Vulnerability Assessment Report

Internal Lab Environment – Network-Based Assessment

Assessment Type: Internal Network Vulnerability Assessment (Unauthenticated)

Prepared By: Saar Yachin
Date: 20 April 2026

Table of Contents

1. Executive Summary

A vulnerability assessment was conducted against an internal lab environment consisting of three Linux hosts. The objective was to identify exposed services, validate security weaknesses, and demonstrate a full vulnerability management workflow including discovery, validation, remediation, and verification. This assessment was conducted from an internal network vantage point (jump server) without the use of host credentials, simulating an attacker with network access but no prior authentication.

The assessment identified:

- **1 Critical** vulnerability (Redis unauthenticated access)
- **1 Medium** vulnerability (SMB signing not enforced)

Both vulnerabilities were successfully:

- manually validated
- remediated through configuration changes
- verified via manual testing and re-scanning

Following remediation, no critical vulnerabilities remain, and the overall risk posture of the environment has been significantly improved.

2. Scope

The assessment included the following hosts:

Host	OS	Role	Description
172.16.1.33	Ubuntu 22.04 (Proxmox LXC)	Multi-Service	Redis, SMB
172.16.1.34	Ubuntu 22.04 (Proxmox LXC)	Directory	LDAP, SSH
172.16.1.35	Ubuntu 22.04 (Proxmox LXC)	Web Server	HTTP, SMTP, SSH

3. Methodology

The assessment followed a structured workflow:

3.1. Service Discovery

- Nmap used to identify open ports and services

3.2. Vulnerability Scanning

- Nessus Essentials used for automated detection

3.3. Validation

- Manual validation was performed to confirm findings and eliminate potential false positives, using:

- redis-cli
- smbclient
- Nmap NSE scripts

3.4. Remediation

- Configuration hardening of exposed services

3.5. Verification

- Manual re-testing
- Follow-up Nessus scan

---

4. Initial Scan Overview

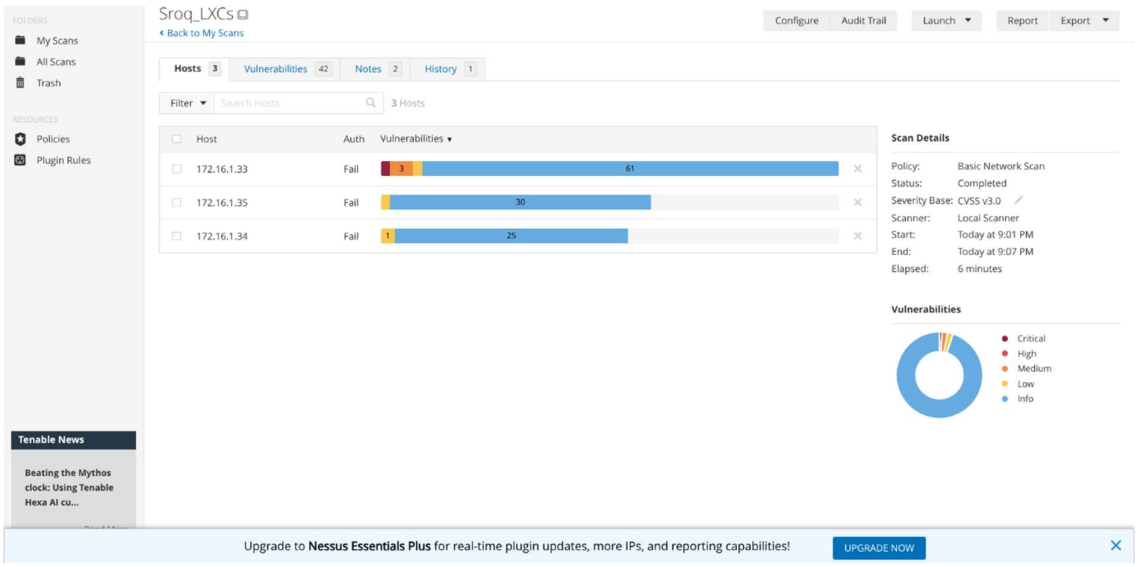

Figure 1a: Initial Nessus scan showing all hosts.

