A CLOUD BASED SOLUTION FOR GUARDING AGAINST SANDBOX AWARE MALWARE

Mruthyunjayam Allakonda[1], Jayanth Betha[2]
[1] Associate professor, [2] Assistant professor, Department of Computer Science & Engineering
St. Martin's Engineering College

Abstract
Due to increase in number of Internet connected devices, malware infections and data breaches have become so common. Recently authored malwares are intelligent enough to detect sandboxing, and they are capable of altering their behavior to evade detection. Sandboxing is a mechanism that runs a program in a secure environment. Visibility, Resistance to detection, and Scalability are must for any malware analysis sandbox. Malware is software that allows criminals to access sensitive information from the computing devices. To detect and guard against advanced malware infections that cause damage to computing devices, one should know how to handle the content that is reaching their device. The objective of proposed research is to suppress sandbox-aware malware from reaching a client. My research is mainly being focused on cloud-based solution for detecting suspicious malware that is aware of sandboxing.
Keywords: Sandbox-aware malware, Cloud Computing, Virtual browser.

1. Introduction

In today's world increase is usage on online applications is directly proportional to malware infections and data breaches. Malware is a software that allows criminals to access sensitive information from the computing devices. Among the present dangers to individual and business data, malware has ended up being the most noteworthy one. Sandboxing is a desirable solution for protecting users from malware. Sandboxing is a mechanism that runs a program in a secure environment. The objective of the proposed study is to present a solution for detecting sandbox-aware malware. In this research paper, I would like to provide a framework for automating the process of detecting sandbox-aware malware. Web browsers are acting as attack vectors for infecting computers. The best solution for protecting users from getting attacked is to stop malware from infecting browsers.

Malware performs different tasks like redirecting users to malicious websites; try to access sensitive information like usernames, passwords stored on the browser by exploiting browser weaknesses. Malware can also help cyber criminals to gain remote level system access. A lot of research is going ahead to fabricate a reliable malware investigation engine. Visibility, Resistance to detection, and Scalability are must for any Malware analysis sandbox. Recently authored malware are intelligent enough to detect sandboxing, and they are capable of altering their behavior to evade detection. The main focus of my research is on Isolation aware malware. In this research paper, I would like to provide a framework for automating the process of detecting sandbox-aware malware without any malicious web content reaching client's computing device. Any browsing request or response is redirected to a cloud-based virtual browser where it is scanned for malicious content, and a signature of anything suspicious is maintained in the cloud database (to guard against zero-day attacks). For this purpose, I will consider the study of Rodriguez, R. J., Gaston, I. R., & Alonso, J. 2016. My research also includes an in-depth study of malware detection methodologies.

Research Objectives:

- The objective of the research is to provide a solution for protecting browsers from sandbox-aware malware which also involves the study of different types of malware.
- Propose the best solution for guarding against sandbox-aware malware using Cloud and virtual browsing technologies.
- Automate the process of detecting sandbox-aware malware and stop any malicious web content from reaching client device.

Sandbox is an isolated computing environment in which a program is tested without affecting the application in which it runs.

Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that provides a minimal management effort and service provider interaction.

Virtual Browser is an Application used to isolate a web-browser from the systems underlying operating system.

2. Literature Review

2.1 Motivation

Malware investigation, which includes malware detection and prevention, remains a rich domain of research for years from now. As the Internet is turning into a critical piece of regular day to day existence, Internet clients confront expanding security dangers postured by malware. Malware is a piece of code that can create pernicious effects, which when enters into a PC performs an undesirable task. The unsafe exercises can degenerate and erase the records on the PC, stealing individual's sensitive pieces of data, taking private or protected innovation data and at a bigger scale decimating PCs. A portion of the cases of the malware incorporates viruses, worms, Trojans, root kits, keyloggers, spyware, ransomware, adware and so on.

