Kron

PAM

3.8.1

Integration Guide

u.trust GP HSM

SecurityServer 6.4.0

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH Krefelder Straße 220 52070 Aachen Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226) EMEA +49 800-627-3081 APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-06-23
Status	PUBLISHED
Document No.	IG-2026-0056

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco u.trust General Purpose (GP) Hardware Security Module (HSM) product can be found in the document directory of the Utimaco SecurityServer product bundle. All Utimaco SecurityServer product documentation is available from Utimaco's website at: https://utimaco.com/.

1.1 About This Guide

This guide describes the integration of Utimaco u.trust GP HSM with Kron Privileged Access Management (PAM) to provide a secure and centralized approach for protecting sensitive cryptographic keys and privileged credentials. It provides an overview of the integration architecture, prerequisites, configuration steps, and validation procedures required to securely deploy Kron PAM with Utimaco HSM.

1.2 Target Audience

This guide is intended for system administrators managing Kron Privileged Access Management (PAM) and Utimaco Hardware Security Modules (HSMs).

1.3 Purpose of the Integration

The purpose of this integration is to combine Kron Privileged Access Management (PAM) with Utimaco Hardware Security Modules (HSMs) to securely manage privileged access and protect cryptographic keys. Utimaco HSMs ensure that keys are generated, stored, and used within a tamper-resistant hardware boundary, preventing exposure and reducing the risks associated with software-based key storage. This ensures that master encryption keys remain protected and never leave the HSM. The integration strengthens overall security, reduces the risk of credential compromise, improves auditability, and supports compliance with standards such as FIPS 140-3. It also enables centralized control of cryptographic operations and enforces strong security practices, including least-privilege access and accountability.

1.4 Abbreviations

Abbreviation	Meaning
HSM	Hardware Security Module
PAM	Privileged Access Management
DEK	Data Encryption Key
JCE	Java Cryptography Extension
PKCS#11	Public-Key Cryptography Standards #11
JVM	Java Virtual Machine
FIPS	Federal Information Processing Standards
IP	Internet Protocol
PIN	Personal Identification Number
HA	High Availability
API	Application Programming Interface
GUI	Graphical User Interface

Table 1: Abbreviations

1.5 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Select **Details** and click on **Properties** button
`Monospaced`	Code that is given for explanation or as an example, file paths	`certreq.exe -new` `request.inf` `IISCertRequest.csr`
Italic	References and important terms	Operating system listed in *Tested Versions*

Table 2: Document conventions

We use special icons to highlight the most important notes and information.

Here you will find important safety information that should be followed.

Here you will find additional notes or supplementary information.

This message marks the result expected after the successful execution of an instruction.

2 Product Overview

2.1 Kron PAM

Kron Privileged Access Management (PAM) is a security solution designed to manage, monitor, and control privileged access across enterprise environments. It enables organizations to securely handle administrative credentials, enforce access controls, and govern privileged sessions across on-premise, cloud, and hybrid infrastructures.

Kron PAM supports key security principles such as least-privilege access, centralized credential management, and controlled access to critical systems, helping reduce the risk of unauthorized access and credential misuse. By providing visibility into privileged activities and enforcing security policies, it helps mitigate insider and external threats while supporting compliance and audit requirements.

The platform also ensures secure storage and management of sensitive credentials and cryptographic material, enabling organizations to maintain strong control over privileged operations and improve the overall security posture of enterprise IT environments.

2.2 Utimaco u.trust GP HSM

Utimaco u.trust GP HSM is a hardware security module developed by Utimaco IS GmbH. It is a physically protected, specialized computer unit designed to perform sensitive cryptographic tasks and securely manage and store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

2.3 Joint Value Proposition

The integration of Kron Privileged Access Management (PAM) with Utimaco Hardware Security Modules (HSMs) delivers a layered, defense-in-depth security model by combining centralized privileged access control with hardware-based key protection. Kron PAM enables organizations to manage and monitor privileged identities, credentials, and access to critical systems with strong governance and visibility, while Utimaco HSMs securely generate, store, and protect cryptographic keys within tamper-resistant hardware boundaries. Together, the solutions strengthen the protection of sensitive credentials and encryption keys, reduce the risk of unauthorized access and key compromise, and support compliance with enterprise and regulatory requirements through centralized control, hardware-rooted trust, and comprehensive audit capabilities.

