Zone: suntowater - ACCEPT: michaelscott, 08:00:27:28:17:23, 192.168.1.100
Apr 19 11:53:35	logportalauth	1111	Zone: suntowater - DISCONNECT: michaelscott, 08:00:27:28:17:23, 192.168.1.100

PowerShell Script for automatic DC Deployment

Present the scenario org chart



Script should demonstrably perform:

- Assigns the Windows Server VM a static IPv4 address
- Assigns the Windows Server VM a DNS

```
1   # Static IP, default Gateway and DNS variables
2
3   $IP = "192.168.1.100"
4   $MaskBits = 24 # This means /24 CIDR block, or subnet mask 255.255.255.0
5   $Gateway = "192.168.1.1"
6   $DNS = "1.1.1.1"
7   $IPType = "IPv4"
8
9   # Retrieve the network adapter that you want to configure
10
11  $adapter = Get-NetAdapter | ? {$_.Status -eq "up"}
12
13  # Remove any existing IP, gateway from our ipv4 adapter
14
15  If (($adapter | Get-NetIPConfiguration).IPv4Address.IPAddress) {
16   $adapter | Remove-NetIPAddress -AddressFamily $IPType -Confirm:$false
17  }
18  If (($adapter | Get-NetIPConfiguration).Ipv4DefaultGateway) {
19   $adapter | Remove-NetRoute -AddressFamily $IPType -Confirm:$false
20  }
21
22   # Configure the IP address and default gateway
23
24  $adapter | New-NetIPAddress `
25   -AddressFamily $IPType `
26   -IPAddress $IP `
27   -PrefixLength $MaskBits `
28   -DefaultGateway $Gateway
29
30  # Configure the DNS client server IP addresses
31
32  $adapter | Set-DnsClientServerAddress -ServerAddresses $DNS
```

Variables were used for this part of the project, for ease of configuration. Part of the excerpt to find the current active network adapter was taken from this website (Source).

Lines 30-32 show what commandlets and variables I used to set up a manual DNS provider for the machine for optimal functionality. In this case, I used Cloudflare's DNS service, at 1.1.1.1.

- Renames the Windows Server VM

```
34    # Rename the machine
35
36    Rename-Computer -NewName "STWServer"
```

Renaming the computer is done with the Rename-Computer commandlet, adding the -NewName and the desired name after quotes. Not that the system name cannot exceed 15 characters.

A restart is not required for this part of the set up, but a restart is necessary in the next step before and after.

- Installs AD-Domain-Services

```
38    # Install AD-Domain-Services
39
40    Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools
```

Install-WindowsFeatures -Name AD-Domain-Services, with the extra parameter -IncludeManagementTools, ensures that all of the necessary and extra features required for Active Directory to run are installed.

This part requires a restart to accept the previous Rename-Computer commandlet before installing AD features into the system. Just run the script again after the restart.

- Creates an AD Forest

```
42  # Elevate to Domain Controller and create AD Forest
43
44  $domainAdmin = "Administrator"
45  $domainPassword = ConvertTo-SecureString "Benfica2024!" -AsPlainText -Force
46  $domainAdminCredential = New-Object System.Management.Automation.PSCredential ($domainAdmin, $domainPassword)
47
48  Install-ADDSForest `
49      -DomainName "suntowater.site" `
50      -DomainNetbiosName "SUNTOWATER" `
51      -DomainMode "WinThreshold" `
52      -ForestMode "WinThreshold" `
53      -InstallDNS `
54      -SafeModeAdministratorPassword (ConvertTo-SecureString "Benfica2024!" -AsPlainText -Force) `
55      -Force `
56
```

I attempted to use a few variables to automate the setup process, or else PowerShell would prompt the user to add their own password. I believe not all of these variables ended up being used, such as $domainAdminCredential, but I will keep those there since that is what I ended up uploading to GitHub. It did not contribute to any errors or strange behavior.
This part requires another system restart once complete. Let the system reboot and finish configuring DC and AD, then log back in as the Domain administrator, not as a local user, before running the script again.

