GUIDE
UTM Kali Linux Shared Folder Configuration

v1.0.0

Author:

Eldon Gabriel

October 17, 2025

TABLE OF CONTENTS

Disclaimer: This report documents my personal work in completing a lab exercise. It reflects my independent understanding and application of publicly available tools and techniques. No proprietary instructional videos, lab guides, or specific content from the original exercise have been posted, shared, or distributed, ensuring full compliance with academic and professional policies.

REVISION HISTORY

Version	Date	Author	Description of Changes
v1.0.0	10/17/2025	Eldon G.	Initial draft.

1.0 SHARED FOLDER CONFIGURATION IN UTM

1.1 Project Description

This project **documents the configuration** of a persistent shared folder between a macOS host and a Kali Linux virtual machine (VM) using the UTM hypervisor. The goal was to make it easy to share files, such as tools, reports, and evidence, between the host folder (*test_folder*) and the Kali system.

1.2 Initial Challenge

Initially, trying to use the new VirtIO-FS method for mounting the shared folder did not work. Even though sharing was enabled in the UTM settings, Kali Linux could not find the shared tags. Despite enabling the shared folder in the UTM settings, the initial attempt using the VirtIO-FS protocol failed, yielding the following critical kernel error messages:

```
cmd

wrong fs type, bad option, bad superblock on share
```

The errors indicate that the guest system did not correctly understand the VirtIO-FS share. This is probably because the Linux kernel modules used by Kali in the UTM environment were not fully compatible.

1.3 Technical Solution

To resolve this kernel-level incompatibility, a different and more robust file system method was leveraged: the **Plan 9 Filesystem (9p)** protocol over the **VirtIO transport layer.** This method allows stable and simple communication between the host and guest, even if the VirtIO-FS does not work

Working Command:

```
cmd

sudo mount -t 9p -o trans=virtio,version=9p2000.L share
~/host_share
```

Explanation:

This command uses the 9p filesystem (`-t 9p`) and sets the transport mode to VirtIO. The `version=9p2000.L` option ensures compatibility with the Linux guest. The `share` tag serves as the UTM's internal, fixed identifier for the host directory (e.g., '*test_folder*'), which is then mapped to the local mount point `~/host_share` on the Kali VM.

By combining VirtIO transport with the 9p protocol, the guest system can access host files, achieving stable integration between macOS and Linux.

1.4 Achieving Persistence

To make the configuration persistent across reboots, a new entry was added to the `/etc/fstab`. This ensures that the shared folder is mounted during system boot **via the kernel initialization process**.

`/etc/fstab` **entry:**

```
cmd

share /home/kali/host_share 9p
trans=virtio,version=9p2000.L,rw,nofail 0 0
```

- `rw` allows read and write access to the device.
- `nofail` ensures that the boot continues even if the share is temporarily unavailable.
- This approach follows the best practices for maintaining reliability in virtual laboratory environments.

1.5 Additional Work

After implementing persistence, I tested the configuration by transferring files between the macOS and Kali Linux systems. File permissions were verified to confirm the correct read/write functionality and ownership within the mounted directory, thereby confirming full operational access.

2.0 CONCLUSION

2.1 Key Takeaways

This task emphasizes the importance of understanding virtualized filesystem protocols and kernel-level compatibility. When VirtIO-FS fails, falling back to the 9p protocol provides a robust and versatile alternative.

2.2 Security Implications and Recommendations

Shared folders enhance efficiency but require careful control of permissions and ownership rights. For sensitive data handling, the following is recommended:

- Limit access to the shared directory using strict permissions (`chmod`, `chown`).
- Sharing should be disabled when not required to reduce exposure risk.
- Encrypted archives should be used when transferring confidential data.

This configuration demonstrates critical system administration and troubleshooting skills, highlighting proficiency in managing cross-platform virtualization environments essential for cybersecurity and IT operations professionals.
