Software Vaccine Technique and Its Application in Early Virus Finding and Tracing

Xiaodong Yuan
Xiaodong_Yuan@trendmicro.com.cn
China Development Center, Trend Micro Inc., Nanjing, P. R. China

Dajiong Yue
John_Yue@trendmicro.com.cn
China Development Center, Trend Micro Inc., Nanjing, P. R. China

Abstract: *The paper introduces a new anti-virus idea of Software Vaccine technique. It can apply the security technique inside the software itself. By inserting the software vaccine into the application software and reconstructing it, we can make the application software have immune ability from malicious attack. Besides, it can also help to find and trace the new viruses and their makers in very early stage. In this paper we will introduce the principle of software vaccine technique, give some samples of encryption algorithms to be used in software vaccine technique, and describe its application in early virus finding and tracing. We believe that the software vaccine technique will be a powerful anti-virus weapon to catch those virus makers.*

Keyword: Software vaccine, anti-virus, virus finding and tracing, encryption algorithm.

1. Introduction

Usually the application software do not have the self-protection ability from the malicious attack, which means, the malicious code such as viruses can infect them by attaching themselves on the application software and threat the computer system. So, they need the other software such as anti-virus software to protect them from the infection. In this paper we introduce a new idea that we can make the application software have immune ability from malicious attack by reconstructing them and adding a special part inside the software. We call this special part software vaccine. Just like people inoculate vaccine against some dangerous viruses, we also want to use software vaccine technique against the computer viruses.

In Section 2 we will introduce the principle of software vaccine technique in detail. Section 3 gives some samples of encryption algorithms that may be applied in software vaccine technique. The biggest benefit of software vaccine technique is that it can not only protect the application software from the virus infection but also find and trace the viruses in very early stage. With its help we are

more capable of catching the viruses makers and judge them in courtroom. We will spread out this benefit in Section 4.

2. Principle of Software Vaccine Technique

To use software vaccine technique, we need to reconstruct the application software to inject the software vaccine and encrypt the file. Let's say the structure for original software is like Figure 1. It consists of the entry section and multiple functional sections to implement various features. After the reconstructing it will be like Figure **2**. Decryption section and software vaccine section are inserted into the application software. Software vaccine section and all functional sections are encrypted and entry section points to decryption section now.

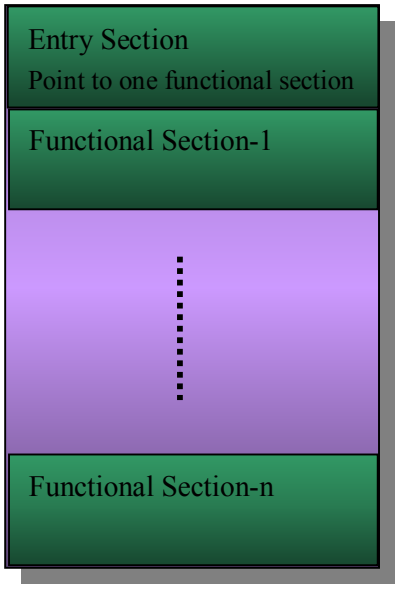

Figure 1: Structure for original software

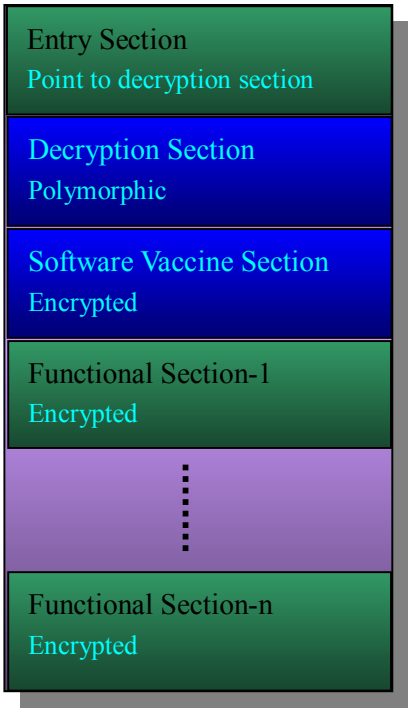

Figure 2: Software structure with software vaccine inside

During reconstructing the following steps are executed:

1. *Insert the software vaccine section into the software*. The software vaccine section will check whether the software was modified by any malicious code when the software is started. If any unwanted modification is found, the software vaccine will send notification to our Security Monitor Center and attach the infected software for further analysis, then it will stop the software from executing any more so that the security threat can be

prevented. We can create multiple software vaccines in our software vaccine library and insert them randomly into different application software. By this way the software vaccine section is polymorphic so that the hackers are almost impossible to develop new viruses to identify and crack all the software vaccines.

