Stopping Ransomware - the SPEEDOS Solution

James Leslie Keedy
Eislebener Str. 20, 28329 Germany
email: keedy@jlkeedy.net
formerly Professor/Honorary Professor at the University of Newcastle, NSW; Monash University, Melbourne, the Technical University of Darmstadt, the University of Bremen and the University of Ulm

Abstract:

The paper explains how the Speedos architecture automatically prevents ransomware via its fundamentally different logging in/out mechanism and its thread identification mechanism.

Keywords:

Ransomware, security, logging in and out, in-process architecture.

1 Introduction

Ransomware is a criminal activity whereby cyber-criminals gain access, then encrypt an entire computer system, making the contents for all practical purposes inaccessible to their owners. The criminals then demand a ransom in return for decrypting the data involved. The ransom must then be paid in an untraceable currency (e.g. Bitcoin). Since a criminal act is involved, it is of the utmost importance to the criminals that they remain anonymous. If they could be traced and apprehended they would probably land in jail, which would ruin their "business" model.

Unfortunately existing computers which are in common use do not contain mechanisms or tools which require an expert criminal to reveal his identity while carrying out his criminal activities; neither do they have mechanisms which prevent these activities. If they did, the ransomware problem would simply cease to exist.

The costs to business and to governments of ransomware attacks are huge and have been increasing annually in recent years. This trend is expected to put an ever higher burden on them. The German Federal Criminal Police Office released a press announcement on 16th August 2023 which stated that damages from cyber-crime in Germany in 2022 totalled **more than 203 billion euros**, which was around double the figure for 2019.

In this paper we describe how the SPEEDOS system can potentially eliminate the ransomware problem. In order to understand this the reader should first familiarise himself with the general architecture of Speedos (see https://www.speedos-security.org/) and especially the paper "Why Speedos executes Threads entirely in-process", which is available on the Speedos website [1].

2 How Ransomware Attacks Take Place

In order to attack a site, ransomware needs a thread or process which the perpetrators can use to encrypt the target. In conventional systems it is a simple matter for criminals to create a new process or thread; there are many possibilities for starting a process or thread. This is the fundamental difference with Speedos systems, as is explained in the paper "Why Speedos executes threads entirely in-process" [1].

3 Logging in and out in Speedos.

The Speedos mechanism for logging in and out is one of the keys to preventing ransomware. It is explained in detail in section 3.3 of [1]. Because Speedos is a persistent system [2] the entire content of the virtual memory is persistent, not only files and programs but also processes and their threads are persistent. This means that the current state of a process and its threads is also preserved automatically when a user logs out. Thus when a user wishes to log in again, he can in principle simply resume his work in the same state that he had left it.

Consequently there is no reason for the system automatically to delete his process on logout and to start a new process for him when he logs in again. That is clearly more efficient and more convenient for users.

But the idea of persistent processes brings a further advantage. It opens up the way for adding greater security to the login mechanism. If the final action which a user takes before logging out is to call a "logout" module (which he must do anyway to warn the process scheduler that his process should be temporarily deactivated) he can do this from a logout module *of his own choice*. Such a module (which is owned by the user, who can also determine what it does) can contain arbitrary checks devised by the user to check his own identity. This need not be a simple password; it can for example be a dynamic password, a cognitive password and/or whether the person attempting to log in has to conform to some required actions[1]. The kernel's role in the login and logout mechanism is trivial. In the case of logging in it simply advises the process scheduler that the user is active. This then activates the user's thread in the latter's logout module, which then validates the user or, if the checks fail, informs the process scheduler to deactivate the process again. Notice also that there is no central file which can be hacked to obtain login information.

4 What is the Relevance for Ransomware?

The primary advantage of the Speedos logging in/out mechanism is that it does not involve creating a process/thread but instead an existing thread is activated, but only when checks determined by the owner of the thread have been carried out. The nature of these checks is not determined by the system but by the user owning the activated thread. These cannot be known to a ransomware attacker (unless of course the owner of the thread reveals them to an attacker). Hence the attacker has no thread in which to encrypt a system.

There is a second advantage in the Speedos system. Even if an attacker were to gain control of a system the thread which he uses has a worldwide identity by which he can easily be found. There is no way to change this. Hence by decrypting a system the attacker would reveal his identity and hence his business model.

References

[1] J. L. Keedy, “Why Speedos executes Threads exclusively In-Process,” Speedos Website (https://www.speedos-security.org/), 2024.

[2] J. L. Keedy, “Persistent Programming with Speedos and Timor,” in *SPEEDOS Website (https://www.speedos-security.org/)*, 2024.

[3] J. L. Keedy, “Protecting and Confining Information with Speedos,” *Speedos Website,* 2024.

[4] J. L. Keedy, Making Computers Secure, Speedos Website, https://www.speedos-security.org/, 2021.

[1] Login Security checking is discussed in considerable detail in Making Computers Secure, volume 1, chapters 4 and 15, and in volume 2, chapter 22 which can be downloaded from the Speedos Website https://www.speedos-security.org/