3 Integration Requirements and Prerequisites

3.1 Tested Versions

Operating System	Kron PAM	Utimaco SecurityServer Version	Utimaco HSM
Redhat 8.10	3.8.1	SecurityServer 6.4.0	u.trust GP HSM CSe-Series/Se-Series

Table 3: List of tested versions

3.2 Hardware and Software Requirements

3.2.1 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	u.trust GP HSM CSe-Series/Se-Series LAN with firmware SecurityServer 6.4.0 or higher
Utimaco PCI-e HSM	u.trust GP HSM CSe-Series/Se-Series PCI-e with firmware SecurityServer 6.4.0 or higher

Table 4: List of hardware requirements

3.2.2 Software Requirements

Software	Software Requirements
HSM Interface	SecurityServer PKCS#11 Provider
Kron PAM	3.8.1

Software	Software Requirements
Java	Java 26 (JDK 26)

Table 5: List of software requirements

3.3 Prerequisites

Before you begin, please ensure that you have:

- Installed and set up the operating system listed in *Tested Versions*.
- Installed and set up the HSM listed in *Tested Versions*.
- Replaced the HSM default admin with a new admin user.
- Created and stored the MBK on each HSM. Refer to the SecurityServer documentation to set up the MBK.
- Set up and configured the SecurityServer. Refer to the SecurityServer documentation to set up the HSM.
- Set up and configured the PKCS#11 library according to your environment. Refer to the SecurityServer documentation for instructions on setting up and configuring the PKCS#11 library.
- Created the Security Officer (SO) user and Crypto user.
- The HSM appliance is powered on and accessible on the network at a known IP address and port (default: 4001–4031).
- A user account exists with sufficient privileges to log in and perform key operations. The username and PIN must be available.
- An AES-256 Secret Key Object has been created on the Utimaco u.trust GP HSM and given a label (alias). This guide uses pam-master-key as the label throughout. Note this value - it will be referenced in multiple configuration files.
- The Utimaco vendor client package has been provided by Utimaco support. This package contains the `libcs_pkcs11_R3.so` shared library and the `CryptoServerJCE.jar` file.

On the Kron PAM server, the following conditions must be satisfied:

- Administrative access to the server with sudo privilege is available.

- All Kron PAM services are stopped. Applying configuration changes while services are running can result in file corruption or mid-operation failures during key migration.
- Network connectivity from the Kron PAM server to the Utimaco u.trust GP HSM appliance has been verified (e.g., via telnet or nc to the HSM port).
- Java is installed and accessible on the server (required to run the .jar tools).

Do not proceed until at least Kron PAM Web UI service is stopped. Run 'systemctl stop pam-gui' (and stop any other Kron PAM service units) before executing any configuration steps below.

3.3.1 Verify the AES Key on the Utimaco u.trust GP HSM

The first action before beginning any PAM configuration is to confirm that the AES key exists on the HSM and note its alias. Log in to the server where the Utimaco PKCS#11 library is accessible and run the following command. Substitute the correct PIN:

```
[UtimacoHSM]# pkcs11-tool --module /usr/lib64/libcs_pkcs11_R3.so --slot-index 0 --
login --pin <YOUR_HSM_PIN> -O
```

The output should include a block similar to the following. Confirm that the key type is AES with a length of 32 bytes (256 bits), and note the label value - this is the alias you will use throughout the configuration:

```
Using slot with index 0 (0x0)
Secret Key Object; AES length 32
  label:      kronpam-master-key
  ID:         01
  Usage:      encrypt, decrypt, verify, wrap, unwrap
  Access:     never extractable, local
```

4 Integration Steps

The following section outlines the procedures required to configure both the Utimaco HSM and Kron PAM components for seamless integration.

4.1 Configuration on Utimaco u.trust GP HSM Se-Series

Create users; SO (Security Officer) and USR (the Crypto user), and initialize a slot.

