Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Лабораторная работа

Дисциплина: Проектирование ОС и компонентов

Тема: Обфускация утилиты Windows и Linux

Выполнил студент гр. 13541/4 \_\_\_\_\_\_\_\_\_\_ Зорин А.Г.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_ Душутина Е.В.

(подпись)

«\_\_»\_\_\_\_\_\_\_\_ 2017 г.

Санкт –Петербург

2017

**Цель работы:**

В данной работе необходимо выполнить:

* Обфускацию утилиты Windows
* Обфускацию утилиты Linux

Все эти пункты необходимо выполнить по отношению к утилите, которая была написана ранее. Суть этой утилиты заключается в отслеживании изменений файлов и директорий. Эти пункты выполняются в ОС двух семейств.

**Выполнение работы:**

Обфускация – это один из методов защиты программного кода, который позволяет усложнить процесс реверсивной инженерии кода защищаемого программного продукта. Суть процесса обфускации заключается в том, чтобы запутать программный код и устранить большинство логических связей в нем, то есть трансформировать его так, чтобы он был очень труден для изучения и модификации посторонними лицами (будь то взломщики, или программисты которые собираются узнать уникальный алгоритм работы защищаемой программы).

Из этого следует, что обфускация одна не предназначена для обеспечения наиболее полной и эффективной защиты программных продуктов, так как она не предоставляет возможности предотвращения нелегального использования программного продукта. Поэтому обфускацию обычно используют вместе с одним из существующих методов защиты, это позволяет значительно повысить уровень защиты программы в целом.

Обфускация соответствует принципу экономической целесообразности, так как ее использование не сильно, увеличивает стоимость программного продукта, и позволяет при этом снизить потери от пиратства, и уменьшить возможность плагиата в результате кражи уникального алгоритма работы защищаемого программного продукта.

Существуют различные определения процесса обфускации. Рассматривая данный процесс с точки зрения защиты ПП, и трансформации кода программы без возможности в последствии вернуться к его первоначальному виду (трансформация "в одну сторону"), можно дать такое определение: Определение. Пусть "TR" будет трансформирующим процессом, тогда при "PR1 =TR=> PR2" программа "PR2" будет представлять собой трансформированный код программы "PR1". Процесс трансформации "TR" будет считаться процессом обфускации если, будут удовлетворены такие требования:

* код программы "PR2" в результате трансформации будет существенно отличаться от кода программы "PR1", но при этом он будет выполнять те же функции что и код программы "PR1", а также будет работоспособным.
* изучение принципа работы, то есть процесс реверсивной инженерии, программы "PR2" будет более сложным, трудоемким, и будет занимать больше времени, чем программы "PR1".
* при каждом процессе трансформации одного и того же кода программы "PR1", код программ "PR2" будут различны.
* создание программы детрансформирующей программу "PR2" в ее наиболее похожий первоначальный вид, будет неэффективно.

Существует много методов определения эффективности применения того или иного процесса обфускации, к конкретному программному коду.

Эти методы принято разделять на две группы: аналитические и эмпирические. Аналитические методы основываются на трех величинах характеризующих насколько эффективен тот или иной процесс обфускации:

1. Устойчивость - указывает на степень сложности осуществления реверсивной инженерии над кодом прошедшим процесс обфускации.
2. Эластичность - указывает на то насколько хорошо данный процесс обфускации, защитит программный код от применения деобфускаторов.
3. Стоимость преобразования - позволяет оценить, насколько больше требуется системных ресурсов для выполнения кода прошедшего процесс обфускации, чем для выполнения оригинального кода программы.

Процессы обфускации можно классифицировать по видам, в зависимости от способа модификации кода программы.

* **Лексическая обфускация**

Наиболее простая, заключается в форматировании кода программы, изменении его структуры, таким образом, чтобы он стал нечитабельным, менее информативным, и трудным для изучения.

Обфускация такого вида включает в себя:

* удаление всех комментариев в коде программы, или изменение их на дезинформирующие
* удаление различных пробелов, отступов которые обычно используют для лучшего визуального восприятия кода программы
* замену имен идентификаторов (имен переменных, массивов, структур, хешей, функций, процедур и т.д.), на произвольные длинные наборы символов, которые трудно воспринимать человеку
* добавление различных лишних (мусорных) операций
* изменение расположения блоков (функций, процедур) программы, таким образом, чтобы это не коим образом не повлияло на ее работоспособность.
* **Обфускация данных**

Такая обфускация связана с трансформацией структур данных. Она считается более сложной, и является наиболее продвинутой и часто используемой. Ее принято делить на три основные группы, которые описаны ниже.

*Обфускация хранения*. Заключается в трансформации хранилищ данных, а также самих типов данных. Ниже приведены основные методы, позволяющие осуществить такую обфускацию:

* изменение интерпретации данных определенного типа. Например, можно сказать, что 16-разрядная переменная целого типа содержащая комбинации разрядов 0000000000001100 представляет целое число 12, но это простое соглашение, данные в такой переменной можно интерпретировать по-разному (не обязательно как 12, а, например, как 1100 и т.д.).
* изменение срока использования хранилищ данных, например, переход от локального их использования к глобальному и наоборот.
* преобразование статических (неменяющихся) данных в процедурные. Строки желательно предать обфускации, это можно сделать, просто записывая каждый символ строки, используя его ASCII код, например символ "A" можно записать как 16-ричное число "0х41", но такой метод банален.
* разделение переменных. Переменные фиксированного диапазона могут быть разделены на две и более переменных. Для этого переменную "V" имеющую тип "x" разделяют на "k" переменных "v1,...,vk" типа "y" то есть "V == v1,...,vk". Потом создается набор функций позволяющих извлекать переменную типа "x" из переменных типа "y" и записывать переменную типа "x" в переменные типа "y".
* изменение представления (или кодирование). Например, целочисленную переменную "i" можно заменить, выражением "i` = c1\*i + c2" где "c1,c2" являются константами.

*Обфускация соединения.* Один из важных этапов в процессе реверсивной инженерии программ, основан на изучении структур данных. Поэтому важно постараться, в процессе обфускации, усложнить представление используемых программой структур данных. Например, при использовании обфускации соединения это достигается благодаря соединению независимых данных, или разделению зависимых.

* объединение переменных. Две или более переменных "v1,...,vk" могут быть объединены в одну переменную "V", если их общий размер ("v1,...,vk") не превышает размер переменной "V".
* реструктурирование массивов, заключается в запутывании структуры массивов, путем разделения одного массива на несколько подмассивов, объединения нескольких массивов в один, сворачивания массива (увеличивая его размерность) и наоборот, разворачивая (уменьшая его размерность).
* изменение иерархий наследования классов, осуществляется путем усложнения иерархии наследования при помощи создания дополнительных классов или использования ложного разделения классов.

*Обфускация переупорядочивания*. Заключается в изменении последовательности объявления переменных, внутреннего расположения хранилищ данных, а также переупорядочивании методов, массивов (использование нетривиального представления многомерных массивов), определенных полей в структурах и т.д.

* **Обфускация управления**

Обфускация такого вида осуществляет запутывание потока управления, то есть последовательности выполнения программного кода.

Большинство ее реализаций основывается на использовании непрозрачных предикат, в качестве, которых выступают, последовательности операций, результат работы которых сложно определить.

Определение. Предикат "Р" считается непрозрачным предикатом, если его результат известен только в процессе обфускации, то есть после осуществления процесса обфускации, определение значения такого предиката, становится трудным.

