# Сокращения

АРМ — Автоматизированное рабочее место

АРМ АБИ — Автоматизированное рабочее место администратора безопасности информации

СВТ — средство вычислительно техники

СЗИ — средства защиты информации

НСД — несанкционированный доступ

КЦ — контроль целостности

ОС СН — операционная система специального назначения

ALD — Astra Linux Directory

# Задание от заказчика

| № | Пункт технико-тактических требований | Как реализовано в АРМ АБИ | Чем можно заменить/Наличие аналогичного функционала в другом ПО | Примечание |
|---|---|---|---|---|
| 1 | Удаленная настройка средств защиты информации | Использование web-интерфейса для удаленного управления данными домена, антивируса, сервера OSSEC и т.д. | Центр управления антивируса dr.Web;<br>Графический интерфейс управления ALD в Astra Linux. | |
| 2 | Блокирование/разблокирование с СВТ работы пользователя, программ и устройств Изделия в случае обнаружения попыток, фактов НСД | Учетные записи блокируются/разблокируются при помощи вкладки Пользователи > Блокировать/разблокировать. Блокирование и разблокирование программ при помощи | Блокирование и разблокирование учетных записей пользователей осуществляется при помощи графического интерфейса домена Astra Linux. Доступ к ресурсам осуществляется штатными средствами Astra Linux. В домене | |

| | | | | |
|---|---|---|---|---|
| | | интерфейса «Доступ к ресурсам», в котором устанавливаются права на доступ к конкретным программам. Блокирование/разблокирование устройств в АРМ АБИ обнаружено не было, устройства добавляются и удаляются автоматически, по мере ввода клиентов в домен. | нет удаленного блокирования устройств, есть только удаленная блокировка учетных записей также как и а АРМ АБИ. | |
| 3 | Просмотр и печать (при необходимости) журналов подсистемы регистрации и учета. | Реализовано во вкладке «Журналы». Происходит сбор логов с сервера обнаружения вторжения OSSEC. При открытии вкладки выбранного журнала, имеется кнопка вывода на печать. | Просматривать, например, в /*var*/log/auth.log | **ВАЖНО**! На Astra Linux v1.6 не реализован механизм OSSEC (его нет даже в дистрибутиве). Сбор логов может осуществлятся вручную из файлов логов. |
| 4 | Ведение таблицы разграничения доступа (с возможностью документирования) пользователей, их прав и полномочий | В АРМ АБИ подобной функции не обнаружено | Разграничение доступа пользователей, их прав и полномочий находится в управлении доменом ALD Astra Linux | |
| 5 | Корректировку параметров идентификации и полномочий прав доступа к защищаемым ресурсам | Реализовано во вкладке Устройства > Аудит ресурсов. Реализуется при помощи стандартных средств Astra Linux. | Реализовано в локальных и доменных учетных записях ALD. Можно разграничивать при помощи стандартных средств Astra Linux: ACL (контроль доступа к ресурсам) и мандатного разграничения доступа. | |
| 6 | Генерацию, установку и смену паролей доступа пользователям с использованием программы | Реализовано во вкладке Пользователи > Пользователь > Сменить пароль. Есть | Генерацию пароля можно произвести штатными средствами Astra Linux, например pwgen. | В Astra Linux 1.6 нет штатного pwgen, можно использовать apg |

| | | | | |
|---|---|---|---|---|
| | генерации пароля | возможность генерации, установки и смены пароля, установка длины пароля и вывода на печать. | Установка паролей производится вручную. | |
| 7 | Формирование и печать списка пользователей с соответствующими им заблаговременно сгенерированными паролями | Формирование и печать списка пользователей реализовано во вкладке Пользователи > Просмотр списка паролей. | Просмотр списка пользователей ALD можно осуществлять при помощи графического интерфейса ALD или командой ald-admin user-list. Просмотр списка паролей (**СОМНЕВАЮСЬ**) | |
| 8 | Стирание защищаемой информации на АРМ изделия по команде АБИ | Вкладка Устройства > Стирание ЗИ. Также имеется кнопка выбора файла со списком защищаемой информации | | |
| 9 | Постановка/снятие на контроль целостности конпонента ПО (каталогов, файлов) Изделия с АРМ АБИ | Реализовано во вкладке КЦ > Пользователь > Компоненты КЦ | Контроль целостности ОС СН Astra Linux можно реализовать в интерфейсе домена ALD | АРМ АБИ использует стандартную утилиту Linux afick для проверки контроля целостности и сбора логов |
| 10 | Отображение и документирование результатов контроля целостности ПО с указанием элементов подвергшихся изменению и характера изменений | Отображение и документирование результатов контроля целостности ПО реализовано во вкладке КЦ > Пользователи | В Astra Linux процедура документирования резельтатов КЦ реализована при помощи утилиты afick | |
| 11 | Проведение антивирусной проверки ПО всех АРМ Изделия с АРМ АБИ по команде администратора | В АРМ АБИ данная функция реализована в связке с Kaspersky Endpoint Security (работает только в Astra Linux 1.5) во вкладке Антивирус | Функция проведения антивирусной проверки реализована в Dr.Web Desktop for Linux. Запуск удаленной проверки осуществляется при помощи Центра Управления Dr.Web | |
| 12 | Отображение и документирование результатов антивирусной проверки | Реализовано во вкладке Антивирус | Функция реализована в Центре Управления Dr.Web | |