The slot must be initialized using `p11tool2`.

First, create the SO using `p11tool2`. Then, using the `p11tool2` command, initialize the slot you want to use and the slot user, as shown below.

```
./p11tool2 slot=<slot no.> Label=<token label> Login=ADMIN,ADMIN.key
InitToken=<SO pin>
```

Initialize the SO user.

```
./p11tool2 slot=<slot no.> LoginSO=<SO pin> InitPin=<Cryptouser pin>
```

Ensure the Utimaco GP HSM is accessible from the partner product control panel.

4.2 Configuration on Kron PAM

The configuration is split into three logical phases. The first phase deploys the Utimaco vendor libraries to the correct locations on the Kron PAM server. The second phase creates the configuration files that establish the connection between the PKCS#11 layer and the Utimaco u.trust GP HSM appliance. The third phase registers the HSM with Kron PAM's security subsystem and performs key migration if necessary.

Each step below includes an explanation of what is being done and why, followed by the exact commands or file contents to use.

4.2.1 Deploy the Utimaco PKCS#11 Shared Library

The PKCS#11 shared library is the core software component that bridges the Kron PAM JVM and the Utimaco u.trust GP HSM. It implements the PKCS#11 v2.20 API but translates those calls into the proprietary CryptoServer protocol understood by the Utimaco hardware. Without this library on the Kron PAM server, SunPKCS11 cannot communicate with the device.

Copy the library to the standard shared library location and set the correct permissions. The file must be executable by the system but does not need to be writable by application users:

```
[pamuser@KronPAM_Instance]# cp libcs_pkcs11_R3.so /usr/lib64/libcs_pkcs11_R3.so
[pamuser@KronPAM_Instance]# chmod 755 /usr/lib64/libcs_pkcs11_R3.so
```

4.2.2 Deploy the Utimaco JCE Provider JAR

Java applications use a provider-based model for cryptographic services. The Java Cryptography Extension (JCE) framework discovers providers at runtime and delegates operations to them. Kron PAM uses the SunPKCS11 built-in provider, which itself requires a vendor-supplied backing library to communicate with the specific hardware. The `CryptoServerJCE.jar` file extends this capability with Utimaco-specific classes and ensures that the Utimaco u.trust GP HSM is correctly recognised by the JVM.

Copy the JAR to the Kron PAM library directory and set ownership and permissions so that only the pamuser service account can read it. This limits the attack surface in case of a filesystem-level intrusion:

```
[pamuser@KronPAM_Instance]# cp CryptoServerJCE.jar /pam/kron/lib/
CryptoServerJCE.jar
[pamuser@KronPAM_Instance]# chown pamuser:pamuser /pam/kron/lib/
CryptoServerJCE.jar
[pamuser@KronPAM_Instance]# chmod 600 /pam/kron/lib/CryptoServerJCE.jar
```

Verify that the permissions are set correctly before proceeding:

```
[pamuser@KronPAM_Instance]# ls -ltr /pam/kron/lib/CryptoServerJCE.jar

-rw-------. 1 pamuser pamuser 2637958 2026-05-07 11:21 /pam/kron/lib/
CryptoServerJCE.jar
```

> Do not set this file world-readable (mode 644 or higher). The JCE provider JAR contains code paths that reference HSM credentials. Restricting access to pamuser reduces the risk of credential extraction through classpath inspection or bytecode analysis.

4.2.3 Create the Kron PAM HSM Configuration File

Kron PAM's security subsystem needs to know which PKCS#11 library to load and which slot on that library to use. This is communicated through a small configuration file placed in the Kron PAM security directory. The file is read by the SunPKCS11 provider at startup to initialise its connection to the PKCS#11 layer.

Create the following file at the path shown. If the `/pam/kron/security/hsm/` directory does not yet exist, create it first.

```
[pamuser@KronPAM_Instance]# mkdir -p /pam/kron/security/hsm/
[pamuser@KronPAM_Instance]# touch /pam/kron/security/hsm/utimaco.conf
```

Edit the file and provide the necessary configuration details.

```
name = Utimaco
library = /usr/lib64/libcs_pkcs11_R3.so
slot = 0
```

The three parameters serve the following purposes: name is an arbitrary label used in JVM logging and diagnostics; library is the absolute path to the Utimaco PKCS#11 shared library deployed in Step 1; slot is the PKCS#11 slot index on the HSM that contains the AES key.

