A Literature Review of Malicious Software Prevention and Threat Mitigation Techniques

Miguel Arindaeng, 100394094
4th-Year Software Engineering Student
2000 Simcoe Street North
Oshawa, ON, CA, L1H 7K4
905-721-8668
miguel.arindaeng@uoit.net

Devan Shah, 100428864
4th-Year Software Engineering Student
2000 Simcoe Street North
Oshawa, ON, CA, L1H 7K4
905-721-8668
devan.shah@uoit.net

ABSTRACT

In recent years, malware has become smarter and very sophisticated. While it might seem that malware might have an upper edge in hacking and thwarting computer systems, there is fantastic research going on that combats malware. Current research involves using the two primary schools of thought with combating malware: prevention and threat mitigation. This paper presents a literature review of the current research done in the both these schools of thought, an analysis of the research being done and how effective these methods are, thoughts of any future work on malware countermeasures, and an overall conclusion of the state of malware research today.

Keywords

Malware, malicious, malware prevention, threat mitigation, system policy, threat detection, behavior analysis, propagation, countermeasures, identification, removal, computer systems, security administrators

1. INTRODUCTION

Malicious software, or malware, is one of the most dangerous threats to computer systems to date. Malware is defined as: a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim [1]. As such, there is an inherent need to detect and prevent malware, as new techniques are being created everyday to disrupt the integrity of computer systems. Malware does not restrict itself to just the typical computer, and can be found in servers, mobile devices, embedded systems, and the like. Thankfully, there is extensive ongoing research that involves the prevention, detection, identification, and removal of these threats.

There are two schools of thought that are prevalent in developing malware countermeasures: prevention and threat mitigation [1]. Prevention from the threat of malware attacks can be thought of bolstering the defense of a computer or system. The idea is to proactively design and maintain systems so that they are not susceptible to malware attacks. This can be done through system policy, awareness, and vulnerability mitigation [1]. There are many techniques being researched that allow prevention to be easily attainable, such as: mimicking attacks on a system to prevent real attacks, understanding current malware threats, and analyzing how malware propagates throughout computer systems. The second school of thought – threat mitigation – can be thought of as how well a system would respond to threats that are already within the system (offense on malware attacks). This can be done through detection, identification, and removal techniques [1]. The idea is to improve these techniques so that malware attacks can be dealt with swiftly and efficiently. There is extensive research being done in this area, particularly detection techniques, that serve as the bulk of research done in malware threat mitigation. Modeling malware behavior, analyzing the data flow of source code, and creating custom applications and frameworks are just some of the topics involved in threat mitigation.

This paper goes over a literature review of the current research done in prevention and threat mitigation techniques (Section 2), an analysis of current research done (Section 3), thoughts of any future work (Section 4), and an overall conclusion of the state of malware research in the field (Section 5).

2. RESEARCH DONE IN PREVENTION & THREAT MITIGATION TECHNIQUES

There is quite an extensive amount of research done when it comes to malware. Some research involves the prevention aspect of malware countermeasures, but most of the research is from detection of malware attacks. The following sections are an in-depth look on these two schools of thought.

2.1 Prevention

Do not allow malware to get into the system in the first place, or block the ability to modify the system. This is the overall goal of using prevention to counter-act malware attacks (thought it is nearly impossible to achieve). There is one paper by the University Sains Malaysia's School of Computer Science that conducts malware behavior analysis using a dynamic approach [2]. This would fall under the

policy and awareness elements of prevention. It posits that dynamic analysis of malware behavior (dynamic analysis here is executing a program and closely observing its activities) is a promising method in strengthening system policy, and conducts an experiment in the local campus system to evaluate the effect of dynamic behavior analysis. The method they used consisted of four major steps which were: malware collection, behavior identification, custom behavior analysis, and statistical report [2]. The first step was to collect samples of current malware spread throughout the campus network. Using these samples they used malware analyzer tools to identify the different malware types that were apparent in the campus system [2]. Next, they used human-behavior analysis to further refine the results from the malware analyzer tools, and categorized the types of malware they saw to Worm and Trojan malware [2]. Finally they created a statistical report to visualize the kind of malware that were most common in the campus system [2]. From these results, they were able to strengthen system policy and awareness of common malware attacks on the campus system by implementing the appropriate security mechanisms to avoid these attacks [2]. While they acknowledge there were limitations to the malware tools that they used (sometimes the results could not even detect what kind of malware type was plaguing a certain system), the experiment gave them valuable information on the malware that plagued their campus system, and were able to adjust as a result. Many system security administrators can benefit greatly from this behavior-based dynamic approach, and is a great example of how prevention is the ideal solution in counteracting malware.

