# Аналог AdaptixC2 на Java

Ботнет — это сеть компьютеров, которые управляются хакерами удаленно.

Ботнеты используются преступниками для распространения программ-вымогателей на ваш ноутбук, телефон, планшет, компьютер. Они могут быть не обнаружены даже антивирусом, и вы можете долгое время не догадываться, что ваше устройство является частью ботнета.

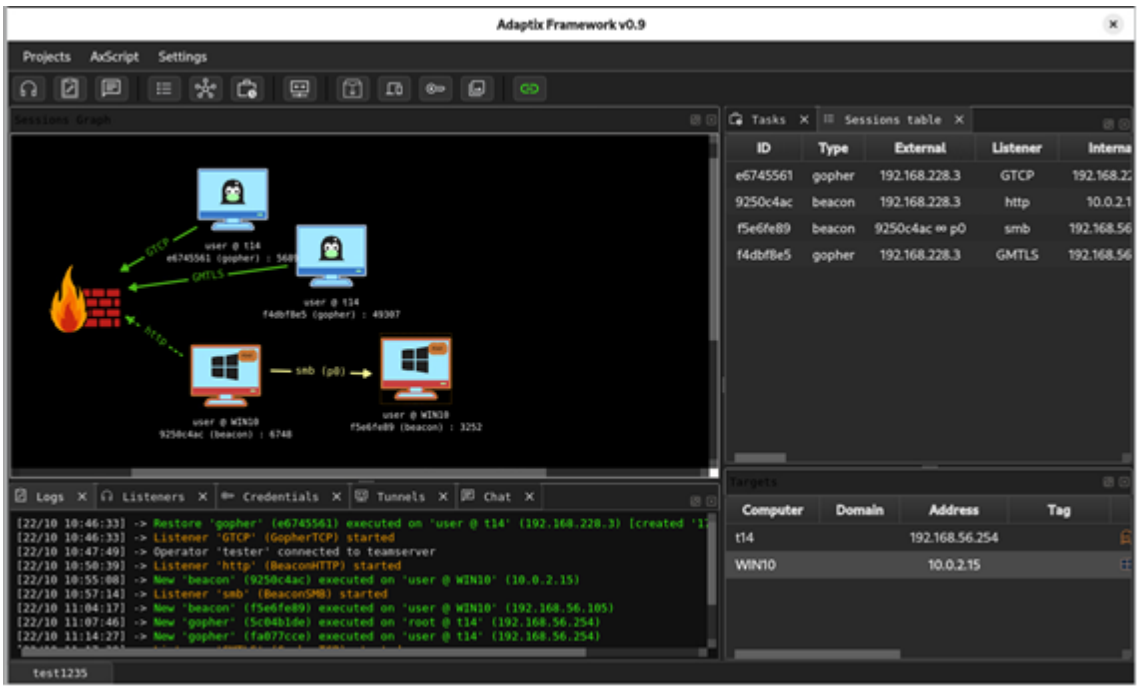
В следующем проекте будет реализован интерфейс оператора для управления ботами, и сам payload, который будет поддерживаться на всех популярных ОС(MacOS, Windows, Linux).

Об существующем аналоге, который уже реализован:

AdaptixC2 — это фреймворк для постэксплуатации, который часто сравнивают с такими известными инструментами, как Cobalt Strike и Brute Ratel. В отличие от них, AdaptixC2 полностью бесплатен и доступен на GitHub.

Публичная версия AdaptixC2 появилась в августе 2024 года, одновременно с созданием канала Adaptix Framework. Фреймворк активно развивается, постоянно расширяются возможности агентов и сервера. 28 сентября 2025 года вышла версия 0.9, 29 августа — версия 0.8. Все обновления подробно описываются в документации и на канале разработчика.

AdaptixC2 сразу привлёк к себе внимание своим открытым исходным кодом, продуманной структурой, кросс-платформенным интерфейсом и богатым возможностями по доработке.




Интерфейс клиента написан на Qt6 и в настоящее время может быть скомпилирован под Windows / Linux / Mac OS. Сервер может быть запущен под Linux Ubuntu / Debian / Arch.

В настройках сервера указывается, какие URI прослушивать и на каком порту, а также какую страницу отображать при ошибке аутентификации. Всё взаимодействие с сервером по умолчанию происходит по HTTPS на сертификатах, сгенерированных при настройке. Также можно разнести прослушивающие интерфейсы сервера и самих сервисов для подключения агентов. В совокупности весь этот комплекс мер позволяет скрыть ключевые элементы AdaptixC2 с радаров публичных сканеров (Shodan, Censys, ZoomEye и т.д.) и минимизировать проактивное обнаружение  инфраструктуры фреймворка.