4.2.4 Create the Utimaco Global Network Configuration File

The Utimaco PKCS#11 library does not have the HSM's network address hardcoded into it. Instead, it reads a global configuration file at runtime to determine where to find the CryptoServer appliance. This file must be created on the Kron PAM server and its path must be provided via an environment variable before any process that uses the library is started.

```
[pamuser@KronPAM_Instance]# mkdir -p /etc/utimaco
[pamuser@KronPAM_Instance]# mkdir -p /var/log/utimaco
```

Edit file `/etc/utimaco/cs_pkcs11_R3.cfg`. The Device parameter uses the format PORT@IP_ADDRESS.

```
[Global]
Logging = 1
Logpath = /var/log/utimaco
[CryptoServer]
Device = <HSM_PORT>@<HSM_IP_ADDRESS>
```

The Logging and Logpath settings instruct the library to write diagnostic output to `/var/log/utimaco`. This is valuable during initial setup and troubleshooting - if an operation fails, the log will typically identify whether the issue is a connectivity problem, an authentication failure, or a key not found error. In production environments, log rotation for this directory should be configured.

> For high-availability deployments, Utimaco supports specifying multiple devices: 'Device = 3001@10.10.10.10,3001@10.10.10.20. The library will attempt the addresses in order, failing over if the primary is unreachable.

4.2.5 Set the Environment Variable

The Utimaco PKCS#11 library locates its global configuration file by reading the CS_PKCS11_R3_CFG environment variable. This variable must be set in any shell or process that will load the library, including the Kron PAM JVM. Setting it in `/etc/environment` ensures it is available system-wide, including to systemd service units:

```
[pamuser@KronPAM_Instance]# export CS_PKCS11_R3_CFG=/etc/utimaco/cs_pkcs11_R3.cfg
[pamuser@KronPAM_Instance]# echo 'export CS_PKCS11_R3_CFG=/etc/utimaco/
cs_pkcs11_R3.cfg' >> /etc/environment
```

> The export command takes effect in the current shell session immediately. The entry in `/etc/environment` is read by PAM (the Linux Pluggable Authentication Modules, not Kron PAM) at login time and by systemd when it sources the file for service units. A system reboot or a 'systemctl daemon-reload' followed by service restart will pick up the change for the PAM service.

4.2.6 Install OpenSC (Optional)

OpenSC is an open-source package that provides the pkcs11-tool command-line utility. It is not required for the Kron PAM integration itself, but it is extremely useful for verifying HSM connectivity and key availability independent of the Kron PAM application. Being able to confirm that the library can reach the HSM and list the key before attempting the Kron PAM configuration significantly reduces diagnostic complexity.

```
[pamuser@KronPAM_Instance]# dnf install opensc -y
```

Once installed, you can use pkcs11-tool to verify the integration at any point. The command from *Verify the AES Key on the Utimaco u.trust GP HSM* - listing objects on slot 0 - is the primary verification test. If this command returns the expected key listing, all network and library configuration is correct.

4.2.7 Configure security.properties

The file `/pam/kron/security/security.properties` is the central configuration file for Kron PAM's cryptographic subsystem. It controls which key provider is active, how to locate the PKCS#11 configuration, and which key alias to use for the master key. The following block must be added to this file. Each parameter is explained in the table that follows:

```
hsm.provider.classname=sun.security.pkcs11.SunPKCS11
hsm.provider.configPath=/pam/kron/security/hsm/utimaco.conf
hsm.keystore.type=PKCS11
hsm.keystore.alias=pam-master-key
hsm.secretkey.algorithm=AES
hsm.keystore.load.password=
hsm.keystore.entry.password=
```

4.2.8 Run the HSM Master Key Persistence Tool

The HSM Master Key Persistence Tool (`hsm-masterkey-registration.jar`) serves two purposes: it validates that Kron PAM can successfully connect to the HSM and locate the named key, and it encrypts the HSM slot PIN before writing it into `security.properties`. This means the PIN is never stored in plaintext anywhere on the filesystem.