Another approach researched in malware prevention is predicting malware propagation. Malware is usually first classified on how it spreads or propagates to reach the desired target(s) [1]. As such, modeling propagation probability would be an intuitive first step in malware prevention. One paper by Xidian University in China uses a proposed malware propagation probability model (MPPM) for online social network services (SNS) [3]. While this paper focuses solely on the MPPM model for SNS, the techniques used can be extrapolated to other types of computer systems (i.e. cloud, mobile) as well. The basic model of MPPM can be summarized thusly: 1) Define several states for nodes and state transitions in an online SNS based on the characteristics of the SNS; 2) Introduce a detection factor that affects the propagation of malware and produce custom evolution equations for that SNS based on 1) and 2); and 3) Describe the relationships between user habits, malware propagation, and malware detection and use these in a simulation to choose the best policy to prevent malware propagation [3]. The results of their simulation using the proposed MPPM chartered the density of the susceptible, immunized, and malware disseminated states of the nodes in the SNS with respect to time. Using this, they analyzed real SNS data and showed that their MPPM model was quite accurate in predicting malware propagation [3].

Another paper created by various universities in the United Kingdom and Pakistan also focuses on malware propagation analysis, specifically in mobile P2P networks. As mobile devices and smartphones become more and more prevalent, the amount of new malware for these devices is growing at an alarming rate. As such, there is a need to analyze malware propagation in these devices so that they can sufficiently protect themselves from such threats. This paper, like the paper from the University Sains Malaysia's School of Computer Science, posits that by identifying which types of mobile malware that is prevalent, mobile devices can receive updated policies to combat the common malware types [4]. While there is much more detail to get into this type of malware propagation analysis, this paper exemplifies further the usefulness of such analysis in preventing malware attacks.

2.2 Threat Mitigation

Threat mitigation is usually the second (and the last) resort in counter-acting malware attacks after malware prevention. There are three main phases of threat mitigation: detection (determine that an infection has occurred and locate the malware), identification (identify the specific malware that has infected the system), and removal (remove all traces of the malware virus so it cannot be spread further) [5]. These phases are dependent on each other and come one after the other. Current research done on threat mitigation mainly focuses on malware detection, as malware identification and removal are relatively easy and simple operations for an infected computer system.

One paper by the University of Luxembourg analyzes malware behavior models. However, unlike the papers that analyze malware behavior to focus on malware prevention, this paper uses it to improve existing malware detection devices, particularly in their detection efficiency. It states that there is a current problem in signature-based malware detection (static analysis of malware) and behavior based malware detection (dynamic analysis of malware) [6]. With static analysis, malware can easily avoid detection by basic obfuscation techniques and by changing its syntax without changing semantics [6]. With dynamic analysis, scalability is a problem, as the number of malware features detected would grow proportionally to the size of execution traces (which can grow quite large) [6]. Thus, it makes detection using dynamic analysis impractical due to the unacceptable high computational complexity and memory consumption. This paper posits a more efficient behavior modeling technique, called BOFM (Bounded Feature Space Behavior Modeling) to combat the inherent inefficiencies of dynamic malware analysis [6]. In essence, BOFM extracts malware features that do not grow in proportion to the number of execution traces, as there is a hard upper bound limit to the extracted features. In addition, BOFM only extracts the minimum amount of malware features to detect a malware

attack, as opposed to traditional malware detection techniques that extract all of those features [6]. While the technical details of the paper delve into very specific research and findings, it was concluded that BOFM was a superior detection method over other studies, as evidenced by the vastly lower computation times and memory usage.

