US 20140359766A1

(19) United States

(12) Patent Application Publication

KLEIN et al.

(10) Pub. No.: US 2014/0359766 A1

(43) Pub. Date: Dec. 4, 2014

(54) METHOD AND SYSTEM FOR PREVENTION OF WINDOWLESS SCREEN CAPTURE

(71) Applicant: **TRUSTEER LTD.**, Tel Aviv (IL)

(72) Inventors: **Amit KLEIN**, Herzliya (IL); **Michael Boodaei**, Givatayim (IL)

(21) Appl. No.: **13/905,650**

(22) Filed: **May 30, 2013**

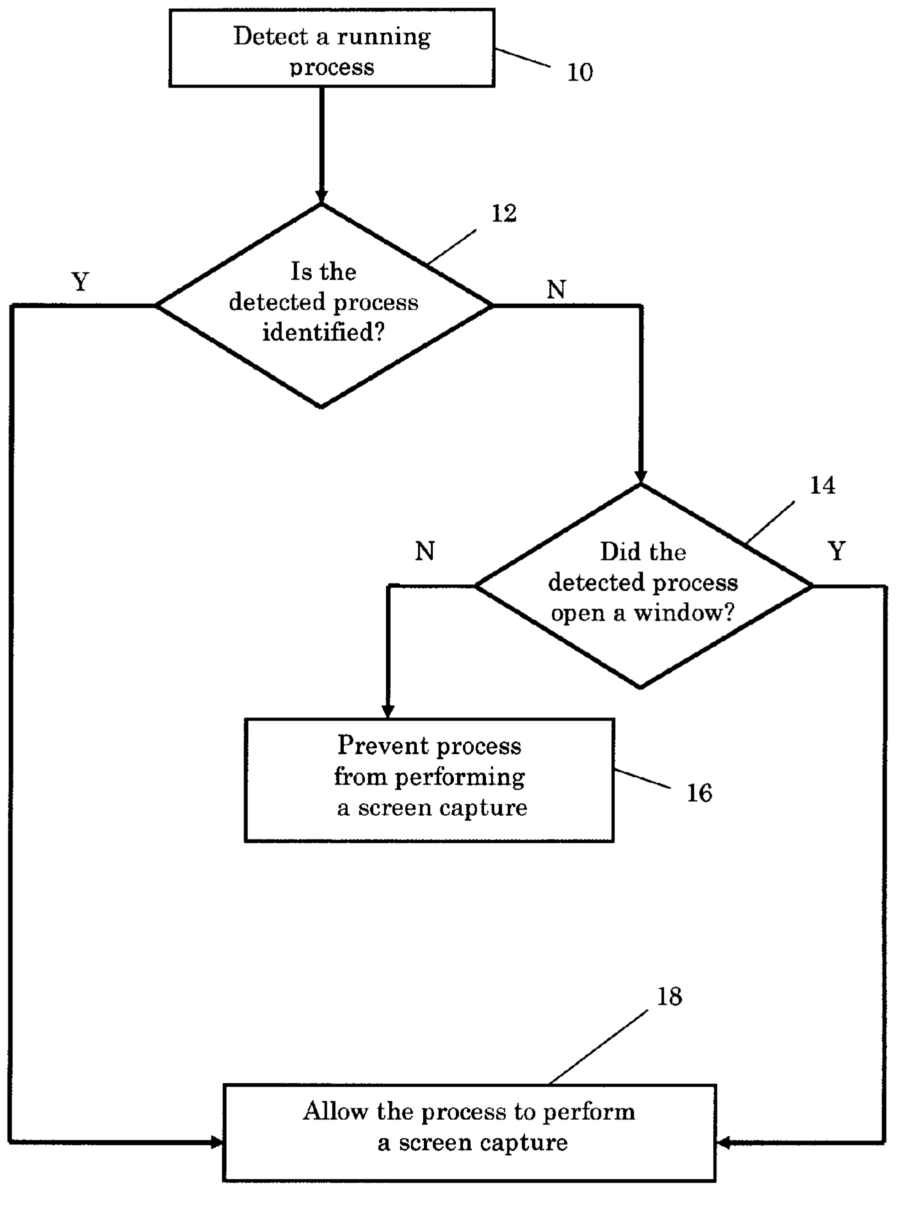
Publication Classification

(51) **Int. Cl.**
G06F 21/56 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 21/56* (2013.01)
USPC **726/23**

(57) **ABSTRACT**

A method for preventing the acquisition of data by a screen capturing malware, comprises preventing an unidentified process that does not open a window from performing screen capture.


