GUIDE

Architectural Guide: Separating Your NAS for Homelab Resilience

> **What you'll have running:** A homelab environment with a securely segmented NAS, enhancing data safety and overall system resilience against common threats through network isolation. **Estimated time:** ~6 hours **Difficulty:** ADVANCED **Power usage:** N/A
>
> **Hardware needed:**

- Existing NAS device (e.g., TrueNAS, unRAID, OpenMediaVault)
- Managed network switch with VLAN support (e.g., UniFi, Cisco, TP-Link Omada)
- Router/Firewall appliance capable of inter-VLAN routing and advanced firewall rules (e.g., pfSense, OPNsense, UniFi Dream Machine)
- Homelab servers/clients that will access the NAS
- Ethernet cables (Cat5e/Cat6 or higher)

Introduction: The Case for NAS Separation

In the exciting world of homelabs, it's easy to get everything up and running on a single flat network. Your shiny new server, your NAS, your desktop, your IoT devices – all living happily together, sharing the same subnet. But as any seasoned homelabber who's been through a "learning experience" will tell you, convenience often comes at the cost of security and resilience.

This guide isn't about simply setting up a NAS; it's about architecting your network to protect your most valuable asset: your data. We're going to dive deep into separating your Network Attached Storage (NAS) from the rest of your homelab, creating a dedicated, isolated network segment. This isn't just a "nice to have"; it's a fundamental step towards a robust, secure, and resilient homelab environment.

Why Separate Your NAS: Risks and Benefits

Think of your NAS as your digital vault. It holds your family photos, critical backups, media library, and potentially sensitive documents. In a flat network, if one of your experimental VMs gets compromised, or an IoT device becomes a pivot point, that attacker could potentially have direct access to your entire NAS. This is a terrifying thought.

Risks of a Flat Network NAS

- **Malware Propagation:** A single compromised server or client on your network can easily spread ransomware or other malware to your NAS, encrypting or destroying your critical data.
- **Accidental Misconfiguration:** A rogue script, an incorrectly configured service, or even human error on a lab server could unintentionally delete or corrupt data on your NAS if it has unrestricted access.
- **Unauthorized Access:** If a service exposed to the internet (e.g., a reverse proxy, a web server) is breached, an attacker could potentially pivot to your NAS.
- **Performance Contention:** High-bandwidth operations from multiple devices on a single network can saturate your NAS's network link, impacting overall performance.
- **Lack of Control:** Without segmentation, you have limited granular control over who or what can access your NAS, and on what ports.

Benefits of NAS Separation

- **Enhanced Security:** By isolating your NAS on its own VLAN, you create a hardened perimeter. Only explicitly allowed devices (e.g., your trusted servers, your admin workstation) can communicate with it, and only on necessary ports. This significantly reduces the attack surface.
- **Improved Resilience:** If a server or client on another VLAN is compromised, the attacker is contained within that segment and cannot directly reach your NAS without bypassing your firewall rules. This buys you time to detect and mitigate.
- **Better Performance:** Dedicated network paths and reduced broadcast traffic on the NAS VLAN can lead to more consistent and higher performance for data transfers.

- **Simplified Troubleshooting:** When issues arise, network segmentation helps you quickly narrow down the scope, as you know exactly which devices should be able to communicate with the NAS.
- **Compliance (Homelab Style):** While not for "official" compliance, it instills good security practices that mirror enterprise environments, making your homelab a better learning ground.

Disaster Scenarios Mitigated by Separation

Let's get concrete. Here are some real-world "oops" moments and malicious attacks that NAS separation helps prevent or contain:

- **The Ransomware Nightmare:** A Docker container on your experimental Proxmox VM gets exploited, and a ransomware payload starts encrypting files. In a flat network, it immediately scans for network shares and encrypts everything on your NAS. With separation, the ransomware is contained to the server VLAN, unable to reach the NAS due to firewall rules.
- **The Rogue Script:** You're testing a new automation script on a lab server. Due to a typo, instead of deleting temporary files, it targets a mounted NAS share. Without proper firewall rules, it could wreak havoc. With separation, you can restrict the server's write access to specific shares or even prevent it from mounting certain volumes.
- **The Compromised IoT Device:** Your smart lightbulb's firmware has a vulnerability, and an attacker gains a foothold. In a flat network, they might be able to scan your network, discover your NAS, and attempt to exploit known vulnerabilities. With an IoT VLAN completely isolated from your NAS VLAN, this attack vector is eliminated.
- **The Misconfigured Public Service:** You're self-hosting a web application with a reverse proxy, and a zero-day exploit allows an attacker to gain root access. If that server also has direct access to your NAS, your data is immediately at risk. With a dedicated server VLAN and strict firewall rules, the attacker's access is limited to that specific server, preventing lateral movement to your NAS.
- **Accidental Exposure:** You temporarily enable SMB guest access for a quick file transfer and forget to disable it. In a flat network, any device could potentially browse your shares. With separation, only devices explicitly allowed by your firewall can even attempt to connect, making accidental exposure less impactful.