Another paper by some IEEE members suggests an interesting challenge that is unique to botnet attacks – mimicking legitimate cyberspace behavior in order to avoid detection. Botnets are a collection of bots capable of acting in a coordinated manner, and uses a variety of different attacks (i.e. spamming, sniffing traffic, keylogging, etc.) to annoy computer systems [1]. As such, botnets are capable of "smarter" malware attacks, and can use their cumulative computer power to avoid detection. Mimicking legitimate cyber behavior is the latest in botnet attempts to avoid detection [7]. Flash crowds (i.e. a large number of users accessing a website or service in a short period of time) are a common way for botnets to subvert a computer system. According to this paper, botnets need three pieces of key information to simulate legitimate behavior: web page popularity of the target website, the time interval of the web page request for the user, and the number of pages a user usually browses in one session (i.e. browsing length) [7]. Also, the one critical condition that botnets must satisfy in order to successfully mimic legitimate cyber behavior was this: the number of active bots in the botnet must not be lower than that of the legitimate number of active users [7]. This is because any number lower than the amount of legitimate users would result in the botnet being detected using current malware detection techniques [7]. In the end, the researchers found it hard to detect this kind of attack based on existing methodologies (feature-based or statistic-based methods). However, researchers found that the aforementioned critical condition was quite difficult to obtain, and so effective botnet attacks like this would come few and far-between. While there is ongoing research in this particular strand of malware detection, it is important to note from this paper that malware is continually evolving and getting smarter, and so malware detection techniques must continue on the same pace in terms of efficiency and time consumption.

While much has been said about malware detection techniques and behavior models, what about the improvement of actual malware detection tools? One paper from the 2012 IEEE Symposium on Security and Privacy describes a new tool – Rozzle (a JavaScript multi-execution virtual machine) – that explores multiple execution paths with a single execution so that environment specific malware will reveal itself [8]. According to the paper, the crux of the problem with detection tools nowadays is that they are very environment-specific, meaning that they only check specific plugins for malware behavior and fail otherwise [8]. As a result, detecting malware is very infrequent, showing itself only when the right environment is present. Using Rozzle in a couple of large-scale experiments, the researchers found that it increases offline runtime detection by almost seven times, and triples the effectiveness of online runtime detection [8]. While the paper goes into specific details about the architecture, design, and implementation of Rozzle, the key idea behind Rozzle is this: it executes all possibilities whenever it encounters some sort of control flow in the program [8]. For example, if there is an IF statement in the program, Rozzle will execute both branches (THEN and ELSE branches) of the program. The researchers have also found a way to reduce runtime overhead with this strategy and expose new attack directions in the program. New detection tools like Rozzle pave the way for future detection tools to become more robust and effective.

3. ANALYSIS OF CURRENT RESEARCH DONE

In the last section, there was a literature review of research done in malware prevention and detection. In this section, there will be an analysis of the ideas of what each paper presented, and an overall conclusion of what each paper means to malware research in general.

3.1 Prevention

In terms of malware prevention, much of the research done focused on analyzing malware propagation behavior. Analyzing such behavior would in theory update system policy and improve malware prevention. The first paper from the University Sains Malaysia's School of Computer Science analyzed their own campus network using dynamic behavior analysis. While they found their method to be effective in updating system policy (gathering malware samples, analyzing the malware samples for specific behavior, identifying the classes of malware that were most common on the campus, and generating a statistical report), this method can take a lot of effort and become time-consuming. This is especially the case, as they required human effort in identifying the types of malware in the system, rather than using automated tools for identification. The second paper from the Xidian University in China used a custom malware propagation probability model (MPPM) to chart how malware would progress in an online SNS network. While in theory this can be extrapolated to other types of computer systems (cloud, mobile), the methods and equations that the MPPM uses to model malware is very specific to SNS networks. In addition, their method required that the SNS system needs to already be infected with malware in order to model malware propagation properly, which might not be an option for mission critical systems. The third paper on malware prevention from various universities from United Kingdom and Pakistan analyzed malware propagation in mobile P2P networks. The main crux of their argument was that by identifying the ways that malware propagates in a system (i.e. via Bluetooth, MMS, SMS), system device policy can be updated easily to thwart malware attacks. This paper further exemplified the usefulness of malware propagation analysis in updating system policy.