| | | | | |
|---|---|---|---|---|
| | ПО с указанием элементов подвергшихся заражению. | | | |
| 13 | Создание резервных копий действующих параметров средств защиты информации, а также рабочих копий машинных носителей информации (CD, DVD, USB и т. д.) | Создание резервных копий действующих параметров СЗИ реализовано во вкладке Резервное копирование ALD > Создать резервную копию | ? (Создание backup для домена ALD) | |
| 14 | Тестирование работоспособности средств защиты информации | Реализовано во вкладке Тестирование > Запустить тестирование СЗИ. Запускает скрипт audit_file.sh | Проверка производится запуском скрипта audit_file.sh (находится в /*usr*/lib/parsec/tests/audit_file.sh) | Тестирование производится на наличие/отсутствие ошибок |

# Требования к модулю

## *. web-приложение

Делаем Qt приложение (сервис или демон — неважно, главное с правами **sudo**) основаное на библиотеке **Qt 5.9.7**, **QtWebApp 1.7.8**, **PostgreSQL**.

Что касается FrontEnd части — **HTML5**, **CSS3**, **JS**. Возможно **jQuery**, **VueJS** — определим в процесса разработки, пока без них

## *. Основное название

Мониторинг и управление удаленными компьютерами в сети (под управлением AstraLinux 1.6).

## 1. Удаленная настройка средств защиты информации

Основные инструменты управления АРМ с сервера scp и ssh для передачи пакетов на АРМ и запуска процессов на АРМ соответственно. Для работы необходим открытый порт 22, а также обмен ключами сервер - АРМ (ssh-key).

### 1.1. установка антивируса на удаленную машину

а) Копируем пакет с Dr.Web или Kaspersky на АРМ (scp -r admin@armN.company.ru:/tmp);

б) Переходим на АРМ

в) Запускаем установку пакета через expect (для автоматизации процесса — все ответы ДА, НЕТ,…, только ести нет quiet режима) с заданными параметрами, которые нужно уточнить у заказчика

г) Для установки kaspersky необходимо выполнить `dpkg -i kesl_1x.x.xxxx.deb`

### 1.2. удаленный запуск антивируса с указанием конкретных каталогов для проверки)

Для удаленного запуска Dr.Web смотри команду `drweb-ctl remotescan (docs/07/09)`

Для местного запуска (непосредственно на АРМ) `drweb-ctl (docs/07/09)`

Для запуска задачи kaspersky `kesl-control [-T] --create-task <имя задачи> --type <тип задачи>`

### 1.3. проведение антивирусной проверки ПО всех АРМ по команде администратора

Собираем все подключенные АРМ к серверу ALD командой `ald-admin host-list` (либо поиском в локальной сети `nmap -sn 10.10.0.1/24`)

Выполняем те же действия, что и в п. 1.2, только в отдельном процессе с ожиданием завершения

### 1.4. отображение и документирование результатов антивирусной проверки ПО с указанием элементов, подвергшихся заражению.

Для Dr.Web необходимо выполнить команду `drweb-ctl threats`

Для Kaspersky необходимо выполнить

2. Блокирование/разблокирование работы пользователя, доступов к конкретным программам, блокирование/разблокирование устройств (флэшек, DVD) в случае обнаружения попыток или фактов несанкционированного доступа.

### 3. Просмотр и печать (при необходимости) журналов подсистемы регистрации и учета

Передача на АРМ АБИ всего вывода из файла журнала `/var/log/auth*.log`

### 4. Архивация журналов регистрации за необходимый промежуток времени;

Нужно сделать выборку из журналов `/var/log/auth*.log` и поместить их в архив `tar cfz auth_arch_`date`.tgz` /var/log/auth*.log

## 5. Интеграция с ALD

### 5.1. ведение журнала учета защищаемых ресурсов (каталогов, файлов, программ);

В руководстве администратора предлагается использовать Zabbix, нужно более точная информация по качеству журналов.

Для работы с Zabbix есть хорошее API через application/json, но Zabbix должен быть установлен и настроен в системе Astra Linux 1.6.

