Strategy and Knowledge-Based XML Attack Detection Systems using Ontology

Gajanan P Bherde, M.A.Pund

Abstract*: Today's highly skilled attackers exploit many of the vulnerabilities in their network. On the other hand, the risk of data leakage has increased dramatically because software or application vulnerability is always left without a fix. Such vulnerability using the (Zero Day), hackers will receive the target network, and can steal sensitive data. Detection of zero day traditional defenses is difficult, because the signature information zero-day attacks are unknown. Therefore, the need for new security solutions, that detect zero-day attacks, and evaluate the identified zero day vulnerability severity. The paper proposed an approach for detecting unknown vulnerabilities. The system is a framework that is a comprehensive approach for detecting and prioritizing zero-day attacks and removes these detected attacks. The proposed framework is based on probabilistic approach, to determine the Zero-Day attack path, and the subsequent degree of severity of the identified zero-day vulnerability. It is a hybrid of detection and removal method based on the detection of unknown defects present in the network, which is not yet detected. The system also shows the file with original size and with an attacked size.

***Keywords*: Zero-day attacks, signature information, Vulnerability analysis.**

I. INTRODUCTION

Current organizations and companies pay close attention to the protection of the network. However, they are still at risk through responsible and sustainable defense investment. This is because attackers can bypass the organization's security system using an unknown flaw that is not listed in security personnel. In a well-protected network, a loophole can be discovered by constant research identified by a hacker. This is the goal, while the attackers infiltration network configuration, where leverage vulnerabilities there [1]. Zero day (0-day) is a software or application vulnerability that can be used to cast the targeted network the residential base, which will allow the hacker attacks or the attacker with sensitive information stolen, such as legal documents or corporate data. Cybercriminals increase the success rate of attacks by uncovering and exploiting the vulnerabilities of Day Zero. This vulnerability information is not available until zero-day attacks have occurred. As a result, it is difficult to identify and analyze attacks that use zero-day exploits. If there is zero day vulnerability, the hacker has two possible options:

- He can help the software vendor by providing him with information about the detected vulnerability.
- It can sell important information to a black market broker who can then sell the identified exploit at the highest rate.

An attacker can embed zero-day vulnerability into a list of vulnerabilities and, when the process and payload of an intrusion program are created, an attack is initiated. In particular, the attacker performs a small crack defense access secretly, revealing the network of the attacker's ability to find zero-days through a few hours, weeks or months of painstaking efforts through a line of code to find some weaknesses. This is how the network is compromised by a zero-day attack [2] [3].

There is no zero - day protection when attacks first appeared. Traditional security approaches to detect vulnerabilities create signatures, but in the case of zero day, signature information is unknown. Thus, it is very difficult to detect zero days with the help of traditional means. Attackers are highly skilled and is detected vulnerability can be society remained unknown for months or years. This fact gives the attackers enough time to cause irreparable harm to the [4]. FireEye's according to the typical zero-day attack can last for an average of 310 days. So zero-day work is certainly a challenge.

The paper, implement a system for detects the zero-day attack and removes or killed these detected attacks. The system builds two methods for detecting the attacks: signature-based and knowledge-based detection system. Once the system detects the attacks it's killed by using a firewall.

- Signature-based detection method:

Signature-based detection techniques are typically used for malware detection by the [legacy] antivirus software. Its name suggests that the technology relies on existing databases of malware signatures, which are used as a reference for scanning systems for viruses. Typically, the signature database is updated very quickly, but since zero-day attacks have no known signatures, it can be used to detect new zero-day attacks.

- Knowledge-based detection system method:

Another way to protect a web service from an injection attack is to use a knowledge-based detection system that applies protection based on previously known cataloged behavior, usually involving normal behavior and unusual actions, including all expected actions that define such profiles.

Revised Manuscript Received on January 15, 2020.
***** Correspondence Author
Gajanan P Bherde*, Department of Computer Science & Engineering, PRMIT & R Badnera, Amravati, India Email: boscompmu@gmail.com
Dr. M.A.Pund, Department of Computer Science & Engineering, PRMIT&R Badnera, Amravati, India Email: pundmukesh@gmail.com

II. LITERATURE REVIEW

The paper [5], propose a framework to detect and respond to attacks. Ports are all online redundant, non-uniform and functionally equivalent modules. There are pools of different execution modules, from new ones to new ones, provided that the stricken execution module can get an abnormal output. By analyzing the abnormal output, it is possible to construct a correspondence between the input and the abnormal output, and the input leading to the abnormal output is written to the Zero-Day attack-related database further correspondence, such as IP blacklist and patching. In paper [6], Analysis and extensive data sets run software's family provides RansomWall, a defense-in-depth system that protects against encryption and Ransomware. Marjan Keramati [7], they propose an innovative method for risk assessment of unknown vulnerabilities. A risk assessment is necessary to have a thorough understanding of the causes of the attack or the characteristics associated with the vulnerability.