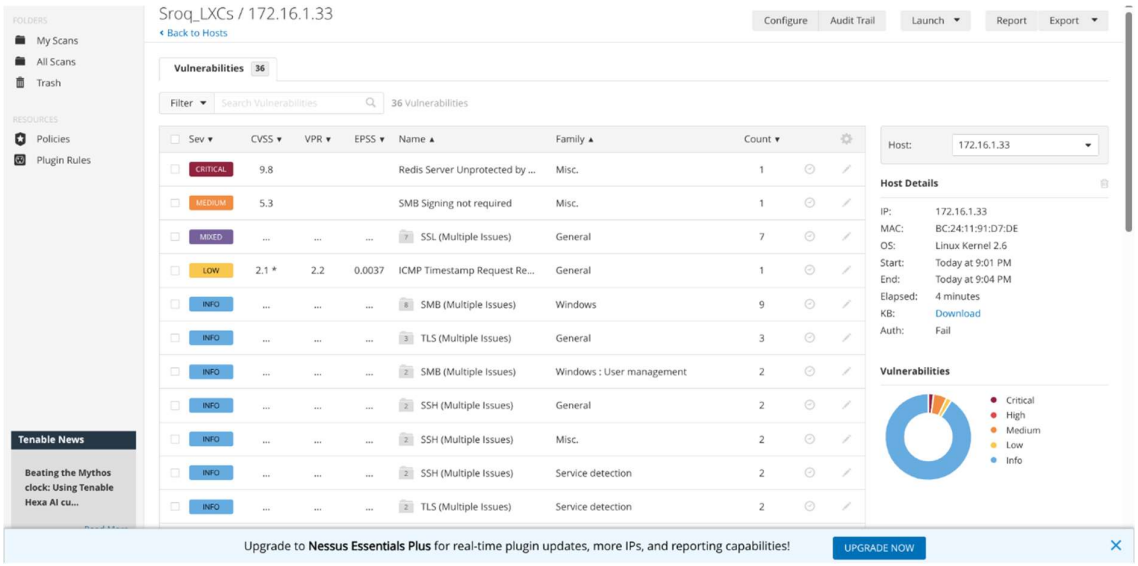

Figure 1a: Initial Nessus scan showing critical Redis exposure and SMB configuration issue on host 172.16.1.33.

5. Findings Summary

Finding	Host	Plugin ID	Status
Redis Server Unprotected by Authentication	172.16.1.33	100634	Remediated
SMB Signing Not Required	172.16.1.33	57608	Remediated

Low and informational findings (e.g., ICMP timestamp, SSL self-signed certificates) were identified but excluded from detailed analysis due to their limited impact in this environment.

6. Detailed Findings

6.1. Redis Server Unprotected by Authentication

Plugin ID: 100634

Severity: **Critical** (CVSS 9.8)

Affected Host: 172.16.1.33 (TCP 6379)

Description

The Redis service allowed unauthenticated access, enabling any network user to interact with the database without credentials. The vulnerability is rated CVSS 9.8 due to: a. network exposure (AV:N); b. no authentication required (PR:N); c. full impact on confidentiality, integrity and availability (C:H/I:H/A:H).

Evidence (Pre-Remediation)

```
saar@jumpingrook:~$ redis-cli -h 172.16.1.33
172.16.1.33:6379> INFO
# Server
redis_version:6.0.16
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:e91d7986ac6d5bb6
redis_mode:standalone
os:Linux 6.17.13-2-pve x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:11.4.0
process_id:117
run_id:3e839f7816ad08d19c89fcff2b6eff81e10313bf
tcp_port:6379
uptime_in_seconds:21
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:15067953
executable:/usr/bin/redis-server
config_file:/etc/redis/redis.conf
io_threads_active:0
```

Figure 2: Unauthenticated INFO command reveals internal Redis configuration and system data.
(See full output in appendix)

Validation

Manual interaction confirmed full access:

```
# Keyspace
172.16.1.33:6379> SET test_key 'test'
OK
172.16.1.33:6379> GET test_key
"test"
```

Figure 3: Ability to create and retrieve keys confirms unauthorized read/write access.

Impact

An attacker could:

- access sensitive cached data
- modify or delete application data
- disrupt service functionality
- potentially pivot to other systems

Remediation

Redis was secured by:

- enabling authentication (requirepass [STRONG_PASSWORD]) in /etc/redis/redis.conf
- restarting the service

Verification (Post-Remediation)

Figure 4: Redis now requires authentication (NOAUTH Authentication required).