3. Studies confirm that a million new pieces of malware hit internet connected devices every day and growing exponentially. Insecure browser environments cause spreading the majority of malware infections. Browser-borne malware is a very significant threat to individuals and organizations today because the browser is the primary source for surfing internet. Older or outdated versions of browsers with many vulnerabilities act as common attack vector that allows attackers to inject malware codes. Injected Malware can damage information stored on computers. Traditional detection techniques are becoming ineffective in guarding against upcoming malware. Researchers say that 41% of web-borne malware was able to bypass intrusion detection systems. These studies reveal that there is a need for an effective solution to ensure that no browser-borne malware can gain access to sensitive pieces of information from individual or organizations computing devices. The amount of new malware out there is tremendous, with new varieties showing up every day. It's new malware that is most dangerous, as it has not been added to the databases of security products yet.

2.2 Disadvantage of the previous literature:

Many users depend on signature based anti-malware systems but, Kevin W. Hamlen, Vishwath Mohan, Mohammad M. Masud, Latifur Khan, Bhavani Thuraisingham (2009) in their research proved that attackers could easily bypass these mechanisms using techniques like encryption and packing. Attackers are penetrating into corporate networks using a bit social engineering to trick employees into visiting compromised websites. By taking advantage of browser vulnerabilities, attackers are injecting malware codes through WebPages to gain access to victim's computer.

One of the traditional solutions to guard against malware is application or browser sandboxing. Sastry Pendyala (2014) in a web security article clearly explained the purpose of sandboxing. Sandbox is a virtual environment created to test malware activity. Sandboxes are used to monitor the isolated suspicious malware samples by executing them in a controlled virtualized environment. This technique allows analysts to identify the behavior of malware based on its propagation. Which would help in determining whether the sample is a variant of already known sample or an unknown but malicious sample that requires further analysis?

Visibility, Resistance to detection, and Scalability are must for any Malware analysis sandbox. But, recently authored malware are intelligent enough to detect sandboxing, and they are capable of altering their behavior to evade detection. Ricardo J. Rodriguez, Inaki Rodriguez Gaston, Javier Alonso (2016) proposed a solution for detecting sandbox-

aware malware. Bickford & Giura, (2015) proposed a method to safely browse the internet using Transparent Virtual Browser, where web requests are redirected to cloud-based Transparent Virtual Browser to protect the user from malware.

3 Design Overview

Malware filtering is critical because browsing the Internet is the first infection vector. Web security administrators firmly believe that web browsing is a serious malware risk. Phishing is the most common Web-borne attack affecting online account holders. Current Defense in depth strategies like, Antivirus and antispyware software's, web gateways, Intrusion Detection and Prevention systems, Malware analysis engines which are supposed to detect and block browser malware are failed to do so. Cyber criminals have the ability to bypass all security layers with the help of crafted browser malware that is completely undetectable. Recent surveys result that current solutions are not efficient to guard against web-borne malware. The existing security tools are not capable of detecting web-borne malware. So there is a need for a more appropriate solution in defending against web-borne malware.

3.1 Types of Malware:

Adware: Adware can automatically deliver advertisements through pop-up windows. Pop-ups will try to attract users by putting ads of fake software like PC Cleaner, free Virus scan that comes bundled with malware. The intention behind most adware programs is to promote the product based on web surfing patterns and generate revenue, but few adware programs are solely designed to deliver exploits. Most of the adware programs come bundled with spyware which can track user's activity for stealing sensitive information. A pop-up will force users to redirect to a malicious website. Visiting a malicious website opens the door for different attacks.

Ransomware: Ransomware is a malware that tries to encrypt files on a computer system. The malware confines client access by necessarily encrypting files on hard drive and displays a message to pay ransom for remove malware or to decrypt files. Ransomware commonly spreads like an ordinary PC worm. The user often gets affected by this kind of malware through clicking links in emails from unauthorized parties.

