Undetectable Metamorphic Viruses

COMP 116

Amit Patel

Abstract

Signature scanning is an efficient technique employed by anti-virus systems to detect known malware. Signature scanning involves scanning files for a specific pattern such that it can uniquely identify a pattern that belongs either only in the virus or other similar malicious generated code. Many viruses use obfuscation to evade signature-based detection from antivirus software. One class of viruses called metamorphic viruses are difficult to detect because they mutate their internal structure and most antivirus engines assume that the structure of an application remains constant. The goal of project is to cover how metamorphic viruses work, metamorphic techniques, and ways that security researchers can detect metamorphic viruses.

Introduction

Before we dive into metamorphic viruses, it is worth discussing viruses in general. A virus is defined as "code that propagates across systems with user intervention" [8]. Viruses over time have evolved to become more dynamic and harder to detect. Encrypted viruses first started with a constant decrypter that is followed by the encrypted virus body. Detection was possible by signature scanning the code pattern of the decrypter since it would unique. Oligomorphic viruses changed the code pattern of the decrypters and detection based on the decrypter's code no longer was practical. Instead, detection became based on the constant code of the decrypted virus body. Polymorphic viruses on the other hand can create an endless number of new decrypters that use different encryption methods to encrypt the constant part of the virus body.

Metamorphic viruses do not have a decrypter, or a constant virus body. These viruses can create new generations that look different and do not use a constant data area filled with string

constants, but instead have one single code body that carries data as code [2]. One example of one of the first known metamorphic virus was the Win95/Regswap virus. This virus implemented metamorphosis by register swapping. Due to the simplicity of its code, this virus would be easy to detect since a wildcard string could uniquely identify it.

The simplest technique used by anti-virus software to detect computer viruses is string scanning. String scanning searches for an exact sequence of bytes that would be found in malicious code that is a part of a virus but not likely to be found in other programs. Each string can be created for a virus to form a signature that uniquely identifies it and can be organized in a database. The anti-virus software would then scan files and memory for the presence of the signature. Shorter signatures run the risk of introducing false positives since the signature could also end up detecting code belonging to a benign program. On the other hand, longer signatures run the risk of being too specific to a form of malicious software and is more likely to fail when trying to detect variations or future generations of the virus. Signature scanning may also feature wildcards. Wildcards are represented by the '?' character and can be used to skip bytes that are unknown [1]. However, signature scanning fails to detect more advanced techniques used by metamorphic viruses.

A metamorphic virus consists of several functional units. A metamorphic engine is composed of units. Below we cover a few units that a metamorphic engine may have. However, all are not necessarily featured in every metamorphic engine [1, 3].

1. **Locate own code:** This unit is responsible for being able to initially locate the code before transforming it.

2. **Decode:** The next unit is responsible for decoding the information needed to execute the transformations.
3. **Analyze:** The third unit is for analysis and constructing a control flow graph of the program.
4. **Transform:** The fourth unit is the transformation unit, which is responsible for converting the code into equivalent code for the next generation of the virus.
5. **Attach:** Finally, the last unit is responsible for attaching the new generation of the virus to a host file.

Metamorphic viruses employ a variety of techniques as outlined below [1]. Not all the techniques are required to be used for a virus to be metamorphic.

1. **Dead or garbage code insertion**: Inserting garbage or junk code is a technique done by metamorphic and polymorphic viruses to make their code appear different and break the ability to signature scan the code. The inserted instructions do not impact the execution of the program.
2. **Permutation of registers**: Metamorphic viruses can also use another technique known as register usage permutation. Each instance of a virus may have the same code but use different registers. This simple technique can be detected using wildcard strings however.
3. **Code permutation**: Two types of metamorphic viruses, Win32/Ghost and Win95/Zperm, introduced permutation techniques. These viruses divided the code section into frames and then permuted the frames. The frames were then connected using branch instructions to maintain the equivalent logical order.

4. **Subroutine permutation**: In addition, an extension of code permutation is subroutine permutation. For example, the Win32/Ghost virus also was able to change the order of its subroutines between generations.
5. **Instruction replacement**: Instruction modification is another technique used by metamorphic viruses. This technique lets these viruses replace their instructions with other equivalent instructions.
6. **Random jump instructions**: Another technique used by metamorphic viruses is randomly removing or inserting jump instructions within the code. For example, the first generation of a metamorphic virus may have a sequence of *n* instructions starting from the entry point. Later generations of the virus may insert random jump instructions such that the order of the instructions is physically rearranged but have the same logical format when executed.

