УДК 004 (056.5+413.5)

Начев А., Иванов Р.П., Жаблянова Г.Б.

НЕКОТОРЫЕ ПОДХОДЫ ДЛЯ БЛОКИРОВКИ ВРЕДИТЕЛЬСКИХ “КАПКАНОВ” В ПРОГРАММАХ

Анотація:

Розглядаються два методи блокування шкідливих “капканів”, які реалізовані за допомогою лічильників та на основі часу з початку запуску програми.

Аннотация:

Рассматриваются два метода блокировки вредительских “капканов”, реализованные с помощью счетчиков и на основе времени с начала запуска программы.

Abstract:

The author presents two methods for generating ‘traps’ for malicious intruders, implemented through counters based on the starting time of the program.

Существуют различные способы ввода так называемых вредительских “капканов”. Из них два являются основными: скрытое задание времени стартирования вредительского кода и программирование скрытого “счетчика”, при достижении содержимого которого стартируется вредительский код.

Первый из них основывается на том, что все современные языки программирования содержат средства для коррекции и синхронизации системных часов. Актуализация времени в сетях – обязательная операция и на это рассчитывает вредительский код.

Второй подход использует программные конструкции для многократного выполнения программного кода, например while, for, foreach; Do…While. “Капкан” в этом случае поставляется с помощью superglobal переменной, т.е. с помощью такой переменной, чьи значения видны «со всех сторон». Значения этой переменной сравниваются не со значениями системных часов, а с количеством рекурсивных процедур, с количеством запуска самой программы и пр.

На рис. 1 представлена обобщенная схема проверки программного обеспечения при наличии вредительских “капканов”.

Существуют несколько методов проверки наличия вредительских “капканов”:

МЕТОД 1 Проверка на соответствие выполняемого кода с основным кодом.

Суть метода заключается в том, что в лабораторных условиях компилируется доставленное программное обеспечение.

МЕТОТ 2 Анализ подключений к базе данных и анализ источников данных.

Параметрами проверки являются программные конструкции, связанные с обращениями к базе данных.

МЕТОД 3 Проверка с помощью автоинкрементального файла.

Метод использует анализ на корректность генерирования автоинкрементального файла (далее «AUTOLOG» файл) за все события, связанные с регистрацией всех действий потребителей. Далее показан пример генерирования такого файла:

```
Implementation
Var
{$IFDEF DEB_FILE}
Dfile: Text;
{$ENDIF}
function GetTitle (Hwnd: THandle; Param: Pointer): Boolean; stdcall;
 var
Text: string;
begin
SetLength (Text, 100);
```

Рис. 1 Обобщенная схема проверки программного обеспечения при наличии вредительских “капканов”

```
GetWindowText (Hwnd, PChar (Text), 100);
 if Pos(UpperCase('Microsoft Word'), UpperCase(text))<>0 then
begin
h := hwnd; end;
Result := True;
end;
procedure Tri1aform.AutologAdd( EventInt:string);
begin
Append(DFile);
WriteLn(DFile, EventNow);
Flush(DFile);
CloseFile(DFile);
end;
initialization
{$IFDEF DEB_FILE}
AssignFile(Dfile, 'AUTOLOG.SEC');
 ReWrite(DFile);
{$ENDIF}
finalization
{$IFDEF DEB_FILE}
CloseFile(Dfile);
{$ENDIF}
```

Рассмотренный метод является одним из лучших и его развитие имеет существенное значение.

Литература:

1. Целков В. Софтуерни средства защита на информацията в компютърните системи за сигурност и отбрана / В. Целков, Н. Стоянов, З. Здравков, М. Божилова // Сборник материали «Първа международна научна конференция ХЕМУС – 2002». – Пловдив, 2002. – С. 134–138.
2. Целков В. Защитени криптографски приложения в компютърни системи и межи / В. Целков, Н. Стоянов. – София: Нова звезда, 2009. – 286 с.
3. Павлов Г. Защита на информацията / Г. Павлов. – София: Издателство «Стопанство», 2010. – 220 с.
4. Стоянов Н. Анализ на някои протоколи и стандарти за информационна сигурност / Н. Стоянов // Сборник материали «Първа международна научна конференция ХЕМУС – 2002». – Пловдив, 2002. – С. 213–218.