Rootkit: A rootkit is a malicious software program designed to operate computer system by hiding deep inside system kernel remotely. Once a rootkit is installed it is possible for the attacker to execute files on the compromised system remotely. The rootkit malware infected system can act as a botnet for DDOS attack. DDOS attack is an attempt to make a machine or network resource unavailable to those trying to access it. Detection and removal of rootkit malware is challenging because of its stealthy nature. To detect and prevent the system from this kind of malware it is compulsory to monitor computer system for any malicious activity.

Spyware: Spyware is a malware that can spy on user activities which include logging keystrokes, capturing monitor screen, harvesting confidential information and more. Spyware can also help an attacker to modify browser security settings. Changing browser security settings can lead to the unauthorized capture of networking information. Spyware often comes bundled with Trojans.

Trojan horse: Trojan horse is a most dangerous Malware. A Trojan can give the attacker remote access to an infected computer. A Trojan will allow an attacker to install more malware which improves the severity of the attack on the targeted system.

The above described five types of malware are the most commonly seen malware that affects computer users. It is important to notice that root causes for systems getting affected by these malwares are vulnerabilities in system applications and lack of security measures. After a preliminary study on different malware, it is noticed that primary cause for malware infection is nothing but a vulnerable web browser. One stop solution for protecting users from malware is to patch vulnerabilities in operating system and application software's on a regular basis.

Table 1. The table given below describes name of the malware, effects of the malware on the computer system and application that acts as injection vector for attacker to install malware.

Type of Malware	Effects of malware	Injection vector
Adware	• System slows down. • Slow internet connection. • Screen flooded with unwanted pop-up advertisements. • New toolbars in the web browser. • Changes in browser homepage. • New programs in start-up program list. • System freezes/crashes for no reason.	• Vulnerabilities in browser. • Visiting malicious websites. • Downloading software from untrusted websites.
Ransomware	• Encrypts data/files on computer. • Demands for ransom to decrypt data/files. • Encrypts data on other computers connected in the LAN (Local Area Network).	• Vulnerabilities in browser. • Visiting malicious websites • Opening links in spam emails.
Rootkit	• Affect performance of the computer. • Directly attacks system kernel. • Allow remote code execution.	• Vulnerabilities in browser. • Visiting malicious websites. • Downloading software from untrusted websites.
Spyware	• System slows down. • Slow internet connection. • The increase in internet traffic.	• Vulnerabilities in browser. • Visiting malicious websites.
Trojan horse	• Corrupts data or applications at the core of the operating system. • Records keystrokes (runs keylogger). • Captures screen. • Allow remote code execution.	• Vulnerabilities in browser. • Visiting malicious websites. • Opening links in spam emails. • Downloading software from untrusted websites.

A far-reaching web security approach is tough to implement. The fact that, there are so many filtering and antivirus solutions on the market fail to stop newly created malware. A Web filter is a program that scans incoming Web traffic to differentiate safe and unsafe content. The designed filter checks content of the Web page against a set of predefined rules to identify malicious content. If the filter suspects any malicious content like viruses, spyware or Trojans on the scanned web page it blocks out the pages from getting displayed on the browser. Using Web filter is one of the standard methods to stop Web-based threats. Researchers proposed several methods for detecting web-based malware. James Harland in his work presented an overview of web-based malware, briefly describing typical architecture encountered when using the Internet as an attack vector. The strength of a malware strongly depends on its ability to remain undetectable. Malware has the power to destroy the computer if it remains undetected during the security checks.

Malware detection attempts to identify programs with malicious intent. Malware detectors can operate using either signature-based or anomaly-based detection, or a combination of both. Antivirus solutions traditionally use Signature-based detection. Anomaly-based detection is performed to

examine the instructions contained in the binary (code). The best way is to execute code in a virtual environment. Information gathered from the program execution is used to determine if a binary is malicious also known as dynamic analysis. But, the above techniques fail to detect Oligomorphic, Polymorphic and Metamorphic malware. However, malware developers adopt various methods to avoid detection. One such technique is code obfuscation. Code obfuscation makes code unclear for detection engines by preserving the programs functionality. No matter what may be the technique malware developers are generating codes and exploits that can bypass active security mechanisms.