Непрозрачные предикаты могут быть:

* локальными - вычисления содержаться внутри одиночного выражения
* глобальными - вычисления содержаться внутри одной процедуры
* межпроцедурными - вычисления содержаться внутри различных процедур

Методы, позволяющие осуществить обфускацию управления, классифицируются на три основных группы:

*Обфускация вычислительная.* Изменение касающиеся главной структуры потока управления. К ним можно отнести:

* расширения условий циклов. Для этого обычно используют непрозрачные предикаты, таким образом, чтобы они не коим образом не влияли на количество выполнений циклического кода.
* добавления недостижимого кода, (который не будет выполняться в процессе работы программы)
* устранение библиотечных вызовов. Имена функций из стандартных библиотек, также желательно придать обфускации, т.е. изменить на наиболее бессмысленные, которые потом будут фигурировать в коде защищаемой программы.
* добавление избыточных операций (мертвого кода) в те участки программного кода, которые наиболее трудные (изначально) для изучения.
* параллелизирование кода, заключается в разделении кода на отдельные независимые участки, которые во время работы программы будут выполняться параллельно

*Обфускация соединения.* Объединение или разделение определенных фрагментов кода программы, для того чтобы убрать логические связи между ними. Ниже приведены основные методы, позволяющие осуществить такую обфускацию:

* встраивание функций, осуществляется путем встраивания кода функции, в места ее вызова.
* извлечение функций, является обратным действием, по отношению к встраиванию функций. Осуществляется в результате объединения некоторой группы взаимосвязанных операторов в коде исходной программы в отдельную функцию, которой потом замещают эти группы операторов. Но следует учесть, что такое преобразование может быть снято компилятором в процессе компиляции кода программы.
* чередование, объединение фрагментов кода программы, выполняющих различные операции, воедино.
* клонирование, данный метод позволяет усложнить анализ контекста использования функций, и объектов используемых в коде исходной программы.
* трансформация циклов. Циклы встречаются в коде различных программ, и их также можно придать трансформации. Блокирование циклов, заключается в добавлении вложенных циклов в существующие, в результате работа существующих циклов будет заблокирована, на какой-то диапазон значений.
* Развертка циклов, повторение тела цикла один или более раз
* Разделение циклов, цикл состоящий из более чем одной независимой операции можно разбить на несколько циклов

*Обфускация последовательности.* Заключается в переупорядочивании блоков (инструкций переходов), циклов, выражений.

* **Первентивная обфускация.**

Превентивная обфускация предназначена для предотвращения применения злоумышленником деобфускаторов, де компиляторов и остальных программных средств деобфускации.

Она нацелена на использование недостатков, особенностей присутствующих в наиболее распространенных программных средствах часто используемых злоумышленниками в процессе деобфускации.

1. Ubuntu

Информация о системе:

|  |
| --- |
| ➜ ~ uname -a  Linux Arseny-HP-Pavilion-dv6-Notebook-PC 4.4.0-66-generic #87-Ubuntu SMP Fri Mar 3 15:29:05 UTC 2017 x86\_64 x86\_64 x86\_64 GNU/Linux |

Для обфускации на Linux была использована утилита CXX-OBFUS. Сама утилита является платной. Однако, имеется пробная версия с ограниченной функциональностью. Пробная версия может выполнять только один вид преобразования для имен. Заключается этот вид в добавлении строки *ReplacementFor\_.* В полной версии утилиты, эта строка заменяется, например, на z78a4b2e. Данная утилита не вносит изменений в исходный код обфусцируемой программы. Можно задать соответствующие настройки, чтобы обфусцированный код сохранился в отдельный файл. По умолчанию код выводится в консоль. В листинге ниже показан пример и результат выполнения утилиты.