Detect a running process
10
Is the detected process identified?
12
Y
N
Did the detected process open a window?
14
N
Y
Prevent process from performing a screen capture
16
18
Allow the process to perform a screen capture

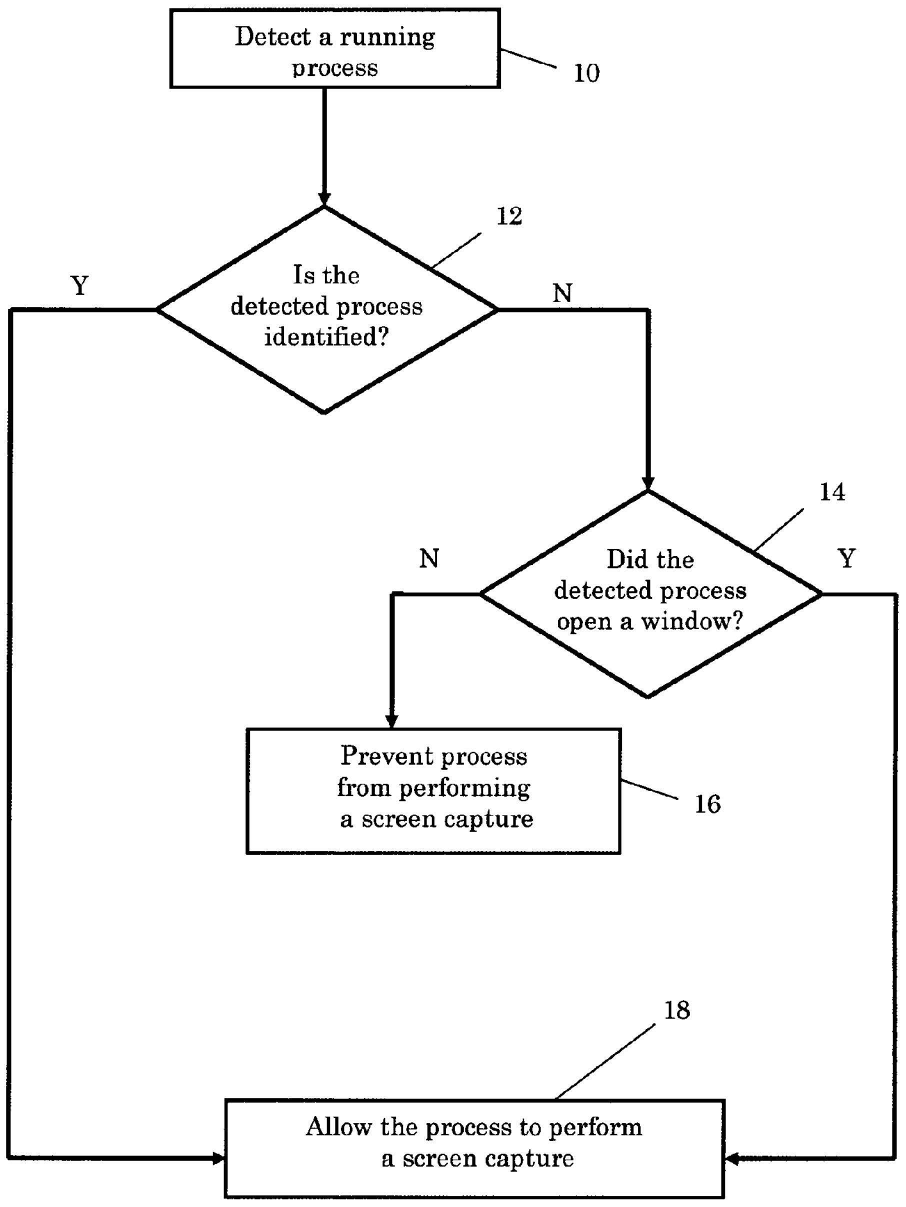

Detect a running process
10
12
Is the detected process identified?
Y
N
14
Did the detected process open a window?
N
Y
Prevent process from performing a screen capture
16
18
Allow the process to perform a screen capture

