A Flexible Framework for Malicious Open XML Document Detection based on APT Attacks

Hung-Min Sun, Chi-En Shen, and Chi-Yao Weng

***Abstract*—The defense against Advanced Persistence Threat (APT) attacks is an important topic in recent years. Many organizations and enterprises even governments have been victims of APT attacks. As APT attacks have a specific objective and are skillfully crafted, motivated, organized and well founded, we should pay more attention on those attacks. Malicious documents have been used with the spear phishing attack in the initial infection phase of an APT attack. The detection of malicious documents is important for an early stage defensive APT attack. The Open XML has a popular document format used in the APT attacks. However, the related malicious document detection research is mostly focused on the PDF file or the traditional OLE Office document format. A specific framework design for malicious Open XML document detection does not exist.**

This article proposes a framework based on malicious Open XML document detection. This framework is designed under the fundamental principle, such as automatic, flexible and configurable. Our proposed framework can analyze Open XML document job automatically and generate analysis reports with information highlighting. The Scanner Module in this framework can be configured and easily extended by adding customized scanners, is flexible. The Configurable framework makes the APT detection more customizable and suitable for user's demand.

***Index Terms*—Open XML; Advanced Persistence Threat; APT; Malicious document;**

I. Introduction

Advanced Persistence Threat (APT) is a popular issue in recent years. Cyber espionage against companies and governments are increasing in complexity. Actors of APT use multiple techniques to break into a network, avoid detection, and harvest valuable information over a long term. The processes involved in APT require high degree of covertness over a long time and the process signifies sophisticated techniques using malware to exploit vulnerabilities in a system. The traditional detection system might not work for APT attacks [2].

The Open XML document is an XML-based file format developed by Microsoft for representing spreadsheets, charts, presentations and word processing documents [3]. Because this kind of document format was rarely used in attacks before, when an Open XML document zero-day exploit has been deployed, it is not easy for a traditional anti-virus to do the detection. The existing malicious analysis tools are not designed for Open XML documents, so the detection and analysis work of malicious documents require manual and complex works. There is no existing open source framework for malicious Open XML document detection so far.

A detection framework for malicious Open XML document is established in this paper. Our design principle is automatic, flexible and configurable. The framework is based on the file format characteristic of Open XML, and has previously analyzed many malicious samples to gain knowledge of the characteristic of malicious Open XML files.

II. Background

Advanced Persistent Threat (APT) is a set of stealthy and continuous computer hacking processes, often orchestrated by human targeting a specific entity [1]. APT Terminology consists of three major components: processes: advanced, persistent, and threat.

APT has few key differences compared to the regular attack, including the technique, target and objective. The characteristics of APT include: unique motivation, targeted, long-term attack and professional techniques.

In a general case, the attack process can be divided into four phases: Reconnaissance, Initial Infection, Penetration, Harvest.

III. The Proposed Framework

Our designed framework can accelerate the process of analysis samples and provide flexible approach to fulfill different research purposes. Because of the special structure property of OpenXML file, our proposed framework consists of these five stages: pre-process, basic process, advance process, analysis and output.

A. Design Principle

To analyze an OpenXML file, the researcher needs to first decompress the file to see the file structure and extract the binary files in this document. Then, they need to open the binary file to see if there are any malicious files put into the activeX binary. There are many different kinds of malicious document types, while some of them use well-known exploits and some others use embedded shellcode or executable files. To make the analysis more flexible and more customizable, we designed our framework to be configurable.

B. Design Framework

To detect malicious OpenXML documents and facilitate the analysis process for researchers and analysts, we designed our framework with our design principle and the five processing stages.

1) OpenXML Dissector: This component will try to decompress the OpenXML document.

2) Object Parser: The "Object Parser" parses the files extracted from the original OpenXML document. Checks whether the document includes an activeX object, flash embedded object, and macros VBA script.

978-1-7281-1878-9/19/$31.00 ©2019 IEEE

3) Object Extractor: The "Object Extractor" does the work of extracting embedded objects.

4) Configuration: The "Configuration" component is an object to set up the scanners. There are three parts in the configuration, "Directory", "Objects", and "Scanners". During the analysis process, several files will be extracted and decompiled.

5) Operator: The "Operator" component is in charge of the operation of all scanners based on the configuration, and manages the scanning result. The Operator will collect scanning results from every scanner and calculate the IOM score of this sample.

6) Scanners: We pre-define five kinds of "Scanners". They are "Flash Scanner", "Mal Structure Scanner", "Shellcode Scanner", "URL Scanner" and "VBA Scanner". Researchers can apply any module to each scanner with the defined interface.