|  |
| --- |
| **➜ bin ./cxx-obfus inot.c**  switching filename/dirname hasher to 'prefix' due to trial limitations. Get non-trial edition to fix this.  #include "lib.h"  char ReplacementFor\_events\_buf[ReplacementFor\_PATH\_MAX+sizeof(struct  ReplacementFor\_inotify\_event)+(0x94d+2715-0x13e7)];ReplacementFor\_\_watchstruct\*  ReplacementFor\_watches;int ReplacementFor\_ifd=(0x185f+600-0x1ab7),  ReplacementFor\_max\_watches;char\*ReplacementFor\_watch\_dir;int main(int  ReplacementFor\_argc,char\*ReplacementFor\_argv[]){int ReplacementFor\_fd,  ReplacementFor\_n;char ReplacementFor\_buf[(0x870+5696-0x1ea4)];  ReplacementFor\_argc=(0x47f+1000-0x865);if(ReplacementFor\_argc!=  (0x132c+1072-0x175a)){ERROR(  "\x45\x52\x52\x4f\x52\x3a\x20\x4e\x6f\x20\x64\x69\x72\x65\x63\x74\x6f\x72\x79\x20\x74\x6f\x20\x77\x61\x74\x63\x68" "\n"  );return(0x1e75+2190-0x2702);}ReplacementFor\_fd=ReplacementFor\_open(  "\x2f\x70\x72\x6f\x63\x2f\x73\x79\x73\x2f\x66\x73\x2f\x69\x6e\x6f\x74\x69\x66\x79\x2f\x6d\x61\x78\x5f\x75\x73\x65\x72\x5f\x77\x61\x74\x63\x68\x65\x73"  ,ReplacementFor\_O\_RDONLY);if(ReplacementFor\_fd<(0x1c76+560-0x1ea6)){  ReplacementFor\_perror(  "\x43\x61\x6e\x6e\x6f\x74\x20\x6f\x70\x65\x6e\x20\x2f\x70\x72\x6f\x63\x2f\x73\x79\x73\x2f\x66\x73\x2f\x69\x6e\x6f\x74\x69\x66\x79\x2f\x6d\x61\x78\x5f\x75\x73\x65\x72\x5f\x77\x61\x74\x63\x68\x65\x73"  );return(0x844+5566-0x1e01);}if((ReplacementFor\_n=ReplacementFor\_read(  ReplacementFor\_fd,ReplacementFor\_buf,sizeof(ReplacementFor\_buf)-  (0x8a1+3804-0x177c)))<(0x34b+4843-0x1636)){ReplacementFor\_perror(  "\x43\x61\x6e\x6e\x6f\x74\x20\x72\x65\x61\x64\x28\x29\x20\x2f\x70\x72\x6f\x63\x2f\x73\x79\x73\x2f\x66\x73\x2f\x69\x6e\x6f\x74\x69\x66\x79\x2f\x6d\x61\x78\x5f\x75\x73\x65\x72\x5f\x77\x61\x74\x63\x68\x65\x73"  );return(0x7c3+798-0xae0);}ReplacementFor\_buf[ReplacementFor\_n]=  (0x1a4b+583-0x1c92);ReplacementFor\_max\_watches=ReplacementFor\_atoi(  ReplacementFor\_buf)\*(0x781+3807-0x165e);if(ReplacementFor\_max\_watches<=  (0x3f5+7054-0x1f83)){ERROR(  "\x49\x6e\x63\x6f\x72\x72\x65\x63\x74\x20\x6e\x75\x6d\x62\x65\x72\x20\x6f\x66\x20\x77\x61\x74\x63\x68\x65\x73\x3a\x20"  );ERROR(ReplacementFor\_buf);ERROR("\n");return(0x123d+565-0x1471);}else{  ReplacementFor\_DEBUG\_PRINT(  "\x4d\x61\x78\x20\x77\x61\x74\x63\x68\x65\x73\x3a\x20\x25\x64" "\n",  ReplacementFor\_max\_watches);}ReplacementFor\_watch\_dir=  "\x2f\x2f\x68\x6f\x6d\x65\x2f\x2f\x61\x72\x73\x65\x6e\x79\x2f\x2f\x44\x69\x72\x46\x6f\x72\x57\x61\x74\x63\x68\x69\x6e\x67"  ;ReplacementFor\_do\_watch(ReplacementFor\_max\_watches);return(0x1169+2567-0x1b70);  }  **➜ bin ./cxx-obfus static.c**  switching filename/dirname hasher to 'prefix' due to trial limitations. Get non-trial edition to fix this.  #include "lib.h"  void ReplacementFor\_debug\_print\_mask(uint32\_t ReplacementFor\_mask){if(  ReplacementFor\_mask&ReplacementFor\_IN\_DELETE\_SELF)ReplacementFor\_PRINT(  "\x57\x41\x54\x43\x48\x49\x4e\x47\x20\x44\x49\x52\x20\x57\x41\x53\x20\x44\x45\x4c\x45\x54\x45\x44\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_MOVE\_SELF)ReplacementFor\_PRINT(  "\x57\x41\x54\x43\x48\x49\x4e\x47\x20\x44\x49\x52\x20\x57\x41\x53\x20\x4d\x4f\x56\x45\x44\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_MOVED\_FROM)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x4d\x4f\x56\x45\x44\x20\x4f\x55\x54\x53\x49\x44\x45\x20\x57\x41\x54\x43\x48\x49\x4e\x47\x20\x44\x49\x52\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_MOVED\_TO)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x4d\x4f\x56\x45\x44\x20\x49\x4e\x53\x49\x44\x45\x20\x57\x41\x54\x43\x48\x49\x4e\x47\x20\x44\x49\x52\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_CLOSE\_WRITE)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x43\x4c\x4f\x53\x45\x44\x20");if(  ReplacementFor\_mask&ReplacementFor\_IN\_MODIFY)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x4d\x4f\x44\x49\x46\x49\x45\x44\x20");if(  ReplacementFor\_mask&ReplacementFor\_IN\_IGNORED)ReplacementFor\_PRINT(  "\x57\x41\x54\x43\x48\x20\x57\x41\x53\x20\x49\x47\x4e\x4f\x52\x45\x44\x20");if(  ReplacementFor\_mask&ReplacementFor\_IN\_ISDIR)ReplacementFor\_PRINT(  "\x4f\x42\x4a\x45\x43\x54\x20\x49\x53\x20\x41\x20\x44\x49\x52\x20");if(  ReplacementFor\_mask&ReplacementFor\_IN\_Q\_OVERFLOW)ReplacementFor\_PRINT(  "\x4f\x56\x45\x52\x46\x4c\x4f\x57\x20\x4f\x46\x20\x45\x56\x45\x4e\x54\x20\x51\x55\x45\x52\x59\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_UNMOUNT)ReplacementFor\_PRINT(  "\x57\x41\x54\x43\x48\x49\x4e\x47\x20\x46\x53\x20\x57\x41\x53\x20\x55\x4e\x4d\x4f\x55\x4e\x54\x45\x44"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_CREATE)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x43\x52\x45\x41\x54\x45\x44\x20\x49\x4e\x20\x57\x41\x54\x43\x48\x49\x4e\x47\x20\x44\x49\x52\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_ACCESS)ReplacementFor\_PRINT(  "\x41\x43\x43\x45\x53\x53\x49\x4e\x47\x20\x54\x4f\x20\x54\x48\x45\x20\x46\x49\x4c\x45\x20"  );if(ReplacementFor\_mask&ReplacementFor\_IN\_DELETE)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x44\x45\x4c\x45\x54\x45\x44\x20");if(  ReplacementFor\_mask&ReplacementFor\_IN\_OPEN)ReplacementFor\_PRINT(  "\x46\x49\x4c\x45\x20\x57\x41\x53\x20\x4f\x50\x45\x4e\x45\x44\x20");}void  ReplacementFor\_wd\_path(int ReplacementFor\_wd,char\*ReplacementFor\_path){if(  ReplacementFor\_wd==(0x19ab+493-0x1b98)){ReplacementFor\_strcpy(  ReplacementFor\_path,ReplacementFor\_watch\_dir);ReplacementFor\_strcat(  ReplacementFor\_path,"\x2f");return;}if(ReplacementFor\_wd<(0x6a5+2179-0xf28)||!  