Architectural Planning: Network Design Principles

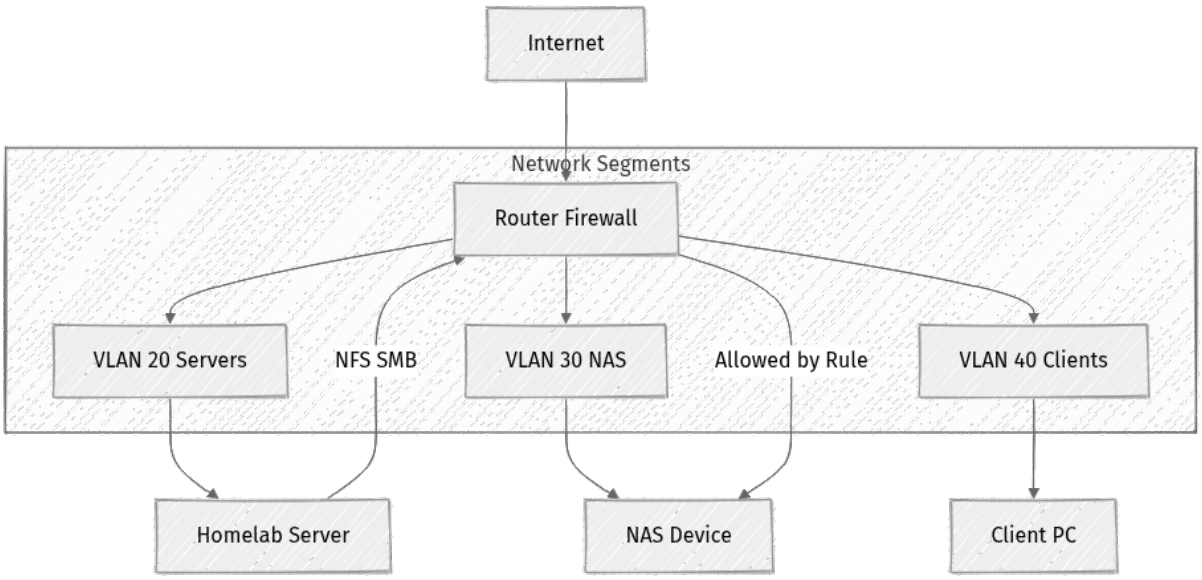
Before we touch any configuration, let's lay out the architectural principles. This isn't just about plugging things in; it's about thoughtful design.

Core Principles

1. **VLANs for Logical Segmentation:** We'll use Virtual Local Area Networks (VLANs) to logically separate traffic on your physical network infrastructure. This means devices on different VLANs cannot communicate directly, even if they're connected to the same physical switch, without passing through a router/firewall.
2. **Dedicated Subnets:** Each VLAN will have its own unique IP subnet (e.g., 192.168.10.0/24, 192.168.30.0/24). This makes routing and firewall rules easier to manage.
3. **Router/Firewall as the Gatekeeper:** Your router/firewall appliance (pfSense, OPNsense, UniFi Dream Machine) will be the central point for all inter-VLAN communication. It will enforce strict firewall rules, acting as a "zero-trust" policy enforcer between your segments.
4. **Least Privilege:** Devices will only be granted the minimum necessary access to the NAS. For example, a media server might only need read access to a media share, while a backup server might need full read/write access to a backup share.
5. **Dedicated Network Interface (Recommended):** If your homelab servers have multiple network interfaces, dedicating one NIC specifically for NAS access can improve performance and further isolate traffic.

Conceptual Network Diagram

Here's a simplified view of the logical network segmentation we're aiming for:



In this diagram:

- `Router_FW` is your central policy enforcement point.
- Each `VLAN` represents a distinct, isolated network segment.
- Traffic between `Homelab_Server`, `NAS_Device`, and `Client_PC` must pass through `Router_FW`, where firewall rules dictate what is allowed.

Hardware and Software Prerequisites

Let's ensure you have the right tools for the job.

Hardware Checklist

- **Existing NAS device:** This is your target. It needs to support a static IP configuration.
    - Example: TrueNAS, unRAID, OpenMediaVault, Synology, QNAP.
- **Managed network switch with VLAN support:** This is crucial for segmenting your physical network into logical VLANs. Ensure it's not just a "smart" switch but explicitly supports 802.1Q VLAN tagging.
    - Example: UniFi Switch (any model), Cisco Catalyst (older enterprise gear), TP-Link Omada (JetStream series), Netgear GS series.

- **Router/Firewall appliance:** This device will handle inter-VLAN routing and enforce your firewall rules. It needs multiple physical interfaces or support for VLAN interfaces on a single physical interface (which is common).
    - Example: pfSense, OPNsense (on dedicated hardware or VM), UniFi Dream Machine (Pro/SE), Sophos UTM, FortiGate (if you're feeling fancy).
- **Homelab servers/clients:** Any machines that need to access your NAS.
    - Example: Proxmox VE host, Docker server, Windows/Linux workstation.
- **Ethernet cables:** Cat5e/Cat6 or higher.

Software Stack

- **NAS Operating System:** The OS running on your NAS device.
    - Example: TrueNAS CORE/SCALE, unRAID OS, OpenMediaVault, Synology DSM, QNAP QTS.
- **Router/Firewall Operating System:** The OS running on your router/firewall appliance.
    - Example: pfSense, OPNsense, UniFi OS.
- **Managed Switch Firmware:** The firmware/OS running on your managed switch.
- **Client OS:** The OS on your servers/clients that will access the NAS shares.
    - Example: Linux (Ubuntu, Debian, CentOS), Windows (10/11, Server), macOS.

> ⚠️ **Warning:** Before proceeding, ensure you have administrative access to all these devices and are comfortable making network configuration changes. Incorrect VLAN or firewall rules can lead to network outages. Have a backup plan or physical access to reset devices if something goes wrong.

Step 1: Network Segmentation with VLANs (Conceptual Design)

Our first step is purely conceptual: defining our VLANs and their associated IP subnets. Choose subnets that don't conflict with your existing network or any other networks you might create later. For this guide, we'll use a common /24 subnet mask, allowing for 254 usable IPs per VLAN.

Let's define our segments:

- **VLAN 10: Management**
  - **Purpose:** For accessing your router/firewall, switch, NAS GUI, and other infrastructure management interfaces. Your admin workstation should reside here.
  - **IP Subnet:** `192.168.10.0/24`
  - **Gateway:** `192.168.10.1`
  - **VLAN ID:** `10`
- **VLAN 20: Servers**
  - **Purpose:** Where your homelab servers (Proxmox, Docker hosts, VMs, etc.) reside. These servers will be the primary consumers of NAS storage.
  - **IP Subnet:** `192.168.20.0/24`
  - **Gateway:** `192.168.20.1`
  - **VLAN ID:** `20`
- **VLAN 30: NAS**
  - **Purpose:** This is our dedicated, isolated network for the NAS device(s).
  - **IP Subnet:** `192.168.30.0/24`
  - **Gateway:** `192.168.30.1`
  - **VLAN ID:** `30`
- **VLAN 40: Clients**
  - **Purpose:** Your general client devices (desktops, laptops, phones) that might need limited access to the NAS (e.g., for media streaming or personal file storage).
  - **IP Subnet:** `192.168.40.0/24`
  - **Gateway:** `192.168.40.1`
  - **VLAN ID:** `40`

> 💡 **Tip:** You might already have a "default" VLAN (often VLAN 1) where your existing devices reside. You can either repurpose that as one of your new VLANs (e.g., VLAN 40) or create new ones and migrate devices. For simplicity, we'll assume you're creating new VLANs and migrating.

