isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6 Issue: 3 May,2019,ISSN_NO: 2321-3337

Antivirus data driven event triggering for IOT for Applications

Authors: R. Saranya. M.E Computer science and engineering
T. Saranya M. E., Assistant Professor,
Jei Mathaajee College of Engineering, Kanchipuram

Abstract: The problem of spam or Unsolicited Bulk Email (UBE) is becoming a pressing issue. In spite of the development of many anti-spam techniques, the war against spam is far from being successful. This is partly due to several characteristics of spam that make it a difficult problem. E-MAIL communication is prevalent and indispensable nowadays. However, the threat of unsolicited junk emails, also known as spams, becomes more and more serious. The primary challenge of spam detection problem lies in the fact that spammers will always find new ways to attack spam filters owing to the economic benefits of sending spams. Note that existing filters generally perform well when dealing with clumsy spams, which have duplicate content with suspicious keywords or are sent from an identical notorious server. Therefore, the next stage of spam detection research should focus on coping with cunning spams which evolve naturally and continuously. we present an open digest technique, that we have adapted to take into account disguising attacks and illustrate its resiliency

Research focus on the dynamic invocation of code and patterns based on the existing antivirus model This project emphasizes an artificial intelligence model in developing the code which will do the following process Invoke automatically and started executing in case of any virus files available 1. The virus files can be identified through the signatures, 2. Can be identified via the type / name of files which already exists.3. In case, the user will train the model for the type of virus. An intelligent system will take care of future detection with those patterns. The operations involve the 3-step process such as, 1. Validating the behavior of the files. 2. An automatic event detection happens in case of any actions needs to be done. 3. Event will be handled to process in case of virus detection.

Introduction: The content of spam email can range from the incomprehensible to the downright obscene. Spam is dangerous to both the computer and its users. Junk mail can contain viruses, key loggers, phishing attacks and more. These types of malware can comprise a user's sensitive private data by

capturing bank account information, usernames and passwords. Spam blocker applications can assist a user in preventing these types of PC contaminations. Reliably blocking and filtering spam is the most valuable feature of any spam filter software. The spam filter software should come equipped with multiple capabilities that prevent junk mail from contaminating the user's inbox. The best spam filtering software has both black and white lists, sensitivity settings, community-based filtering, challenge and response techniques, and quarantine settings. Additional features to evaluate are blocking by IP address, server, email address, and country code. The fact that the same information is sent to many users, though spammers try to disguise it by creating a specific version of the message for each user. Reliably blocking and filtering spam is the most valuable feature of any spam filter software. The spam filter software should come equipped with multiple capabilities that prevent junk mail from contaminating the user's inbox. The best spam filtering software has both black and white lists, sensitivity settings, community-based filtering, challenge and response techniques, and quarantine settings. Additional features to evaluate are blocking by IP address, server, email address, and country code.

Previous researchers have developed various methods on near-duplicate spam detection; these works are still subject to some drawbacks. To achieve the objectives of small storage size and efficient matching, prior works mainly represent each e-mail by a succinct abstraction derived from e-mail content text. Moreover, hash-based text representation is applied extensively. One major problem of these abstractions is that they may be too brief and thus may not be robust enough to withstand intentional attacks.

An automated Artificial intelligence system which will take care of finding the

- Custom signature based files
- Custom patterned names(File formats)

The scope of this project is to prevent the customized viruses which can be analyzed via the name and also the signature of the viruses.