### 5.2. ведение таблицы разграничений доступов (с возможностью документирования) пользователей, их прав и полномочий;

Сначала берем список пользователей ald-admin user-list

Затем выбираем всю доступную информацию о пользователях последовательно командой `ald-admin user-get username`

### 5.3. корректировка прав доступа к защищаемым ресурсам;

Осуществляется командой `ald-admin user-ald-cap`

### 5.4. генерация, установка и смена паролей доступа пользователям.

Для установки/смены пароля используется команда `ald-admin user-passwd username`

Затем через `expect` и `apg `parameters`` можно задать пароль и повторить его

## 8. Формирование и печать списка пользователей с соответствующими им заблаговременно сгенерированными паролями.

Все пароли и логины находятся в файле `/etc/shadow`, но пароли в зашифрованном виде. Можно сохранять пароли при вводе в БД, но это дыра в безопасности.

## 9. Удаление и копирование защищаемой информации на удаленных компьютерах.

Переходим в АРМ и выполняем команду(ы) `ssh -l admin` `` `ARM-name` `` `'command'`

## 10. Создание резервных копий действующих параметров средств защиты информации, а также рабочих копий машинных носителей информации. (CD, DVD и т.п.).

Необходимо использовать команду типа `sudo tar czf /backup.tar.gz --exclude=/backup.tar.gz --exclude=/home --exclude=/media --exclude=/dev --exclude=/mnt --exclude=/proc --exclude=/sys --exclude=/tmp /`

Где exclude — игнорируемые папки

## 11. Постановка/снятие на контроль целостности компонентов программного обеспечения (каталогов, файлов).

Для управление целостностью компонентов необходимо править файл конфигурации `/etc/afick.conf`

Затем необходимо выполнить `afick -i` для создания БД с контрольными суммами

## 12. Отображение и документирование результатов контроля целостности программного обеспечения с указанием элементов подвергшихся изменению и характера изменений.

Для отображения журнала необходимо выполнить `afick --stat-secu`

### 13. Тестирование работоспособности средств защиты.

Необходимо перейти в АРМ и выполнить команду `ssh -l admin `ARM-name` '/usr/lib/parsec/tests/audit_file.sh'` полсле чего нужно проанализировать информацию на предмет сообщений **PASS** — проверка безопасности успешно пройдена и **FAIL** — проверка не прошла

## Реализация

### 1. Валидация админа без https

а) От АРМ АБИ при вводе логина и пароля сначала на сервер отправляется запрос **SALT** по логину

б) Если на сервере есть в `/etc/shadow` такой пользователь, то в ответ присылается **SALT** вида **$x$xxxxxxxx$** или 0

в) АРМ АБИ формирует **HASH** сумму для данного логина, пароля и **SALT** и отправляет на сервер

г) Сервер проверяет хэш на совпадение с данными в `/etc/shadow`

```
python3 -c 'import crypt; print(crypt.crypt("password", "$6$SALT$"))'
```

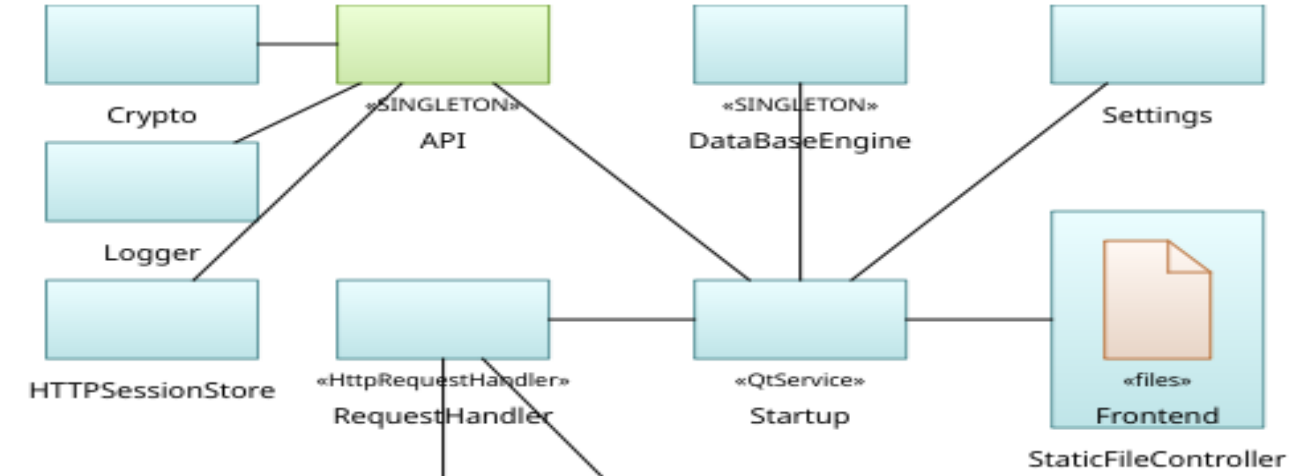
#### 1.2. Проверка пароля внутри приложения