7) IOM: The "Indicator of Malicious" represents the possibility of malicious in the framework. The result contains an IOM score and the scanning report. The higher the IOM, the higher the possibility the file could be malicious.

8) Report Generator: The "Report Generator" gets the sample's scanning result from the operator and writes the report to the configured location.

IV. Implementation

Our design principle is to make the proposed framework as flexible as possible and easy to deploy. We implement the configuration as a XML [4] file. Three parts are considered into our configuration component, such as "Output Directory", "Objects" and "Scanners". There are five kinds of pre-defined objects in our framework; including "ActiveX", "VBA Script", "XML", "Action Script" and "PE".

The "Indicator of Malicious" tells the researcher the possibility of a malicious file in this sample. A good IOM score regulation will help the researcher better distinguish between an ordinary and malicious Open XML document.

V. Analysis

The detection rate is a key factor on evaluating the detection performance of proposed framework.

1) Specify IOM Implication: We collected 30 malicious samples and 30 clean samples to test. The scanning results of these malicious samples show that 10% of the samples (3 samples) has been detected as "Moderate malicious risk", 90% of the samples (27 samples) was detected as "Extreme malicious risk". Since all malicious samples were detected above "Moderate Malicious Risk", our experiment will consider IOM score over 100 as malicious.

2) True Positive Rate: After we specify the IOM implication, we experimented on more samples to test the true positive rate of our framework and scanners. We downloaded 250 malicious Open XML samples from VirusTotal to do this experiment. In our result, 6% of samples (15 samples) were detected as "Clean Document", 62.4% of samples (156 samples) were detected as "Moderate Malicious Risk", 31.6% of samples (79 samples) were detected as "Extreme Malicious Risk". Our experiment shows that the true positive rate of detection is 94%.

3) False Positive Rate: We collected 1752 clean samples from VirusTotal and deployed our framework and scanners on these samples. In our result, 94.3% (1652 samples) samples were detected as "Clean Document", 2.7% of the samples (48 samples) were detected as "Moderate Malicious Risk", 3% samples (52 samples) were detected as "Extreme Malicious Risk". The false positive rate of our framework and scanners is about 5.7%.

4) Compare with Antivirus: The 250 malicious samples used on our previous true positive rate experiment are taken into testing on 13 antivirus software. This experiment shows that our malicious Open XML document detection framework has a better detection rate than that of existing antivirus detection tools.

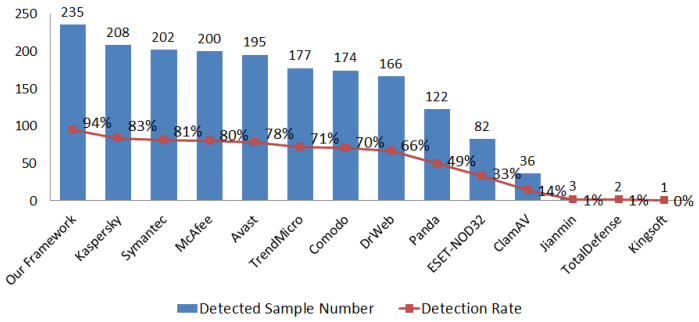

Fig. 1. Detection Rate of Our Framework and 13 Antiviruses

VI. Conclusion

The detection of a malicious document is important to defend in an early stage attack of an APT. There is no existing open-source detecting framework designed for Open XML document. To provide an efficient Open XML document tools, our proposed framework possesses not only automatic analysis of the malicious document but also flexible to extend by adding other scanner module and scanning program or modify the scanning process. Furthermore, our proposed framework is configurable and friendly for researchers to configure the scanner according to their demands in different research or focus on different scanning objects or exploit. Our framework has better detection ratio than that of existing commercial detection tools.

The novel detection framework has become a standard technology since the research works on developing the malicious document detection method to defense the APT attacks. An effectual detection method held on flexible structure, high detection rate and speedy the performance of scanning large-size binary objects, would be fully taken into consideration in the future.

References

[1] Wikipedia, *"Advanced Persistent Threat"*, http://en.wikipedia.org/wiki/advanced_persistent_threat.

[2] M.P. Collins, and M.K. Peiter, *" On the Limits of Payload-oblivious Network Attack Detection"*, in Proceedings of 11th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2008, pp. 251-270.

[3] Wikipedia,*"Office Open XML"*, http://en.wikipedia.org/wiki/Office_Open_XML.

[4] Wikipedia, *"XML"*, http://en.wikipedia.org/wiki/XML.