To the Community

I choose this topic as I have been interested in the idea of self-modifying code and methods of obfuscating code to evade antivirus detection. Metamorphic viruses are particularly interesting to me due to their complexity and difficulty in detecting them. As an everyday user, we are to believe that antivirus software keeps us safe and that they can detect nearly all forms of malware. However, metamorphic viruses use techniques that commonly evade anti-viruses. Because they are considered one of the most infectious types of viruses, they pose a serious security risk to systems that lack detection mechanisms.

As a user, it is important to know that just analyzing the contents of a file is not enough, it is also necessary to understand the runtime behavior of a program to truly know what it is doing. As a security researcher, it is also important to consider the extent that may malware go in trying to evade detection. For example, while running malware in a virtual machine is a safe way to try to observe what it does, the malware itself can try to detect if it is running in a virtual machine and instead of immediately shutting down, it could perform primitive, dummy infection tasks to attempt to fool a security researcher and lead him or her astray. The researcher would then ultimately fail to uncover what the virus truly does.

Defenses against Metamorphic Viruses

While there is no guaranteed way to detect metamorphic viruses, using a combination of techniques greatly increases the probability and allows redundancy if certain techniques fail [2].

1. **Do not only rely on signature scanning**. Beyond a simple level of metamorphosis, signature scanning no longer is a viable method to detect metamorphic viruses. The ideal way to detect a metamorphic virus is to create a detection method that can regenerate the core instructions of the virus body from instances of the virus.
2. **Use geometric detection.** Geometric detection is detection based on changes that a virus made to its own file structure. One example is from W95/ZMist, which is a virus that increases the virtual size of the data section by over 32 KB while keeping the physical size of the section the same [2]. Nevertheless, geometric detection may yield false positives.

3. **Use code emulation**. An emulator is an application that simulates the behavior of a CPU. An emulator can allow a virus to execute in an observable environment that it cannot escape out of. To detect viruses, a scanner examines the memory of the virtual machine after a set number of iterations.
4. **Try machine learning based approaches**. Given the ability of metamorphic viruses to mutate their entire code body, some researchers have resorted to using more advanced detection techniques such as using hidden markov models (HMM) and genetic algorithms. A technique for detecting metamorphic viruses using HMMs was done by two researchers, Wong and Stamp, in [5]. A HMM is a state machine in which the transitions between states have fixed probabilities. An HMM can be trained to recognize common features between different generations of a metamorphic virus and output a high probability given an instance of a particular metamorphic virus it may have never encountered before.

Conclusion

Metamorphic viruses are a powerful form of viruses and require techniques beyond signature scanning to detect them. While the main application of metamorphic code has been for hiding malicious code to evade detection from antiviruses, metamorphic code can also be used to protect or obfuscate code in applications besides malware such as commercial software. While the paper discussed metamorphic viruses, similar techniques can be used in worms. One potential future development is a set of viruses that communicate with each other and exchange information [2].

While we have shown several ways to detect metamorphic viruses, there are also many ways to evade these techniques. A solution that detects a metamorphic virus one day perhaps may not necessarily be future proof. It is always a cat and mouse game between virus and antivirus developers. The evolution of these viruses over the years has been and will be a great challenge for security researchers and antivirus software developers.

Works Cited

[1] Konstantinou, Evgenios. “Metamorphic Virus: Analysis and detection,” 2008, Technical Report RHUL-MA-2008-2, Search Security Award M.Sc. thesis, 93 pages.

[2] Szor, Peter and and Peter Ferrie. “Hunting for Metamorphic,” *Virus Bulletin Conference*, September 2001, pp 123-144.

[3] Priti Desai, “Towards an Undetectable Computer Virus”, Master’s report, Department of Computer Science, San Jose State University (2008).

[4] Szor, Peter. "Advanced Code Evolution Techniques and Computer Virus Generator Kits." InformIT. N.p., 25 Mar. 2005. Web. 1 Dec. 2017. <http://www.informit.com/articles/article.aspx?p=366890>.

[5] Wong, Wing and Mark Stamp. “Hunting for metamorphic engines.” Journal in Computer Virology (2006) 2: 211. https://doi.org/10.1007/s11416-006-0028-7.

[6] Anderson, Hyrum S., Anant Kharkar, Bobby Filar, and Phil Roth. “Evading Machine Learning Malware Detection.” *Black Hat USA* (2017). Web. 1 Dec. 2017.

[7] Beaucamps, Philippe. “Advanced Metamorphic Techniques in Computer Viruses”. *International Conference on Computer, Electrical, and Systems Science, and Engineering* (2017). Web 1 Dec. 2017.

[8] Rousseau, Amanda. "Malware Techniques." *Malware Unicorn*. N.p., n.d. Web. 1 Dec. 2017. <https://securedorg.github.io/RE101/section2/>.