Step 2: Configuring Your Managed Switch for VLANs

Your managed switch is where the physical network meets the logical segmentation. We'll configure ports as either "access" ports (for devices that don't understand VLAN tags, like your NAS or servers) or "trunk" ports (for connections to your router/firewall, carrying multiple VLANs).

General Concepts

- **Access Port:** A port configured for a single VLAN. Any device connected to it will be on that VLAN, and traffic leaving the port will be untagged.
- **Trunk Port:** A port configured to carry traffic for multiple VLANs. Traffic leaving a trunk port is typically tagged with its respective VLAN ID (802.1Q). This is how your router/firewall receives traffic from all your VLANs on a single physical cable.
- **PVID (Port VLAN ID):** For access ports, this specifies the VLAN ID that untagged traffic entering the port belongs to.

Configuration Examples

Example 1: Generic CLI Managed Switch (e.g., Cisco SG series, some TP-Link/Netgear)

Let's assume:

- `port 1` is connected to your Router/Firewall (trunk).
- `port 5` is for your NAS (VLAN 30 access).
- `port 10` is for a Homelab Server (VLAN 20 access).
- `port 15` is for your Admin Workstation (VLAN 10 access).

```
# --- Global VLAN Creation ---
vlan 10
 name Management
vlan 20
 name Servers
vlan 30
 name NAS
```

```
vlan 40
 name Clients

# --- Configure Trunk Port (Port 1 to Router/Firewall) ---
# This port will carry all VLANs (10, 20, 30, 40)
interface gigabitethernet 1/0/1
 switchport mode trunk
 switchport trunk allowed vlan add 10,20,30,40
 # If your management interface for the switch is on VLAN 10,
 # you might also set the native VLAN to 10 so untagged management traffic
 # from the router on the trunk port is handled correctly.
 # switchport trunk native vlan 10
!

# --- Configure Access Port for NAS (Port 5) ---
# The NAS device itself doesn't tag traffic, so it's an access port for VLAN
30.
interface gigabitethernet 1/0/5
 switchport mode access
 switchport access vlan 30
!

# --- Configure Access Port for Homelab Server (Port 10) ---
interface gigabitethernet 1/0/10
 switchport mode access
 switchport access vlan 20
!

# --- Configure Access Port for Admin Workstation (Port 15) ---
interface gigabitethernet 1/0/15
 switchport mode access
 switchport access vlan 10
!

# Save configuration (command varies by switch, e.g., 'write memory' or 'copy
running-config startup-config')
```

Example 2: UniFi Switch (GUI Concept)

In UniFi, you'd typically:

1. **Create Networks/VLANs:** Go to `Settings > Networks > Create New Network`.
    - Name: `Management`, VLAN ID: `10`, Gateway IP/Subnet: `192.168.10.1/24` (this creates the VLAN interface on the UniFi Dream Machine/Gateway).
    - Repeat for `Servers` (VLAN 20), `NAS` (VLAN 30), `Clients` (VLAN 40).

2. **Configure Switch Ports:** Go to `Devices > [Your Switch] > Ports`.
    - **Trunk Port (to Router/Firewall):** The port connected to your UniFi Dream Machine (or other router/firewall) will automatically be configured as a trunk port, allowing all defined VLANs. You usually don't need to explicitly configure this unless you want to restrict which VLANs traverse it.
    - **NAS Port:** Select the port connected to your NAS. Go to `Port Profile > Select` NAS `network`. This sets it as an access port for VLAN 30.
    - **Server Port:** Select the port connected to your Homelab Server. Go to `Port Profile > Select` Servers `network`. This sets it as an access port for VLAN 20.
    - **Admin Workstation Port:** Select the port connected to your Admin Workstation. Go to `Port Profile > Select` Management `network`. This sets it as an access port for VLAN 10.

Verification (Pre-Router Configuration)

After configuring your switch, connect your NAS to its dedicated VLAN 30 port and a server to its VLAN 20 port. Assign static IPs to these devices within their intended subnets (e.g., NAS to 192.168.30.10, Server to 192.168.20.10), but do not configure a gateway yet.

- **Test 1: Intra-VLAN Connectivity (should fail because no gateway configured yet)**
    - From your Admin Workstation (VLAN 10, e.g., 192.168.10.10), try to `ping 192.168.10.1` (its intended gateway). This should fail.
    - From your Admin Workstation (VLAN 10, e.g., 192.168.10.10), try to `ping 192.168.20.10` (the server on VLAN 20). This should fail.
    - From your Admin Workstation (VLAN 10, e.g., 192.168.10.10), try to `ping 192.168.30.10` (the NAS on VLAN 30). This should fail.

This failure is expected and good! It means your VLANs are isolated at the switch level. If you could ping between VLANs at this stage, your switch configuration is incorrect.

Step 3: Configuring Your Router/Firewall for Inter-VLAN Routing and Rules

This is the brain of your segmented network. Your router/firewall will create virtual interfaces for each VLAN, assign itself as the gateway for those VLANs, and most importantly, enforce firewall rules to control traffic flow.

We'll use pfSense/OPNsense as examples, but the concepts apply to UniFi Dream Machines or other advanced routers.

1. Create VLAN Interfaces

You'll need to create a VLAN interface for each of our defined VLANs (10, 20, 30, 40) on the physical interface connected to your managed switch's trunk port.

pfSense/OPNsense Example:

1. **Navigate:** Go to `Interfaces > Assignments > VLANs` (or `Interfaces > Other Types > VLANs` in OPNsense).
2. **Add VLAN:**
    - Click `+ Add`.
    - **Parent Interface:** Select the physical interface connected to your managed switch (e.g., `igb1` or `LAN`).
    - **VLAN Tag:** Enter `10` (for Management).
    - **Description:** `MGMT_VLAN`.
    - Repeat for VLAN `20` (Servers), `30` (NAS), `40` (Clients).
3. **Assign Interfaces:**
    - Go to `Interfaces > Assignments`.
    - For each new VLAN you created (e.g., `VLAN 10 on igb1`), click `+ Add` to assign it as a new interface (e.g., `OPT1`, `OPT2`, etc.).
    - Rename them to something meaningful (e.g., `MGMT_VLAN`, `SERVERS_VLAN`, `NAS_VLAN`, `CLIENTS_VLAN`).