в unix-like системах проверка пароля осуществляется библиотекой libcrypt

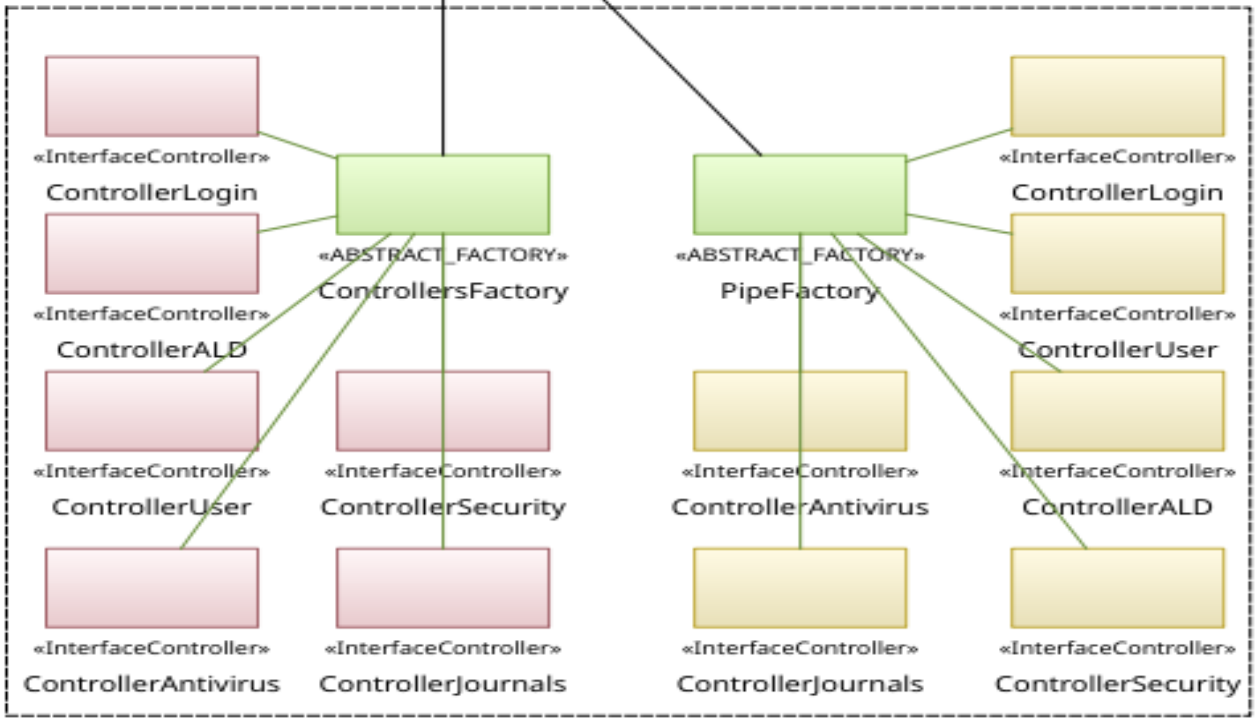
примерный код для моей системы выглядит так

```
const char * Password_str = crypt(«PASSWORD», «$X$SALT$»);
```
// где X — тип шифрования (6 - SHA512)

## 2. Предварительная архитектура ПО




**Базовая сервисная часть.
Обеспечивает доступ Controller'ов к
coocke, базе данных, настройкам.
А также простой веб-сервер.**




**Обработка команд
администратора через
Web интерфейс или
через pipe на стороне сервера.
Единый интерфейс для
обработки запросов.
Каждый Controller создается
исходя из path запроса,
Например: /login -
создается экземпляр LoginController
и параметры request и response
передаются в обработчик**

## 2.1. Основные положения

Все приложение построено на базе **Qt** и **QtWebApp**, а соответственно только динамические библиотеки (ограничение **LGPL v3**) с распространением через прогон на **ldd** для сборки конечного приложения. Для нормальной работы с **PostgreSQL** возможно потребуется сделать свою сборку **Qt** или же использовать **libpg**, возможно с какой-нибудь оберткой на C++.

Pipe контроллеры нужны для ускорения разработки сервисной части — создается файл **/tmp/osds_command.pipe** в который в режиме реального времени можно руками внести команду вида **«/login $USER $PASSWORD; /ald $USER delete;...»** и сервис ее обработает также как команду от АБИ АРМ. После считывания набора команд сервис должен удалить файл.

## 3. Frontend мысли

## Итог (мотиватор)

Код не должен превратиться в полное говно