Fig. 1

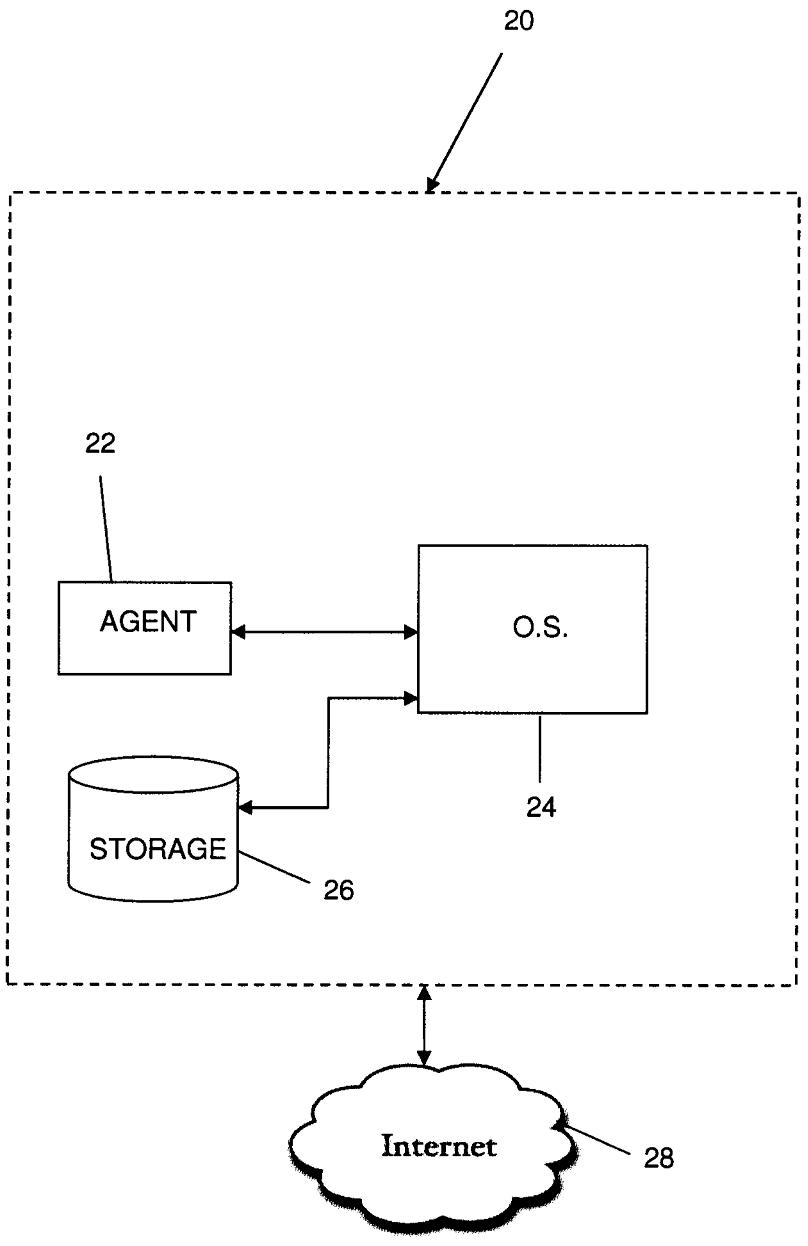


Fig. 2

METHOD AND SYSTEM FOR PREVENTION OF WINDOWLESS SCREEN CAPTURE

FIELD OF THE DISCLOSURE

[0001] The present disclosure relates to the field of computer security. More particularly, the disclosure relates to the prevention of the acquisition of private user data by malware that employs screen capture techniques.

