В настоящее время активно идет процесс внедрения в деятельность органов государственной охраны информационных систем на базе ОС Astra Linux. Данная ОС применяется на серверах и абонентских пунктах сети ведомственной сети передачи данных. Кроме того, ОС Astra Linux внедряется и в учебный процесс академии. Для обеспечения эффективного и безопасного функционирования ИС на базе Astra Linux требуется обеспечить качественную настройку и конфигурирование ПО и самой ОС.

Одним из важных элементов обеспечения информационной безопасности является **управление конфигурациями**. Оно позволяет контролировать текущее состояние системы и выбирать будущее. Управление конфигурациями обеспечивает непротиворечивость, полноту и точность ОИБ. Также, оно включает в себя механизм определения текущего состояния ИБ и средства оценки воздействия на это состояние.

Назначение управления конфигурациями – предоставлять точные и актуальные данные о состоянии активов, сохранять требуемое состояние, анализировать и контролировать изменения в выделенных конфигурационных единицах.

Цель процесса УК – сохранить контроль над установленными конфигурациями элементов системы.

Основные функции процесса УК:

1. Планирование УК;
2. Идентификация конфигураций;
3. Контроль и управление изменениями конфигурациями;
4. Учет состояния конфигураций;
5. Взаимодействие и обмен данными о конфигурации.

Для осуществления процесса управления конфигурациями применяются системы управления конфигурациями.

Система УК – это программный комплекс, обеспечивающий автоматизацию процесса УК.

В зависимости от масштабов системы УК, она может не включать некоторые функции процесса УК, перенося ответственность их выполнения на администратора.

В настоящий момент, в составе Astra Linux не имеется системы управления конфигурациями. Изменение конфигураций машин под управлением данной ОС осуществляется вручную локально или удаленно. Отсюда выходят следующие недостатки:

1. Для применения какого-либо набора требуемых изменений на большом количестве машин требуется большое количество времени, что в некоторых случаях является критичным;
2. Возможность совершения ошибок администратором.

Из вышесказанного можно вывести следующую цель создания системы УК для ОС Astra Linux:

* Повышение оперативности процесса управления конфигурацией серверов и рабочих станций;

Ограничения разрабатываемой системы:

1. На рабочих станциях и серверах применяется Astra Linux;
2. Управление конфигурацией должно осуществляться удаленно;
3. Для осуществления требуемых изменений необходимо знать административный пароль.

Требования к разрабатываемой системе определяются критериями оперативности, результативности и ресурсоемкости.

Сформулируем критерий оперативности.

Пусть – множество требуемых наборов конфигураций, – некоторый набор конфигураций, входящий в множество требуемых конфигураций.  *-* функция применения набора конфигураций на множестве машин.

Критерий оптимальности определяется тем, что время применения набора конфигурации не превышает некоторое допустимое время, т. е.:

Так как процесс применения конфигурации на целевых машинах является случайным процессом, то критерий оперативности преобразуется к следующему виду:

Вероятность того, что время применения набора конфигураций не превысит допустимое время, не меньше некоторого значения.

При рассмотрении применения данной системы в учебном процессе Академии допустимое время определяется длительностью перерыва между учебными занятиями, в течение которого возможно изменение конфигурации. Вероятность = 0.9.

Критерий результативности определяется как отношения числа успешно примененных наборов конфигураций к общему числу наборов.

Исходя из раннее описанных ограничений определим требования к системе УК:

* Возможность удаленного управления целевыми машинами
* Возможность группового применения конфигураций
* Осуществление контроля за результатом применения конфигурации
* Возможность изменения конфигурации в соответствии с некоторым планом, расписанием

Предлагается следующая архитектура системы УК: ----схема

Управляющий и веб-сервера могут быть объединены в одной машине.

Уже существует несколько программных продуктов, решающих задачи управления конфигурацией. Все они имеют общий принцип построения, но у каждого имеются свои особенности. Сравнение существующих программных продуктов представлено:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Лицензия | Язык программирования | Архитектура | Язык написания сценариев | Наличие GUI |
| Ansible | GPL | Python | Без агента | YAML | Да (проприетарное решение) |
| Сhef | Apache | Ruby, Erlang | Клиент-Серверная | Собственный | Нет |
| CFEngine | Проприетарная | С | Клиент-Серверная | Собственный | Да (проприетарное решение) |
| Puppet | GPL, Apache | Ruby | Клиент-Серверная | Собственный | Нет |
| SaltStack | Apache | Python | Клиент-Серверная | YAML | Да |