Improved file analyses for Real-time attack source

DaeHee Seo[1], JongHyun Kim[1], JoongYong Choi[1], ByungGil Lee[1], KabSeung Kou[2]
Jang-Mi Baek[3]

[1] Electronics and Telecommunications Research Institute
[2]Korea System Assurance Co. R&D Team
[3]SoonChunHyang University
[1]{dhseo, jhkim, choijy bglee}@etri.re.kr
[2]kabseung@kosyas.com
[3]bjm1453@sch.ac.kr

Abstract. This, propose scheme suggests multi analyses method on the files uploaded to the file sharing site for tracking down the circulating area and the source of real-time attack. The propose scheme is divided into the method for real-time analysis for real-time tracking and the method of cooperative analysis method for non-real time analysis. By allowing the supervisors to choose the relevant analyses method selectively, we can conduct variable analyses depending on the network threat.

Keywords: Cyber-attack, Web Security, File analysis, log analysis, Global Cooperation

1 Introduction

While the rapid advancement of the internet provides diversified communication and services the social elements, there is also an increasing number of cyber-attacks ranging from DDoS(Distributed Denial of Service) attack to the illegal leakage of user privacy against certain social infrastructure, leading to the social chaos[1]. Especially, in the process that many users use the file sharing system universally, the illegal user with malicious purpose distributes files, making normal user's zombie PC, therefore utilizing it as network invasion attack[2][3]. Thus, tracking the attack source and web site is utmost needed in social chaotic situation, however, there exists limitations to the trace technology that is reliable and in real time, it is problematic to the social integration[4][6]. In particular, known malignant code can be blocked by existing network security system, on the other hand, it is required that we need to conduct a practical analysis on suspicious unknown malignant file [5]. Therefore, this paper proposes the method that can analyze independently each type of files that are uploaded to the file sharing site. Also, we distinguished between the real-time analysis and non-real-time analysis, thus enabling detailed analysis. For this, the second section proposes improved file analyses for tracking the attack source. Finally, the third section derives the conclusion and provides a direction to the follow-up studies.

ISA 2013, ASTL Vol. 21, pp. 37 - 40, 2013

2 Improved file analyses for Real-time attack source

2.1 Assumptions

For the file multi analyses method to trace the web-based real-time attack source, we make the assumptions as the following:

- Web-based file sharing site has the information on the known attack source or black list (IP, ID, MAC) of web site registered as Rule in advance.

- Database on Header signature, Footer signature and BFD to known files is constructed in advance.

2.2 Improved file analyses for real-time attack source

1) Pattern test

Pattern test consists of Simple test and SMART test as such:

A) Simple test

① Based on known Rule, simple test registers IP, ID or MAC of previously known black list in advance and executes black list-based Rule test

② In case the new file is uploaded, vaccine program installed at file sharing site detects it on the basis of reputation and notifies reputation-based detected[1] files.

③If the file that needs reputation-based detection is uploaded, fuzzy theory is used for Rule-based comparing and searching as such:

Ⓐ If unknown file is uploaded, we define the fitness on R_k as the following:

$$u_k(P_i) = u_{k_1}(P_1) \cdot u_{k_2}(P_2) \cdot \ldots \cdot u_{k_n}(P_{i_m})$$

We decide result class C_k and certainty gc_k on a fuzzy set M_{k_j}. According to the fitness of fuzzy rule R_k,

$$\beta_c = \sum_{P_i \in Class\ C} u_k\,(P_i), (C = 1,2,\ldots,n)$$

We decide a class (c′) that has the maximum amount of the sum of fitness on result class C_k of fuzzy rule R_k, then we calculate $\beta_{c\prime} = \max\{\beta_1, \beta_2, \ldots, \beta_n\}$. If we cannot decide c′, we define C_k as dummy class(empty class)

Ⓑ The certainty of all dummies is set as $g_{C_k} = 0$ and the certainty on other rules is defined as such :

$$g_{C_k} = \frac{\beta_{c\prime} - \bar{\beta}}{\sum_{c=1}^{n} \beta_c}, \; \bar{\beta} = \frac{\sum_{c \neq c\prime} \beta_c}{n-1}$$

Ⓒ We can guarantee the certainty of Rule based on current network information by defining g_{C_k} in the option field

[1] Through the user's reputation information, the technology confirms the credibility of unknown files and detects the dangerous elements, providing the reputation information after mining data and evaluating the reputation. Nowadays, mainly anti-virus programmed adopts that technology and utilizes it in the analysis and detection of virus.