Farag Azzedin, Husam Suwad, Zaid Alyafeai [8], focuses on security domains and considers new ways to look at the security lifecycle. The authors take advantage of the vision to propose an asset-based approach to zero-day attack prevention. Stephen Taylor [9], recently, a quiet revolution in embedded system technology has been taking place with the advent of system-on-chip (SoC) devices. These innovations, which include a wide range of peripherals, cryptographic hardware, and high-performance multi-core processors within the boundaries of a single chip, are accompanied by the arrival of the era of high-level synthesis and are expected to contribute to the development of hardware and software in the C and other system programming languages. At the same time, the appearance of a lightweight OCI-compliant container has revolutionized the distribution and maintenance of vertically integrated software stacks.

Nisreen Innab, Eman Alomairy and Lamya Alsheddi [10], Mention some techniques to avoid zero-day attacks. They then analyze the strengths and weaknesses of the approach, which is based on the honeypot and anomaly detection. As a result, to integrate the approaches in a hybrid model as an advanced solution to detect zero-day attacks that may occur in the system. Vivek Bardia [11], CRS Kumar, the investigation of various threat detection, protection and mitigation systems, users have been completely ignored or the system is heavily dependent on user input for the correct functionality compiled above they designed research taken in addition to the independent detection and prevention system, user input to identify and reduce risks. In paper [12], a stochastic approach for Zero-Day attack path discrimination is proposed and a prototype system Zepro is implemented. To capture zero-day attacks, a dependency graph named object instance graph first analyzes the system call and then displays a graph of supergraph, an instance based on the Bayesian network building system, hidden in the Zero-Day attack path. By entering the intrusion proofs as an object instance in the Bayesian network to be computed, the infected. When a high probability instance is connected through a dependency relationship, a path is formed that is the Zero-Day attack path. The paper [13], describe a system developed to extract necessary signatures and an algorithm to solve this problem. The algorithm finds the length of messages. In paper [14], propose a Prophetic Defender (PD) which can minimize ZDAP. Prior to the actual attack, hackers may have vulnerable ports to host the network. If this port scan can be detected early, a zero-day attack can be detected. The PD architecture used honeypot-based pseudo-server placement to detect and scan for malicious ports. The paper [15], it has many variations of the signature of the game. Because of the signature-based defense capabilities, the traditional security layer here is your preference; persistent threats to browse also miss. This paper provides a detailed study to outline the research effort, which is associated with modern zero-day malware detection zero-day polymorphic worms.

III. METHODOLOGY

This Figure 1 shows Strategy and Knowledge-Based XML Attack detection Systems Using Ontology, the system is based on the signature of the XML file path. It is an ontology database that stores all known attacks in the signature format. And it is the owl file obtained through the ontology database. Then the attacker adds an attack. An attacker is a security threat who attempts to delete, destroy, or modify information without permission or access. The contents of the input file are converted to the signature format. The contents of the signature format are compared with the memory attacks in the ontology database. If the signature coincided with an attack, an attack was detected. Otherwise, no attack is detected. The knowledge-based detection system detects new attacks and stores them in a database. The system also shows the file with the original size and with an attacked size. The system removes the detected attack by using a firewall.