2. *Encrypt all functional sections and software vaccine section*. The encryption algorithm is not fixed but randomly selected from the encryption algorithm library, so it's also polymorphic. In this way we again make it much more difficult for virus makers to decrypt these sections. We will give some samples of encryption algorithms in Section 3.
3. *Insert the decryption section into the software*. The decryption section will decrypt the software vaccine section and functional sections after the software is started so that they can run properly. The decryption algorithm is the right one corresponding to the encryption algorithm used in step 2. Since the encryption algorithm is polymorphic the decryption algorithm is also polymorphic.
4. *Make the entry section point to the decryption section*. In the original software the entry section will point to some functional section after OS loads the software into memory. Now it will point to the decryption section and the decryption section will decrypt the software vaccine section and functional sections in the memory. Then the software vaccine will check whether the software is modified. If it is not modified, it will go to the original functional section to execute as usual.

After we reconstruct the application software with the above four steps to insert the software vaccine, its startup process will be like Figure 3.

There are many ways for the software vaccine to check whether the software is modified. For example, it can check the size of binary code such as .lib and .dll files. It can also calculate the CRC and compare it with the original one. Many other algorithms can be applied to check the completeness of software. We can easily create many software vaccines and apply them randomly to the software so that it's difficult for hackers to recognize and get rid of them. To make the software vaccine more complex we can refer to many skills currently used by virus programs.

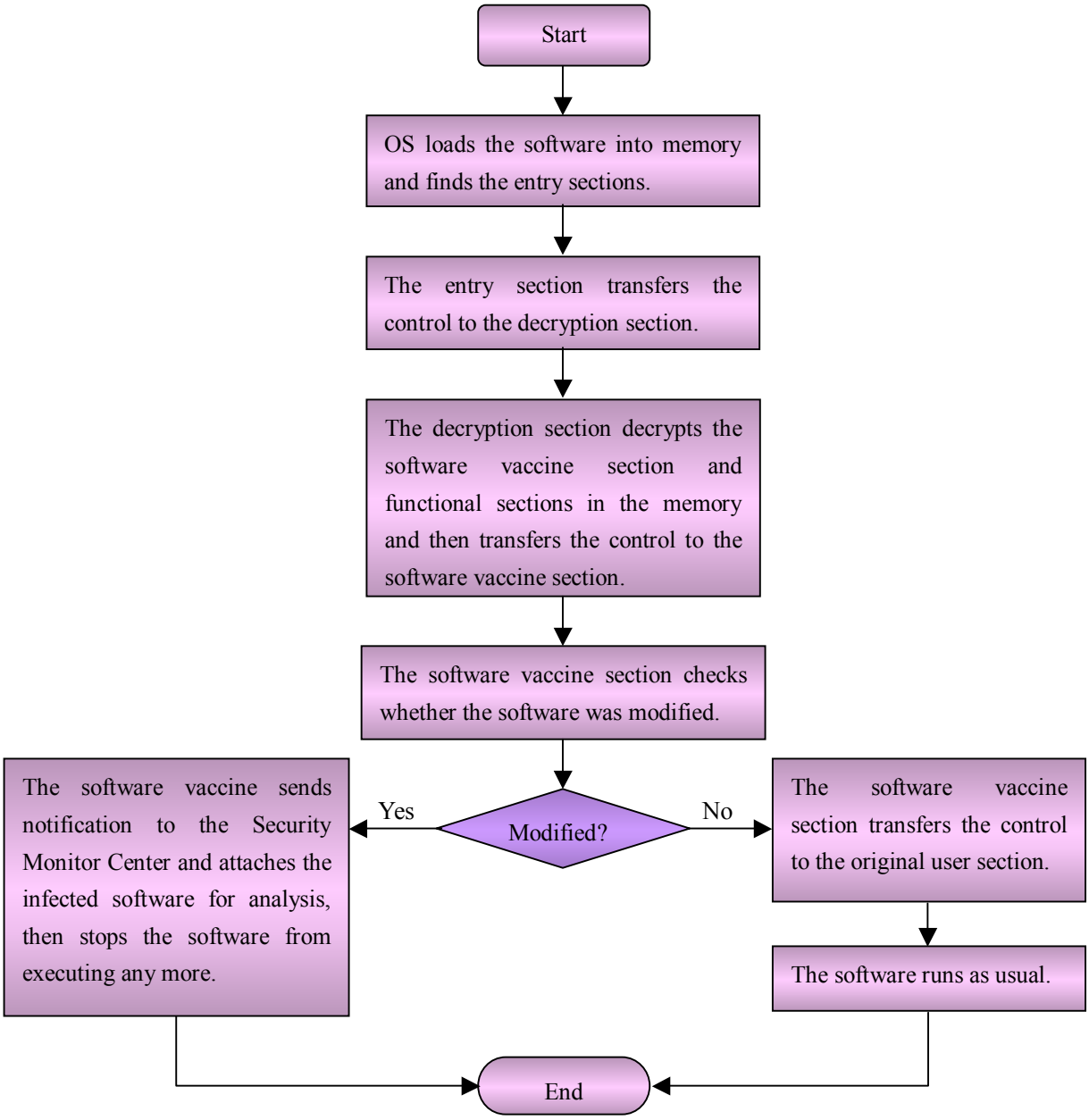

Figure 3: The startup process for software with software vaccine inside.