BACKGROUND OF THE DISCLOSURE

[0002] Screen capture is a very useful option available in most computer operating systems and one which the vast majority of users would not be willing to give up. Unfortunately, the screen capture option also opens the door for malware to acquire private information entered by the user on his screen. For instance, when a user accesses a bank account or other sensitive web location, the user may type confidential information into specific fields to acquire access. Malware that is installed on said user's computer can be activated when such a window is displayed and may capture the screen, thus also capturing confidential information that the malware may then forward to its creator.

[0003] Some security software deals with this problem simply by preventing the system from performing a screen capture under a variety of conditions which give rise to suspicion. However, this approach has the great disadvantage of resulting in many "false positive" identifications of potential malware attacks, which lead to a frustrating situation in which the user is not able to perform a screen capture when he needs to. These and other drawbacks exist.

SUMMARY OF THE DISCLOSURE

[0004] The present disclosure illustrates methods and software agents to alleviate the aforementioned problem and greatly reducing the number of false positives determinations. This will improve the user experience and maintain a high level of security.

[0005] The present disclosure may provide a simple and yet efficient method of preventing malicious screen capture events.

[0006] The present disclosure may provide simple software agents suitable to carry out the method of the disclosure.

[0007] Other objects and advantages of the disclosure will become apparent as the description proceeds.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] In the drawings:

[0009] FIG. **1** is a flowchart illustrating the process of preventing the acquisition of data by a malware through the use of a screen capture; and

[0010] FIG. **2** illustrates a system for preventing the acquisition of data by a malware through the use of a screen capture.

DETAILED DESCRIPTION

[0011] One aspect the disclosure relates to a method for preventing the acquisition of data by a malware through the use of a screen capture approach, comprising preventing an unidentified process that does not open a window from performing screen capture.

[0012] According to an exemplary embodiment, the process is unidentified if it is not comprised in a white list. In this embodiment of the disclosure the white list may be constantly updated, to avoid too many instances in which legitimate process may be considered a potential threat.

[0013] According to another embodiment, a software agent may be useful in preventing a malware from acquiring data by performing a screen capture, where the software agent is configured to analyze a process to determine whether it does or does not open a window.

[0014] Another embodiment is directed to a system for preventing the acquisition of data by a malware through the use of a screen capture, comprising the software agent of the disclosure and a white list of non-malicious processes that do not open a window. In an exemplary embodiment, the system operates such that the white list may be a dynamic list, which may be updated by external input, or according to data generated within the system.

[0015] While reference will be made in the description to follow to a Microsoft Windows-based system, the skilled person will appreciate that the method of the disclosure can be applied, mutatis mutandi, to other operating systems. According to the disclosure, the only two types of processes that may perform a screen capture are 1) processes that by themselves open a window; and 2) processes that have been white listed, as further discussed below.

[0016] It has been found that the vast majority of malicious processes that are programmed to steal information by performing screen captures do not open a window in the user's system. Such malicious agents (i.e., malware) operate by stealth and without being noticed by the user or by a system that is not equipped with appropriate software protection.

[0017] A variety of software utilities exist, which may be used to detect that a process has an open window and, thus, is deemed to be legitimate. An example of such a utility supplied by Microsoft Corporation is Spy++, which is a part of Visual Studio and is a Win32-based utility that gives a graphical view of the systems processes, threads, windows and window messages. For ease of reference, a third-party utility that identifies the opening the windows is the well-known Spy++ utility. However, for the purpose of disclosure it is not necessary to use third-party utilities and the embodiments may be carried out to create an agent according to the disclosure, which may determine whether a process that is being inspected is or is not opening a window.

[0018] The system according to an exemplary embodiment may comprise at least the software agent that inspects processes to determine whether there is an open window, and a white list of non-malicious processes. The system may contain additional components, agents and elements, which are not part of the present disclosure and, therefore, are not discussed herein in detail for the sake of brevity.

[0019] In one embodiment, the danger of determining false positives may be reduced by providing and updating a "white list" of programs that do not open windows and yet are safe and non-malicious. According to this embodiment, when a process is attempting to perform a screen capture and the software agent determines that the process does not own an open window, the screen capture may be prevented if said process is not in the white list.