Run the tool as root from the Kron PAM server, pointing it at the `security.properties` file edited in the previous step.

```
[pamuser@KronPAM_Instance]# java -jar hsm-masterkey-registration.jar /pam/kron/
security/security.properties
```

The tool presents an interactive menu. Select option **2-) Update passwords** and enter the HSM keystore PIN when prompted. Both the load password and entry password prompts typically require the same slot PIN. A successful session looks like this:

```
=== HSM Master Key Persistence Tool ===
HSM properties will be loaded from /pam/kron/security/security.properties

Menu:
1-) Save/update master key
2-) Update passwords
0-) Exit

Choose an option: 2
Enter HSM keystore load password: [enter slot PIN]
Enter HSM keystore entry password: [enter slot PIN]

HSM key successfully found with alias pam-master-key
Some parameters updated, overwrite? [y]/n: y
security.properties file updated successfully.

Choose an option: 0
```

The line "HSM key successfully found with alias pam-master-key" is the critical confirmation. It means the tool was able to connect to the HSM over the network, authenticate with the provided PIN, and locate the key object identified by the alias in `security.properties`. If this message does not appear, the most likely causes are incorrect PIN, wrong alias, a network connectivity issue, or an incorrect Device address in `cs_pkcs11_R3.cfg`. Check `/var/log/utimaco/` for detailed error output.

> After the tool exits, security.properties will contain encrypted representations of the passwords. These are specific to the current server and key configuration - do not copy security.properties to another server as-is.

4.2.9 Finalizing the Integration

With the libraries deployed, configuration files created, and passwords registered, the final phase is to activate HSM-based key storage within Kron PAM. The correct procedure depends on whether this is a new installation or an existing deployment. The distinction is important because an existing deployment has DEK records in the database that are currently encrypted under the software-based master key; those records must be re-encrypted under the new HSM key before the software key provider is deactivated.

4.2.9.1 Fresh Installation (No Existing DEK Records)

If Kron PAM has just been installed and the t_dek table in the database is empty, there are no DEK records to migrate. In this case, activating HSM-based key storage is a one-line change to `security.properties`:

```
kron.crypto.keyProvider = hsm
```

Add or update this line in `/pam/kron/security/security.properties`. Once this change is saved, start Kron PAM services and the system will begin generating new DEK records backed by the HSM master key from first use.

4.2.9.2 Existing Installation - Migrating DEK Records

If Kron PAM has been running with a software-based master key, the database contains DEK records encrypted under that key. Simply switching `kron.crypto.keyProvider` to hsm without migrating those records would leave Kron PAM unable to decrypt them - it would look for them in the HSM, which does not have the software key.

The correct process is to use the DEK Rotation Utility (`dek-rotator.jar`) to re-encrypt every DEK record. The tool reads each DEK from the database, decrypts it using the current (software) master key, re-encrypts it using the HSM master key, and writes the updated record back. The process is atomic per record and can be interrupted and resumed.

Run `dek-rotator.jar` only on the master Kron PAM instance. Do not execute it on replica or standby nodes. Running it concurrently on multiple nodes risks database corruption.

Execute the tool, pointing it at the Netright properties file:

```
[pamuser@KronPAM_Instance]# java -jar dek-rotator.jar /pam/gui/netright/
netright.properties
```

The tool displays a warning and then presents a menu. Read the warning carefully - this tool modifies every DEK record in the database. Ensure you have a recent database backup before proceeding. Select option **2-) Migrate master key to HSM** to migrate the master key to HSM:

```
WARNING: BE CAREFUL WHILE USING THIS TOOL. IT WILL UPDATE ALL T_DEK
         RECORDS IN THE DATABASE BY ENCRYPTING THEM WITH THE NEW MASTER KEY.
Menu:
1-) Renew master key
2-) Migrate master key to HSM
3-) Migrate master key from HSM
0-) Exit

Choose an option: 2
Enter current master key: [enter existing software master key]
```

The tool will process all DEK records and confirm when migration is complete. After it exits successfully, update `security.properties` to activate the HSM provider:

```
kron.crypto.keyProvider = hsm
```

5 Verification and Testing

Before starting Kron PAM services, open `security.properties` and confirm that kron.crypto.keyProvider is set to hsm. This is the switch that tells the Kron PAM application to route all master key operations through the HSM. Any other value - such as the default 'file' - means the software key is still in use regardless of all other configuration.

