**REPORT OF PROJECT**

**Use any available open-source software to generate a report of your system to check how many new software is installed, how many are deleted, and how many updates have been done in the last 3 months.**

1. **INTRODUCTION:-**

The new software installations, deletions, and updates is critical for maintaining the stability, security, and performance of a computer system. Being able to generate reports on these changes can provide valuable insights for system administrators or IT personnel to ensure compliance with software policies, identify potential issues, and troubleshoot problems. In this modern age, various open-source tools are available to facilitate the generation of such reports, allowing for efficient monitoring of software changes in a system over a specified period, such as the last 3 months.

With the use of open-source software, system administrators can access tools that interact with the system's package manager, which is responsible for managing software installations and updates. These package managers maintain metadata about software packages, including information such as installation dates, versions, and statuses. By leveraging open-source software, system administrators can retrieve and analyze this metadata to generate comprehensive reports on new software installations, deletions, and updates that have occurred within a specified time frame.

**1.1 Objective of the project:-**

1. Tracking new software installations: The system should be able to identify and report on all new software installations that have occurred within the last 3 months. This includes capturing relevant metadata such as the name of the software, version number, installation date, and other relevant details.
2. Monitoring software deletions: The system should be able to detect and report on any software that has been deleted or uninstalled from the system during the specified period. This information can be useful for identifying potential security risks or compliance issues.
3. Recording software updates: The system should be able to track and report on any software updates or patches that have been applied to the system within the last 3 months. This includes capturing details such as the updated version number, date of the update, and any relevant information about the changes made.
4. Generating comprehensive reports: The system should be capable of generating comprehensive reports that provide an overview of all the new software installations, deletions, and updates that have occurred within the specified timeframe. These reports should be easy to understand and provide relevant information for system administrators or IT personnel to effectively manage software changes.
5. Ensuring compliance and security: The system should aid in ensuring compliance with organizational policies, licensing agreements, and security best practices by providing accurate and up-to-date information on software installations, deletions, and updates. This can help identify any unauthorized software installations or potential security risks.
6. Supporting system management: The system should provide insights into the overall management of the system's software inventory, helping system administrators to keep track of changes, troubleshoot issues, and maintain a stable and secure computing environment.

**1.2 Description of the project:-**

A system to check new software installations, deletions, and updates in the last 3 months typically involves several components working together to collect and analyze data related to software changes in a computer system. Here is a high-level description of such a system:

1. Data Collection: The system collects data from various sources, such as the system's package manager, log files, or other relevant sources. The package manager is responsible for managing software installations, updates, and deletions, and maintains metadata about software packages, including installation dates, versions, and statuses.
2. Data Processing: The collected data is processed to extract relevant information about new software installations, deletions, and updates. This may involve parsing and analysing log files, querying package manager databases, or other methods of data extraction and processing.
3. Reporting and Visualization: The processed data is then used to generate comprehensive reports that provide insights into the software changes that have occurred in the last 3 months. These reports may include details such as the names of the installed, deleted, or updated software, version numbers, installation or deletion dates, and other relevant information. The reports may also be visualized using graphs, charts, or other visual aids to facilitate understanding and analysis.
4. Compliance and Security Checks: The system may include checks to ensure compliance with organizational policies, licensing agreements, and security best practices. For example, it may verify that all software installations are authorized and compliant with licensing requirements and identify any potential security risks associated with software changes.
5. Alerts and Notifications: The system may generate alerts or notifications to inform system administrators or IT personnel of significant software changes, such as unauthorized software installations or critical updates. This can help ensure prompt actions to address any issues or discrepancies.
6. User Interface: The system may include a user-friendly interface that allows system administrators or IT personnel to interact with the system, view reports, configure settings, and perform other relevant tasks related to monitoring and managing software changes.
7. Data Storage and Retention: The system may store collected data for a specified period, allowing for historical analysis and trend identification. Data retention policies and privacy considerations should be considered as per organizational requirements.
8. Scalability and Flexibility: The system should be designed to handle a large volume of data, be scalable to accommodate growing software inventory, and flexible enough to adapt to different environments and configurations.