[0020] The white list may be a dynamic list that is updated as additional windowless processes that are not malicious. In an exemplary embodiment, the dynamic list may be updated with external data, for instance periodic updates received from a repository external to the system employed to guard the user's operating system from malicious attacks. In another embodiment, however, the dynamic list may be updated using data generated by the system of disclosure itself. For example, one embodiment may, upon detection of a screenshooting attempt from a windowless process, consult the user whether this process should be white-listed or not. According to another embodiment of disclosure, the Agent can keep track of processes and monitor them for window usage. So, if a process does not own a window at present, but did own a visible window at some point in the past, it will not be deemed a windowless process for purposes of screenshooting protection.

[0021] FIG. **1** is a flowchart illustrating the process of preventing the acquisition of data by a malware through the use of a screen capture. At the first step **10**, the agent monitors the computer and detects running processes. At the next step **12**, the agent checks whether or not an examined process is identified. If an examined process is identified, the process is allowed to perform a screen capture. If not, at the next step **14** the agent checks if the examined process attempts to open a window. If not, at the next step **16** the agent prevents the examined process from performing a screen capture. If the examined process did open a window, at the next step **18** the examined process is allowed to perform a screen capture.

[0022] FIG. **2** illustrates a system for preventing the acquisition of data by a malware through the use of a screen capture. The system **20** comprises a data storage unit **26** that houses a white list of non-malicious processes that do not open a window, a software agent **22** that is configured to analyze a process that runs on a computer, to determine whether the process opens a window and if the process is not listed on the white list and does not open a window, to prevent the process from performing a screen capture.

[0023] The dynamic list may be updated by external inputs received over a data network **28** such as the internet, to which the system **20** may be connected.

EXAMPLE

Discriminating Between Malicious and Non-Malicious Processes

[0024] An example of a malicious process is the recently discovered FAKEM RAT (http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-fakem-rat.pdf), which uses its own process ("the malware typically copies itself using the name, tpframe.exe, to the %System% folder"), and is capable of taking screenshots ("takes a snapshot of the desktop").

[0025] In contrast, various remote management software (e.g. Symantec PC-Anywhere) need to take screenshots of the user's desktop, while (in some cases) not having a visible window at such time.

[0026] By using the exemplary embodiments, it is possible to discriminate between a non-malicious management software, which is identified as non-malicious by its behavior, because at least some of its windows are visible and/or because they have been white-listed and malicious processes. The exemplary embodiments provide a simple and yet efficacious tool to fight malicious attacks of the type discussed above.

[0027] All of the above examples and explanations have been provided for the purpose of illustration and are not intended to limit the disclosure in any way. For instance, many different ways can be devised to determine whether a process does or does not open a window, with different operating systems or versions of operating systems, all without exceeding the scope of the disclosure.

1. A method for preventing the acquisition of data by a malware through the use of a screen capture approach, comprising:

detecting, on a computer, a process;

determining, on the computer, that the process is an unidentified process;

preventing the unidentified process from performing a screen capture when the unidentified process does not open a window.

2. The method according to claim **1**, wherein determining that the process is the unidentified process when the process is not in a while list.

3. A non-transitory computer readable medium for preventing a malware from acquiring data by performing a screen capture, said non-transitory computer readable medium configured to:

analyze a process running on a computer to determine whether the process opens a window; and

where the process does not open a window, preventing the process from performing a screen capture.

4. A system for preventing the acquisition of data by a malware through the use of a screen capture, the system comprising:

a data storage unit housing a white list of non-malicious processes that do not open a window; and

a software agent configured to:

analyze a process running on a computer to determine whether the process opens a window;

determine if the process is listed on the white list; and

prevent the process from performing a screen capture when the process does not open the window and is not listed on the white list.

5. The system according to claim **4**, wherein the white list is a dynamic list.

6. The system according to claim **5**, wherein the dynamic list is updated by external input.

7. The system according to claim **5**, wherein the dynamic list is updated according to data generated within the system.

* * * * *