Undetectable Malwares: A Brief Survey

2016. 1

Soonchunhyang University

Dr. IIsun You
0. Outline of This Talks

- Signature based Malware Detection
- History of Undetectable Malwares
- Malware obfuscation techniques
- Today’s Research Missions
This talk is based on the paper “Malware Obfuscation Techniques: A Brief Survey” presented at the 2010 International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA 2010), Fukuoka, Japan, in November 2010.

It is cited 107 times according to the google scholar.
I. Signature based Malware Detection

• Signature
  ◦ A malware can be distinguished by using its unique string patterns in its binary code or a unique hash value of its binary code
  ◦ Such information are called “Signature”

• Traditional Malware Detection
  ◦ Traditional virus scanners mainly used signatures to detect malwares.
  ◦ Malware authors have developed “Undetectable Malwares” to avoid such a signature based detection.
I. Signature based Malware Detection

- Undetectable Malwares
  - Encrypted Malware
  - Polymorphic Malware
  - Metamorphic Malware
2. History of Undetectable Malwares

- Encrypted Malware
  - First approach to elude signature based detection.
  - An encrypted malware is composed of the decryptor and the encrypted main body.
  - For infection, a new key is generated and used to hide signature by encrypting the original main body.
  - In execution, the encrypted body is decrypted.
2. History of Undetectable Malwares

- Encryption Malware
  - Encrypted malwares have the main drawback: the decryptor remains constant from generation to generation.
  - Antivirus scanners can still detect these malwares based on the descriptor’s code pattern (i.e., signature).
2. History of Undetectable Malwares

- Polymorphic Malwares
  - Polymorphic malwares were developed to overcome the problem of the encrypted ones. With the help of “Obfuscation”, they can mutate their decryptor from one generation to the next.
2. History of Undetectable Malwares

- Polymorphic Malwares
  - Due to the powerful toolkits such as “The Mutation Engine (MtE)”, the polymorphic malwares were a critical problem.
  - The toolkits help a malware writer to easily convert her normal malware into a polymorphic version.
  - Important hint for detecting polymorphic malwares: Their constant body appears after decryption, thus being able to be used as an important source for detection.
2. History of Undetectable Malwares

• Polymorphic Malwares
  ◦ Antivirus tools adopt the emulation technique. The tools execute a malware in an emulator (called “Sandbox”) without resulting in any harm.
  ◦ Once the constant body is loaded into memory after decrypted, the conventional detection, i.e., signature based, can be applied.
2. History of Undetectable Malwares

- Metamorphic Malwares
  - As polymorphic malwares are not effective any more, metamorphic malwares are developed.
  - They make best use of obfuscation techniques to evolve its body into new generations, which look different but work essentially the same.

![Diagram showing metamorphic malware evolution](image)

The malware is obfuscated from one generation to the next.
2. History of Undetectable Malwares

- Metamorphic Malwares
  - It is important that the metamorphic malware never reveals its constant body in memory due to not using encryption or packing.

That makes it so difficult for the antivirus scanners to detect this malware.
What is “Obfuscation”?

- Obfuscation is a technology making programs harder to understand.
- Through this technology, a program is converted to a new different version without any change in its original function.

The original goal of this technology was to defend against Software Reverse Engineering.
3. Malware obfuscation techniques

- Obfuscation: Malware’s best friend
  - Generally virus scanners analyze and detect malwares based on their signatures.
  - Malware authors try to elude such a signature based detection by using “Obfuscation”.

![Diagram showing original malware and obfuscated malware with signatures are different]
3. Malware obfuscation techniques

- Obfuscation technologies can be divided into:
  - Dead-Code Insertion
  - Register Reassignment
  - Subroutine Reordering
  - Instruction substitution
  - Code transposition
  - Code Integration
3. Malware obfuscation techniques

- **Dead-Code Insertion**
  - Dead-code insertion is a simple technique that adds some ineffective instructions to a program.

A original malware is obfuscated by adding “NOP instructions”.
3. Malware obfuscation techniques

*Dead-Code Insertion*

- In order to make detection more difficult, some ineffective code sequences can be used.

A original malware is obfuscated by adding “ineffective code sequences”.
3. Malware obfuscation techniques

- Register Reassignment
  - Register reassignment is another simple technique that switches registers from generation to generation.
  - Note that the wildcard searching can make this technique useless.

A original malware is obfuscated by reassigning **EAX** and **EBX** to **EBX** and **EDX** respectively.
3. Malware obfuscation techniques

- Subroutine Reordering
  - Subroutine reordering obfuscates an original code by changing the order of its subroutines in a random way.

  This technique can generate $n!$ different variants, where $n$ is the number of subroutines. For example, Win32/Ghost had ten subroutines, leading to $10! (= 3628800)$ different generations.
3. Malware obfuscation techniques

- **Instruction Substitution**
  - Instruction substitution evolves an original code by replacing some instructions with other equivalent ones.

  For example, xor can be replaced with sub and mov can be replaced with push/pop.
3. Malware obfuscation techniques

- Instruction Substitution

A Sample Code

Instruction Substitution
3. Malware obfuscation techniques

- Code Transposition
  - Code transposition reorders the sequence of the instructions of an original code without having any impact on its behavior.
  - There are two methods:
    - **Using unconditional branches**
      This method randomly shuffles the instructions, and then recovers the original execution order by inserting the unconditional branches or jumps.
    - **Using independent instructions**
      This method creates new generations by choosing and reordering the independent instructions that have no impact on one another.
3. Malware obfuscation techniques

- Code Transposition

A Sample Code

Code Transposition based on Unconditional Branches
3. Malware obfuscation techniques

- **Code Integration**
  - This technique was introduced by the Win95/Zmist malware (called Zmist).
  - The malware knits itself to the code of its target program. In order to apply this technique, Zmist firstly decompile its target program into manageable objects, seamlessly adds itself between them, and reassembles the integrated code into a new generation.

  **As one of the most sophisticated obfuscation techniques, code integration can make detection and recovery so difficult.**
4. Today’s Research Challenge

- PLEASE SOLVE A REAL WORLD PROBLEM BY USING CRYPTOGRAPHIC OR SECURITY TECHNOLOGIES
  - Find a real world problem and explain why it is important.
  - Design an innovative model based on cryptographic or security technologies to solve the identified problem.
  - Briefly but concisely describe your model in 5 slides.
  - Submit the result to me via ilsunu@gmail.com until the midnight (24:00 on Jan. 07)
  - Present the model during the research contest
4. Today’s Research Challenge

• Examples
  ◦ How to detect Ransomware?
  ◦ How to allow more than two users to have a group chatting based on the End-to-End security for mobile messengers such as WeChat, KaKao Talk or Line? That is, the messenger server cannot know the messages exchanged among the users.
  ◦ How to know if a computer indeed exists at its claimed address? For example, a malicious user can try to launch attacks by using a victim’s address to hide his or her real address.