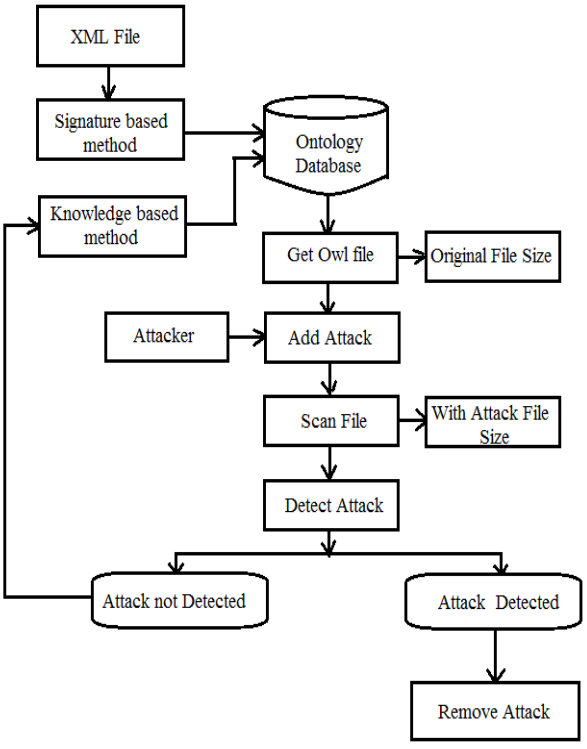

Fig 1: Strategy and Knowledge-Based XML Attack detection Systems Using Ontology

IV. RESULT AND DISCUSSION

To evaluate the proposed system, all experiments were conducted in a research laboratory on an isolated network. The system is built using Java within the Windows platform. Netbeans IDE is used as a development tool. The system does not require any specific hardware to run; any standard machine capable of running the program. We have developed four experiment scenarios in order to evaluate the ontology knowledge-based approach, the scalability and performance of signature-based databases against which ensure that the new attacks conclusion the use do not endanger the proposal of vitality. We applied an ontology database consisting of 128 known attacks (plus a protege) for evaluation. The database initially contains four attack categories (XMLInjection, XPathInjection, XQueryInjection, and XSSInjection) and four attack cases (xpathInjection1, XQueryInjection1, XQueryInjection2, and XSSInjection1). To create a 128-attack database, we simulated several attack and action cases mimic the attack of variations.

We observed that the prototype warns of an attack immediately after reaching the axiom limit. Correctly displaying this page also detected instances of the attack correctly. These instances had the AttackAction of the class detection, the attack of the ProbeXPath class, and one of the actions of the InjectXQuery class. This set of three AttackActions should alert the XQueryInjection attack according to the defined axioms of this attack class. However, for each instance, the prototype warned XPathInjection and XMLInjection attacks (a total of 14 inaccurately warned attacks). Discrepancies occur because the first part of this simulated cases (discovery AttackActions and ProbeXPath) satisfy the constraints XPathInjection class Axiom, and the rest (InjectXQuery) satisfy the constraints XMLinjection universal class of an axiom. After the XID engine does not take into account the full set of actions that satisfy axiom limits the most specific attack classes, in other words, where the attack messages would be the wrong conclusion, only consider the subset of attacks that satisfy axiom limits generic class. The following figure shows the implemented Strategy and Knowledge-Based XML Attack detection Systems Using Ontology.