Note: The service can be further hardened where possible by restricting network exposure (e.g., binding to localhost).

Result

- Unauthorized access blocked
- Vulnerability removed in follow-up scan

6.2. SMB Signing Not Required

Plugin ID: 57608

Severity: **Medium**

Affected Host: **172.16.1.33 (TCP 445)**

Description

SMB message signing was enabled but not enforced, allowing potential man-in-the-middle attacks.

Evidence (Pre-Remediation)

```
saar@jumpingrook:~$ nmap --script smb2-security-mode -p 445 172.16.1.33
Starting Nmap 7.93 ( https://nmap.org ) at 2026-04-22 11:48 IDT
Nmap scan report for 172.16.1.33
Host is up (0.00063s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
saar@jumpingrook:~$
```

Figure 5: Nmap output shows signing enabled but not required.

Unauthenticated share enumeration:

```
saar@jumpingrook:~$ smbclient -L //172.16.1.33 -N

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (mihai1 server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available
saar@jumpingrook:~$
```

Figure 6: Unauthenticated share enumeration was possible, confirming lack of enforced message integrity.

Remediation

SMB was secured by:

- Setting ‘server signing = mandatory’ in /etc/samba/smb.conf
- Service restarted to apply changes.

Verification (Post-Remediation)

```
saar@jumpingrook:~$ nmap --script smb2-security-mode -p 445 172.16.1.33
Starting Nmap 7.93 ( https://nmap.org ) at 2026-04-22 12:07 IDT
Nmap scan report for 172.16.1.33
Host is up (0.00062s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb2-security-mode:
|   311:
|_    Message signing enabled and required

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
saar@jumpingrook:~$
```

Figure 7: SMB signing is now enabled and required.

While the service remained accessible, enforcing SMB signing ensures integrity of communications and mitigates man-in-the-middle attack risk within the network.

Result

- SMB traffic integrity enforced
- MITM risk significantly reduced

7. Post-Remediation Scan

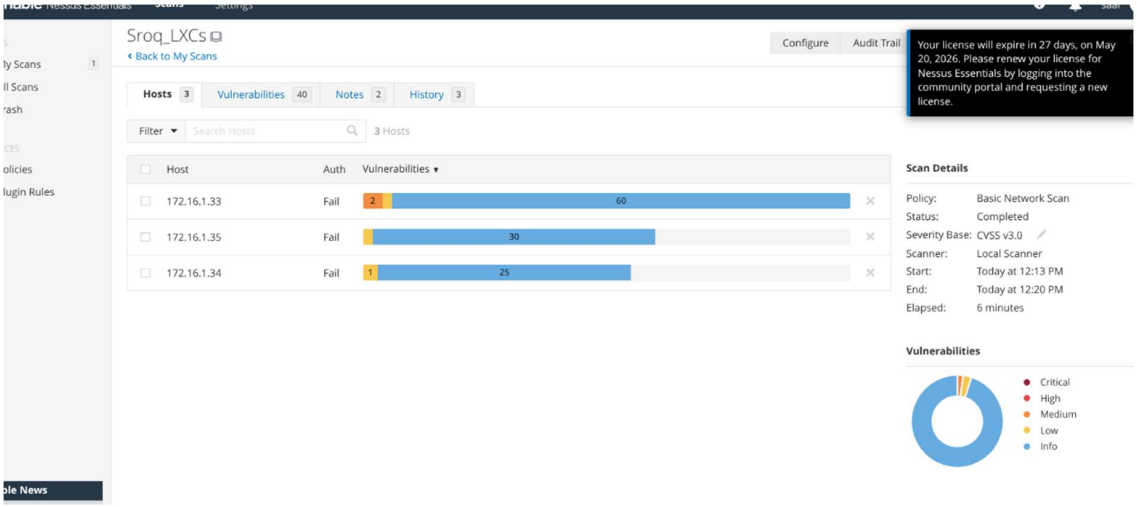

Figure 8a: Follow-up scan, all hosts.