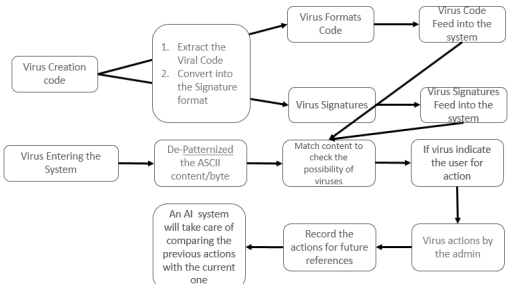


Fig 1 : Antivirus Procedural Techniques

3. Literature Survey:

Automatic Inference and Enforcement of Kernel Data Structure Invariants, Introduce a narrative method for Fast SVM Training detection, called Fast SVM Training . It uses a Bayesian network to resolve the chance of two Fast SVM Training elements being duplicates, making an allowance for not only the information within the elements, except the way that information is ordered. A novel pruning strategy, capable of significant gains over the un-optimized version of the algorithm, is used to improve the efficiency of the network evaluation. Here we are able to do better than another state-of-the-art duplicate detection solution, equally in conditions of efficiency and of effectiveness. Fast SVM Training needs petite user intervention, in view of the fact that the user only needs to offer the attributes to be considered, their individual default probability parameter, and a similarity threshold is the main drawback of the system. Control-Flow Integrity Principles, Implementations, and Application, majorly focusing on computation acceleration Sequential Minimal Optimization. They present a revision of Sequential Minimal Optimization and decide that memory-side data loading in the parsing phase incurs a major performance overhead, as much as the computation does.

Suggests memory-side acceleration which incorporates of data prefetching techniques, and can be practical on top of computation-side quickening to speed up the Sequential Minimal Optimization data parsing. They put into practice a prefetcher on an display place in an effort to appraise its execution feasibility in conditions of area and energy overhead. The Merits of the system is, Memory-side accelerators carry substantial effectiveness athwart existing parsing models. Applying a two-layer prefetcher may guide to up to 4 percent additional energy consumption. Sequential Minimal Optimization performance is hurt by the latency, due to bandwidth becomes the drawback of the system. Detecting kernel-level rootkits Through Binary Analysis take a unusual approach—deploying interoperable Simple Object Access Protocol (SOAP)-based web services straight on the nodes and not by means of gateways.

- It enables standard based and straight application-layer integration flanked by web service- enabled IT systems and resource-constrained sensor nodes.

- Establishes a novel pruning strategy, competent of significant gains over the un-optimized version of the algorithm, is used to perk up the efficiency of the network evaluation.

- Able to do better than another state-of-the-art duplicate detection solution, equally in conditions of efficiency and of effectiveness. The demerits involves • The transparency related to SOAP message processing is very small compared to message transmission.

- Dissimilar ways to lower the related visual projection should be investigated.

3. Exiting Methodology: "A computer virus is a program that recursively and explicitly copies a possibly evolved version of itself"[1]. A virus copies itself to a host file or system area. Once it gets control, it multiplies itself to form newer generations. Over the past two decades, the number of viruses has been increasing rapidly. In 1999, the infamous Melissa virus infected thousands of computers and caused damage close to $80 million; while the Code Red worm outbreak in 2001 affected systems running Windows NT and Windows 2000 server and caused damage in excess of $2 billion[2]. To simplify the virus creation process, virus writers have made virus construction kits readily available on the Internet[3][17]. Computer malwares can be classified according to their different characteristics in several various manners, such as classification by target or classification by infection mechanism. One of these classification types is according to concealment techniques employed. To make viruses more resistant to emulation, virus writers developed numerous advanced metamorphic techniques. According to Muttik,[4] "Metamorphics are bodypolymorphics". A metamorphic virus not only changes it decryptor on each infection but also its virus body. New virus generations look different from one another and they do not decrypt to a constant virus body[5][6][18]. A metamorphic virus changes its "shape" but not its behavior. This is illustrated diagrammatically by Szor in [7], and is shown in Figure 1. Because all polymorphic viruses[19] carry a constant virus body, detection is still possible based on the decrypted virus code.

isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6 Issue: 3 May,2019,ISSN_NO: 2321-3337

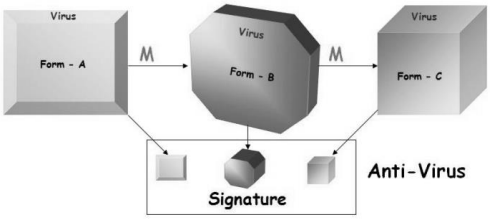


Fig 2 : Signature based Antivirus Procedural Techniques

4. Proposed Methodology: Signature based virus detection - Signature based detection systems scan the files for specific signatures that are present in them.