ReplacementFor\_watches[ReplacementFor\_wd].ReplacementFor\_name){  ReplacementFor\_DEBUG\_PRINT(  "\x52\x65\x63\x75\x73\x69\x76\x65\x20\x25\x64\x2c\x20\x25\x78" "\n",  ReplacementFor\_wd,ReplacementFor\_watches[ReplacementFor\_wd].ReplacementFor\_name)  ;ERROR(  "\x4d\x65\x6d\x6f\x72\x79\x20\x63\x6f\x72\x72\x75\x70\x74\x65\x64\x3a\x20\x61\x73\x6b\x65\x64\x20\x70\x61\x74\x68\x20\x6f\x66\x20\x64\x65\x6c\x65\x74\x65\x64\x20\x65\x76\x65\x6e\x74" "\n"  );ReplacementFor\_exit((0x491+4635-0x16ab));}ReplacementFor\_wd\_path(  ReplacementFor\_watches[ReplacementFor\_wd].ReplacementFor\_parent\_wd,  ReplacementFor\_path);if(ReplacementFor\_watches[ReplacementFor\_wd].  ReplacementFor\_name[(0x1347+3714-0x21c9)]==(0x895+7034-0x240f))return;  ReplacementFor\_strcat(ReplacementFor\_path,ReplacementFor\_watches[  ReplacementFor\_wd].ReplacementFor\_name);ReplacementFor\_strcat(  ReplacementFor\_path,"\x2f");}  **➜ bin ./cxx-obfus dynamic.c**  switching filename/dirname hasher to 'prefix' due to trial limitations. Get non-trial edition to fix this.  #include "lib.h"  int ReplacementFor\_do\_watch(int ReplacementFor\_max\_watches){struct  ReplacementFor\_inotify\_event\*ReplacementFor\_ev;ReplacementFor\_ssize\_t  ReplacementFor\_n=(0xb63+916-0xef7),ReplacementFor\_wd;char ReplacementFor\_path[  ReplacementFor\_PATH\_MAX+(0x1ba+4878-0x14c7)];ReplacementFor\_watches=(  ReplacementFor\_\_watchstruct\*)ReplacementFor\_calloc(ReplacementFor\_max\_watches,  sizeof(ReplacementFor\_\_watchstruct));if(!ReplacementFor\_watches){ERROR(  "\x43\x61\x6e\x6e\x6f\x74\x20\x61\x6c\x6c\x6f\x63\x61\x74\x65\x20\x6d\x65\x6d\x6f\x72\x79" "\n"  );ReplacementFor\_exit((0x17c2+2674-0x2233));}ReplacementFor\_DEBUG\_PRINT(  "\x44\x6f\x69\x6e\x67\x20\x69\x6e\x69\x74\x69\x61\x6c\x20\x77\x61\x74\x63\x68\x65\x73\x20\x73\x65\x74\x75\x70" "\n"  );ReplacementFor\_ifd=ReplacementFor\_inotify\_init();if(ReplacementFor\_ifd==-  (0x1081+836-0x13c4)){ReplacementFor\_perror(  "\x43\x61\x6e\x6e\x6f\x74\x20\x69\x6e\x69\x74\x20\x69\x6e\x6f\x74\x69\x66\x79");  ReplacementFor\_exit((0x5f7+4199-0x165d));}ReplacementFor\_wd=  ReplacementFor\_add\_dir\_watch((0x12e9+1178-0x1783),ReplacementFor\_watch\_dir,"",  (0x140f+823-0x1745));if(ReplacementFor\_wd<(0xa88+5624-0x2080)){ERROR(  "\x43\x61\x6e\x6e\x6f\x74\x20\x61\x64\x64\x20\x64\x69\x72\x20\x77\x61\x74\x63\x68" "\n"  );ReplacementFor\_exit((0x17b1+236-0x189c));}ReplacementFor\_add\_dir(  ReplacementFor\_wd,ReplacementFor\_watch\_dir,(0x3df+1533-0x9db),  (0x5e3+2902-0x1138));while((ReplacementFor\_n=ReplacementFor\_read(  ReplacementFor\_ifd,ReplacementFor\_events\_buf,sizeof(ReplacementFor\_events\_buf)))  >(0x1b1d+2427-0x2498)){ReplacementFor\_ev=(struct ReplacementFor\_inotify\_event\*)  ReplacementFor\_events\_buf;while(ReplacementFor\_n>(0x136+3667-0xf89)){if(  ReplacementFor\_ev->ReplacementFor\_mask&ReplacementFor\_IN\_Q\_OVERFLOW){ERROR(  "\x51\x75\x65\x75\x65\x20\x6f\x76\x65\x72\x66\x6c\x6f\x77\x2c\x20\x72\x65\x73\x74\x61\x72\x74\x20\x6e\x65\x65\x64\x65\x64" "\n"  );ReplacementFor\_exit((0x6c4+6779-0x213c));}if(ReplacementFor\_ev->  ReplacementFor\_mask&ReplacementFor\_IN\_IGNORED){ReplacementFor\_free(  ReplacementFor\_watches[ReplacementFor\_ev->ReplacementFor\_wd].ReplacementFor\_name  );ReplacementFor\_watches[ReplacementFor\_ev->ReplacementFor\_wd].  ReplacementFor\_parent\_wd=-(0x98b+3010-0x154c);ReplacementFor\_watches[  ReplacementFor\_ev->ReplacementFor\_wd].ReplacementFor\_name=ReplacementFor\_NULL;  goto ReplacementFor\_loop\_end;}ReplacementFor\_wd\_path(ReplacementFor\_ev->  ReplacementFor\_wd,ReplacementFor\_path);ReplacementFor\_PRINT(  "\x50\x61\x74\x68\x3a\x20");ReplacementFor\_PRINT(ReplacementFor\_path);  ReplacementFor\_PRINT("\n" "\x46\x69\x6c\x65\x3a\x20");ReplacementFor\_PRINT(  ReplacementFor\_ev->ReplacementFor\_name);ReplacementFor\_PRINT(  "\n" "\x45\x76\x65\x6e\x74\x3a\x20");ReplacementFor\_debug\_print\_mask(  ReplacementFor\_ev->ReplacementFor\_mask);ReplacementFor\_PRINT("\n");  #ifdef DEBUG  if(ReplacementFor\_ev->ReplacementFor\_len){ReplacementFor\_DEBUG\_PRINT(  "\x20\x7c\x20");ReplacementFor\_DEBUG\_PRINT("\x25\x73",ReplacementFor\_ev->  ReplacementFor\_name);}ReplacementFor\_DEBUG\_PRINT("\x20\x7c\x20");  ReplacementFor\_debug\_print\_mask(ReplacementFor\_ev->ReplacementFor\_mask);  #endif  if((ReplacementFor\_ev->ReplacementFor\_mask&ReplacementFor\_IN\_DELETE)||(  ReplacementFor\_ev->ReplacementFor\_mask&ReplacementFor\_IN\_MOVED\_FROM)){goto  ReplacementFor\_loop\_end;}if(ReplacementFor\_ev->ReplacementFor\_mask&  ReplacementFor\_IN\_ISDIR&&ReplacementFor\_ev->ReplacementFor\_len>  (0x422+4097-0x1423)&&ReplacementFor\_strcmp(ReplacementFor\_ev->  ReplacementFor\_name,"\x2e")){if(ReplacementFor\_ev->ReplacementFor\_len+  ReplacementFor\_strlen(ReplacementFor\_path)>sizeof(ReplacementFor\_path)-  (0x228d+122-0x2306)){ERROR(  "\x54\x6f\x6f\x20\x64\x65\x65\x70\x20\x64\x69\x72\x65\x63\x74\x6f\x72\x79\x3a\x20"  );ERROR(ReplacementFor\_path);ERROR(ReplacementFor\_ev->ReplacementFor\_name);ERROR  ("\n");goto ReplacementFor\_loop\_end;}ReplacementFor\_strcat(ReplacementFor\_path,  ReplacementFor\_ev->ReplacementFor\_name);ReplacementFor\_wd=  ReplacementFor\_add\_dir\_watch(ReplacementFor\_ev->ReplacementFor\_wd,  ReplacementFor\_path,ReplacementFor\_ev->ReplacementFor\_name,(0xb0d+5853-0x21ea));  if(ReplacementFor\_wd<(0x5b4+4300-0x1680))goto ReplacementFor\_loop\_end;  ReplacementFor\_add\_dir(ReplacementFor\_ev->ReplacementFor\_wd,ReplacementFor\_path,  (0x10d8+919-0x146f),(0xb06+2733-0x15b3));}ReplacementFor\_loop\_end:  ReplacementFor\_n-=sizeof(struct ReplacementFor\_inotify\_event)+ReplacementFor\_ev  ->ReplacementFor\_len;ReplacementFor\_ev=(struct ReplacementFor\_inotify\_event\*)((  char\*)ReplacementFor\_ev+sizeof(struct ReplacementFor\_inotify\_event)+  ReplacementFor\_ev->ReplacementFor\_len);}ReplacementFor\_PRINT("\x2d" "\n");}  