Ⓓ After defining g_{C_k} in the option field, data on each of option value $\sigma(\mathrm{N})$ are calculated as the following. We calculate the data $\sigma(\mathrm{N})$ on each of option value as the following and decide whether we generate the trace log on uploaded file according to $\sigma(\mathrm{N})$.

$$\sigma(\mathrm{N}) = \prod_{p|N}\left(1 - \frac{1}{(p-1)^2}\right)\prod_{p|N}\left(1 + \frac{1}{(p-1)^3}\right) > \frac{1}{2}, \begin{cases}\sigma(N) > \frac{1}{2}\\ \sigma(N) \le \frac{1}{2}\end{cases}$$

If $\sigma(\mathrm{N}) > \frac{1}{2}$, then we do not generate trace log. If we do not generate trace log, post uploaded file to the server and $\sigma(\mathrm{N}) \le \frac{1}{2}$, we send the attack source and web site for generating trace log on this.

B) SMART test

SMART test analyze behavior analysis process based on unique format that each of file has as following:

①The uploader uploads Sample.jpg file to the file sharing site.

② File sharing site checks Header signature and Footer signature of uploaded Sample.jpg file.

JPEG's Header signature: ″FF D8 FF ED″

③If the signature of uploaded file and known signature are the same, we upload and post that file to the board. If it is not the same Header signature, we use BFA (Byte Frequency Analysis) algorithm to extract the specific pattern of BFD(Byte Frequency Distribution) depending on types of the same file

④The frequency distribution for extracting a specific BFD pattern is as such:

$$\mathrm{NFPS} = \frac{(OFPS \times PNF) + NFS}{PNF + 1}$$

- NFPS(New File Print Score), OFPS(Old File Print Score), PNF(Previous Number of Files), NFS(New File Score)

⑤After calculating the frequency distribution for extracting a specific pattern of that uploaded file, we decide whether to generate trace log on that file depending on types of file through the comparison of identification DB of known BFD.

3 Conclusions

The rapid advancement of the internet makes it possible the real-time information sharing of the data, supporting the diversity and universality of user approach and providing enhanced information service. Especially, web-based file sharing site has a strong point that the user can be provided with the information whatever and whenever he or she wants, however, the analysis of the uploaded file is needed since it is used as an attack web site of network invasion attack. Therefore, proposed scheme suggests file multi analyses that can be utilized in the web-based file sharing site for constructing the trace system. To generate the trace log that is required in the trace system of the attack source and web site, the proposed scheme suggests the method for real-time cooperative analysis on the uploaded file in the web-based file

sharing site. Also, depending on the types of network security, it is possible not only to process it in real time but also to analyze it in detail, which enables the application on the attack source and web site through the analysis on the file itself.
In the future, proposed scheme will departmentalize the proposed scheme and design the optimum analyses schedule for the real-time and non-real-time analysis and minimizing calculation overhead according to each of analysis method.

Acknowledgments. This work was supported by ETRI through Maritime Safety & Maritime Traffic Management R&D Program of the MLTM/KIMST(D10902411H360000110, Development of u-VTS for Maritime Safety)

References

1. David Evans, "LCLint User's Guide", University of Virginia, Department of Computer Science, May 2000.
2. John Viega, J.T. Bloch, Yoshi Kohno, Gary McGraw, “ITS4: A Static Vulnerability Scanner for C and C++ Code,” ACSAC’00, pp. 257-267, 2000.
3. Martin Elsman, Jerey S. Foster, and Alexander Aiken. “Carillon - A System to Find Y2K Problems in C Programs”, 1999. http://bane.cs.berkeley.edu/carillon
4. Renico Koen, Martin S. Olivier, “The user of file timestamps in digital forensics,” ISSA 2008, pp.1-17, 2008.
5. Rutkowska Joanna, "Advanced Windows 2000 Rootkit Detection Execution Path Analysis", 2003.
6. Wankyung Kim , WooYoung Soh, ″ Design and Implementation of the Detection Tool of API Hooking Based on Window XP Kernel,″ Journal of Security Engineering, vol 7, No.4, pp.385-397, 2010.