## Агенты

В терминах фреймворка AdaptixC2 агент — это программный компонент, который устанавливается на скомпрометированную машину и обеспечивает связь между этой машиной и инфраструктурой управления злоумышленника (Command and Control-сервером). Агент выполняет команды, отправляемые с управляющего сервера, позволяет передавать данные, запускать удалённые

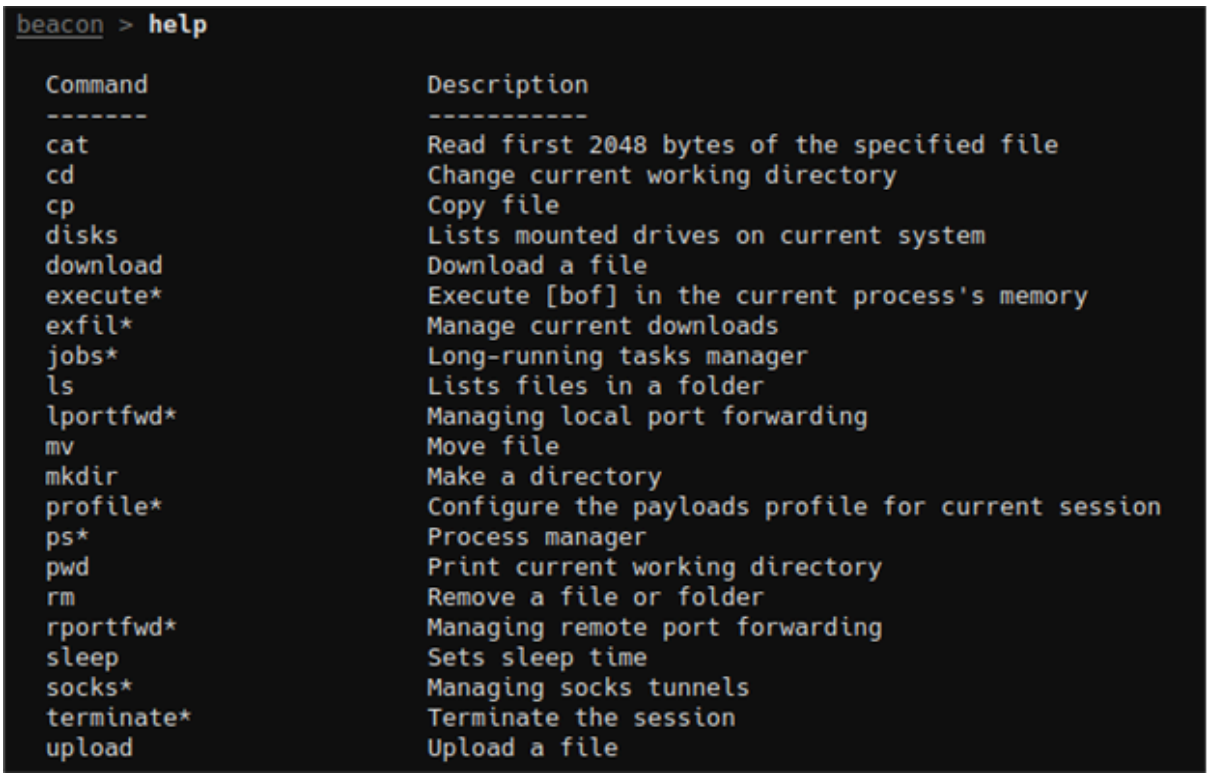
задачи, обеспечивать скрытность и устойчивость присутствия во взломанной системе.

Агенты AdaptixC2 в настоящее время написаны на C++ (Beacon) и Go (Gopher), поддерживают операционные системы Linux, Windows и Mac OS. Для Windows доступно больше вариантов агентов — как Beacon, так и Gopher, в то время как для Linux и Mac OS доступны только агенты Gopher. Каждый агент адаптирован под особенности соответствующей платформы и обеспечивает широкий функционал для управления скомпрометированными системами.

Поскольку AdaptixC2 — это фреймворк постэксплуатации, реализация автозагрузки агентов в самих компонентах отсутствует. Запуск агентов требуется организовывать вручную после получения доступа к системе, что соответствует типичной модели работы таких инструментов.