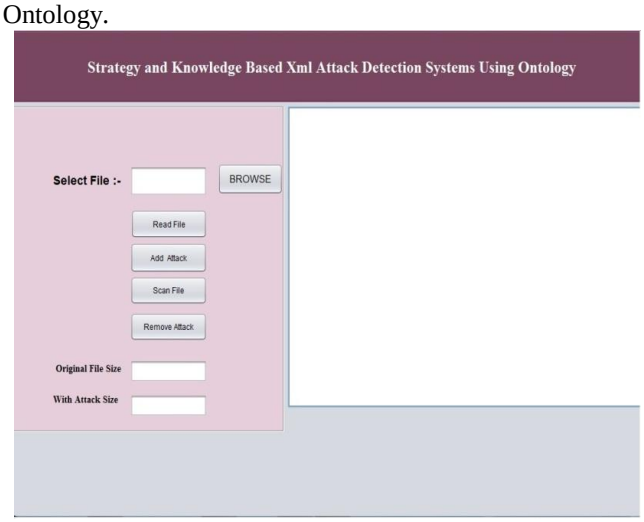

Fig 2: Homepage

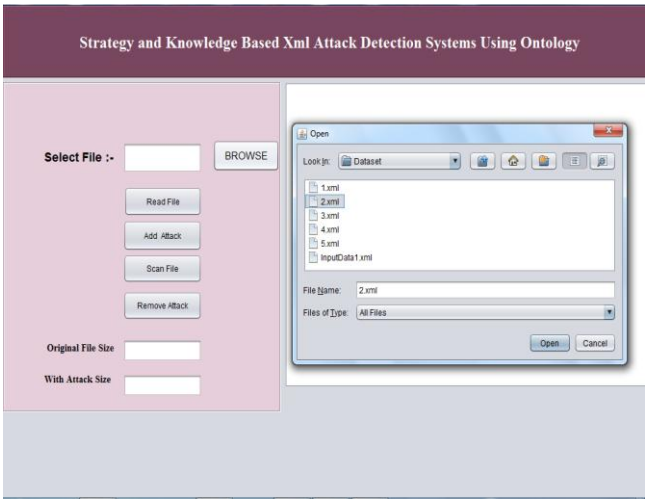

Fig 3: Browse Owl file

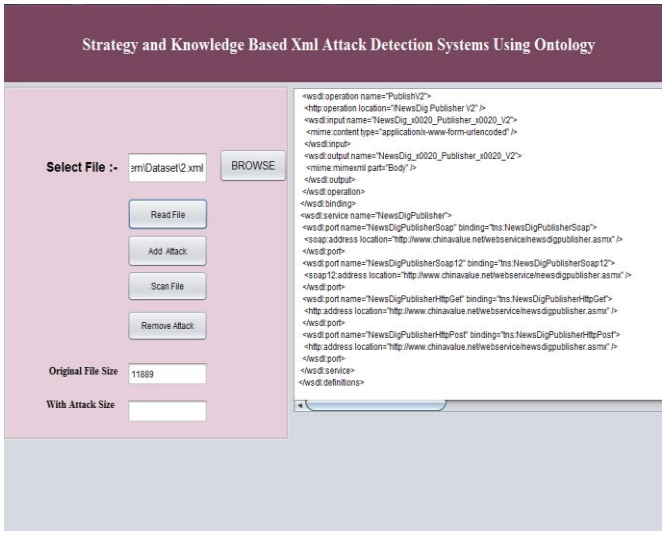

Fig 4: Read File

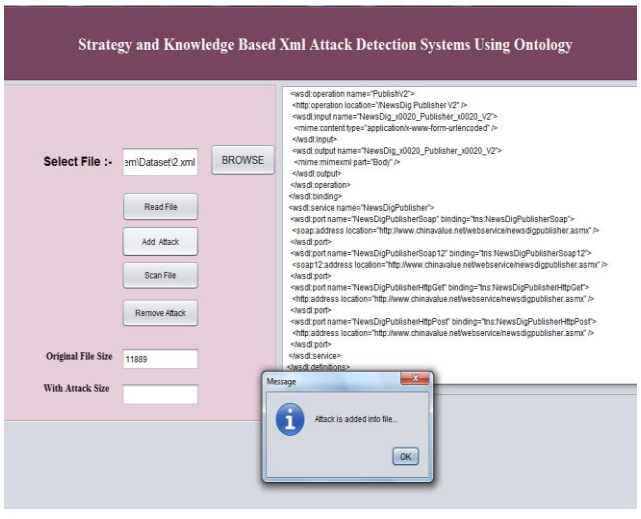

Fig 5: Add Attack

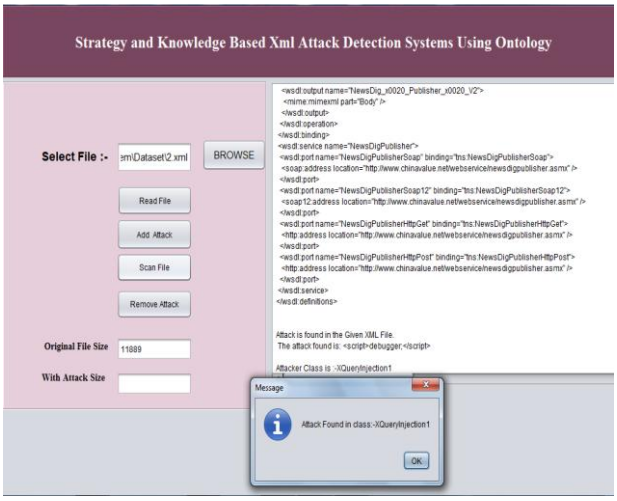

Fig 6: Scan File

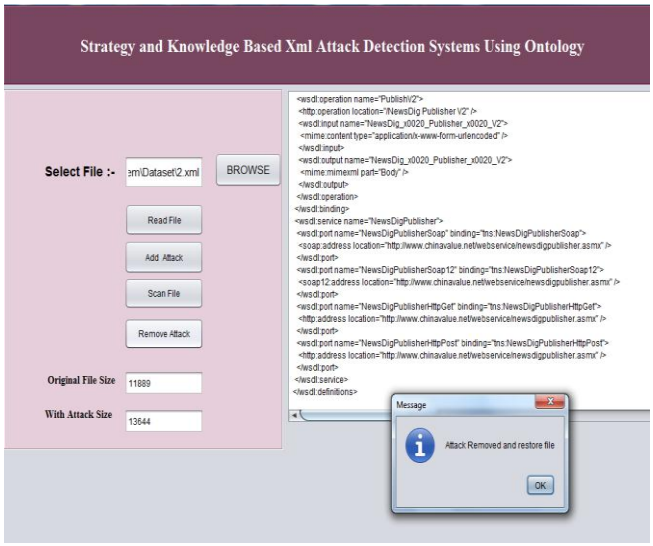

Fig 7: Remove Attack

V. CONCLUSION

Most organizations have the most attractive vulnerabilities for attackers and are widely used in the software. Most of these vulnerabilities contained in the software, such as Internet Explorer and Adobe Flash and is often used by many consumers and experts. After discovery, a zero-day attack is immediately added to the attacker's toolkit and exploited. The paper proposed an approach for the analysis and detection of zero-day attacks. This system is a combination of a signature-based system and a knowledge-based system. The proposed method solves the unsupervised learning technique and identifies the known and unknown attacks and removes this detected attack by using a firewall.

REFERENCES

1. Paul Maxwell, "Stockpiling Zero-Day Exploits: The Next International Weapons Taboo", Research gate, Feb 2017.
2. M. Masthan1 and R. Ravi, "Prevention of zero-day vulnerability in network using ensemble fuzzy association and cuttle fish detection", IJCT june 2017.
3. Umesh Kumar Singh, and Chanchala Joshi, "Scalable Approach towards Discovery of Unknown Vulnerabilities", IJONS Sept. 2018.
4. C. Joshi, and U. K Singh, "Performance Evaluation of Web Application Security Scanners for More Effective Defense", IJSRP, June 2016
5. Wenyan Liu ; Fucai Chen ; Hongchao Hu ; Guozhen Cheng ; Shumin Huo ; Hao Liang, "A Novel Framework for Zero-Day Attacks Detection and Response with Cyberspace Mimic Defense Architecture" IEEE 2017, DOI 10.1109/CyberC.2017.39
