

# malware/windows/gozi-isfb at master · gbrindisi/malware · GitHub

github.com/gbrindisi/malware/tree/master/windows/gozi-isfb

gbrindisi

## gbrindisi/malware

malware source codes



 1  
Contributor

 1  
Issue

 470  
Stars

 295  
Forks



master

[malware](#) / [windows](#) / **[gozi-isfb](#)** /

Failed to load latest commit information.

..

[AcDII](#)

[BcClient](#)

[Builder](#)

Common

Handle

KeyLog

Lib32

Lib64

RsaKey

SocksLib

ZipLib

apdepack

client

crypto

cryptor

dname

release(builder)

x64/release(builder)

zconv

Config.exe

ReadMe.txt

appack.exe

av.c

bkinst.h

build.bat

build\_crm.bat

builder.h

client.ini

config.h

config.txt

crm.h

crm.rc

crm.vcproj

crm\_b.rc

crm\_dbg.rc

crm\_u.rc

cschar.c

desktop.c

dll.c

dll.def

dll.h

dll.vcproj

dname.txt

encrypts.bat

exe.c

format.txt

id.h

install.c

isfb.sln

masm64.rules

ntdll.h

pack\_dll.bat

pack\_exe.bat

private.key

private.key.txt

public.key

public.key.txt

resource.h

testconf.bin

testconf.txt

unpack\_dll.bat

zconv.txt

**ReadMe.txt**

---

ISFB - программа-бот предназначенная для анализа и модификации HTTP трафика на компьютере клиента.

Поддерживает все 32x и 64x битные Windows, начиная с Windows XP.  
Поддерживает все 32x и 64x битные версии Internet Explorer, начиная с 6.0.

Поддерживает все 32x и 64x битные версии Mozilla Firefox.

Поддерживает все 32x битные версии Google Chrome.

Программа способна устанавливаться и работать без привелегий администратора.

Обрабатывает весь HTTP трафик браузера в том числе и шифрованный HTTPS.

Бот управляется с удаленного сервера, с помощью файлов конфигурации и команд.

Файлы конфигурации и команд подписываются посредством RSA. При получении файлов, бот проверяет цифровую подпись, и, в случае несоответствия подписи, файл игнорируется.

При первом запуске бот иницирует таймер. В дальнейшем, по таймеру, бот обращается на управляющий сервер за файлами.

Поддерживается 2 способа поиска управляющего сервера:

- перебор заданного списка доменных имен и выбор активного;
- генерация динамического списка доменных имен в зависимости от текущей даты и конфигурации системы;

Анализ трафика производится на основе специально сформированного файла конфигурации, который бот получает с сервера.

Такой файл может содержать следующие инструкции:

- подмена HTML страницы целиком
- замена фрагмента HTML страницы
- скопировать фрагмент страницы и отправить на сервер
- найти файл по маске и отправить на сервер
- сделать скриншот экрана и отправить на сервер

Кроме файла конфигурации бот получает с сервера команды:

GET\_CERTS - экспортировать и выслать сертификаты, установленные в системном хранилище Windows.

Для XP выгружает, также, неэкспортируемые сертификаты.

GET\_COOKIES - собрать cookie FF и IE, SOL-файлы Flash, упаковать их с сохранением структуры

каталогов и выслать на сервер.

CLR\_COOKIES - удалить cookie FF и IE, SOL-файлы Flash.

GET\_SYSINFO - собрать системную информацию: тип процессора, версию ОС, список процессов, список

драйверов, список установленных программ.

KILL - убить ОС (работает только с правами администратора)

REBOOT - перезагрузить ОС

GROUP=n - сменить ID группы бота на n

LOAD\_EXE=URL - загрузить файл с указанного URL и запустить его

LOAD\_REG\_EXE=URL - загрузить файл с указанного URL, зарегистрировать его в autirun и запустить