- Creates Organizational Units (OU)

```
57  # Create Organizational Units
58
59  New-ADOrganizationalUnit -Name "CEO" -Path "DC=suntowater,DC=site"
60  New-ADOrganizationalUnit -Name "CSO" -Path "DC=suntowater,DC=site"
61  New-ADOrganizationalUnit -Name "HR" -Path "DC=suntowater,DC=site"
62  New-ADOrganizationalUnit -Name "IT" -Path "DC=suntowater,DC=site"
63  New-ADOrganizationalUnit -Name "Security" -Path "DC=suntowater,DC=site"
64  New-ADOrganizationalUnit -Name "Finances" -Path "DC=suntowater,DC=site"
65  New-ADOrganizationalUnit -Name "Marketing" -Path "DC=suntowater,DC=site"
66  New-ADOrganizationalUnit -Name "Management" -Path "DC=suntowater,DC=site"
```

To add a new organization unit, the New-ADOrganizationalUnit commandlet is used, with the -Name and -Path parameters, defining them clearly for the Active Directory's database. Note that, since the Domain is suntowater.site, it is required to separate the path as shown above. Using DC=suntowater or DC=suntowater.site is not recognized as an acceptable path, and it will produce errors and prevent these commands and the next ones from working.

- Creates users

```powershell
# Define the CSV file location and import the data
$Csvfile = "C:\temp\ImportADUsers.csv"
$Users = Import-Csv $Csvfile

# The password for the new user
$Password = "P@ssw0rd1234"

# Import the Active Directory module
Import-Module ActiveDirectory

# Loop through each user
foreach ($User in $Users) {
    try {
        # Retrieve the Manager distinguished name
        $managerDN = if ($User.'Manager') {
            Get-ADUser -Filter "DisplayName -eq '$($User.'Manager')'" -Properties DisplayName |
            Select-Object -ExpandProperty DistinguishedName
        }

        # Define the parameters using a hashtable
        $NewUserParams = @{
            Name                = "$($User.'First name') $($User.'Last name')"
            GivenName           = $User.'First name'
            Surname             = $User.'Last name'
            DisplayName         = $User.'Display name'
            SamAccountName      = $User.'User logon name'
            UserPrincipalName   = $User.'User principal name'
            StreetAddress       = $User.'Street'
            City                = $User.'City'
            State               = $User.'State/province'
            PostalCode          = $User.'Zip/Postal Code'
            Country             = $User.'Country/region'
            Title               = $User.'Job Title'
            Department          = $User.'Department'
            Company             = $User.'Company'
            Manager             = $managerDN
            Path                = $User.'OU'
            Description         = $User.'Description'
            Office              = $User.'Office'
            OfficePhone         = $User.'Telephone number'
            EmailAddress        = $User.'E-mail'
            MobilePhone         = $User.'Mobile'
            AccountPassword     = (ConvertTo-SecureString "$Password" -AsPlainText -Force)
            Enabled             = if ($User.'Account status' -eq "Enabled") { $true } else { $false }
            ChangePasswordAtLogon = $true # Set the "User must change password at next logon"
        }

        # Add the info attribute to OtherAttributes only if Notes field contains a value
        if (![string]::IsNullOrEmpty($User.Notes)) {
            $NewUserParams.OtherAttributes = @{info = $User.Notes }
        }

        # Check to see if the user already exists in AD
        if (Get-ADUser -Filter "SamAccountName -eq '$($User.'User logon name')'") {

            # Give a warning if user exists
            Write-Host "A user with username $($User.'User logon name') already exists in Active Directory." -ForegroundColor Yellow
        }
        else {
            # User does not exist then proceed to create the new user account
            # Account will be created in the OU provided by the $User.OU variable read from the CSV file
            New-ADUser @NewUserParams
            Write-Host "The user $($User.'User logon name') is created successfully." -ForegroundColor Green
        }
    }
    catch {
        # Handle any errors that occur during account creation
        Write-Host "Failed to create user $($User.'User logon name') - $($_.Exception.Message)" -ForegroundColor Red
    }
}
```

This part of the script was more complex and required some specific target directories, file names, and a specific .CSV file with all the user information required for it to be successfully added to Active Directory. Note that this portion of the script can be run standalone once DC is set up, for future changes in the users database of Active Directory.
This script was taken in large part from this source, as it streamlined the process and made it very easy to understand.