## Поддерживаемый функционал:

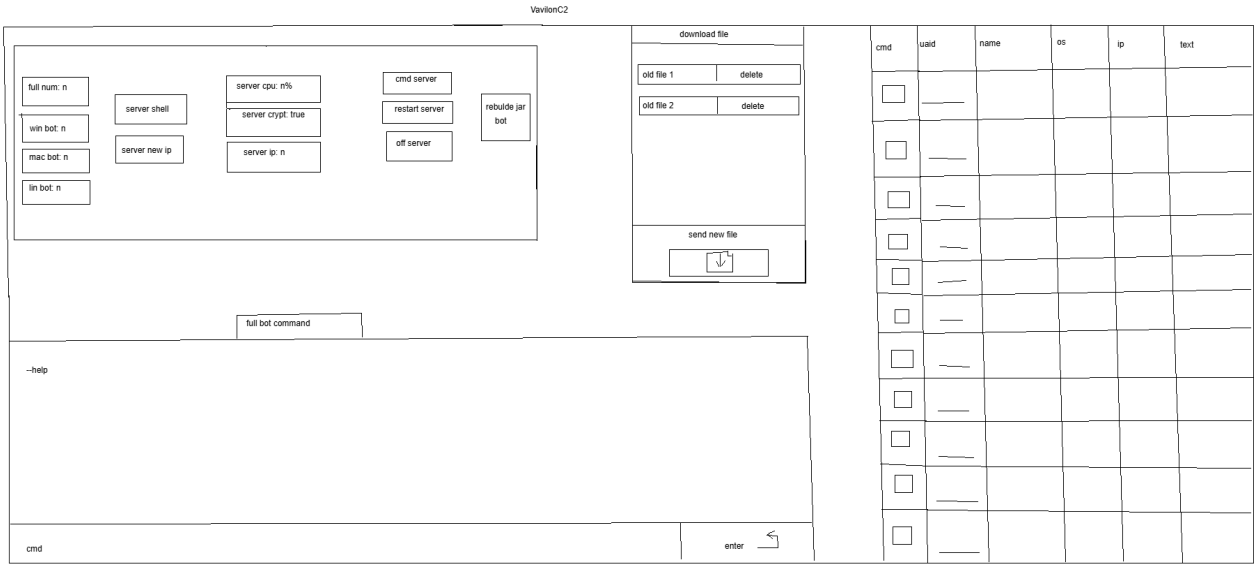
Функционал агентов широкий даже в базовой комплектации. Для каждой команды есть справка, доступная из консоли клиента. Для операций с файлами и процессами есть отдельные визуальные вкладки в интерфейсе, которые существенно упрощают процесс работы. Ну а для всего остального есть консоль.



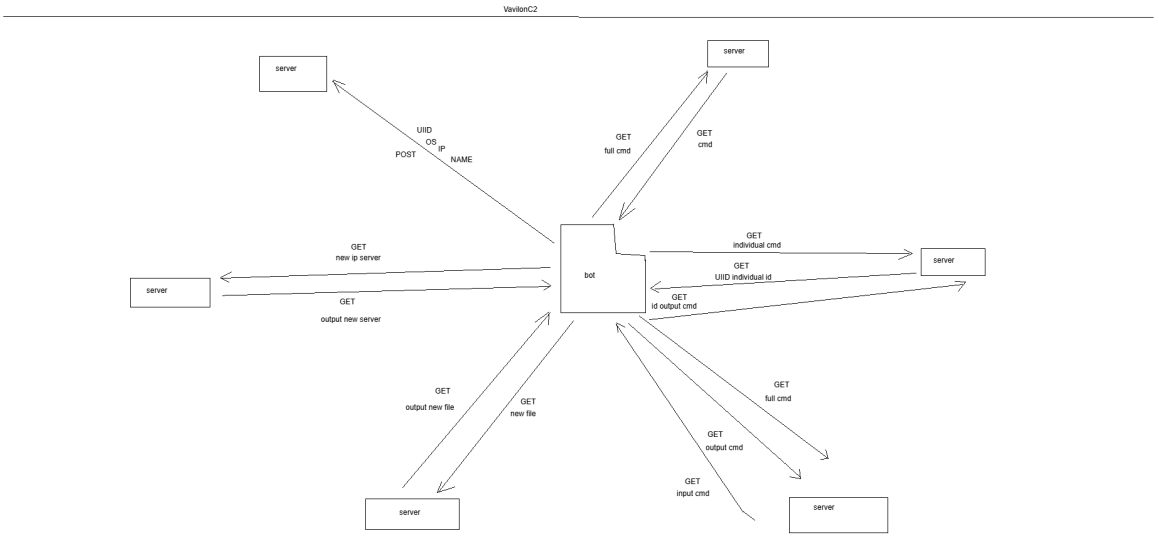

- Выполнение команд через интерактивную оболочку заражённого компьютера (shell).
- Чтение, запись и удаление произвольных файлов и каталогов.
- Скачивание и загрузка файлов.
- Создание скриншотов экрана.
- Выполнение BOF в контексте текущего процесса (расширение возможностей агента налету, без изменения исходного кода).
- Операции с процессами (просмотр, создание и прекращение).
- Создание туннелей (SOCKS4/5) и пробрасывание портов.
- Соединение нескольких агентов в цепь для обхода сегментации сети и межсетевых экранов (pivoting) по SMB/TCP.
- Изменение текущих настроек профиля (даты окончания работы, рабочих интервалов и периодичности запроса команд).

Интерфейс и базовая логика будущего аналога(не считая функций аналога, и того что весь код будет на Java, а так же один билд на все ОС)

# Сервер:


VavilonC2
full num: n
win bot: n
mac bot: n
lin bot: n
server shell
server new ip
server cpu: n%
server crypt: true
server ip: n
cmd server
restart server
off server
rebulde jar bot
download file
old file 1
delete
old file 2
delete
send new file
cmd
uaid
name
os
ip
text
full bot command
--help
cmd
enter

# Бот:


VavilonC2
server
server
UIID
OS
IP
POST
NAME
GET
full cmd
GET
cmd
GET
new ip server
GET
output new server
server
bot
GET
individual cmd
GET
UIID individual id
GET
id output cmd
server
GET
output new file
GET
new file
server
GET
full cmd
GET
output cmd
GET
input cmd
server