6. Saiyed Kashif Shaukat ; Vinay J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransom ware attacks using machine learning", IEEE 2018, 10.1109/COMSNETS.2018.8328219
7. Marjan Keramati, "An attack graph based procedure for risk estimation of zero-day attacks", IEEE 2016, DOI: 10.1109/ISTEL.2016.7881918
8. Farag Azzedin ; Husam Suwad ; Zaid Alyafeai, "Countermeasureing Zero Day Attacks: Asset-Based Approach", IEEE 2017, DOI: 10.1109/HPCS.2017.129
9. Stephen Taylor, "Protecting Embedded Systems from Zero-Day Attacks", IEEE 2018, DOI: 10.1109/NAECON.2018.8556791
10. Nisreen Innab ; Eman Alomairy ; Lamya Alsheddi, "Hybrid System Between Anomaly Based Detection System and Honeypot to Detect Zero Day Attack", IEEE 2018, DOI: 10.1109/NCG.2018.8593030
11. Vivek Bardia ; CRS Kumar, "End Users Can Mitigate Zero Day Attacks Faster", IEEE 2017, DOI: 10.1109/IACC.2017.0190
12. Xiaoyan Sun ; Jun Dai ; Peng Liu ; Anoop Singhal ; John Yen, "Using Bayesian Networks for Probabilistic Identification of Zero-Day Attack Paths" IEEE 2018, DOI: 10.1109/TIFS.2018.2821095
13. Yehuda Afek ; Anat Bremler-Barr ; Shir Landau Feibish, "Zero-Day Signature Extraction for High-Volume Attacks", IEEE 2019, DOI: 10.1109/TNET.2019.2899124
14. Chia-Nan Kao ; Yung-Cheng Chang ; Nen-Fu Huang ; I Salim S ; I-Ju Liao ; Rong-Tai Liu ; Hsien-Wei Hung, "A predictive zero-day network defense using long-term port-scan recording", IEEE 2015, DOI: 10.1109/CNS.2015.7346890
15. Ratinder Kaur ; Maninder Singh, "A Survey on Zero-Day Polymorphic Worm Detection Techniques" IEEE 2014, DOI: 10.1109/SURV.2014.022714.00160

AUTHORS PROFILE

Gajanan P Bherde, Department of Computer Science & Engineering,
PRMIT & R Badnera, Amravati, Mumbai, India
Email: boscompmu@gmail.com

Dr. M.A.Pund, Department of Computer Science & Engineering, PRMIT & R Badnera, Amravati, India