The .CSV was made by our team, using the company hierarchy picture provided by the project assignment. After many iterations that could not be successfully imported, this is the final version:

First name	Last name	Display na	User logon	User princ	Street	City	State/prov	Zip/Postal	Country/r	Job Title	Departmei	Company	Manager	OU	Descriptio	Office	Telephone	Other Tele	E-mail	Mobile	Notes	Account status
Bill	Lumbergh	Bill Lumbe	Bill.Lumbe	Bill.Lumbergh@suntowater.site						CEO	CEO	SunToWater		OU=CEO,DC=suntowater,DC=site								Enabled
Minerva	McGonagi	Minerva M	Minerva.M	Minerva.McGonagall@suntowater.site						Executive	HR	SunToWater		OU=HR,DC=suntowater,DC=site								Enabled
Sybill	Trelawney	Sybill Trela	Sybill.Trel	Sybill.Trelawney@suntowater.site						Director Te	HR	SunToWater		OU=HR,DC=suntowater,DC=site								Enabled
Remus	Lupin	Remus Lup	Remus.Lup	Remus.Lupin@suntowater.site						Manager T	HR	SunToWater		OU=HR,DC=suntowater,DC=site								Enabled
Severus	Snape	Severus Sr	Severus.Sr	Severus.Snape@suntowater.site						Human Re	HR	SunToWater		OU=HR,DC=suntowater,DC=site								Enabled
Bade	Habib	Bade Habil	Bade.Habi	Bade.Habib@suntowater.site						Chief Infor	IT	SunToWater		OU=IT,DC=suntowater,DC=site								Enabled
Samir	Naheenan	Samir Nah	Samir.Nah	Samir.Naheenanajar@suntowater.site						Chief Tech	IT	SunToWater		OU=IT,DC=suntowater,DC=site								Enabled
John	Dorian	John Dori	John.Dori	John.Dorian@suntowater.site						Senior Eng	IT	SunToWater		OU=IT,DC=suntowater,DC=site								Enabled
Elliot	Reid	Elliot Reid	Elliot.Reid	Elliot.Reid@suntowater.site						Frontend I	IT	SunToWater		OU=IT,DC=suntowater,DC=site								Enabled
Chris	Turk	Chris Turk	Chris.Turk	Chris.Turk@suntowater.site						Backend D	IT	SunToWater		OU=IT,DC=suntowater,DC=site								Enabled
Courtney	Hans	Courtney I	Courtney.	Courtney.Hans@suntowater.site						Chief Infor	Security	SunToWater		OU=CSO,DC=suntowater,DC=site								Enabled
Milton	Waddams	Milton Wa	Milton.Wa	Milton.Waddams@suntowater.site						Facilites M	Security	SunToWater		OU=Security,DC=suntowater,DC=site								Enabled
Brian	Krebs	Brian Kreb	Brian.Kreb	Brian.Krebs@suntowater.site						Directory I	Security	SunToWater		OU=Security,DC=suntowater,DC=site								Enabled
Perry	Cox	Perry Cox	Perry.Cox	Perry.Cox@suntowater.site						Defensive	Security	SunToWater		OU=Security,DC=suntowater,DC=site								Enabled
Bob	Kelso	Bob Kelso	Bob.Kelso	Bob.Kelso@suntowater.site						Offensive	Security	SunToWater		OU=Security,DC=suntowater,DC=site								Enabled
Bob	Slydell	Bob Slydel	Bob.Slyde	Bob.Slydell@suntowater.site						Chief Finar	Finances	SunToWater		OU=Finances,DC=suntowater,DC=site								Enabled
Luca	Pacioli	Luca Pacio	Luca.Pacio	Luca.Pacioli@suntowater.site						Senior Auc	Finances	SunToWater		OU=Finances,DC=suntowater,DC=site								Enabled
Kevin	Malone	Kevin Mali	Kevin.Mali	Kevin.Malone@suntowater.site						Jr Account	Finances	SunToWater		OU=Finances,DC=suntowater,DC=site								Enabled
Angela	Martin	Angela Ma	Angela.Ma	Angela.Martin@suntowater.site						Sr Account	Finances	SunToWater		OU=Finances,DC=suntowater,DC=site								Enabled
Oscar	Martinez	Oscar Mar	Oscar.Mar	Oscar.Martinez@suntowater.site						Ap Admini	Finances	SunToWater		OU=Finances,DC=suntowater,DC=site								Enabled
Peter	Gibbons	Peter Gibb	Peter.Gibb	Peter.Gibbons@suntowater.site						Chief Mark	Marketing	SunToWater		OU=Marketing,DC=suntowater,DC=site								Enabled
Ron	Weasley	Ron Weas	Ron.Weas	Ron.Weasley@suntowater.site						Sr Manage	Marketing	SunToWater		OU=Marketing,DC=suntowater,DC=site								Enabled
Harry	Potter	Harry Pott	Harry.Pott	Harry.Potter@suntowater.site						Sr Manage	Marketing	SunToWater		OU=Marketing,DC=suntowater,DC=site								Enabled
Hemione	Granger	Hemione (	Hemione.(	Hemione.Granger@suntowater.site						Sr Manage	Marketing	SunToWater		OU=Marketing,DC=suntowater,DC=site								Enabled
Michael	Scott	Michael Sc	Michael.Sc	Michael.Scott@suntowater.site						Chief Oper	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled
Dwight	Schrute	Dwight Sc	Dwight.Sc	Dwight.Schrute@suntowater.site						Divisional	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled
Andy	Bernard	Andy Bern	Andy.Bern	Andy.Bernard@suntowater.site						Regional M	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled
Phyullis	Lapin-Van	Phyullis La	Phyullis.La	Phyullis.Lapin-Vance@suntowater.site						Regional M	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled
Jim	Halpert	Jim Halper	Jim.Halper	Jim.Halpert@suntowater.site						Divisonal V	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled
Kelly	Kapoor	Kelly Kapo	Kelly.Kapo	Kelly.Kapoor@suntowater.site						Regional M	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled
Meredith	Palmer	Meredith I	Meredith.	Meredith.Palmer@suntowater.site						Regional M	Managem	SunToWater		OU=Management,DC=suntowater,DC=site								Enabled