LOAD\_UPDATE=URL - загрузить апдейт программы и запустить  
 GET\_LOG - отправить внутренний лог на сервер  
 GET\_FILES=\* - найти все файлы, соответствующие заданной маске, и  
 отправить на сервер  
 SLEEP=n - остановить обработку очереди команд на n  
 миллисекунд. (используется при долгих операциях)  
 SEND\_ALL - отправить все данные из очереди на отправку  
 немедленно. В противном случае, данные отправляются  
 по таймеру.  
 LOAD\_DLL=URL[,URL] - загрузить по указанному URL DLL и инжектировать её в  
 процесс explorer.exe.  
 первый URL для 32x-битной DLL, второй - для  
 64x-битной.  
 SOCKS\_START=IP:PORT - запустить сокс4\5 сервер (при его наличии)  
 SOCKS\_STOP - остановить сокс4\5 сервер  
 GET\_KEYLOG - отправить данные кейлоггера (при его наличии)  
 GET\_MAIL - активировать граббер E-Mail (при наличии) и  
 отправить, полученные от него, данные  
 GET\_FTP - активировать граббер FTP (при наличии) и отправить,  
 полученные от него, данные  
 SELF\_DELETE - удалить софт из системы, включая все файлы и ключи  
 реестра  
 URL\_BLOCK=URL - заблокировать доступ ко всем URL удовлетворяющим  
 заданной маске  
 URL\_UNBLOCK=URL - разблокировать доступ к URL, удовлетворяющим  
 заданной маске, ранее заблокированным командой URL\_BLOCK  
 FORMS\_ON - включить граббер HTTP форм (если есть  
 дефайн \_ALWAYS\_HTTPS, то граббер HTTPS остаётся включен всегда)  
 FORMS\_OFF - отключить граббер HTTP форм  
 KEYLOG\_ON[= list] - включить кейлог, для заданного списка  
 процессов  
 KEYLOG\_OFF - отключить кейлог  
 LOAD\_INI=URL - загрузить упакованный INI-файл с указанного URL,  
 сохранить его в реестре и использовать вместо INI-файла,  
 прикрепленного к софту с  
 помощью билдера. INI-файл должен быть упакован и подписан.  
 LOAD\_REG\_DLL = name, URL[,URL] - загрузить DLL по указанному URL,  
 сохранить её под заданным именем и зарегистрировать для  
 автоматической загрузки после  
 каждого запуска системы  
 UNREG\_DLL = name - удалить из автоматической загрузки DLL с  
 заданным именем

#### Технические детали

Дропер - программа установки.  
 Дропер представляет собой исполняемый файл Windows (PE32). В файле, в  
 виде бинарного ресурса, содержатся  
 две упакованные DLL: 32x битный и 64x-битный бот.  
 При старте дропер распаковывает DLL и регистрирует их для

автозапуска.

DLL распаковываются и регистрируются таким образом, чтобы иметь возможность выполняться при любом уровне привелегий:  
как при администраторе, так и при пользователе.

DLL - бот.

Бот представляет собой динамически загружаемую библиотеку (DLL). Для каждой архитектуры собирается своя, соответствующая DLL.

DLL-бот загружается во все запускаемые процессы.

Бот состоит из 2х логических компонентов: парсер и сервер. Парсер активируется в контексте процесса-браузера.

Сервер активируется в контексте процесса оболочки (как правило explorer.exe).

Парсер выполняет следующие функции:

- отправка/получение данных (получение команд, конфигов; отправка форм, файлов)
- непосредственный перехват, анализ, и модификация HTTP трафика

Сервер (в контексте explorer.exe) выполняет:

- файловые операции (поиск, создание и удаление файлов)
- запуск программ, обновление
- системные функции (перезагрузка, блокировка ОС)

Таким образом, все операции, требующие привелегий, выполняются сервером в контексте explorer.exe,  
а все операции с сетью исключительно из браузера.

Сборка и настройка

Проект собирается при помощи Microsoft Visual Studio 2005, либо более поздней версии.

В проект интегрирован криптор, который используется по-умолчанию.

В результате сборки и криптовки получаются следующие файлы:

Release\crm\_p.exe

Release\client\_p.dll

x64\Release\client\_p.dll

это упакованные и криптованные версии бота и дропера, причем дропер (файл crm\_p.exe) содержит в себе два других.

Некрипованные версии бота лежат там же:

Release\crm.exe

Release\client.dll

x64\Release\client.dll

Кроме бота, проект включает в себя:

Release\dnname.exe - утилита для генерации псевдо-случайных доменных имен;

Release\rsakey.exe - утилита для подписывания файлов команд и конфига;

config.exe - программа конфигуратор.

Основные настройки программы находятся в файлах id.h и config.h.

id.h содержит номер группы бота.

config.h содержит такие параметры как: список управляющих серверов,

названия URL-ов для получения команд и конфигов,  
и для отсылки данных, а также различные ключи и параметры влияющие  
на настройку программы.

#### Сборка с билдером

Существует возможность собрать ISFB так, чтобы в дальнейшем  
прикреплять к DLL ключи и файлы настроек,  
не пересобирая проект.