Once confirmed, start Kron PAM Web GUI and other services:

```
[pamuser@KronPAM_Instance]# systemctl restart pam-gui
```

After the service starts, log in to the Kron PAM console and verify that session creation, credential retrieval, and vault access all work correctly. These operations exercise the full key hierarchy - DEK retrieval, decryption via the master key, and data access - and a successful round-trip confirms that HSM integration is working end-to-end.

Check /var/log/utimaco/ after PAM starts. You should see entries corresponding to the key lookup and initial cryptographic operations. Ongoing activity in this log confirms that the PAM application is actively using the HSM rather than falling back to any cached or software-based key material.

6 Troubleshooting

6.1 Rollback Procedure

If the HSM integration must be reversed - for example, due to HSM hardware failure or a network outage that prevents the Kron PAM server from reaching the device - the `dek-rotator.jar` tool provides a **Migrate master key from HSM** option (menu option 3). This operation performs the reverse migration: it re-encrypts all DEK records using a software-based master key, allowing the key provider to be switched back to 'file'.

Before initiating rollback, ensure you have the software master key value available. The tool will prompt for it during the migration. After the migration completes, set kron.crypto.keyProvider back to its original value in `security.properties` and restart Kron PAM services.

7 Contact and Support Information

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH
Krefelder Straße 220
52070 Aachen
Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

- Mail (preferred contact method)
  support@utimaco.com
  Attach the diagnostic information to your email.
- Web portal
  https://support.hsm.utimaco.com/support/cases/new/
  The diagnostic information will be requested in our response if necessary.
- By phone
  AMERICAS +1-844-UTIMACO (+1 844-884-6226)
  EMEA +49 800-627-3081
  APAC +81 800-919-1301
  The diagnostic information will be requested in our response if necessary.

8 Further Information

This document forms a part of the information and support which is provided by the Utimaco IS GmbH. Additional documentation can be found on the product CD in the Documentation directory.

All CryptoServer product documentation is also available at the Utimaco IS GmbH website: https://utimaco.com/.

9 Command Summary

Command used	Purpose
`pkcs11-tool --module /usr/lib64/libcs_pkcs11_R3.so --slot-index 0 --login --pin <HSM_PIN> -O`	Verify AES key presence on HSM
`systemctl stop pam-gui`	Stop Kron PAM GUI service before configuration
`cp libcs_pkcs11_R3.so /usr/lib64/`	Deploy PKCS#11 shared library
`chmod 755 /usr/lib64/libcs_pkcs11_R3.so`	Set execute permissions for PKCS#11 library
`cp CryptoServerJCE.jar /pam/kron/lib/`	Copy Utimaco JCE provider to Kron PAM
`chown pamuser:pamuser /pam/kron/lib/CryptoServerJCE.jar`	Set ownership for JCE provider
`chmod 600 /pam/kron/lib/CryptoServerJCE.jar`	Restrict access to JCE provider
`mkdir -p /pam/kron/security/hsm/`	Create HSM configuration directory
`mkdir -p /etc/utimaco`	Create Utimaco config directory

Command used	Purpose
`mkdir -p /var/log/utimaco`	Create log directory for HSM library
`export CS_PKCS11_R3_CFG=/etc/utimaco/cs_pkcs11_R3.cfg`	Set environment variable for HSM config
`dnf install opensc -y`	Install OpenSC tools (optional verification)
`java -jar hsm-masterkey-registration.jar /pam/kron/security/security.properties`	Register HSM key and encrypt PIN
`java -jar dek-rotator.jar /pam/gui/netright/netright.properties`	Migrate DEK records to HSM
`systemctl restart pam-gui`	Restart Kron PAM service after configuration

Table 6: List of commands used