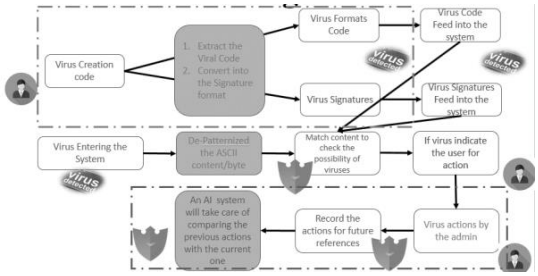


Fig 3 : Data Flow Diagram

The pattern of instructions present in a virus code is identified as the signature of the virus file. This will raise an alarm for virus if the signature of a virus is detected in any of the files scanned.

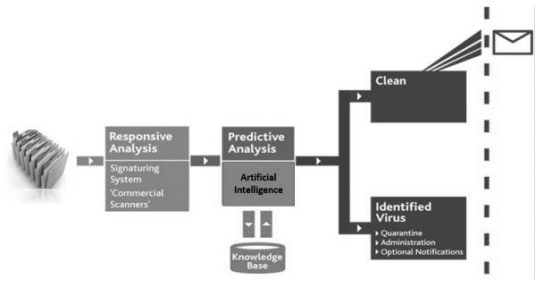
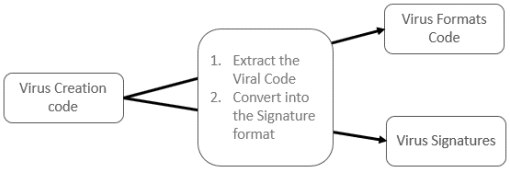
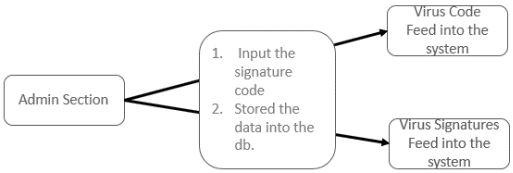


Fig 4 : Related Data Flow Diagram

This method of intrusion detection is fast and accurate since the chances of false alarms are very low in this system. The main requirement of the system is to have an updated database of all the signature files of malware.



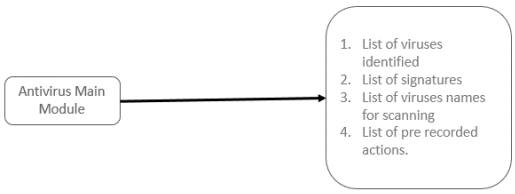
The accuracy is totally dependent on the signature database of the system.



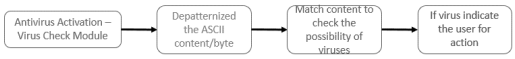
Signature based detection systems cannot detect a new virus since the database will not have any information about the new virus. An antivirus scanner extracts the opcode pattern from an executable file and searches the signature database for the input opcode pattern. Below are the list,

- List of viruses identified with the dates and everything
- List of signatures added
- List of names of the viruses added – In case of executable files, the only option is to scan through the names.

- List of pre-recorded actions taken on the signature and viruses.



The input opcode pattern is considered as the signature of the input file. If a match is found in the signature database, the input file is classified as the corresponding virus family matched in the signature database.

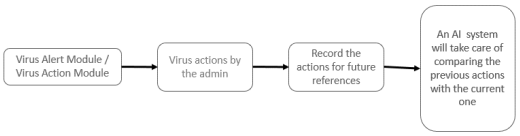


For example, if the signature of the input file is 83EB 0274 EB0E 740A 81EB 0301 0000, then this will be searched in the signature database and the file will be classified as W32/Beast virus since 83EB 0274 EB0E 740A 81EB 0301 0000 is the signature of the W32/Beast virus [8][9].

Fig 4 : Antivirus Signature

Anomaly based virus detection: Anomaly based detection systems monitor the processes on a host machine for any abnormal activity. If any abnormal activity is identified, the system raises an alarm signaling the possible presence of malware [10].

In this detection technique, the system uses the collected heuristics to categorize an activity as normal or malicious. Even though chances of false alarm are relatively higher in this method, it is more reliable because it is also capable of detecting new viruses.



