A Survey for an Efficient Secure Guarantee in Network Flow

S.Kalaiselvi[1], S.Yoga[2]

[1] Research Scholar, Department of Computer Science, Sakthi College of Arts and Science for Women, Oddanchatram, India
[2] Assistant Professor, Department of Computer Science, Sakthi College of Arts and Science for Women, Oddanchatram, India

---***---

Abstract - *Network flows carry extremely confidential information that should not be released for privacy reasons. Existing techniques for network flow sanitization are vulnerable to different kinds of attacks, and solutions proposed for micro data anonymity cannot be directly applied to network traces. In our previous research, we proposed an obfuscation technique for network flows, providing formal confidentiality guarantees under realistic assumptions about the adversary's knowledge. Put forward an obfuscation technique that leads to confidential guarantee of IP address thus securing the sensitive data. In this paper, we identify the threats posed by the incremental release of network flows and by using Secured Hash algorithm 3 (SHA3) we formally prove the achieved confidentiality guarantees. For this operation, a fingerprint is created which is based on the configuration of the system. Then, the process of grouping is done using the generated signature. Group intimation is done and the set of IP addresses and signature are compared and the requested signature is send as response. All this processes occur with an intermediate router. Only, the obfuscated signature will be visible to the hacker.*

***Key Words*: Security, Incremental release, Obfuscation, Code Security, Code obfuscation techniques**

1. INTRODUCTION

Obfuscation is the obscuring of intended meaning in communication, making the message confusing, will fully ambiguous, or harder to understand. It may be intentional or unintentional (although the former is usually connected) and may result from circumlocution (yielding wordiness) or from use of jargon or even argot (yielding economy of words but excluding outsiders from the communicative value). Unintended obfuscation in expository writing is usually a natural trait of early drafts in the writing process, when the composition is not yet advanced, and it can be improved with critical thinking and revising, either by the writer or by another person with sufficient reading comprehension and editing skills. Data obfuscation (DO) is a form of data masking where data is purposely scrambled to prevent unauthorized access to sensitive materials. This form of encryption results in unintelligible or confusing data. DO is also known as data scrambling and privacy preservation.

There are two types of DO encryption:

1. Cryptographic DO: Input data encoding prior to being transferred to another encryption schema.
2. Network security DO: Payload attack methods are purposely enlisted to avoid detection by network protection systems.

To provide high security in network flows. To obfuscate the high sensible incremental data using obfuscation techniques. With respect to our previous work, the original contributions of this paper consist in:

1) The identification of confidentiality threats that may arise from the incremental release of network traces.

2) A novel defense algorithm to apply obfuscation to incremental releases of network traces.

3) A theoretical proof of the confidentiality guarantees provided by the defense algorithms.

4) An extensive experimental evaluation of the algorithm for incremental obfuscation, carried out with billions of real flows generated by the border router of a commercial autonomous system.

2. RELATED WORK

Early techniques for network flow obfuscation were based on the encryption of source and destination IP addresses. However, those techniques proved to be ineffective since an adversary might be able to re-identify message source and destination by other values of network flows against network flow sanitization methods these techniques, fall into two main categories are Fingerprinting and Injection.

Fingerprinting:

Messages re-identification is performed by matching fields' values to the characteristics of the target environment (knowledge of network topology and settings, OS and services of target hosts, etc.).Typical re-identifying values for network flows are type of service, TCP flags, number of bytes, and number of packets per flow.

Injection:

The adversary injects a sequence of flows in the target network that are easily recognized due to their specific characteristics; e.g., marked with uncommon TCP flags, or following particular patterns.

3. LITERATURE REVIEW

J. King, K. Lakkaraju, and A. J. Slagell[1] Many new attacks have been created, in parallel, that try to exploit weaknesses in the anonymization process. In this paper, we present a taxonomy that relates similar kinds of attacks in a meaningful way. We also present a new adversarial model which we can map into the taxonomy by the types of attacks that can be perpetrated by a particular adversary. This has helped us to negotiate the trade-offs between data utility and trust, by giving us a way to specify the strength of an anonymization scheme as a measure of the types of adversaries it protects against.

S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, and M. K. Reiter[6] Encouraging the release of network data is central to promoting sound network research practices, though the publication of this data can leak sensitive information about the publishing organization. To address this dilemma, several techniques have been suggested for anonymizing network data by obfuscating sensitive fields. In this paper, we present new techniques for inferring network topology and deanonymizing servers present in anonymized network data, using only the data itself and public Information. Quantify the effectiveness of our techniques, showing that they can uncover significant amounts of sensitive information. We also discuss prospects for preventing these deanonymization attacks.

Y. Song, S. J. Stolfo, and T. Jebara[4]Modern network security research has demonstrated a clear necessity for open sharing of traffic datasets between organizations a need that has so far been superseded by the challenges of removing sensitive content from the data beforehand. Network Data Anonymization is an emerging field dedicated to solving this problem, with a main focus on removal of identifiable artifacts that might pierce privacy, such as usernames and IP addresses. However, recent research has demonstrated that more statistical artifacts, also

Present, may yield fingerprints that are just as differentiable as the former. This result highlights certain short comings in current anonymization frameworks particularly, ignoring the behavior of network protocols, applications, and users. Network traffic synthesis (or simulation) is a closely related complimentary approach which, while more difficult to accurately execute, has the potential for far greater flexibility. This paper leverages the statistical-idio syncrasies of network behavior to augment anonymization and traffic-synthesis techniques through machine-learning models specifically designed to capture host-level behavior. We present the design of a system that can automatically learn models for network host behavior across time, then use these models to replicate the original behavior, to interpolate across gaps in the original traffic, and demonstrate how to generate new diverse behaviors. Further, we measure the similarity of the synthesized data to the original, providing us with a quantifiable estimate of data fidelity.