**1.3 Scope of the project:-**

1. Software Inventory Management: Implement a software inventory management system that keeps track of all software installed on your system. This can be a database or a specialized software that automatically scans and records all software installations, deletions, and updates.
2. Logging and Auditing: Enable logging and auditing on your system to capture relevant events related to software installations, deletions, and updates. This can include system logs, event logs, or application logs that record activities related to software changes.
3. Configuration Management: Utilize a configuration management tool, such as Chef, Puppet, or Ansible, which can help you automate software installations, deletions, and updates on your system. These tools often come with built-in tracking and reporting features that can provide information on changes made to the system.
4. Version Control System: If you are using a version control system, such as Git, you can keep track of changes made to software installations, deletions, and updates by maintaining a repository for your system's configuration files. This allows you to track changes over time and see what software has been added, removed, or updated.
5. Monitoring and Reporting Tools: Utilize monitoring and reporting tools, such as system monitoring software, log analyzers, or security information and event management (SIEM) systems, to generate reports or alerts based on software installation, deletion, and update events. These tools can provide insights into changes made to your system's software environment.

**System description**

**2.1 Target system description:-**

Target system uses a package manager, such as apt, yum, or pacman on Linux, or the Windows Installer (MSI) on Windows, you can review the package manager logs or history to track software installations, deletions, and updates. These package managers often maintain logs or history of software changes, which can provide insights into the software modifications made on the system.

**3. Analysis Report**

**3.1 System snapshots and full analysis report**

Open Workstation and login to kali.

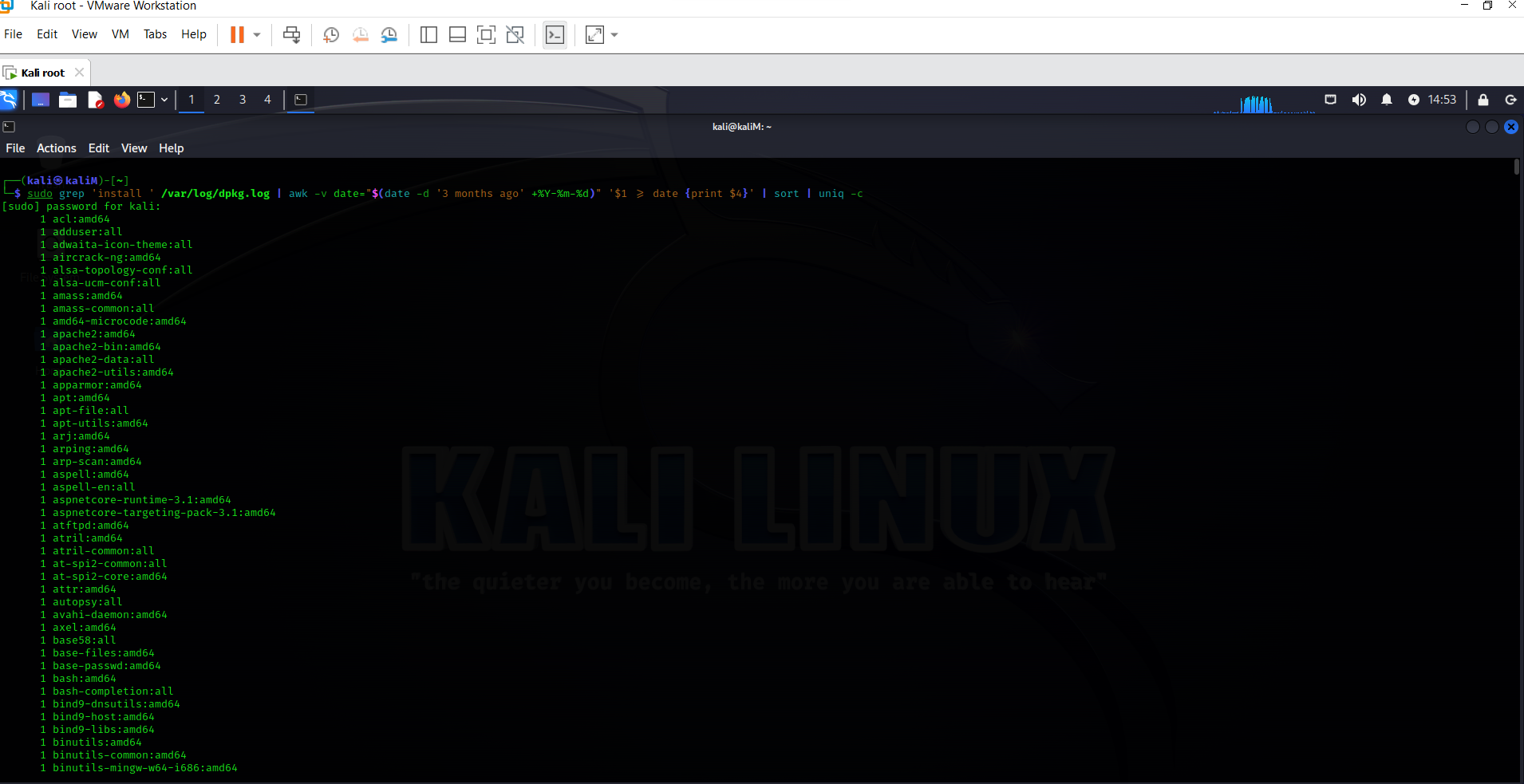
1. Open a terminal window in Kali Linux.
2. Run the following command to update the package lists and upgrade the installed packages to their latest versions:

Graphical user interface

Description automatically generated

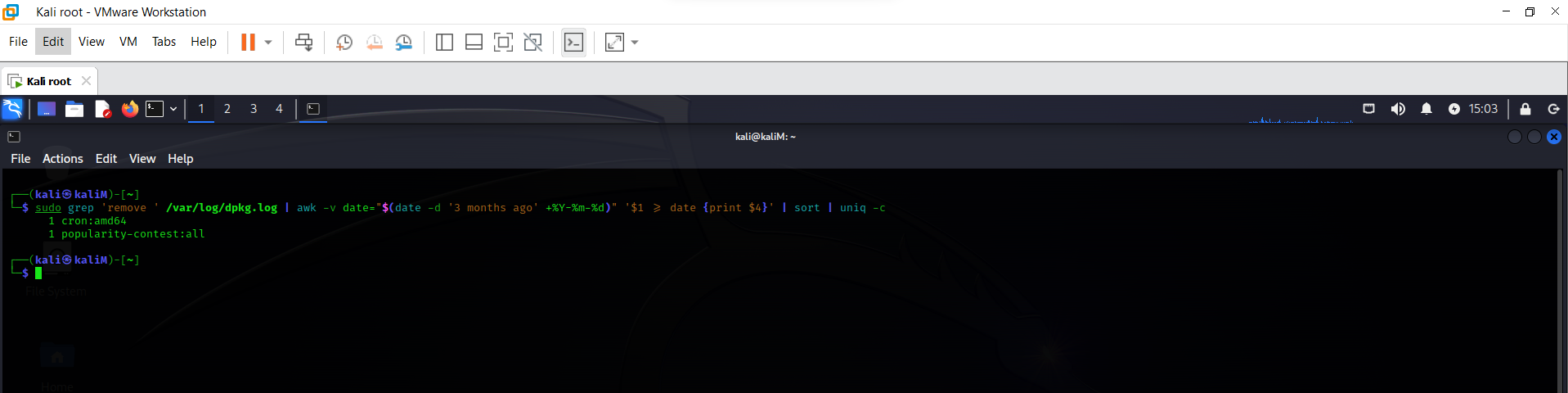
**To check newly installed packages**

**Type the command in terminal and check new installed packages.**



1. **To check deleted packages:**

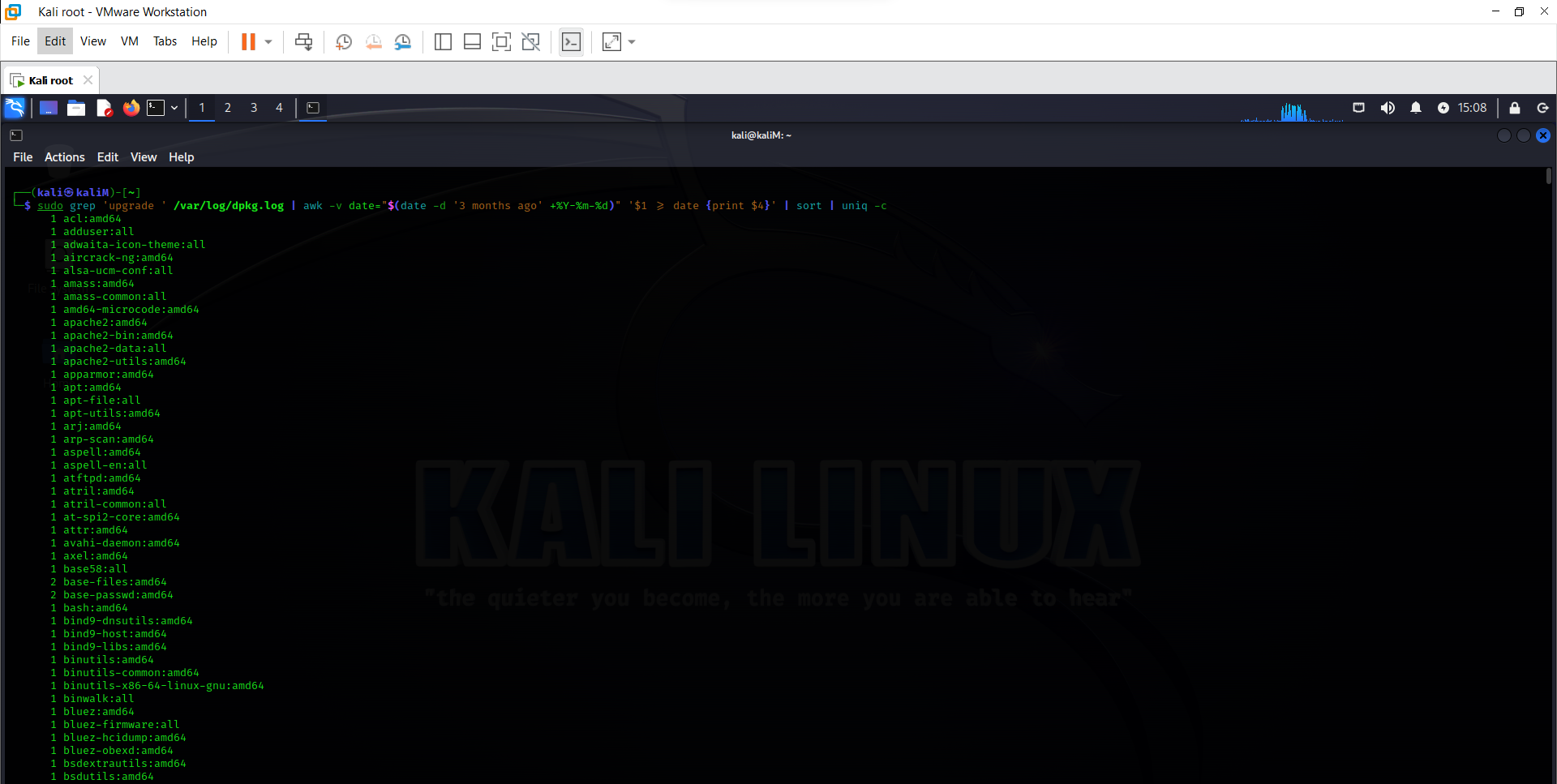
**Type the command and check the deleted packages.**

****

Top of Form

1. **To check updated packages:**

**Type the command in terminal and check updated packages.**

****

Bottom of Form

**Reference/Bibliography**

We can use open-source software such as package managers like APT (Advanced Package Tool) for Linux-based systems, PowerShell for Windows-based systems, or tools like Homebrew for macOS-based systems. These tools provide commands or APIs to list installed software, check for updates, and track changes in the software.

**Github link- https://github.com/sheshank367/Int301CA3**

Top of Form

Bottom of Form

.