4. **Configure Each Interface:**
    - Go to `Interfaces > [Your New VLAN Interface, e.g., MGMT_VLAN]`.
    - **Enable Interface:** Check this box.
    - **IPv4 Configuration Type:** Select `Static IPv4`.
    - **IPv4 Address:** Enter the gateway IP for that VLAN (e.g., `192.168.10.1/24` for MGMT_VLAN).
    - **IPv4 Upstream gateway:** `None`.
    - Repeat for `SERVERS_VLAN` (`192.168.20.1/24`), `NAS_VLAN` (`192.168.30.1/24`), `CLIENTS_VLAN` (`192.168.40.1/24`).
    - **DHCP Server (Optional but Recommended):** For `MGMT_VLAN`, `SERVERS_VLAN`, and `CLIENTS_VLAN`, you might want to enable a DHCP server under `Services > DHCP Server > [VLAN Interface]` to automatically assign IPs to devices. For the `NAS_VLAN`, we'll use static IPs on the NAS itself, so DHCP is typically not needed here.

2. Configure Firewall Rules

This is the most critical part for security. By default, most firewalls will block all inter-VLAN traffic until you explicitly permit it. We want a "default deny" posture and then create specific "allow" rules.

General Rule Philosophy for NAS_VLAN (192.168.30.0/24)

1. **Allow NAS to Internet:** For updates, NTP sync, etc. (Outbound).
2. **Allow NAS to DNS:** For name resolution (Outbound).
3. **Allow Servers to NAS:** For NFS/SMB file access (Inbound to NAS_VLAN).
4. **Allow Management to NAS:** For GUI/SSH administration (Inbound to NAS_VLAN).
5. **Allow Clients to NAS:** For limited SMB/media access (Inbound to NAS_VLAN).
6. **Deny All Other:** Explicitly block anything not covered.

pfSense/OPNsense Firewall Rule Examples

Go to `Firewall > Rules > [Your NAS_VLAN Interface]`. Add rules in order from most specific to most general.

```
# --- Rules on NAS_VLAN Interface (192.168.30.1/24) ---
# These rules control INBOUND traffic TO the NAS_VLAN.
# The default rule is usually "block all" which is good.
```

```
# Rule 1: Allow NAS to reach the Internet (for updates, NTP)
# This is an OUTBOUND rule from NAS_VLAN. You'd typically put this on the
NAS_VLAN interface
# or on a floating rule, or rely on the default allow-all rule on the LAN if
it exists
# (but for security, we'll assume default deny).
# Let's assume your WAN interface has a default allow-all outbound rule.
# If not, you'd add a rule on the NAS_VLAN interface:
# Action: Pass
# Interface: NAS_VLAN
# Direction: Out
# IPv4 Protocol: Any
# Source: NAS_VLAN net
# Destination: Any
# Description: Allow NAS VLAN outbound to Internet

# Rule 2: Allow NAS to reach DNS server (e.g., your router's IP or a specific
DNS server)
# Action: Pass
# Interface: NAS_VLAN
# Direction: Out
# IPv4 Protocol: TCP/UDP
# Source: NAS_VLAN net
# Source Port: Any
# Destination: (Your Router's IP, e.g., 192.168.10.1, or a Public DNS like
1.1.1.1)
# Destination Port: 53 (DNS)
# Description: Allow NAS to DNS

# Rule 3: Allow Homelab Servers (VLAN 20) to access NAS (NFS/SMB)
# Action: Pass
# Interface: NAS_VLAN
# Direction: In
# IPv4 Protocol: TCP
# Source: SERVERS_VLAN net (192.168.20.0/24)
# Source Port: Any
# Destination: NAS_VLAN net (or specific NAS IP: 192.168.30.10)
# Destination Port: 111 (NFS portmapper), 2049 (NFS), 445 (SMB/CIFS), 139
(NetBIOS/SMB)
# Description: Allow Servers to NAS (NFS/SMB)

# Rule 4: Allow Management Workstation (VLAN 10) to access NAS GUI/SSH
# Action: Pass
# Interface: NAS_VLAN
# Direction: In
# IPv4 Protocol: TCP
# Source: MGMT_VLAN net (192.168.10.0/24)
# Source Port: Any
# Destination: NAS_VLAN net (or specific NAS IP: 192.168.30.10)
# Destination Port: 80 (HTTP for GUI), 443 (HTTPS for GUI), 22 (SSH)
# Description: Allow Management to NAS GUI/SSH

# Rule 5: Allow Client PCs (VLAN 40) to access NAS (Limited SMB/Media)
# Action: Pass
# Interface: NAS_VLAN
# Direction: In
# IPv4 Protocol: TCP
# Source: CLIENTS_VLAN net (192.168.40.0/24)
# Source Port: Any
# Destination: NAS_VLAN net (or specific NAS IP: 192.168.30.10)
# Destination Port: 445 (SMB/CIFS), 139 (NetBIOS/SMB)
# Description: Allow Clients to NAS (SMB)
```

```
# Rule 6: Block all other traffic to NAS_VLAN (Implicit or Explicit Deny)
# Most firewalls have an implicit "deny all" at the end of the rule list.
# If not, you'd add an explicit block rule as the last rule:
# Action: Block
# Interface: NAS_VLAN
# Direction: In
# IPv4 Protocol: Any
# Source: Any
# Destination: Any
# Description: Block all other inbound to NAS_VLAN
```

> ⚠️ **Warning:** Be extremely careful with firewall rules. Start with the most restrictive rules and gradually open only what's absolutely necessary. Incorrect rules can lock you out of your NAS or entire network.

