Arch Linux Installation Script Analysis

Overview

The existing script at `~/Uploads/arch-install.sh` is a comprehensive Arch Linux installation script designed for specific hardware (AMD Ryzen 9 9950X, ASRock X670 Taichi, RTX 5090). It implements an automated installation with Secure Boot, NVIDIA Open drivers, KDE Plasma with Wayland, and Norwegian localization.

Script Structure

The script follows a well-organized modular approach with the following main components:

1. **Configuration Section** (lines 16-22): Hardcoded system parameters
2. **Utility Functions** (lines 24-44): Logging, warnings, errors, and user confirmation
3. **System Checks** (lines 46-58): UEFI and internet connectivity validation
4. **Disk Operations** (lines 70-102): Partitioning, formatting, and mounting
5. **Base Installation** (lines 104-125): Core system and swapfile setup
6. **System Configuration** (lines 127-277): Chroot-based configuration with embedded script
7. **Main Execution Flow** (lines 284-324): Orchestrates the entire installation process

What Works Well

Strong Foundation

- **Error Handling**: Uses `set -e` and proper error checking throughout
- **UEFI Validation**: Ensures system is booted in UEFI mode before proceeding
- **Internet Check**: Validates connectivity before attempting downloads
- **User Confirmation**: Multiple confirmation prompts for destructive operations
- **Colored Output**: Clear visual feedback with color-coded messages
- **Cleanup Handling**: Proper trap for cleanup on script exit

Hardware-Specific Optimizations

- **AMD Microcode**: Includes `amd-ucode` for Ryzen processors
- **NVIDIA Open Driver**: Uses modern `nvidia-open` instead of proprietary driver
- **LTS Kernel**: Stable `linux-lts` kernel choice for reliability
- **Norwegian Localization**: Proper keyboard and locale setup for Norwegian users

Security Features

- **Secure Boot Implementation**: Complete sbctl setup with key generation and enrollment
- **Module Signing**: Automatic signing of NVIDIA kernel modules
- **Pacman Hooks**: Auto-signing hooks for kernel and NVIDIA module updates
- **EFI Binary Signing**: Signs GRUB and kernel binaries

Modern Desktop Environment

- **KDE Plasma**: Full KDE installation with applications
- **Wayland Support**: Includes `plasma-wayland-session` and `xorg-xwayland`

- **Display Manager**: SDDM configuration and enablement
- **Thunderbolt Support**: Includes `bolt` for Thunderbolt device management

Issues and Areas for Improvement

Critical Issues

1. Hardcoded Configuration (High Priority)

- All system parameters are hardcoded (lines 17-22)
- No user prompts for hostname, username, passwords, or disk selection
- Violates the requirement to “prompt for user details”

2. Security Vulnerabilities (High Priority)

- **Hardcoded Root Password**: Uses “password123” as default (line 161)
- **No User Account Creation**: Only sets up root account
- **No sudo Configuration**: Mentions it in post-install but doesn’t implement it

3. Limited Hardware Support (Medium Priority)

- **Fixed Disk Path**: Assumes `/dev/nvme0n1` (line 17)
- **No Wireless Setup**: Missing MediaTek MT7927 wireless configuration
- **No Multi-GPU Support**: Doesn’t handle integrated + discrete GPU scenarios

Missing Features

1. Development Tools (High Priority)

The script lacks the comprehensive development environment requested:
- **Programming Languages**: No Python, Node.js, Rust, Go, etc.
- **Development Tools**: Missing Git, Docker, IDEs, text editors
- **Build Tools**: No make, cmake, gcc toolchain beyond base-devel
- **Version Managers**: No nvm, pyenv, rustup, etc.

2. Essential Applications (Medium Priority)

- **Web Browsers**: No Firefox, Chromium, or other browsers
- **Media Tools**: No multimedia codecs, players, or editors
- **Office Suite**: No LibreOffice or productivity applications
- **System Utilities**: Missing htop, neofetch, file managers, etc.

3. Shell Configuration (Medium Priority)

- **Default Shell**: Uses default bash, no zsh option
- **Shell Customization**: No oh-my-zsh, starship, or other enhancements
- **Terminal Emulator**: Relies on KDE default, no alternatives

Technical Limitations

1. Partition Scheme (Medium Priority)

- **No Swap Partition**: Uses swapfile only (may impact hibernation)
- **Simple Layout**: Only EFI + root, no separate /home partition option
- **Fixed Sizes**: EFI partition size not optimized for multiple kernels

2. Package Management (Low Priority)

- **No AUR Helper**: Missing yay, paru, or other AUR helpers

- **No Flatpak/Snap**: No alternative package managers
- **No Reflector**: Missing mirror optimization

3. Network Configuration (Medium Priority)

- **Basic NetworkManager**: No advanced network configurations
- **No VPN Setup**: Missing VPN client configurations
- **No Firewall**: No UFW or iptables configuration

User Experience Issues

1. Limited Customization (Medium Priority)

- **No Desktop Themes**: Uses default KDE appearance
- **No Font Configuration**: Missing font installations
- **No Icon Themes**: Basic icon set only

2. Post-Install Manual Steps (Medium Priority)

- **Manual User Creation**: Requires manual user account setup
- **Manual sudo Configuration**: User must configure sudo access
- **Manual Password Change**: Relies on user to change default password

Recommended Improvements

High Priority Enhancements

1. **Interactive Configuration System**
   - Add prompts for hostname, username, passwords
   - Disk selection menu with validation
   - Timezone and locale selection options

2. **Comprehensive Package Installation**
   - Development tools and languages
   - Essential applications and utilities
   - AUR helper installation and configuration

3. **Security Hardening**
   - Automatic user account creation with sudo access
   - Secure password generation or prompting
   - Firewall configuration

4. **Hardware-Specific Optimizations**
   - MediaTek MT7927 wireless driver setup
   - Multi-GPU configuration detection
   - Advanced AMD CPU optimizations

Medium Priority Enhancements

1. **Advanced Partitioning Options**
   - Optional separate /home partition
   - Swap partition vs swapfile choice
   - Encryption support (LUKS)

2. **Shell and Terminal Customization**
   - Bash vs Zsh selection

   - Terminal emulator options
   - Shell enhancement frameworks

3. **Network and System Services**
   - Advanced NetworkManager configurations
   - VPN client setup
   - System monitoring tools

Low Priority Enhancements

1. **Desktop Customization**
   - Theme and appearance options
   - Font and icon pack installations
   - Wallpaper and cursor themes

2. **Backup and Recovery**
   - Timeshift or similar backup solutions
   - System recovery tools
   - Configuration backup scripts

Conclusion

The existing script provides a solid foundation with excellent security features and hardware-specific optimizations. However, it requires significant enhancements to meet the user's requirements for interactive configuration, comprehensive development tools, and user-friendly setup. The script's modular structure makes it well-suited for expansion and improvement.

Overall Assessment: Good foundation requiring substantial feature additions and user interaction improvements.