The important thing to note is that raising a false alarm is not as potential harmful as allowing a new virus. However, these systems can be trained gradually by intruders to consider abnormal behavior as normal. Thus, system will fail to detect the abnormal activity in such cases [10].

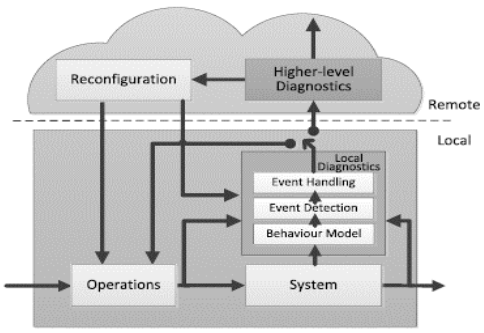


Fig 5: Event flow Techniques

Proposed system mainly focusses on the Artificial Intelligence system

In this project,

- Custom file format files will be deleted based on the user feedback provided.

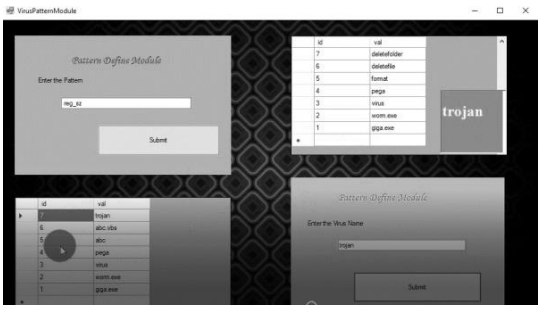
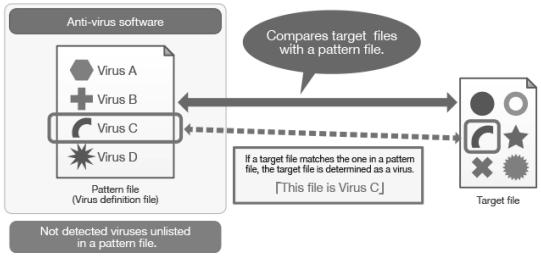


Fig 6 : Virus Patents injection methodology

- Custom virus signature can be added and it will be evaluated on the files and based on the outcome the files will be deleted.
- Automated email system in case of file detection.



The main advantage of our proposed system is,

- Proper reporting of viruses were done in the proposed format through emails
- Custom signature viruses can be detected. User feed signatures were considered in detecting the viruses.
- Custom formats will be added and the actions were marked for future detections and actions.

Fig 7 : Custom File Checks

4.1 **Algorithm Implemented:** Malware bytes anti-malware scanner algorithm involves The Malware anti-malware scanner algorithm is configured to use the interpreted virtual memory states and the interpreted virtual disk states to detect system's malware and the affected files.

- Based on non-intrusive machine introspection without perturbing their execution, the resources extrapolating guest functions by interpreting the virtual memory states and the virtual disk states.
- The object malware algorithm detects the system and enables out-of-the box, tamper-resistant virus detection without losing the semantic view.

An Anti-malware Scanner Algorithm is a state machine where the transitions between states have fixed probabilities. Each state in an Anti-malware Scanner Algorithm is associated with a probability distribution for observing a set of observation symbols. We can “train” an Anti-malware Scanner Algorithm to represent a set of data, which is usually in the form of observation sequences. The states in the trained Anti-malware Scanner Algorithm then represent the features of the input data, while the transition and the observation probabilities represent the statistical properties of these features. Given any observation sequence, we can match it against a trained Anti-malware Scanner Algorithm to determine the probability of seeing such a sequence. The probability will be high if the sequence is “similar” to the training sequences.