3.2 Threat Mitigation

There was a lot more literature and research done in terms of malware threat mitigation, specifically for malware detection. The first paper reviewed in the previous section was from the University of Luxembourg. They analyzed malware behavior models to improve malware detection tools, or specifically, they proved impracticality of current malware detection tools and suggested a new way (i.e. Bounded Feature Space Behavior Modeling, BOFM) to improve them. Right now, detection tools extract malware features in proportion to the amount of execution traces that have been done. BOFM imposes an upper limit on the number of features that these execution features extract, and only extracts features that are essential in identifying the malware. The critique that comes with this method is that BOFM might not be flexible to all types of computer systems, as it requires a lot of effort to customize BOFM to a specific computer system. Also, it might be difficult to determine exactly which types of features to extract, as malware can be changed to accommodate all types of environments. The second paper by senior IEEE members suggests that botnet attacks nowadays try to mimic legitimate cyber behavior to thwart systems. In the end however, it was proved that it was hard to mimic legitimate cyber behavior with botnet attacks. The saving grace here is that these botnet attacks must satisfy a critical condition of having at least the same number of bots as legitimate users, which is quite hard to achieve. The third paper on malware detection from the 2012 IEEE Symposium on Security and Privacy goes into depth with Rozzle, a Javascript multi-execution virtual machine. The experiments with Rozzle proved to be quite successful, and paved the way for the future of malware detection tools.

4. THOUGHTS OF FUTURE WORK

The future of malware research techniques is very bright. There is extensive research going on in a variety of universities and there have been many interesting strides in malware prevention and threat mitigation. Behavior analysis seems to be quite the trend in updating malware system policies and provide the best defense against malware attacks. In terms of threat mitigation, there are a plethora of methods researched, particularly in the threat detection field. A lot of these methods are quite promising, and should give computer systems and security administrators even more options to defend against malware attacks. Overall, the research done in countering malware has been outstanding so far, and is at least keeping up with the pace at which malware evolves.

5. CONCLUSION

This paper presents an overall literature review of the state of malware research in the computer science field. It introduces the need for malware research, as malware continually evolves and becomes smarter. It then delves into six different papers that focus on malware prevention and threat mitigation, the two ways in which to combat malware. Many methods to thwart malware were discussed in those six papers, but it seems that the most prevalent method of dealing with malware is behavior analysis. It then analyzed the current research that was done on both malware prevention and threat mitigation, which summed up the overall state of each strategy. Finally, it provided thoughts of future work on thwarting malware, and the future seems very bright. With this paper, it is with hope that there is more awareness among computer professionals about the current state of malware, more knowledge on how to prevent and detect malware, and more enthusiastic development of future techniques to combat malware.

6. ACKNOWLEDGEMENTS

We would like to thank Dr. Khalid Hafeez in giving us the inspiration to create and finish this paper. Writing and researching for this paper gave us an enlightening education in computer security, and serves as one of the highlights in our Software Engineering education.

7. REFERENCES

[1] K. Hafeez, 'Malicious Software', University of Ontario Institute of Technology, 2015.

[2] M. Zolkipli and A. Jantan, 'Malware Behavior Analysis: Learning and Understanding Current Malware Threats', Ph.D, Universit Sains Malaysia, 2010.

[3] Z. Hui, C. Huang and H. Li, 'MPPM: Malware Propagation and Prevention Model in Online SNS', Ph.D, Xidian University, 2014.

[4] M. Adeel, L. Tokarchuk, M. Azam, S. Khan and M. Khalil, 'Propagation Analysis of Malware Families in Mobile P2P Networks', Ph.D, Cromwell College of IT and Management, Kohat University of Science and Technology, Queen Mary University of London, Middlesex University, 2014.

[5] W. Stallings, L. Brown, M. Bauer and M. Howard, *Computer Security Principles and Practice (2nd Edition)*. Upper Saddle River, N.J.: Prentice Hall, 2008.

[6] M. Chandramohan, H. Tan, L. Briand, L. Shar and B. Padmanabhuni, 'A Scalable Approach for Malware Detection through Bounded Feature Space Behavior Modeling', Ph.D, University of Luxembourg, 2013.

[7] S. Yu, S. Guo and I. Stojmenovic, 'Fool Me If You Can: Mimicking Attacks and Anti-Attacks in Cyberspace', Ph.D, IEEE, 2015.

[8] C. Kolbitsch, B. Livshits, B. Zorn and C. Seifert, 'Rozzle: De-Cloaking Internet Malware', Ph.D, IEEE, 2012.