This .CSV was imported with no errors.

Script additions

- **DNS**

```
# Configure DNS service (with Google DNS and Cloudflare DNS as forwarders)

Install-WindowsFeature -Name DNS -IncludeManagementTools
Add-DnsServerPrimaryZone -Name "suntowater.site" -ZoneFile "suntowater.site.dns"
Set-DnsServerForwarder -IPAddress 1.1.1.1, 8.8.8.8
Add-DnsServerResourceRecordA -Name "www" -ZoneName "suntowater.site" -IPv4Address "192.168.1.100" -CreatePtr
```

I noticed that I did not configure a proper DNS service for the Windows Server above, and so it is now set with a basic configuration. All DNS traffic should be forwarded through the VPN tunnel, to the Server's DNS, and then forwarded to the listed providers.

- **RADIUS**

```
# Initial setup for Windows Server RADIUS (couldn't verify if it's working fully due to not having the same virtual environment as the project target)

Install-WindowsFeature -Name NPAS -IncludeManagementTools
New-NpsRadiusClient -Name "SunToWater" -Address "192.168.2.1" -SharedSecret "secretosdeporco"
Set-NpsRadiusServerSettings -AccountingOnOff $true -NpsDnsDomain "suntowater.site"
Set-NpsRadiusAuthentication -EapTls $true -EapTlsCipherSuite "TLS_RSA_WITH_AES_128_CBC_SHA256"
Start-Service "IAS"

# RADIUS check
Get-NpsRadiusClient
Get-NpsConnectionRequestPolicy
Get-NpsNetworkPolicy
```

This part of the script will set up the basic RADIUS server configuration and point it towards the pfSense to be used as the RADIUS client. Note that I believe some extra manual configuration is required with policies that were not included in the script because of the level of granularity required for a proper configuration.

Demonstration

Script part 1

The videos are long because the virtual machine was very slow, so feel free to skip ahead. In the first part, I changed the IP address settings and the basic DNS lookup provider. I also changed the system name, as shown. Since this is one single script that's meant to do everything, it aborted after it tried to install Active Directory because it still needed a name change. Those errors are there for that reason, but after a reboot the setup will continue.

Script part 2

Part 2 installs Active Directory and all necessary tools and features required to run it. Note that installing and the initial setup of Active Directory requires a reboot, so the rest of the script could not be completed in this part.

Script part 3

This final part skips the steps that were already implemented, verifies the installation of Active Directory, and finishes setting up the Forest and the Domain. It also imports all users to the AD database, to be used with RADIUS and for domain logins.