Verification (Post-Router Configuration)

1. **Connect Devices:** Ensure your Admin Workstation is on VLAN 10, a Homelab Server on VLAN 20, and the NAS on VLAN 30.
2. **Test Gateway Reachability:**
    - From Admin Workstation (VLAN 10): `ping 192.168.10.1` (should succeed).
    - From Homelab Server (VLAN 20): `ping 192.168.20.1` (should succeed).
    - From NAS (VLAN 30): `ping 192.168.30.1` (should succeed).
3. **Test Inter-VLAN Pings (without specific rules):**
    - From Homelab Server (VLAN 20): `ping 192.168.30.10` (the NAS IP). This should still fail if you haven't configured the specific "allow" rules yet. If it succeeds, double-check your firewall's default behavior or existing rules.

4. **Test with Firewall Rules:**
    - Once you've added the "allow" rules (e.g., Rule 3, 4, 5 above), re-test:
        - From Homelab Server (VLAN 20): `ping 192.168.30.10` (should now succeed).
        - From Admin Workstation (VLAN 10): `ping 192.168.30.10` (should now succeed).
        - From Client PC (VLAN 40): `ping 192.168.30.10` (should now succeed).
    - From a device on an unauthorized VLAN (e.g., an IoT device on a separate VLAN), try to `ping 192.168.30.10`. This should consistently fail.

```
    # Example verification from a Homelab Server (192.168.20.10)
    ping 192.168.30.10
```

```
Expected output (after rules are in place):
```

```
    PING 192.168.30.10 (192.168.30.10) 56(84) bytes of data.
    64 bytes from 192.168.30.10: icmp_seq=1 ttl=63 time=0.876 ms
    64 bytes from 192.168.30.10: icmp_seq=2 ttl=63 time=0.793 ms
    ^C
    --- 192.168.30.10 ping statistics ---
    2 packets transmitted, 2 received, 0% packet loss, time 1004ms
    rtt min/avg/max/mdev = 0.793/0.834/0.876/0.041 ms
```

```
Expected output (from an unauthorized VLAN, or before rules):
```

```
    PING 192.168.30.10 (192.168.30.10) 56(84) bytes of data.
    From 192.168.20.1 icmp_seq=1 Destination Host Unreachable
    From 192.168.20.1 icmp_seq=2 Destination Host Unreachable
    ^C
    --- 192.168.30.10 ping statistics ---
    2 packets transmitted, 0 received, +2 errors, 100% packet loss, time
1015ms
```

Step 4: Re-IPing and Configuring Your NAS for the Dedicated VLAN

Now that your network infrastructure is ready, it's time to move your NAS onto its dedicated VLAN.

1. **Physically Connect:** Connect your NAS's network interface to the switch port you configured as an access port for VLAN 30 (e.g., `port 5` in our example).
2. **Access NAS Interface:** Initially, you might need to connect a monitor and keyboard to your NAS, or temporarily connect it to your Management VLAN (if you have an available port) to access its web GUI.
3. **Configure Static IP:** Navigate to your NAS's network settings.
    - **IP Address:** Assign a static IP from your NAS VLAN subnet (e.g., `192.168.30.10`).
    - **Subnet Mask:** `255.255.255.0` (for a /24 subnet).
    - **Gateway:** Set this to your NAS VLAN's gateway (e.g., `192.168.30.1`).
    - **DNS Servers:** Use your router's IP (e.g., `192.168.10.1` if your router handles DNS for all VLANs, or a public DNS like `1.1.1.1`). Make sure your firewall allows the NAS to reach this DNS server.

Example: TrueNAS CORE/SCALE Network Configuration

1. **Log in to TrueNAS Web UI.**
2. Go to `Network > Interfaces`.
3. Click `Add` or `Edit` an existing interface (e.g., `em0` or `eno1`).
4. **Interface Name:** Select the physical interface.
5. **IPv4 DHCP:** Uncheck this.
6. **IPv4 Address:** `192.168.30.10`
7. **IPv4 Netmask:** `24`
8. Click `Apply`.
9. Go to `Network > Global Configuration`.
10. **Default Gateway IPv4:** `192.168.30.1`
11. **Nameserver 1:** `192.168.10.1` (or your preferred DNS server)
12. Click `Save`.

Example: OpenMediaVault Network Configuration

1. **Log in to OMV Web UI.**
2. Go to `Network > Interfaces`.
3. Select your network interface (e.g., `eth0`) and click `Edit`.
4. **Method:** Select `Static`.
5. **IP Address:** `192.168.30.10`
6. **Netmask:** `255.255.255.0`
7. **Gateway:** `192.168.30.1`
8. **DNS Servers:** `192.168.10.1` (or your preferred DNS server)
9. Click `Save` and then `Apply` the configuration changes.

Verification

- **Access NAS GUI:** From your Admin Workstation (VLAN 10), try to access the NAS web GUI (e.g., `<https://192.168.30.10>`). This should succeed if your firewall rule (Rule 4 in Step 3) is correct.
- **Ping from Server:** From your Homelab Server (VLAN 20), `ping 192.168.30.10`. This should succeed if your firewall rule (Rule 3 in Step 3) is correct.
- **NAS Internet Access:** From the NAS itself (e.g., via SSH), try to `ping google.com` or `apt update`. This should succeed if your firewall rules allow outbound traffic and DNS resolution.

```
# From NAS SSH session
ping 8.8.8.8
ping google.com
```

```
Expected output (pinging 8.8.8.8):
```

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=15.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=15.1 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 15.143/15.174/15.205/0.031 ms
```

```
Expected output (pinging google.com):
```

```
    PING google.com (142.250.186.142) 56(84) bytes of data.
    64 bytes from lhr48s28-in-f14.1e100.net (142.250.186.142): icmp_seq=1
ttl=118 time=15.3 ms
    64 bytes from lhr48s28-in-f14.1e100.net (142.250.186.142): icmp_seq=2
ttl=118 time=15.2 ms
    ^C
    --- google.com ping statistics ---
    2 packets transmitted, 2 received, 0% packet loss, time 1001ms
    rtt min/avg/max/mdev = 15.234/15.286/15.339/0.052 ms
```

Step 5: Configuring Homelab Servers/Clients for NAS Access

With the NAS securely segmented, you can now configure your servers and clients to access its shares. Remember, access is now governed by your firewall rules.

NFS (Network File System) for Linux Servers

NFS is commonly used for Linux-based servers (e.g., Proxmox, Docker hosts, VMs) needing high-performance access. Ensure your NAS has NFS shares configured and that the firewall allows NFS traffic from your Server VLAN (ports 111, 2049, plus potentially others for older NFS versions).

`/etc/fstab` Entry for Persistent Mounts

Edit `/etc/fstab` on your Linux server to automatically mount the NFS share at boot.