ReplacementFor\_perror(  "\x43\x61\x6e\x6e\x6f\x74\x20\x72\x65\x61\x64\x28\x29\x20\x69\x6e\x6f\x74\x69\x66\x79\x20\x71\x75\x65\x75\x65"  );ReplacementFor\_exit((0xf5c+83-0xfae));}int ReplacementFor\_add\_dir\_watch(int  ReplacementFor\_parent\_wd,char\*ReplacementFor\_dir,char\*ReplacementFor\_dir\_name,  int ReplacementFor\_no\_print){int ReplacementFor\_wd=  ReplacementFor\_inotify\_add\_watch(ReplacementFor\_ifd,ReplacementFor\_dir,  ReplacementFor\_EVENT\_MASK);if(ReplacementFor\_wd<(0x168a+1933-0x1e17)){ERROR(  "\x43\x61\x6e\x6e\x6f\x74\x20\x61\x64\x64\x20\x77\x61\x74\x63\x68\x20\x74\x6f\x20\x27"  );ERROR(ReplacementFor\_dir);ERROR(  "\x27\x20\x75\x73\x69\x6e\x67\x20\x69\x6e\x6f\x74\x69\x66\x79\x3a\x20");if(  ReplacementFor\_errno==ReplacementFor\_ENOSPC){ERROR(  "\x74\x6f\x6f\x20\x6d\x61\x6e\x79\x20\x77\x61\x74\x63\x68\x65\x73" "\n" "\x59\x6f\x75\x20\x63\x61\x6e\x20\x69\x6e\x63\x72\x65\x61\x73\x65\x20\x6e\x75\x6d\x62\x65\x72\x20\x6f\x66\x20\x75\x73\x65\x72\x20\x77\x61\x74\x63\x68\x65\x73\x20\x75\x73\x69\x6e\x67\x20\x2f\x70\x72\x6f\x63\x2f\x73\x79\x73\x2f\x66\x73\x2f\x69\x6e\x6f\x74\x69\x66\x79\x2f\x6d\x61\x78\x5f\x75\x73\x65\x72\x5f\x77\x61\x74\x63\x68\x65\x73"  );}else{ERROR(ReplacementFor\_strerror(ReplacementFor\_errno));}ERROR("\n");if(  ReplacementFor\_errno!=ReplacementFor\_EACCES&&ReplacementFor\_errno!=  ReplacementFor\_ENOENT)ReplacementFor\_exit((0xe21+1855-0x155f));return  ReplacementFor\_wd;}if(ReplacementFor\_wd>=ReplacementFor\_max\_watches){ERROR(  "\n" "\x54\x6f\x6f\x20\x6d\x61\x6e\x79\x20\x65\x76\x65\x6e\x74\x73\x3b\x20\x72\x65\x73\x74\x61\x72\x74\x20\x72\x65\x71\x75\x69\x72\x65\x64\x20\x74\x6f\x20\x70\x72\x65\x76\x65\x6e\x74\x20\x77\x61\x74\x63\x68\x20\x64\x65\x73\x63\x72\x69\x70\x74\x6f\x72\x20\x6f\x76\x65\x72\x66\x6c\x6f\x77\x2e" "\n"  );ReplacementFor\_exit((0x27d+4611-0x147d));}ReplacementFor\_dir\_name=  ReplacementFor\_strdup(ReplacementFor\_dir\_name);if(!ReplacementFor\_dir\_name){  ERROR(  "\x43\x61\x6e\x6e\x6f\x74\x20\x73\x74\x72\x64\x75\x70\x28\x64\x69\x72\x5f\x6e\x61\x6d\x65\x29" "\n"  );ReplacementFor\_exit((0x1c73+2015-0x2451));}ReplacementFor\_watches[  ReplacementFor\_wd].ReplacementFor\_wd=ReplacementFor\_wd;ReplacementFor\_watches[  ReplacementFor\_wd].ReplacementFor\_parent\_wd=ReplacementFor\_parent\_wd;if(  ReplacementFor\_watches[ReplacementFor\_wd].ReplacementFor\_name)  ReplacementFor\_free(ReplacementFor\_watches[ReplacementFor\_wd].  ReplacementFor\_name);ReplacementFor\_watches[ReplacementFor\_wd].  ReplacementFor\_name=ReplacementFor\_dir\_name;if(!ReplacementFor\_no\_print){  ReplacementFor\_PRINT(ReplacementFor\_dir);ReplacementFor\_PRINT("\n");}return  ReplacementFor\_wd;}void ReplacementFor\_add\_dir(int ReplacementFor\_dir\_wd,char\*  ReplacementFor\_dir,int ReplacementFor\_errors\_fatal,int ReplacementFor\_no\_print){  char ReplacementFor\_path[ReplacementFor\_PATH\_MAX+(0xd1d+4628-0x1f30)];  ReplacementFor\_DIR\*ReplacementFor\_dh=ReplacementFor\_opendir(ReplacementFor\_dir);  struct ReplacementFor\_dirent\*ReplacementFor\_ent;struct ReplacementFor\_stat  ReplacementFor\_st;int ReplacementFor\_dirl=ReplacementFor\_strlen(  ReplacementFor\_dir),ReplacementFor\_n=sizeof(ReplacementFor\_path)-  (0x255+2983-0xdfb)-ReplacementFor\_dirl,ReplacementFor\_had\_errors=  (0x18d7+2577-0x22e8),ReplacementFor\_wd;if(ReplacementFor\_dirl>sizeof(  ReplacementFor\_path)-(0x5f+4534-0x1212)){ERROR(  "\x54\x6f\x6f\x20\x6c\x6f\x6e\x67\x20\x70\x61\x74\x68\x20\x28\x6e\x6f\x74\x20\x77\x61\x74\x63\x68\x65\x64\x29\x3a\x20"  );ERROR(ReplacementFor\_dir);ERROR("\n");if(ReplacementFor\_errors\_fatal)  ReplacementFor\_exit((0x3fa+2269-0xcd6));return;}if(!ReplacementFor\_dh){ERROR(  "\x43\x61\x6e\x6e\x6f\x74\x20\x6f\x70\x65\x6e\x64\x69\x72\x28");ERROR(  ReplacementFor\_dir);ERROR("\x29\x3a\x20");ERROR(ReplacementFor\_strerror(  ReplacementFor\_errno));ERROR("\n");if(ReplacementFor\_errors\_fatal)  ReplacementFor\_exit((0x856+6987-0x23a0));return;}ReplacementFor\_strcpy(  ReplacementFor\_path,ReplacementFor\_dir);while((ReplacementFor\_ent=  ReplacementFor\_readdir(ReplacementFor\_dh))!=ReplacementFor\_NULL){if(!  ReplacementFor\_strcmp(ReplacementFor\_ent->ReplacementFor\_d\_name,"\x2e")||!  ReplacementFor\_strcmp(ReplacementFor\_ent->ReplacementFor\_d\_name,"\x2e\x2e")||!  ReplacementFor\_strcmp(ReplacementFor\_ent->ReplacementFor\_d\_name,  "\x2e\x75\x6e\x72\x65\x61\x6c\x73\x79\x6e\x63"))continue;ReplacementFor\_path[  ReplacementFor\_dirl]=((char)(0xa71+5687-0x2079));ReplacementFor\_path[  ReplacementFor\_dirl+(0x1ab5+1893-0x2219)]=(0x1193+2214-0x1a39);  ReplacementFor\_strncat(ReplacementFor\_path+ReplacementFor\_dirl,  ReplacementFor\_ent->ReplacementFor\_d\_name,ReplacementFor\_n);ReplacementFor\_path[  sizeof(ReplacementFor\_path)-(0x665+6397-0x1f61)]=(0x425+8356-0x24c9);if(  ReplacementFor\_lstat(ReplacementFor\_path,&ReplacementFor\_st)){ERROR(  "\x43\x61\x6e\x6e\x6f\x74\x20\x6c\x73\x74\x61\x74\x28");ERROR(  ReplacementFor\_path);ERROR("\x29\x3a\x20");ERROR(ReplacementFor\_strerror(  ReplacementFor\_errno));ERROR("\n");ReplacementFor\_had\_errors=  (0x167b+3173-0x22df);continue;}if(ReplacementFor\_S\_ISDIR(ReplacementFor\_st.  ReplacementFor\_st\_mode)){ReplacementFor\_wd=ReplacementFor\_add\_dir\_watch(  ReplacementFor\_dir\_wd,ReplacementFor\_path,ReplacementFor\_ent->  ReplacementFor\_d\_name,ReplacementFor\_no\_print);if(ReplacementFor\_wd<  (0x833+6829-0x22e0))continue;ReplacementFor\_add\_dir(ReplacementFor\_wd,  ReplacementFor\_path,ReplacementFor\_errors\_fatal,ReplacementFor\_no\_print);}}  ReplacementFor\_closedir(ReplacementFor\_dh);if(ReplacementFor\_errors\_fatal&&  ReplacementFor\_had\_errors)ReplacementFor\_exit((0x14e2+21-0x14f6));} |