1. Собрать ISFB в конфигурации Release(Builder) под x86 и x64.
2. Отредактировать файлы: \public.key и \client.ini, содержащие RSA-ключ и настройки программы соответственно.
3. В консольном окне выполнить build.bat из папки \Builder
4. Забрать готовый installer.exe из папки \Builder\Release

Батник build.bat запускает билдер, который прикрепляет к каждой DLL  
(для x86 и x64) файлы:  
public.key и client.ini.

В последствии обе DLL прикрепляются к инсталлеру.

Готовый инсталлер сохраняется в файл \Release\install.exe

#### Сборка с ВК

Существует возможность собрать ISFB вместе ВК в один исполняемый  
файл-установщик, так, чтобы в случае  
ошибки при установке ВК, устанофчик извлекал DLL и устанавливал их  
отдельно.

Примечание: папка, содержащая солюшен с ВК2 должна находиться в той  
же директории, что и папка, содержащая ISFB.

1. Собрать ВК в конфигурации Release под x86 и x64.
2. Собрать ISFB в конфигурации Release(Builder) под x86 и x64.
3. Отредактировать файлы: \public.key и \client.ini.
4. В консольном окне запустить bkbuild.bat из папки \Builder
5. Забрать собранный bksetup.exe, содержащий ВК, ISFB-DLL и ISFB-инсталлер, из \Builder\Release

#### Работа в режиме инжекта из памяти

Для работы в режиме инжекта из памяти необходимо установить значение  
флага \_INJECT\_AS\_IMAGE в файле \common\main.h в

TRUE, и пересобрать проект. В этом случае инсталлер не создает DLL  
на диске, а копирует себя в одну из системных папок  
и регистрируется в Windows AutoRun.

При запуске инсталлер инжектит образ DLL, соответствующей  
архитектуры, в Explorer.exe, откуда, в свою очередь,  
соответствующий образ DLL инектится во все поражаемые процессы,  
разных архитектур.

#### Плагины

ISFB поддерживает плагины: специально собранные, DLL, экспортирующие функцию PluginRegisterCallbacks и вызывающие внутренние функции софта (например, функции отправки данных). Для загрузки плагина используется команда:  
LOAD\_PLUGIN=URL[,URL] - где первый URL для 32x-битной версии DLL, второй - 64x-битной.  
Софт скачивает DLL соответствующей архитектуры и инжектит её в explorer.exe, затем вызывается функция PluginRegisterCallbacks, в которую передаётся указатель на список коллбэков (функций), реализованных внутри софта, которые может использовать плагин.  
Описание структур и прототипов функций для создания плагинов находится в файле \common\plugin.h

### Состав проекта

\AcDLL - библиотека инжектов. Реализует механизм инжекта DLL во все поражаемые процессы, независимо от архитектуры.  
Поддерживает два режима работы: инжект, непосредственно DLL и инжект образа DLL из памяти без создания файла на диске.  
\ArDpack - библиотека на основе APLIB, релизующая функции распаковки.  
\VcClient - библиотека клиента для бэкконект сервера.  
\Client - основная DLL приложения  
\Common - библиотечка, реализующая общие функции, используемые в разных частях проекта. Такие как: чтение файлов, ключей реестра, операции с потоками данных, со строками, с XML, хуки и т.п.  
\Crypto - библиотека криптографических функций. Реализует следующие алгоритмы: CRC32, BASE64, MD5, RSA, RC6, AES, DES, SHA1.  
Используется для подписи конфиг-файлов и файлов команд, а также, для саршифровки информации e-mail и ftp аккаунтов.  
\Dname - программа генерации доменных имён на основе номера группы софта и текущей даты.  
\Ftp - библиотека FTP-грабберов.  
\Handle - библиотека, реализующая хэш таблицу. Используется для привязки хэндлов HTTP запросов к внутреннему контексту ISFB.  
Также, используется кейлоггером, для группировки клавиатурных логов по PID-ам и HWND.  
\IM - DLL-плагин, реализующая граббер Instant Messengers.  
\Install - программа-установщик ISFB.  
\KeyLog - библиотека кейлоггер.  
\Mail - библиотека E-mail грабберов.  
\RsaKey - программа для шифрования и цифровой подписи конфиг-файлов и файлов команд.  
\SocksLib - библиотека, реализующая SOCKS4\5-сервер.  
\Sqlite3 - библиотека для работы с БД SQLite. Используется IM-грабберами.  
\ZConv - программа-конвертер конфигов Zeus в конфиг-файлы ISFB.