Sandboxing is a method implemented by latest browsers to make internet users stay away from web malware. Sandbox is a virtual environment where the malware activity can be tested. In fact, we can use Sandboxes to execute malware codes without harming existing computing environment. Sandboxes are used to run application data such as Adobe Flash and JavaScript etc. Once running its behavior can be analyzed which allows analysts to identify how malware propagates and make changes to file system and registry. Cuckoo Sandbox is one of the widely used sandboxes for malware analysis. The Increased utilization of Sandboxes for the dynamic analysis of malware has incited the malware developers to write codes that can detect the presence of sandbox and sidestep the sandbox level investigation of the malware. Researchers stated that a well-crafted malware could avoid sandbox detection. Intelligent Malware can detect the environment it is operating in, and do not exhibit its malicious function to evade analysis. Malicious software has increased in number and complexity over the past decade. That is why malware samples are analyzed in an isolated environment to prevent infection from spreading and the proper execution of the machines. In this sense, sandbox and virtualization have become a trend in malware analysis. The more time passes the sample without being detected, the greater the benefit. Thus, to maintain the criminal enterprise, malware developers have begun to incorporate code to recognize when the code starts executing in a test environment. This kind of malware is known as analysis-aware malware. These samples change their behavior or even completely stop execution when they recognize isolation environment analysis. Rodriguez, R. J., Gaston, I. R., & Alonso, J. (2016) reviewed the techniques used to identify context-aware analysis focusing on detecting isolation- aware malware. They proposed a tool PinVMShield which makes use of Dynamic Binary Instrumentation, which allows analyzing the runtime behavior of a binary and executing arbitrary code. They first reviewed the most relevant evasion techniques used by malicious software conscious; that is, those methods which are used to detect and recognize an isolated environment analysis. Then the proposed dynamic binary analysis tool (DBA) that allows obfuscating the isolated environment analysis and observed that known malware fails to detect such an environment, and it shows all its malicious behavior. A DBA tool uses dynamic binary instrumentation (DBI) to analyze the behavior of executable while maintaining complete control of their performance.

This approach to the problem offers two main advantages:

1. It is entirely independent of the programming language and compiler used to generate the binary code.
2. No need to recompile every time. The instrumentation code changes, as the instrumentation code is added at runtime.

The results clearly show that the proposed tool can fool the malware into thinking it is in a non-isolated environment and executing all its malicious functions. However, the game of cat and mouse never ends and eventually improving their malware evasion technique is needed. Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that provides a minimal management effort and service provider interaction.These advantages of cloud computing technology allow users to makes the process of malware analysis and detection easy and automatic. Malware analysis engine on the cloud provided as Software as a Service for users. Jeffrey Bickford, & Paul Giura, (2015) proposed a new method to safely browse the internet with the help of Transparent Virtual Browser (TVB). A secure way to surf the Internet by redirecting the web request to a remote virtual machine (VM) located on the cloud. Isolated virtual machine located on the

cloud is responsible for rendering web pages and streaming the content back to user's native browser. If the users try to visit a compromised website, as the content is rendered in a Virtual machine the chances of compromising users system is negligible. Once the user closes their native browser window, the Virtual browser session destroys.

Steps carried out by the attacker to attack a computer:

1. The user visits a website that is compromised by the attacker.
2. The user is redirected to attacker's machine or server hosted on the internet.
3. Attacker gathers the necessary information to create malware that can exploit victim's computer system.
4. If the attack is successful, the malware is downloaded onto victim's computer.

A traditionally way of implementing a web security solution is by adding a hardware appliance to the existing infrastructure. The leaders in Web Security appliances, BlueCoat, are also moving from hardware implementations to The Cloud-based solutions for web security.

The proposed solution to the above-described problems is to develop an automated dynamic malware analysis engine and deploy it on the cloud. This automatic engine is customized based on user requirement to scan and analyze web browsing content. The automatic engine is designed to detect Isolation aware malware as described above. Fig.1. Shows the existing system, where a user visiting a malicious website automates the process of downloading malware.