```
# Example /etc/fstab entry for NFS mount
# NAS_IP:/path/to/share /mnt/my_nfs_share nfs
defaults,noatime,hard,intr,tcp,wsize=8192,rsize=8192 0 0

192.168.30.10:/mnt/pool/datasets/vm_storage /mnt/vm_storage nfs
defaults,noatime,_netdev 0 0
192.168.30.10:/mnt/pool/datasets/docker_data /mnt/docker_data nfs
defaults,noatime,_netdev 0 0
```

- `192.168.30.10`: Your NAS IP.
- `/mnt/pool/datasets/vm_storage`: The exported path from your NAS (check your NAS's NFS share configuration).
- `/mnt/vm_storage`: The local mount point on your Linux server.
- `nfs`: Specifies NFS filesystem type.

- `defaults`: Common mount options (rw, suid, dev, exec, auto, nouser, async).
- `noatime`: Don't update access times on files, improves performance.
- `_netdev`: Ensures the network is up before attempting to mount.
- `0 0`: fs_freq and fs_passno (usually 0 for NFS).

Manual Mount Command (for testing)

```
# Create mount point if it doesn't exist
sudo mkdir -p /mnt/vm_storage

# Mount the share
sudo mount 192.168.30.10:/mnt/pool/datasets/vm_storage /mnt/vm_storage
```

SMB/CIFS (Server Message Block) for Linux, Windows, macOS Clients

SMB is the standard for Windows file sharing and is widely supported by macOS and Linux. Ensure your NAS has SMB shares configured and that the firewall allows SMB traffic from your Server/Client VLANs (ports 139, 445).

Linux `/etc/fstab` Entry for SMB Mounts

You'll need `cifs-utils` installed: `sudo apt install cifs-utils` (Debian/Ubuntu).

```
# Example /etc/fstab entry for SMB mount
# NAS_IP:/share_name /mnt/my_smb_share cifs
username=your_user,password=your_password,uid=1000,gid=1000,iocharset=utf8,fil
e_mode=0770,dir_mode=0770,_netdev 0 0

//192.168.30.10/Media /mnt/media_share cifs credentials=/etc/cifs-
credentials,uid=1000,gid=1000,iocharset=utf8,file_mode=0770,dir_mode=0770,_net
dev 0 0
```

- `//192.168.30.10/Media`: The SMB share path.
- `/mnt/media_share`: Local mount point.
- `credentials=/etc/cifs-credentials`: Recommended for security, store credentials in a separate file.
    - Create `/etc/cifs-credentials` with:

```
        username=your_smb_username
        password=your_smb_password
```

```
- Set permissions: `sudo chmod 600 /etc/cifs-credentials`.
```

- `uid=1000,gid=1000`: Maps file ownership to your local user ID/group ID.
- `file_mode=0770,dir_mode=0770`: Sets permissions for files/directories.

Windows Clients

1. **Open File Explorer.**
2. In the address bar, type `\\192.168.30.10` and press Enter.
3. You should see the available SMB shares. You may be prompted for credentials.
4. To map a network drive: Right-click `This PC` > `Map network drive`.
    - Choose a drive letter.
    - Folder: `\\192.168.30.10\YourShareName`.
    - Check `Reconnect at sign-in`.
    - Enter credentials if prompted.

macOS Clients

1. **In Finder, go to `Go > Connect to Server...` (or `Cmd+K`).**
2. Enter `smb://1192.168.30.10`.
3. Click `Connect`.
4. You'll be prompted for a username and password.
5. Select the share you wish to mount.

Verification

After configuring `fstab` (on Linux) or mapping network drives (Windows/macOS):

```
# On Linux server/client
sudo mount -a # Mounts all entries in fstab
df -h         # Check mounted filesystems
```

Expected output (example for `df -h`):

```
Filesystem                                  Size  Used Avail Use% Mounted on
udev                                        7.8G     0  7.8G   0% /dev
tmpfs                                       1.6G  1.7M  1.6G   1% /run
/dev/sda1                                   234G   12G  210G   6% /
tmpfs                                       7.8G     0  7.8G   0% /dev/shm
tmpfs                                       5.0M     0  5.0M   0% /run/lock
tmpfs                                       7.8G     0  7.8G   0% /sys/fs/cgroup
192.168.30.10:/mnt/pool/datasets/vm_storage  10T  1.2T  8.8T  12% /mnt/
```

```
vm_storage
192.168.30.10:/mnt/pool/datasets/docker_data 10T  500G  9.5T   5% /mnt/
docker_data
//192.168.30.10/Media                    10T  3.5T  6.5T  35% /mnt/media_share
tmpfs                                    1.6G     0  1.6G   0% /run/user/1000
```

- **Test File Operations:** Create a file, copy a file, delete a file on the mounted share from the client/server. Ensure permissions are correct.
- **Check NAS Logs:** Verify that the NAS logs show successful connections from the expected IP addresses (e.g., 192.168.20.x for servers, 192.168.40.x for clients).

Step 6: Testing and Validation of Connectivity and Rules

You've built the system, now rigorously test it. Don't just assume it works; prove it.

Comprehensive Testing Matrix

Use a device on each VLAN to perform the following tests. Record success/failure.

Source VLAN (IP)	Destination (IP/Service)	Expected Outcome	Actual Outcome	Notes
Admin (10.x)	NAS GUI (30.10:443)	SUCCESS		
Admin (10.x)	NAS SSH (30.10:22)	SUCCESS		
Admin (10.x)	NAS SMB (30.10:445)	SUCCESS		
Admin (10.x)	NAS NFS (30.10:2049)	SUCCESS (if enabled)		
Server (20.x)	NAS SMB (30.10:445)	SUCCESS		
Server (20.x)	NAS NFS (30.10:2049)	SUCCESS		
Server (20.x)	NAS GUI (30.10:443)	FAIL (by rule)		
Client (40.x)	NAS SMB (30.10:445)	SUCCESS		
Client (40.x)	NAS NFS (30.10:2049)	FAIL (by rule)		
Client (40.x)	NAS GUI (30.10:443)	FAIL (by rule)		
NAS (30.10)	Internet (8.8.8.8)	SUCCESS		
NAS (30.10)	DNS (1.1.1.1 or 10.1)	SUCCESS		
NAS (30.10)	Server SSH (20.x:22)	FAIL (by rule)		NAS should not initiate to servers
Any Other VLAN (e.g., IoT)	NAS (30.10:Any)	FAIL		Critical isolation test

Tools for Validation

- **`ping`:** Basic connectivity test.