A.J.Slagell, K.Lakkaraju, and K.Luo[7] FLAIM (Framework for Log Anonymization and Information Management) addresses two important needs not well addressed by current log anonymizers. First, it is extremely modular and not tied to the specific log being anonymized. Second, it supports multi-level anonymization, allowing system administrators to make fine-grained trade-offs between information loss and privacy/security concerns. In this paper, we examine anonymization solutions to date and note the above limitations in each. Specifically, they have been one-size-fits-all tools, addressing only one type of log, often anonymizing only one field in one way. We desperately need more flexible tools that are highly extensible, multi-level, supporting many options for each field, allowing one to customize the level of anonymization for their needs, multi-log capable, being flexible enough to support the anonymization of most security relevant logs without major modification andhave a rich supply of anonymization algorithms available for use on various fields.

S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter [8] Anonymization of network traces is widely viewed as a necessary condition for releasing such data for research purposes. For obvious privacy reasons, an important goal of trace anonymization is to suppress the recovery of web browsing activities .While several studies have examined the possibility of reconstructing web browsing activities from anonymized packet-level traces, we argue that these approaches fail to account for a number of challenges inherent in real-world network traffic, and more so, are unlikely to be successful on coarser Net- Flow logs. By contrast, we develop new approaches that identify target web pages within anonymized Net Flow data, and address many real-world challenges, such as browser caching and session parsing. The major benefit in web browsing privacy in anonymized netflows is, anonymized net flow traces poses to the privacy of web browsing behaviors. Identify the flows that constitute a web page retrieval and the effects of browser caching, content distribution networks, dynamic web pages, and HTTP pipelining and the drawback are anonymization offers less privacy to web browsing traffic. Non-trivial amount of information about web browsing behaviors is leaked in anonymized network.

4. CONCLUSIONS

The basic advantage of code obfuscation is the low cost and the flexibility of this technique. It is important to note that code obfuscation is largely an “art” rather than science. The main goal of it is to make attacking complicated enough to repulse attackers, rather than formally proving the strength of algorithms; for example, in games industry, the majority of revenue happens during the first few weeks of releasing new game software. Thus, code obfuscation is used only to resist attacking during such small time window. We are arguing for its utility for the opposite motive of securing cloud computing environments. In this paper we surveyed the need for code obfuscation to make the resulting code unintelligible for human and hard to reverse engineer or being tampered by automated tools. We also briefly reviewed code obfuscation methods and techniques and quantitatively evaluate their benefits in accordance to a set of reasonable criteria. We hope that continuous study in this field would help to find more defenses that would finally lead to the total endorsement of this new paradigm in our everyday life without any concerns.

ACKNOWLEDGEMENT

The authors would like to thank Dr. Zhengguo Li, Dr. Mingli Song and Dr. Bo Gu for providing the fusion results for comparison, Zhengfang Duanmu and the anonymous reviewers for constructive comment.

REFERENCES

[1] J. King, K. Lakkaraju, and A. J. Slagell, "A taxonomy and adversarial model for attacks against network log anonymization," in Proc. ACMSAC, 2009, pp. 1286–129

[2] Sorna Sukanya G, "Rattle Adversary In IP Address Race Of Puzzler Networks", in International Journal of Research in Computer and Communication Technology, Vol 4, Issue 3 , March -2015

[3] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, "Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme," Comput. Netw., vol. 46, no. 2, pp.253– 272, 2004.

[4] Y. Song, S. J. Stolfo, and T. Jebara, "Behavior-based network trafficsynthesis," in Proc. IEEE HST, 2011, pp. 338–344.

[5] G. Dewaele, Y. Himura, P. Borgnat, K. Fukuda, P. Abry, O. Michel, R. Fontugne, K. Cho, and H. Esaki, "Unsupervised host behavior classification from connection patterns," Int. J. Netw. Manag., vol. 20, no.5, pp. 317–337, Sep. 2010

[6] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, and M. K. Reiter, "Playing devil's advocate: Inferring sensitive information from anonymized network traces," in Proc. NDSS, 2007.

[7] A. J. Slagell, K. Lakkaraju, and K. Luo, "FLAIM: A multi-level anonymization framework for computer and network logs," in Proc.LISA, 2006, pp. 63–77.

[8] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter, "On Web browsing privacy in anonymized NetFlows," in Proc.USENIX Security, 2007, pp. 339–352.

[9] J. Mirkovic, "Privacy-safe network trace sharing via secure queries,"in Proc. ACM NDA, 2008, pp. 3–10.

[10] J. C. Mogul and M. F. Arlitt, "SC2D: An alternative to trace anonymization," in Proc. MineNet, 2006, pp. 323–328.