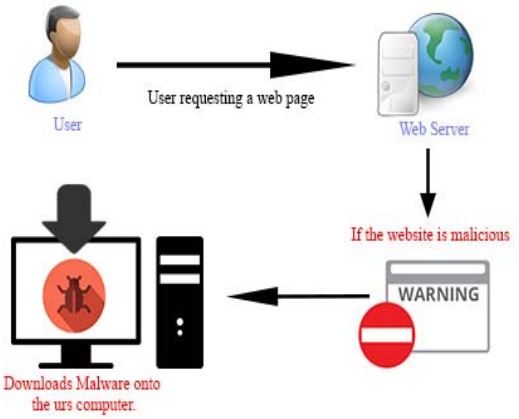

Fig. 1. The Existing System

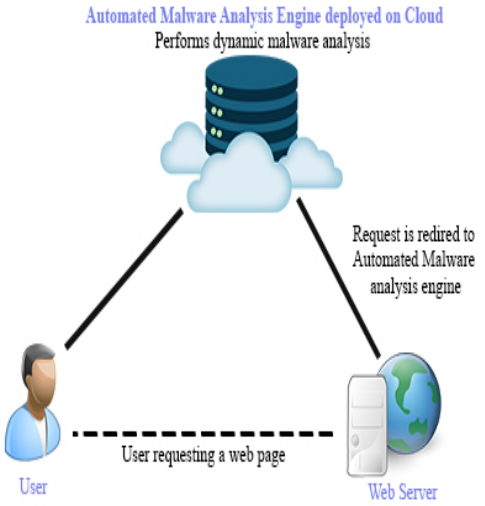

Fig. 2. The Proposed System

Fig.2. Depicts the proposed system, when a user requests for a website the site content is redirected to an automated malware analysis engine deployed on cloud and is capable of detecting sandbox-aware malware. If the Website is clean from malware and not trying to pose any threat to the use, then the website content reaches users browser. If the content on the requested site is unauthorized and containing some malware that can harm the user's computer, then the malware analysis engine blocks the content from reaching users browser environment.

Problems caused by malware stated earlier in this paper can be solved with the help of proposed system. As the malware analysis engine is capable of detecting sandbox-aware malware, malware that can evade detection mechanisms cannot reach client side environment. The preliminary cause of any attack is lack of user awareness and outdated or unpatched applications running on the computer system. In the proposed system the malware analysis engine is deployed in the cloud environment, maintained by the third party whose job is to keep the system updated by regularly patching the security loopholes. Virtual environment created on the cloud for testing requested content destroys for every new web request made by the user. The proposed system described above can be implemented with the help of existing programming

languages and methodologies, which is out of the scope of this report.

4 Methodology and Findings

The literature for this study was extracted from the Internet, journals, and University Library. An in-depth study of different types of malware is done and identified that existing security solutions are not capable of meeting the requirement. Based on the literature study it is observed that there is a need for protecting web browsers from malware mainly focused on sandbox-aware malware. The methodology for this study combines two commonly used security methods:

1. Browser sandboxing
2. Malware Analysis

This study is a theoretical study with a proposed design for implementation. The project report briefly provides a theoretical solution for protecting browsers from sandbox-aware malware. In conducting this research, testing had not been implemented. But by considering the results in the literature, implementing the proposed design can protect individuals from specially crafted sandbox-aware malware.

These days everything is happening in the cloud. The concept of moving applications to cloud environment makes developers to run and maintain applications in a comfortable way. Storing data on cloud removes the burden of maintenance cost for user. If data or application is on cloud then it means that it is somewhere at the other end of internet connection, which can be accessed from anywhere and anytime. Data or Application on the cloud can be accessed over the Internet with the help of a web browser. Intentionally attackers take this as an advantage to exploit browsers.