```
ping 192.168.30.10
```

- **`traceroute` / `tracert`:** Shows the path packets take, confirming routing.

```
traceroute 192.168.30.10
```

```
Expected output should show hops through your router's VLAN interface (e.g.,
`192.168.20.1` then `192.168.30.10`).
```

- **`nmap`:** Scan the NAS from different VLANs to ensure only expected ports are open.
    - From a Server (VLAN 20):

```
nmap -p 22,111,139,2049,445 192.168.30.10
```

```
    Expected output: Ports 22, 111, 139, 445, 2049 should show `open`.
- From a Client (VLAN 40):
```

```
nmap -p 22,111,139,2049,445 192.168.30.10
```

```
    Expected output: Only ports 139, 445 should show `open`. Others should be
`filtered` or `closed`.
- From an unauthorized VLAN:
```

```
nmap -p- 192.168.30.10
```

```
    Expected output: All ports should show `filtered` (meaning the firewall is
blocking).
```

- **Firewall Logs:** On pfSense/OPNsense, check `Status > System Logs > Firewall` to see if any `block` rules are being hit or if `pass` rules are correctly logging traffic. This is invaluable for troubleshooting.

Security Hardening and Best Practices for Your NAS VLAN

Separation is a great start, but it's not the end. Here's how to further harden your NAS and its dedicated VLAN.

- **Strong, Unique Passwords & SSH Keys:**
  - For all NAS user accounts and administrative interfaces.
  - Use SSH keys for administrative access instead of passwords where possible. Disable password authentication for SSH.
- **Disable Unused Services:**
  - If you don't use FTP, SFTP, rsync over SSH, WebDAV, etc., disable them on your NAS. Every open port is a potential attack vector.
  - On your router/firewall, ensure no unnecessary ports are forwarded to your NAS from the internet. Ideally, your NAS should never be directly exposed to the WAN.
- **Keep OS and Software Updated:**
  - Regularly update your NAS operating system (TrueNAS, unRAID, OMV).
  - Update your router/firewall OS (pfSense, OPNsense, UniFi OS).
  - Update your managed switch firmware.
  - Automate updates where safe, or set reminders for manual checks.
- **Implement Intrusion Detection/Prevention (IDS/IPS):**
  - Tools like Suricata or Snort on pfSense/OPNsense can monitor traffic on your NAS VLAN for suspicious patterns and alert you, or even block malicious activity.
- **Restrict Admin Access:**
  - Limit NAS administrative access (SSH, Web GUI) only to your Management VLAN and specific Admin Workstation IPs.
  - Avoid using the `root` user for daily tasks. Create a separate admin user with sudo privileges.

- **Principle of Least Privilege for Shares:**
    - Configure NAS share permissions (NFS exports, SMB shares) to grant only the necessary read/write access to specific users or groups from specific IP addresses (e.g., your Server VLAN IPs).
    - Avoid guest access or overly broad permissions like `Everyone` or `0777`.
- **Regular Backups & Snapshots:**
    - Implement a 3-2-1 backup strategy: 3 copies of data, 2 different media, 1 offsite.
    - Utilize NAS features like ZFS snapshots (TrueNAS) or Btrfs snapshots (unRAID, OMV) for quick recovery from accidental deletion or ransomware. These snapshots should be immutable and ideally replicated to another location.
- **Network Time Protocol (NTP):**
    - Ensure your NAS and all network devices synchronize their time with a reliable NTP server. Accurate timestamps are critical for logs and security analysis.
- **Logging and Alerting:**
    - Configure your NAS, switch, and router to send logs to a central syslog server (e.g., an ELK stack, Splunk, or Graylog in your homelab).
    - Set up alerts for critical events (e.g., failed login attempts, disk errors, unexpected network traffic).

Ongoing Maintenance, Monitoring, and Backup Strategies

Your homelab is a living system. Regular care ensures its continued health and security.

Maintenance Schedule

- **Weekly:**
    - Check NAS disk health (SMART reports).
    - Review firewall logs for unusual activity.
    - Verify backup jobs completed successfully.

- **Monthly:**
    - Apply NAS OS updates.
    - Apply router/firewall OS updates.
    - Apply managed switch firmware updates.
    - Test a backup restore (critical for verifying your backups are actually usable).
- **Quarterly:**
    - Review all firewall rules for necessity and correctness.
    - Change administrative passwords.
    - Review NAS share permissions.

Example Update Commands

- **TrueNAS CORE/SCALE:** Use the Web UI (`System > Update`).
- **OpenMediaVault:**