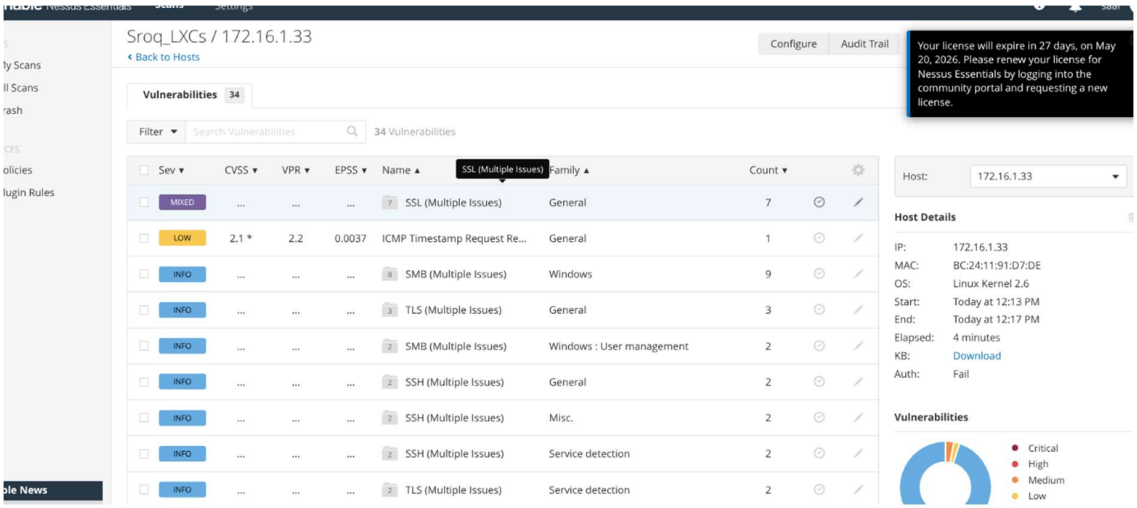

Figure 8b: Follow-up scan showing removal of Redis vulnerability.

Observations

- Redis vulnerability remediated and verified via manual validation
- SMB issue remediated and verified via manual validation
- Remaining findings are low-risk or informational (e.g., SSL self-signed certificates)

8. Conclusion

This assessment demonstrated a complete vulnerability management lifecycle:

- identification through automated scanning
- manual validation of real exposure
- targeted remediation
- verification through re-testing

The removal of the Redis vulnerability eliminated a critical security risk, while enforcing SMB signing improved overall network security posture.

This process highlights the importance of combining automated tools with manual validation and prioritizing high-impact vulnerabilities.