3. Samples for Encryption Algorithm

To apply software vaccine technique, engineer can still develop software with their accustomed method as usual. After they finish the normal software development process, the security enhancement tool can help to reconstruct the software as the steps mentioned in last section.

As we mentioned in step 2 and 3 of last section, the security enhancement tool needs to choose encryption and decryption algorithms. The following are some samples for encryption algorithms can

be chosen.

1. Simple XOR transformation.
2. Any public encryption algorithm[1]. No requirement to algorithm intensity if only it has corresponding decryption algorithm.
3. Any encryption algorithm developed by yourself. No requirement to algorithm intensity if only it has corresponding decryption algorithm.
4. The polymorphic and metamorphic algorithms[2][3][4][5].

We can collect many encryption algorithms and save them in the library. Then our tool can choose one encryption algorithm dynamically and randomly while encrypting the application software. To make the encrypted software executable in computer, each encryption algorithm needs to have a corresponding decryption algorithm and the decryption section will decrypt the application software after it's loaded into memory. The decryption section is not encrypted since it will be executed first. As for the software encryption, we have the following viewpoints:

1. No matter how intense the encryption algorithm is, it can only protect the encrypted sections. Since the decryption section itself is not encrypted, we can only protect the decryption section by using various encryption algorithms dynamically and randomly. Even the different copies of the same application software may adopt different encryption algorithms.
2. The polymorphic and metamorphic algorithms are the best algorithms we can adopt. The attackers are very difficult to develop an automatic analysis algorithm to decrypt each software copy and infect it.
3. Though the decryption section is not encrypted, it is still safe enough by using the above strategy.

4. Virus Early Finding and Tracing

One of the biggest benefits of software vaccine technique is that it can be applied to find and trace the potential viruses at very early stage. If most application software protect themselves with the software vaccine technique, it will be a hard time for those virus makers. Most viruses will be sent to our security monitor center and analyzed by the professional engineer as soon as they are just created and infect the first few application software. Since the application software in hackers' machines may also be protected by software vaccine, some viruses may even be sent to us directly from hackers' machines while they are still testing or debugging the new viruses.

With software vaccine technique definitely we can also prevent the new virus outbreak. Since we can get the virus code at the very early stage, we can analyze it and adopt the corresponding outbreak prevention policy and publish it to all customers.

Once the software vaccine technique is used widely, the hackers are sure to hate it and want to antagonize it. However, if they want to antagonize the software vaccine technique, they have to analyze the software vaccine code. We can develop a lot of software vaccines and apply them randomly to the application software. Because of the polymorphism of software vaccine, the price for hackers to analyze all software vaccines will be very high and not realistic. Since we can keep creating new software vaccines, they will have to maintain a pattern library for software vaccines if

they want to antagonize software vaccine technique effectively. If one day this is the case, we can say we are near the victory in the battle with computer viruses.

So, with the help of software vaccine technique we are much easier to trace the source of the new viruses and catch the virus makers. They will be judged in the court and punished by the law. Then the situation between hackers and security protectors will change a lot and the virus makers will have to consider the security for themselves. They cannot hide far behind our sight to attack the information network world any more. That is to say, they will pay much more efforts to avoid being found and punished by the electronic police and finally may find no way to escape. After more and more virus makers are traced, found, arrested and judged, most of them will give up evil and turn to good, contributing their great talent to serve our information network world.

5. Summary

This paper introduces a new anti-virus technique, software vaccine technique, describes its principal, implementation steps and its application in early virus finding and tracing. It is a general solution to self-protect the software from malicious attack. Making full use of the benefit of this technique, we can turn to attack from defense in the battle with computer viruses makers. We are expecting to move a big step toward the final victory. The further work includes implementing and consummating the software vaccine technique, applying it to more and more popular application software, and establishing software vaccine library and encryption algorithm library.

The technique itself does not have any faults. It depends on how people use them. Hackers use many advanced techniques to attack and destroy the computer system and information network, but now we can use the same techniques to protect the system and catch them.

Reference

[1] Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition. John Wiley & Sons, 1996

[2] http://vx.netlux.org/lib/static/vdat/eppoldis.htm

[3] http://vx.netlux.org/lib/static/vdat/epmetam2.htm

[4] http://vx.netlux.org/29a/29a-6/29a-6.316

[5] http://vx.netlux.org/29a/29a-6/29a-6.205