```
sudo apt update
sudo apt upgrade -y
sudo omv-update
```

- **pfSense/OPNsense:** Use the Web UI (`System > Update`).
- **UniFi Devices:** Use the UniFi Controller UI (`Devices > [Device] > Settings > Manage Device > Update`).

Monitoring

- **NAS:** Utilize built-in dashboards for CPU, RAM, disk I/O, and network usage. Many NAS platforms support SNMP for external monitoring.
- **Router/Firewall:** Monitor interface traffic, CPU/RAM usage, and connection states.
- **External Monitoring:** Consider setting up a dedicated monitoring solution in your homelab like:
    - **Prometheus + Grafana:** For collecting metrics and visualizing dashboards.
    - **Netdata:** Real-time performance monitoring.
    - **Zabbix/Nagios:** Comprehensive monitoring and alerting.

Backup Strategies (Beyond NAS Snapshots)

While NAS snapshots are great for quick recovery, they are not a substitute for full backups.

- **Local Backup:** Replicate critical data from your primary NAS to a secondary local storage device (e.g., another NAS, an external HDD).
- **Offsite Backup:** For truly critical data, push encrypted backups to an offsite location (e.g., cloud storage like Backblaze B2, Wasabi, or a friend's homelab).
- **Offline Backup:** For ultimate protection against ransomware, have a physically disconnected backup copy that is only connected for backup/restore operations.

Common Pitfalls and Troubleshooting

Even with careful planning, things can go wrong. Here are some common issues and their fixes.

1. "No route to host" or "Destination Host Unreachable"

Cause:

- **Firewall Blocking:** Most common cause. Your router/firewall is explicitly blocking the traffic, or there's no `pass` rule and an implicit `deny` is active.
- **Incorrect Gateway:** The device trying to connect has the wrong gateway configured, so it doesn't know how to reach other subnets.
- **Incorrect Subnet Mask:** The device thinks the destination is on its local subnet when it's not.
- **Switch VLAN Configuration:** The switch port for the source or destination device is not correctly assigned to its VLAN, or the trunk port to the router isn't configured correctly.

Fix:

- **Check Firewall Logs:** On pfSense/OPNsense, check `Status > System Logs > Firewall`. Look for `block` entries with the source/destination IPs and ports you're testing. Adjust firewall rules accordingly.
- **Verify IP/Gateway/Subnet:** Double-check the network configuration on both the source and destination devices, ensuring correct static IPs, subnet masks, and gateways.

- **Inspect Switch Ports:** Confirm the switch port connected to the source device is an access port for its correct VLAN, and similarly for the destination. Ensure the trunk port to the router is configured to carry all necessary VLANs.

2. "Connection refused"

Cause:

- **Service Not Running:** The target service (NFS, SMB, SSH, HTTP/S) on the NAS is not running or crashed.
- **Firewall Blocking Port:** The firewall is allowing the initial connection but blocking the specific port the service uses. This is different from "no route to host" as the target IP is reachable.
- **Incorrect Port:** You're trying to connect to the wrong port.
- **Service Misconfiguration:** The service on the NAS is configured to listen only on a specific IP address or interface, not the one being used.

Fix:

- **Check Service Status:** Log into your NAS (via direct console or a temporary network connection if needed) and verify the service status.

```
# On Linux-based NAS (e.g., OMV)
sudo systemctl status smbd
sudo systemctl status nfs-kernel-server
```

- **Verify Firewall Rules:** Ensure your firewall rules explicitly allow the specific port(s) required by the service (e.g., 445 for SMB, 2049 for NFS, 22 for SSH, 80/443 for GUI).
- **NAS Service Bindings:** Check your NAS service configuration to ensure it's listening on the correct network interface (the one assigned to VLAN 30).

3. "Permission denied" or "Access denied"

Cause:

- **Incorrect Share Permissions:** The NFS export or SMB share on the NAS has incorrect permissions, not allowing the connecting user/IP to access it.
- **User Mapping Issues:** For SMB, the user account or credentials provided don't match a valid user on the NAS, or there are issues with user ID/group ID mapping for NFS.

- **Firewall Blocking:** While less common for this specific error message, a firewall could block specific aspects of the authentication handshake.

Fix:

- **Review NAS Share Permissions:**
  - **NFS:** Check `/etc/exports` on your NAS (or the NFS share configuration in the GUI). Ensure the client IP (e.g., 192.168.20.0/24 for your Server VLAN) is permitted.
  - **SMB:** Verify the SMB share permissions and user/group access in your NAS GUI.
- **Verify Credentials:** Ensure the username and password (or SSH key for SSH access) are correct and that the user exists on the NAS.
- **UID/GID Mapping (NFS):** If you're having issues with NFS, ensure the UID/GID of the user accessing the share on the client matches a user/group on the NAS, or that `no_root_squash` (use with caution!) or `all_squash` with `anonuid`/`anongid` are configured appropriately.

4. VLAN Tagging Issues / Misconfigured Trunk Port

Cause:

- **Switch Trunk Port:** The port connecting your switch to your router/firewall is not configured as a trunk port, or it's not allowing all the necessary VLANs.
- **Router VLAN Interface:** The VLAN interfaces on your router/firewall are not correctly configured (e.g., wrong VLAN ID, wrong parent interface).
- **Device Access Port:** An end device (NAS, server) is connected to a switch port that's not assigned to its intended VLAN, or it's configured as a trunk when it should be an access port.

Fix:

- **Check Switch Port Configuration:** Revisit Step 2. Use `show running-config` (CLI) or the GUI to verify that the trunk port has `switchport mode trunk` and `switchport trunk allowed vlan add 10,20,30,40`.
- **Verify Router VLAN Interfaces:** Revisit Step 3. Ensure the VLAN IDs on your router's virtual interfaces match the VLAN IDs you defined (10, 20, 30, 40) and that they are assigned to the correct physical parent interface.

- **Confirm Device Access Ports:** Ensure all end devices are connected to switch ports configured as `access` ports for their respective VLANs (e.g., NAS on VLAN 30 access port).

> 🧠 **Important:** When troubleshooting network issues, always follow a systematic approach:
>
> 1. **Physical Layer:** Is everything plugged in correctly? Are cables good?
> 2. **Link Layer:** Is the switch port up and configured correctly (access/trunk, VLAN)?
> 3. **Network Layer:** Are IPs, subnet masks, and gateways correct? Can you ping the gateway? Can you ping the destination?
> 4. **Transport Layer:** Is the firewall allowing the specific port?
> 5. **Application Layer:** Is the service running on the destination? Are permissions correct?

By methodically checking each layer, you'll isolate the problem quickly. This advanced setup provides immense security benefits, but it does require a deeper understanding of networking. Embrace the challenge, and your homelab will be more resilient than ever!