---

9. Appendix

9.1. Redis Validation Output

See full command output:

```
172.16.1.33:6379> INFO

Server

redis_version:6.0.16
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:e91d7986ac6d5bb6
redis_mode:standalone
os:Linux 6.17.13-2-pve x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:11.4.0
process_id:117
run_id:3e839f7816ad08d19c89fcff2b6eff81e10313bf
tcp_port:6379
uptime_in_seconds:81
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:15068013
executable:/usr/bin/redis-server
config_file:/etc/redis/redis.conf
io_threads_active:0

Clients
```

```
connected_clients:1
client_recent_max_input_buffer:8
client_recent_max_output_buffer:0
blocked_clients:0
tracking_clients:0
clients_in_timeout_table:0

Memory

used_memory:873176
used_memory_human:852.71K
used_memory_rss:10743808
used_memory_rss_human:10.25M
used_memory_peak:933800
used_memory_peak_human:911.91K
used_memory_peak_perc:93.51%
used_memory_overhead:830176
used_memory_startup:809680
used_memory_dataset:43000
used_memory_dataset_perc:67.72%
allocator_allocated:1283800
allocator_active:1540096
allocator_resident:3874816
total_system_memory:33265258496
total_system_memory_human:30.98G
used_memory_lua:41984
used_memory_lua_human:41.00K
used_memory_scripts:0
used_memory_scripts_human:0B
number_of_cached_scripts:0
maxmemory:0
maxmemory_human:0B
maxmemory_policy:noeviction
allocator_frag_ratio:1.20
allocator_frag_bytes:256296
allocator_rss_ratio:2.52
allocator_rss_bytes:2334720
rss_overhead_ratio:2.77
rss_overhead_bytes:6868992
mem_fragmentation_ratio:12.93
mem_fragmentation_bytes:9913152
mem_not_counted_for_evict:0
mem_replication_backlog:0
mem_clients_slaves:0
mem_clients_normal:20496
mem_aof_buffer:0
mem_allocator:jemalloc-5.2.1
active_defrag_running:0
lazyfree_pending_objects:0

Persistence

loading:0
rdb_changes_since_last_save:0
rdb_bgsave_in_progress:0
rdb_last_save_time:1776675612
rdb_last_bgsave_status:ok
rdb_last_bgsave_time_sec:-1
rdb_current_bgsave_time_sec:-1
rdb_last_cow_size:0
```

```
aof_enabled:0
aof_rewrite_in_progress:0
aof_rewrite_scheduled:0
aof_last_rewrite_time_sec:-1
aof_current_rewrite_time_sec:-1
aof_last_bgrewrite_status:ok
aof_last_write_status:ok
aof_last_cow_size:0
module_fork_in_progress:0
module_fork_last_cow_size:0

Stats

total_connections_received:1
total_commands_processed:3
instantaneous_ops_per_sec:0
total_net_input_bytes:99
total_net_output_bytes:22442
instantaneous_input_kbps:0.00
instantaneous_output_kbps:0.00
rejected_connections:0
sync_full:0
sync_partial_ok:0
sync_partial_err:0
expired_keys:0
expired_stale_perc:0.00
expired_time_cap_reached_count:0
expire_cycle_cpu_milliseconds:0
evicted_keys:0
keyspace_hits:0
keyspace_misses:0
pubsub_channels:0
pubsub_patterns:0
latest_fork_usec:0
migrate_cached_sockets:0
slave_expires_tracked_keys:0
active_defrag_hits:0
active_defrag_misses:0
active_defrag_key_hits:0
active_defrag_key_misses:0
tracking_total_keys:0
tracking_total_items:0
tracking_total_prefixes:0
unexpected_error_replies:0
total_reads_processed:6
total_writes_processed:5
io_threaded_reads_processed:0
io_threaded_writes_processed:0

Replication

role:master
connected_slaves:0
master_replid:c473fff8e72b861b8122c9e79bf5550ba0ac6b0d
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:0
second_repl_offset:-1
repl_backlog_active:0
repl_backlog_size:1048576
```

```
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0

CPU

used_cpu_sys:0.046643
used_cpu_user:0.038869
used_cpu_sys_children:0.000000
used_cpu_user_children:0.000000

Modules

Cluster

cluster_enabled:0

Keyspace

172.16.1.33:6379> SET test_key 'test'
OK
172.16.1.33:6379> GET test_key
"test"
172.16.1.33:6379>
```

On the VM:

```
root@172.16.1.33:~# cat /etc/redis/redis.conf | grep requirepass


# If the master is password protected (using the "requirepass" configuration

# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility

# requirepass foobared

# So use the 'requirepass' option to protect your instance.

root@172.16.1.33:~#

root@172.16.1.33:~# cat /etc/redis/redis.conf | grep requirepass

# If the master is password protected (using the "requirepass" configuration

# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility

# requirepass foobared

# So use the 'requirepass' option to protect your instance.

root@172.16.1.33:~# sed -i "s/^# requirepass foobared/requirepass
[STRONG_PASSWORD]/" /etc/redis/redis.conf

root@172.16.1.33:~# cat /etc/redis/redis.conf | grep requirepass

# If the master is password protected (using the "requirepass" configuration

# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility

requirepass [STRONG_PASSWORD]

# So use the 'requirepass' option to protect your instance.

root@172.16.1.33:~# sudo systemctl restart redis

root@172.16.1.33:~#
```

AFTER FIX:

```
saar@jumpingrook:~$ redis-cli -h 172.16.1.33

172.16.1.33:6379> INFO

NOAUTH Authentication required.

172.16.1.33:6379>
```

9.2. SMB Validation Output

See validation steps and configuration:

```
saar@jumpingrook:~$ nmap --script smb2-security-mode -p 445 172.16.1.33
Starting Nmap 7.93 ( https://nmap.org ) at 2026-04-22 11:48 IDT
Nmap scan report for 172.16.1.33
Host is up (0.00063s latency).

PORT STATE SERVICE
445/tcp open microsoft-ds

Host script results:
| smb2-security-mode:
| 311:
|_ Message signing enabled but not required

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
saar@jumpingrook:~$

remediation:
added to /etc/samba/smb.conf: server signing = mandatory
```