УДК 004.076

О. П. Войтович
І. В. Павленко

Створення дампу оперативної пам’яті

Вінницький національний технічний університет

Анотація

У даній роботі розробляється програмний засіб для створення дампу пам’яті в операційній системі Windows. Програмний засіб даватиме можливість створювати дамп оперативної пам’яті, в якому відображатимуться усі запущені процеси в операційній системі, їхня адреса в пам’яті, розмір та бібліотеки які вони використовують.

Ключові слова: Дамп пам’яті, оперативна пам'ять, API-функції.

Abstract

In this paper, we developed a software tool to create a memory dump of the operating system Windows. The software tool will give an opportunity to create a dump of memory, which will show all running processes in the operating system, their address in memory size and libraries they use.

Keywords: Memory dump, RAM, API-functions.

Дамп пам'яті - це копія вмісту оперативної пам'яті, що знаходиться на жорсткому диску або іншому енергонезалежному пристрої пам'яті. Природно, дампом може бути не вся оперативна пам'ять, а тільки якась певна її частина, яка, так би мовити, цікавить в даний момент ту програму, яка робить цей дамп [1].

Використовуючи дамп пам'яті, можна легко отримати повноцінний профіль системи.

В останні роки дамп оперативної пам'яті використовується при дослідженні комп’ютерних інцидентів [2]. Дослідники використовують спеціальне програмне забезпечення для захоплення оперативної пам'яті і зазвичай роблять це після авторизації на пристрої. Подібні дампи пам'яті використовуються для пошуку шкідників і прихованих процесів.

Дамп пам'яті, як правило, створюється при різних збоях в програмному забезпеченні (наприклад, в операційній системі), що призводять до його краху, але ще дозволяють запустити ту частину програми, яка призначена для збору інформації про причини збою. Власне, сама істотна область застосування дампів пам'яті якраз і полягає в зборі інформації про причини фатальних для програмного забезпечення збоїв в його власній роботі. Також за допомогою дампа пам’яті можна виявити шкідливі програми [3].

Метою дослідження є виявлення сторонніх процесів у пам’яті операційної системи Windows, при дослідженні комп’ютерних інцидентів.

Оскільки найбільш поширеною операційною системою є Windows, і більшість проблем та інцидентів з інформаційною безпекою трапляється з нею, то доцільним є розробка програмного засобу саме під цю операційну систему.

Серед API-функцій є функція «`CreateToolhelp32Snapshot`», яка виконує дамп процесів, а також купи, модулів і тем, використовуваних цими процесами [4].

Дамп, зроблений за допомогою цієї функції розглядається іншими функціями для виводу окремих результатів.

Прототип цієї функції:

```
HANDLE WINAPI CreateToolhelp32Snapshot (
  _In_ DWORD dwFlags,
  _In_ DWORD th32ProcessID);
```

Аргумент `dwFlags` визначає, який дамп необхідно отримати. Деякі можливі значення – `TH32CS_SNAPPROCESS`, `TH32CS_SNAPMODULE`, `TH32CS_SNAPTHREAD`. Параметр `dwProcessId` задає ідентифікатор процесу, для якого створюється дамп. Може бути непотрібним і встановлюватися в нуль залежно від `dwFlags`. У разі успіху процедура повертає дамп, який повинен бути потім закритий за допомогою процедури `CloseHandle`.

У випадку помилки повертається значення INVALID_HANDLE_VALUE, а подробиці допоможе дізнатися GetLastError.

Для того щоб витягнути інформацію про перший процес, що зустрічається в дампі, який зробила функція CreateToolhelp32Snapshot, використовується Process32First (hSnapshot,&peProcessEntry), синтаксис має такий вигляд:

```
BOOL WINAPI Process32First (
  _In_ HANDLE hSnapshot,
  _Inout_ LPPROCESSENTRY32 lppe);
```

Функція повертає TRUE, якщо перший запис у списку процесів був скопійований в буфер або FALSE в іншому випадку. ERROR_NO_MORE_FILES – значення помилки повертається функції GetLastError, якщо не існує ніяких процесів або дамп не містить інформації про процес [5].

Для отримання інформації про інші процеси, записані в тому ж дампі, використовується функція Process32Next.

Також використовуються функції Module32First та Module32Next, які використовуються для виводу модулів пов'язаних з процесом.

Прототип функції Module32First:

```
BOOL WINAPI Module32First (
  _In_ HANDLE hSnapshot,
  _Inout_ LPMODULEENTRY32 lpme);
```

Функції Module32First и Module32Next працюють аналогічно функціям Process32First та Process32Next. Функція Module32First повертає TRUE, якщо перший запис у списку модулів був скопійований в буфер або FALSE в іншому випадку. ERROR_NO_MORE_FILES – значення помилки повертається функції GetLastError, якщо не існує жодного модуля або дамп не містить інформації про модуль. Для отримання інформації про інші модулі, пов'язані із зазначеним процесом, використовують Module32Next функцію.

У роботі досліджено задачу створення дампу пам’яті в операційній системі Windows. Результатом проведеного дослідження є перелік функцій для створення дампу оперативної пам’яті, за допомогою якого можна відобразити усі запущені процеси в операційній системі, їхня адреси в пам’яті, розмір та бібліотеки які вони використовують. Досліджені та використані функції для отримання даму пам’яті, а саме: функція для отримання дампу процесів CreateToolhelp32Snapshot, функції Process32First та Process32Next для отримання першого і наступних процесів, що ідуть за ним, функції для отримання списку модулів, які використовує процес, Module32First та Module32Next.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дамп памяти [Електронний ресурс]. – Режим доступу: URL: http://www.kv.by/index2008061109.htm - Назва з екрану
2. Dieterle, D. (2011). Memory Forensics: How to Pull Passwords from Memory Dump. Cyber Arms – Computer Security. Retrieved 2013-11-18.
3. Дамп памяти Windows 7 [Електронний ресурс]. – Режим доступу: URL: http://www.winline.ru/os/windows_7/Damp_pamyati_Windows_7.php - Назва з екрану.
4. CreateToolhelp32Snapshot function [Електронний ресурс].–Режим доступу: URL: https://msdn.microsoft.com/enus/library/windows/desktop/ms682489(v=vs.85).aspx - Назва з екрану
5. Process32First function Електронний ресурс]. – Режим доступу: URL: https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms684834(v=vs.85).aspx - Назва з екрану

Павленко Іван Валентинович, студент, Вінницький національний технічний університет, м. Вінниця, факультет інформаційних технологій та комп’ютерної інженерії, 1БС-12б, vanyapavlenko7@gmail.com.

Войтович Олеся Петрівна, к.т.н., доцент, доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця.

Pavlenko Ivan Valentinovich, student, Vinnytsia National Technical University. Vinnitsa, Department of Information Technology and Computer Engineering, 1BS-12b, vanyapavlenko7@gmail.com.

Voitovych Olesya Petrivna, Ph.D. docent, docent of Vinnytsia National Technical University, Vinnytsia, Faculty for Information Technologies and Computer Engineering, chair of information security.