Fig 8 : Final actions module

5. Conclusion and Future Enhancement: The developed the metamorphic engine producing morphed copies of the base virus that are highly dissimilar and includes some opcodes of the normal program. These were the two main

criteria described in which are required in metamorphic virus to defeat Anti-malware Scanner Algorithm. In our new engine, employ code obfuscation techniques such as equivalent instruction substitution, dead code insertion, and transpose. This Paper introduce floating point opcodes in morphed copies which are commonly found in normal programs. The similarity showed that the morphed copies are highly metamorphic with 2.5% similarity index. Even with such a high metamorphism, Anti-malware Scanner Algorithm was able to classify the morphed copies of the base virus as the family virus. The base virus was compared with model of morphed copies, Anti-malware Scanner Algorithm was still able to classify the base virus as the same family. This fact proves that even with high metamorphism, Anti-malware Scanner Algorithm is able to identify a common statistical pattern across all morphed copies and the base virus. Anti-malware Scanner Algorithm has proved very difficult to defeat. Ideally, would

like to find viruses that are similar to normal programs to a degree that the similarity index alone cannot distinguish the viruses from normal code. Only with such data can we evaluate the effectiveness of the HMM approach to detecting metamorphic viruses. However, it appears that no metamorphic kit available today is capable of producing such challenging viral code.

References:

[1] Leonard Adleman. An abstract theory of computer viruses. In Lecture Notes in Computer Science, vol 403. Springer-Verlag, 2018.

[2] M. Stamp, "Information Security: Principles and Practice," August 2005.

[3] Fred Cohen. Computer Viruses. PhD thesis, University of Southern California, 2017.

[4] Peter J. Denning, editor. Computers Under Attack: Intruders, Worms and Viruses. ACMPress (Addison-Wesley), 1990.

[5] Christopher V. Feudo. The Computer VirusDesk Reference. Business One Irwin, Homewood, IL, 2015.

[6] Harold Joseph Highland, editor. Computer Virus Handbook. Elsevier Advanced Technology, 2016.

[7] J. Aycock, "Computer Viruses and malware," Springer Science+Business Media, 2018.

[8] E. Daoud and I. Jebril, "Computer Virus Strategies and Detection Methods," Int. J. Open Problems Compt. Math., Vol. 1, No. 2, September 2008. http://www.emis.de/journals/IJOPCM/files/IJOPCM(vol.1.2.3.S.08).pdf

[9] Lance J.Hoffman, editor. Rogue Programs:Viruses,Worms, and Trojan Horses. VanNostrand Reinhold, New York, NY, 2012.

[10] Jan Hruska. Computer Viruses and Anti-VirusWarfare. Ellis Horwood, Chichester, England, 1990.

[11] Filiol, E., G. Jacob, M.L. Liard, 2007. Evaluation methodology and theoretical model for antiviral behavioral detection strategies. J. Comput. Virol., 3(1): 27-37

[12] Ye, Y., D. Wang, T. Li and D. Ye, 2018. An intelligent pe-malware detection system based on association mining. In Journal in Computer Virology,

[13] Zakorzhevsky, 2016. Monthly Malware Statistics. Available from: http://www.securelist.com/en/analysis/204792182/Monthly_Malware_Statistics_June_ [Accessed 2 July.]

[14] W. Wong, "Analysis and Detection of Metamorphic Computer Viruses," Master's thesis, San Jose State University, 2016. http://www.cs.sjsu.edu/faculty/stamp/students/Report.pdf

[15] Eugene H. Spafford. Computer viruses. In John Marciniak, editor, Encyclopedia of Software Engineering. JohnWiley & Sons, 2017.

[16] S. Attaluri, "Profile hidden Markov models for metamorphic virus analysis," Master's thesis, San Jose State University, 2007. http://www.cs.sjsu.edu/faculty/stamp/students/Srilatha_cs298Report.pdf

[17] Cristian Barría Huidobro ; David Cordero ; Claudio Cubillos ; Héctor Allende Cid ; Claudio Casado Barragán " Obfuscation procedure based on the insertion of the dead code in the crypter by binary search , IEEE 2018

[18] Ignacio Martín ; José Alberto Hernández ; Sergio de los Santos - SignatureMiner: A Fast Anti-Virus Signature Intelligence Tool, 2018 IEEE Conference on Communications and Network Security (CNS)

[19] Xing Wang ; Derek Pao, Memory-Based Architecture for Multicharacter Aho–Corasick String Matching, EEE Transactions on Very Large Scale Integration (VLSI) Systems 2018