2. Windows 10.

|  |
| --- |
| C:\Users\zorin>systeminfo  Имя узла: DESKTOP-1LD1IT8  Название ОС: Майкрософт Windows 10 Pro (Registered Trademark)  Версия ОС: 10.0.14393 Н/Д построение 14393  Изготовитель ОС: Microsoft Corporation  Параметры ОС: Изолированная рабочая станция  Построение ОС: Multiprocessor Free  Зарегистрированный владелец: Пользователь Windows |

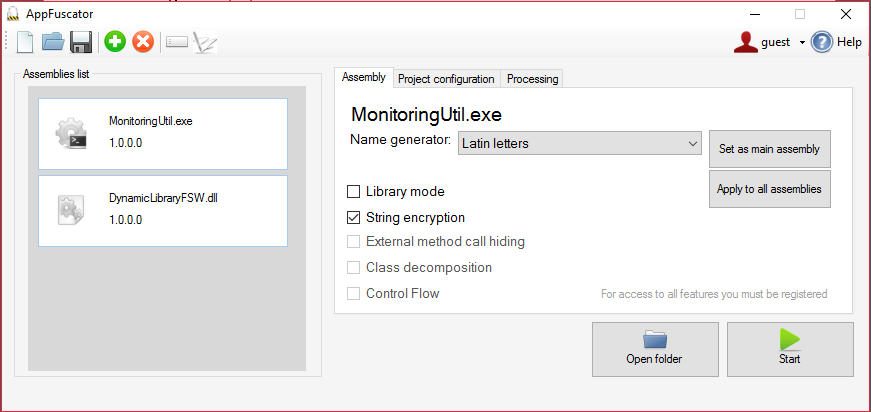
Для выполнения обфускации на ОС Windows была использована утилита AppFuscator. AppFuscator - это мощный протектор кода и обфускатор .NET приложений. Сочетая в себе ряд принципиально новых алгоритмов защиты, автоматический анализ кода и выгодную ценовую политику, он представляет собой оптимальное решение для защиты вашей интеллектуальной собственности. Данную программу можно скачать или воспользоваться демо версией на сайте.

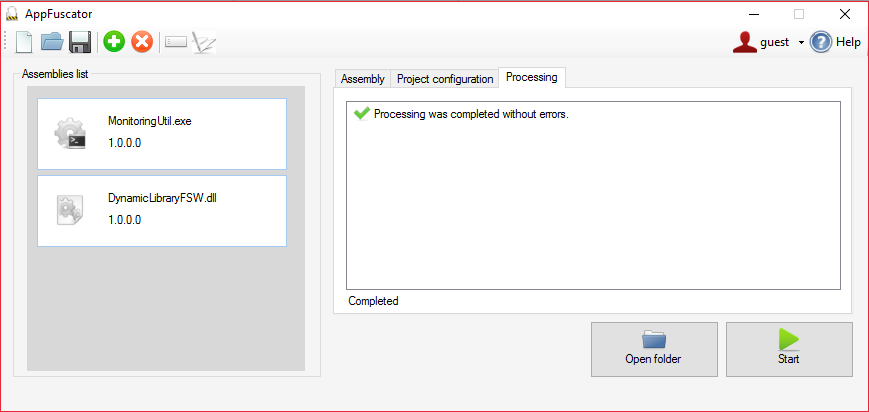
Утилита позволяет произвести:

* Переименование методов
* Шифрование строк
* Декомпозиция
* Создание водяного знака
* Запутывание потока управления
* Скрытие вызовов внешних методов
* Режим библиотеки
* Объединение сборок

Однако, большая часть из этих возможностей станут доступны только после приобритения полной версии. Пробная версия на месяц стоит 575 рублей. Допустимые расширения файлов для обфускации - .exe и .dll. Так как обфусцируются только исполняемые файлы и файлы библиотек, то для того, чтобы убедиться в корректной обфускации, воспользуемся утилитой от JetBrains, которая называется dotPeek. Эта утилита является бесплатной и позволяет произвести декомпиляцию.

Ниже показаны этапы обфускации с помощью используемой утилиты.





После произведения обфускации и декомпиляции, исходный код будет выглядеть следующим образом:

|  |
| --- |
| using StaticLibraryFSW;  using System;  using System.IO;  using System.Runtime.InteropServices;  internal class a  {  private static void a()  {  bool flag1;  bool flag2;  while (true)  {  string str1;  int num1;  do  {  \u003CModule\u003E.a = (object) null;  flag2 = true;  int num2 = 34693;  int num3 = 7424;  int n1 = \u003CModule\u003E.n;  int num4 = n1;  int num5 = (num3 & num4 ^ 1752323181) != 0 ? sizeof (ushort) + 35615 : ((1 & n1 - (n1 + 449)) == 1 ? Type.EmptyTypes.Length - 2113855040 : sizeof (ulong) - 1761516878);  int num6 = Type.EmptyTypes.Length + 64;  Console.WriteLine(\u003CModule\u003E.c(num2, num5, num6));  str1 = Console.ReadLine();  int q;  int num7;  try  {  \u003CModule\u003E.i = 2081625616;  int num8 = (((int) ((uint) n1 >> 4) | -1928415313) & 1315) == 1315 ? ((268435455 | (int) ((uint) n1 / 17U)) == 268435455 ? Type.EmptyTypes.Length + 23760 : Type.EmptyTypes.Length - 1991152871) : 504190888 ^ 999864120;  int num9 = checked (1218900621 - 1218875912);  int num10;  if ((int) ((uint) n1 / 16039U) != -1845109675)  {  int num11 = 5009;  q = \u003CModule\u003E.q;  int num12 = q << 20;  num10 = num11 + num12 + 483840 == (int) ((uint) ~(q \* 1073741824) >> 17) ? Type.EmptyTypes.Length - 1963321438 : Type.EmptyTypes.Length + 213;  }  else  {  int o = \u003CModule\u003E.o;  num10 = o \* 12966 - -131 != (int) ((uint) o % 256U / 1972U >> 23) ? Type.EmptyTypes.Length - 884098835 : Type.EmptyTypes.Length + 1457581078;  }  string str2 = \u003CModule\u003E.c(num8, num9, num10);  \u003CModule\u003E.l = -1592258590;  \u003CModule\u003E.r = -1871252905;  // ISSUE: variable of the null type  \_\_Null local1 = null;  \u003CModule\u003E.m = -1437277352;  \u003CModule\u003E.a = (object) local1;  \u003CModule\u003E.e = (object) 1386028750;  string str3 = str1;  int num13 = str2.Equals(str3) ? 1 : 0;  \u003CModule\u003E.o = 2136656571;  flag1 = num13 != 0;  if (flag1)  {  // ISSUE: variable of the null type  \_\_Null local2 = null;  \u003CModule\u003E.q = -1852116043;  \u003CModule\u003E.e = (object) local2;  throw new FileNotFoundException();  }  string str4 = \u003CModule\u003E.c(44666, sizeof (int) + 45636, Type.EmptyTypes.Length + 219);  // ISSUE: variable of a boxed type  \_\_Boxed<bool> local3 = (ValueType) flag2;  \u003CModule\u003E.l = -1410905245;  \u003CModule\u003E.k = 1845842485;  \u003CModule\u003E.h = (object) local3;  string str5 = str1;  \u003CModule\u003E.c = (object) ex;  if (str4.Equals(str5))  {  int num11 = sizeof (double) + 21715;  num7 = 22728;  int num12 = num7;  int num14 = sizeof (Guid) + 95;  string str6 = \u003CModule\u003E.c(num11, num12, num14);  \u003CModule\u003E.b = (object) null;  \u003CModule\u003E.i = 1057425350;  Console.WriteLine(str6);  \u003CModule\u003E.d = (object) flag2;  break;  }  \u003CModule\u003E.h = (object) null;  string str7 = \u003CModule\u003E.c(Type.EmptyTypes.Length + 8801, sizeof (uint) + 9765, sizeof (float) + 89);  \u003CModule\u003E.i = 1308380089;  \u003CModule\u003E.i = -1411494653;  \u003CModule\u003E.d = (object) str7;  \u003CModule\u003E.e = (object) null;  }  catch (FileNotFoundException ex)  {  int num8 = 1203310366;  \u003CModule\u003E.k = 1657774894;  \u003CModule\u003E.p = num8;  \u003CModule\u003E.m = -1557401652;  \u003CModule\u003E.q = 744302617;  Console.WriteLine(\u003CModule\u003E.c(((num7 << 16) + -2090598400 \* num7 & 59591) != 0 ? Type.EmptyTypes.Length - 1280742505 : sizeof (long) + 51678, Type.EmptyTypes.Length + 52731, Type.EmptyTypes.Length + 117));  continue;  }  catch (Exception ex)  {  Exception exception = ex;  c.b = (object) str1;  Console.WriteLine(exception.Message);  continue;  }  string str8 = \u003CModule\u003E.c(Type.EmptyTypes.Length + 41225, Type.EmptyTypes.Length + 42307, (\u003CModule\u003E.j & -2093841770 ^ 536876702) == 0 ? checked (-564057709 \* 2) : Type.EmptyTypes.Length + 177);  \u003CModule\u003E.o = -2051646939;  Console.WriteLine(str8);  int num15 = checked (6628 \* 2);  int num16 = 14319;  int num17;  if (((int) ((uint) n1 / 570U) ^ 33562096) != 0)  {  num17 = Type.EmptyTypes.Length + 32;  }  else  {  int m = \u003CModule\u003E.m;  num17 = ((int) ((uint) (~(int) ((uint) m >> 3) | ~m) >> 12) & -1915879424) != 917504 ? sizeof (ulong) - 1592093318 : checked (302205267 - 2054197228);  }  Console.WriteLine(\u003CModule\u003E.c(num15, num16, num17));  string str9 = Console.ReadLine();  \u003CModule\u003E.d = (object) \u003CModule\u003E.c(sizeof (byte) + 66540, 67633, checked (119 \* 2));  string str10 = str9;  bool flag3;  \u003CModule\u003E.a = (object) flag3;  Console.WriteLine();  string str11 = \u003CModule\u003E.c(Type.EmptyTypes.Length + 51532, sizeof (int) + 52405, sizeof (short) + 133);  \u003CModule\u003E.n = -1040838703;  c.b = (object) 1876936332;  string lower = str10.ToLower();  flag3 = str11.Equals(lower);  int num18 = flag3 ? 1 : 0;  \u003CModule\u003E.d = (object) ex;  if (num18 != 0)  {  string str2 = str1;  \u003CModule\u003E.e = (object) null;  int num8 = a.c(str2) ? 1 : 0;  \u003CModule\u003E.o = -1978466511;  int num9 = (11096 ^ (5499 | (int) ((uint) \u003CModule\u003E.n % 616U))) == 0 ? sizeof (Guid) + 822465074 : 21231;  int num10 = 33554432;  int r = \u003CModule\u003E.r;  int num11 = (int) ((uint) r / 711941971U);  int num12;  if (num10 \* num11 - int.MinValue != (int) ((uint) r / 5U))  {  num12 = Type.EmptyTypes.Length + 21913;  }  else  {  int num13 = -4862;  int o = \u003CModule\u003E.o;  int num14 = o \* 134217728 | 7869 | o;  int num19;  if ((num13 & num14) == 0)  {  int n2 = \u003CModule\u003E.n;  num19 = (int) ((uint) n2 / 349549U) + 1400504320 - 229896191 == (int) ((uint) (986 & n2) >> 8) ? ((int) ((uint) num7 % 14253U / 436U) != -863038997 ? Type.EmptyTypes.Length - 703588150 : Type.EmptyTypes.Length + 683881886) : sizeof (int) - 2077715009;  }  else  num19 = sizeof (ulong) + 1123732964;  num12 = num19;  }  int num20 = (q \* 111230976 | -9937) != -9937 ? Type.EmptyTypes.Length + 720440157 : Type.EmptyTypes.Length + 45;  \u003CModule\u003E.d = (object) \u003CModule\u003E.c(num9, num12, num20);  int num21 = 0;  \u003CModule\u003E.f = (object) 1957620381;  int num22 = num8 == num21 ? 1 : 0;  int num23 = 2097519326;  \u003CModule\u003E.n = -1932913121;  \u003CModule\u003E.r = num23;  \u003CModule\u003E.a = (object) null;  bool flag4 = num22 != 0;  \u003CModule\u003E.c = (object) flag1;  num1 = flag4 ? 1 : 0;  \u003CModule\u003E.q = -1950879357;  \u003CModule\u003E.a = (object) flag2;  c.b = (object) null;  \u003CModule\u003E.i = 1503776956;  \u003CModule\u003E.h = (object) str1;  }  else  goto label\_22;  }  while (num1 == 0);  break;  label\_22:  string str12 = str1;  \_\_Null local = null;  c.a = (object) ex;  \u003CModule\u003E.r = 1809257038;  \u003CModule\u003E.h = (object) local;  c.b = (object) 1952428595;  int num24 = a.b(str12) ? 