Website scams are increasing in number. The web browser is acting as current injection vector for attackers to perform sophisticated attacks. A user redirected to a malicious site can unknowingly damage PC and loose personal information. A browser is most exposed and holds valuable information. If an attacker manages to break in, chances are he can take full control of the system or steal user's confidential information. Browsers use tools for various tasks, such as Flash Player, Java, etc. But older versions of these tools often come with security loopholes, which attackers exploit to get access to PC. To guard against attacks initiated via the internet like auto download malware (driven-by-downloads) it is necessary to use a technique proposed by Jeffrey Bickford, & Paul Giura, (2015) to safely browse the internet with the help of Transparent Virtual Browser (TVB).

Attackers focus on exploiting systems through various ways like finding vulnerabilities in websites or browser and exploiting those vulnerabilities. Among them the most common way is to inject malware into targets system through the browser. Malware authors are using different techniques to bypass security mechanisms and infect computers. Researchers Rodriguez, R. J., Gaston, I. R., & Alonso, J. (2016) recently found that malware can even detect the presence of security mechanisms like sandbox and then, behave like a non-harmful application.

Users rely on Anti-malware engines to protect data and resources from attacks. But, these anti-malware systems are failing in detecting well-crafted malwares that can evade the security controls. So, there is a need for building a malware analysis engine which can continuously monitor user activities and block anything that is trying to bypass security mechanisms. One such technique proposed by Rodriguez, R. J., Gaston, I. R., & Alonso, J. (2016) will obfuscate the isolated environment analysis and make conscious malware unable to detect virtual environment. In our research, we found that a combination of the above technique can help user protect system from advanced malware threats. Our intention is to analyze malware in a sandbox environment. A sandbox provides an environment where computer resources are controlled and monitored very strictly, which is perfect for analysis. Based on findings it is clear that a good automated malware analysis engine maintained on the cloud can solve the problem of analysis and detection easy. Many researchers address this problem, but most of their solutions rely on analyzing malware pieces based on signatures and analyzing inbound and outbound traffic. Our solution of analysis is similar to that but, implementation is different, which can make the complete process of malware analysis perfect.

5 Conclusion

The proposed system combines already existing methods of malware analysis and cloud computing that make the process of guarding users against threats efficient. In the proposed system, the malware analysis engine is deployed in the cloud environment, maintained by the third party whose job is to keep the system updated by regularly patching the security loopholes. Virtual environment created on the cloud for testing requested content destroys for every new web request made by the user.

Researchers stated that shortly all Internet-connected devices would face new challenges in defeating malware. The proposed design can be extended by adding an automated dynamic malware analysis engine like Cuckoo Sandbox. There is a need to extend proposed system by adding machine learning algorithms to the automated malware analysis and detection engine. Machine learning can help the malware detection engine to learn from analyzing malware and act smart to defeat newly crafted malware and guard computers from major threats.

6 References

1. Rodriguez, R. J., Gaston, I. R., & Alonso, J. (2016). Towards the Detection of Isolation-Aware Malware. *IEEE Latin America Transactions,14*(2), 1024-1036. doi:10.1109/tla.2016.7437254
2. Bickford, J., & Giura, P. (2015). Safe Internet Browsing Using a Transparent Virtual Browser. *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*. doi:10.1109/cscloud.2015.58
3. SANS Institute InfoSec Reading Room. (n.d.). Retrieved October 28, 2016, from https://www.sans.org/reading-room/whitepapers/forensics/detecting-malware-sandbox-evasion-techniques-36667
4. Sastry Pendyala. (2014, November 16). Sandbox aware Malware. Retrieved from https://securitycommunity.tcs.com/infosecsoapbox/articles/2014/11/16/sandbox
5. Lee. Badger, Tim. Grance, Robert Patt-Corner, & Jeff Voas. (2012, May). Cloud Computing Synopsis and Recommendations. Retrieved from http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-146.pdf
6. Dilshan Keragala. (2016, January 16). Detecting Malware and Sandbox Evasion Techniques. Retrieved from https://www.sans.org/reading-room/whitepapers/forensics/detecting-malware-sandbox-evasion-techniques-36667