1 : 0;  \u003CModule\u003E.i = -563903361;  \u003CModule\u003E.k = -1529522494;  \u003CModule\u003E.f = (object) 1818084011;  int num25 = 0;  if (num24 != num25)  \u003CModule\u003E.k = 1987339265;  else  break;  }  // ISSUE: variable of a boxed type  \_\_Boxed<bool> local4 = (ValueType) flag2;  \u003CModule\u003E.m = 796469985;  int num = -1051365525;  \u003CModule\u003E.p = 1335196033;  \u003CModule\u003E.r = num;  \u003CModule\u003E.o = -1980982856;  \u003CModule\u003E.g = (object) local4;  c.a = (object) flag1;  }  private static bool b([In] string obj0)  {  string str;  int num1;  int num2;  int num3;  do  {  int num4 = -3461;  int l = \u003CModule\u003E.l;  int num5 = ~(l \* (l - -7573 & l));  Console.WriteLine(\u003CModule\u003E.c((num4 & num5) != 0 ? (((int) ((uint) l / 83U) | 67108863) == 67108863 ? Type.EmptyTypes.Length + 63219 : sizeof (long) - 1746278992) : sizeof (long) + 1964230553, Type.EmptyTypes.Length + 63928, sizeof (byte) + 84));  Console.WriteLine();  Console.WriteLine(\u003CModule\u003E.c(checked (1033286927 - 1033232943), (l \* 26345472 - -5363 & l) - -15126 == 2368 + l ? (((l + l \* 3 ^ -8657) & 1) != 1 ? ((int) ((uint) (l \* 3 + 29 \* l - 6837) >> 5) - -4569 != l ? sizeof (long) - 636894119 : sizeof (Guid) - 1692008988) : -2095318113 ^ 163419891) : sizeof (double) + 54712, sizeof (uint) + 198));  Console.WriteLine(\u003CModule\u003E.c(38025 - -18909, ((int) ((uint) l / 1753U) & l \* 2 + 62 \* l ^ -1494) != 0 ? ((8655 ^ l \* 7 + l) != 0 ? sizeof (uint) + 58159 : ((l + l - -3545 | -2) == -1 ? (int) (IntPtr) (1246548022 ^ 86095112) : -1085959746)) : sizeof (ushort) - 943959491, sizeof (ulong) + 33));  Console.WriteLine();  WatcherEven.WatcherEvents(b.a(obj0));  str = Console.ReadLine();  Console.WriteLine();  int num6 = Type.EmptyTypes.Length + 2621;  int num7 = 4704;  int q = \u003CModule\u003E.q;  int num8 = q \* 1117 + 931 \* q;  int num9 = (num7 ^ num8) == 0 ? sizeof (float) + 1951619264 : ((128 & 2342656 \* l) != (128 & (l \* 20 + 12 \* l) \* 32) ? sizeof (int) - 830710559 : Type.EmptyTypes.Length + 3903);  int num10 = (1 & l) != ((int) ((uint) (1626296320 \* l - 921131914) >> 12) & 1) ? ((-2048 & (int) ((uint) l / 2097152U)) == 0 ? Type.EmptyTypes.Length + 1804492253 : sizeof (byte) - 1984861031) : 1073555530 ^ 1073555645;  if (!\u003CModule\u003E.c(num6, num9, num10).Equals(str.ToLower()))  {  num1 = Type.EmptyTypes.Length + 49528;  num2 = checked (1803311438 - 1803261351);  int num11 = 16;  int n = \u003CModule\u003E.n;  int num12 = n - 4313;  num3 = ((num11 \* num12 ^ (n << 4) - 2308) & 16) == 0 ? checked (1278922390 - 1278922346) : checked (682437489 - 1617698579);  }  else  goto label\_1;  }  while (!\u003CModule\u003E.c(num1, num2, num3).Equals(str.ToLower()));  goto label\_3;  label\_1:  return false;  label\_3:  return true;  }  private static bool c([In] string obj0)  {  string str;  do  {  Console.WriteLine();  Console.WriteLine(\u003CModule\u003E.c(51579, 52167, Type.EmptyTypes.Length + 160));  Console.WriteLine(\u003CModule\u003E.c(checked (1382980571 - 1382934679), Type.EmptyTypes.Length + 46505, \u003CModule\u003E.q \* 205472 != 4785 ? sizeof (int) + 42 : -358078368 ^ 781552629));  WatcherEven.WatcherEvents(b.b(obj0));  str = Console.ReadLine();  Console.WriteLine();  int num1 = 47683;  int num2 = Type.EmptyTypes.Length + 48829;  int minValue = int.MinValue;  int r = \u003CModule\u003E.r;  int num3 = (int) ((uint) r / 15U / 4U);  int num4 = minValue + num3 != -95248 \* r << 28 ? Type.EmptyTypes.Length + 200 : Type.EmptyTypes.Length + 2031769873;  if (\u003CModule\u003E.c(num1, num2, num4).Equals(str.ToLower()))  goto label\_1;  }  while (!\u003CModule\u003E.c(Type.EmptyTypes.Length + 24178, (3407 & (2863 | \u003CModule\u003E.l) - 7781) == 0 ? sizeof (uint) + 1804092126 : 25613, Type.EmptyTypes.Length + 171).Equals(str.ToLower()));  goto label\_3;  label\_1:  return false;  label\_3:  return true;  }  }  using System.Runtime.InteropServices;  public class c  {  public static object a;  public static object b;  public static string a([In] string obj0)  {  return obj0;  }  }  using System;  using System.IO;  using System.Reflection;  using System.Resources;  using System.Runtime.InteropServices;  using System.Text;  internal class \u003CModule\u003E  {  public static object a;  public static object b;  public static object c;  public static object d;  public static object e;  public static object f;  public static object g;  public static object h;  public static int i;  public static int j;  public static int k;  public static int l;  public static int m;  public static int n;  public static int o;  public static int p;  public static int q;  public static int r;  public static ResourceManager a([In] string obj0, [In] Assembly obj1)  {  string baseName;  switch (obj1.FullName)  {  case "MonitoringUtil, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null":  baseName = global::b.a(obj0);  break;  case "DynamicLibraryFSW, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null":  baseName = global::c.a(obj0);  break;  default:  baseName = global::b.a(obj0);  break;  }  return new ResourceManager(baseName, obj1);  }  internal static int b([In] string obj0)  {  int num1 = -1;  foreach (byte num2 in Encoding.UTF8.GetBytes(obj0))  {  num1 ^= (int) num2 << 24;  for (int index = 0; index < 8; ++index)  {  if (((long) num1 & 2147483648L) == 2147483648L)  num1 = num1 << 1 ^ 79764919;  else  num1 <<= 1;  }  }  return num1;  }  internal static string c([In] int obj0, [In] int obj1, [In] int obj2)  {  obj0 += 593;  Assembly executingAssembly = Assembly.GetExecutingAssembly();  obj1 -= 331;  Stream manifestResourceStream = executingAssembly.GetManifestResourceStream("resource");  int num1 = (obj0 ^ obj1) \* 17 / 27;  manifestResourceStream.Seek((long) (7 + num1), SeekOrigin.Begin);  byte[] buffer = new byte[8];  manifestResourceStream.Read(buffer, 0, 4);  int num2 = (BitConverter.ToInt32(buffer, 0) ^ 2100157544) - 100;  manifestResourceStream.Read(buffer, 0, 4);  int count = BitConverter.ToInt32(buffer, 0) - 5 ^ 485648943;  manifestResourceStream.Seek((long) num2, SeekOrigin.Begin);  byte[] numArray = new byte[count];  manifestResourceStream.Read(numArray, 0, count);  for (int index = 0; index < numArray.Length; ++index)  numArray[index] = (byte) ((uint) numArray[index] ^ (uint) obj2);  return Encoding.UTF8.GetString(numArray);  }  } |

**Вывод:**

В ходе данной лабораторной работы были изучены методы создания и запуска демона на UNIX подобных системах. Также, были изучены методы создания и установки служб и инсталяторов в Windows.

Помимо изучения теоретической части, была произведена практическая реализация. То есть, был создан демон, который работает как консольная утилита, но без захвата консоли. В системе Windows была создана служба, которая, пока она запущена, записывала в файл изменения наблюдаемой директории. И, в дополнение ко всему, был создан инсталятор, который в выбранную директорию устанавливает утилиту.

**Список литературы:**

1. Обфускация и защита программных продуктов http://citforum.ru/security/articles/obfus/
2. AppFuscator http://appfuscator.com
3. CXX-OBFUS http://stunnix.com/prod/cxxo/