Journal of Information Security and Applications 38 (2018) 28–39

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Towards privacy preserving threat intelligence

Sashank Dara[a,b,*], Saman Taghavi Zargar[c], VN Muralidhara[d]

[a] Security Business Group, Cisco Systems Inc. Bangalore, India
[b] International Institute of Information Technology, Bangalore, India
[c] Technical Staff in Security Business Group, Cisco Systems Inc. San Jose, USA
[d] International Institute of Information Technology, Bangalore, India

ARTICLE INFO

Article history:

Keywords:
Threat intelligence
Privacy preserving
Private information retrieval
Intrusion detection

ABSTRACT

As modern threats become more sophisticated, it is imperative for organizations to defend with the global context. Many cloud based services provide *threat intelligence* pertaining to modern advanced persistent threats (APTs). Cloud services such as: Google Safe Browsing, PhishTank, and Malwr offer black lists of known malicious URLs, domains, emails *etc*. Querying such services require users to share their browsing history and files in order to know whether their machines got infected or not. One of the major concerns/hindrances remained to be addressed to benefit from such services is the users' privacy. In this paper, we concretely identify various privacy concerns in different threat intelligence services. We introduce the general notion of *Privacy Preserving Threat Intelligence (PPTI)* to address such concerns.

As one of the major efforts towards addressing the users' privacy concerns while querying public databases, *Private Information Retrieval (PIR)* techniques have been proposed. They enable a *User* to retrieve an element from a public database privately. Many of the traditional *PIR* techniques assume that *User* is aware of the address of the element to be retrieved. In this paper, we identify two major advancements that are needed for *PIR* in designing the privacy preserving threat intelligence services: *(i)* private retrieval of the elements using keyword(s), and *(ii)* private retrieval of matching documents. In doing so, we introduce relevant schemes needed and propose a generic architecture. We also identify a specific use case for privacy preserving spam intelligence and present our experimental results. Although our experimental evidence show some limitations, we believe our work aides in formulating and advancing the technology and we present our future direction towards addressing the limitations presented.

All our source code is open sourced and publicly available.

1. Introduction

Modern threat actors are well funded, organized, and their attacks happen at a global scale. In order to defend against such powerful adversaries, organizations seek help of publicly available (both free and commercial) threat intelligence services at the cost of privacy of their users. Examples of such services are Google Safe Browsing [1], PhishTank [2], Email Spam Test [3] and many others [4].

Attack artifacts like known *black listed domains, URLs, IP addresses, and malicious files/email hashes etc.* are collectively identified as Indicators of Compromise (IoCs). IoCs together with other security data such as: IP reputations, user reputations, vulnerabilities, signatures *etc.* are broadly termed as *Threat Intelligence (TI)* [5]. Threat intelligence services are provided by commercial vendors, open source communities, government agencies, consortia *etc.*

1.1. Threat Intelligence

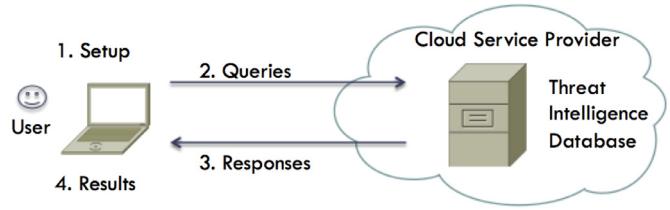
TI services are considered vital for modern intrusion detection [6–8]. We briefly list them below to familiarize the readers with such services:

1.1.1. URL Reputation

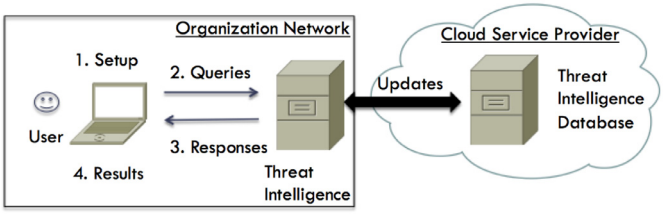
Modern sophisticated attacks like phishing, domain generated algorithms, command and control (C&C) communications, data exfiltration are predominantly web based [9,10]. In order to safe guard the users from such malicious web based attacks, it is critical to know the reputation of the URLs. Many prominent TI services exist that provide URL scanning service [4]. Given a particular URL, such services determine whether it is *known bad* from its corpus of information.

* Corresponding author.
E-mail addresses: sadara@cisco.com (S. Dara), staghavi@cisco.com (S.T. Zargar), murali@iiitb.ac.in (V. Muralidhara).

https://doi.org/10.1016/j.jisa.2017.11.006
2214-2126/



(a) TI Lookups



(b) TI Download

Fig. 1. Threat Intelligence's Users' *Telemetry* consumption options.

1.1.2. Malware attributes

Similar services exist for determining malicious public IP addresses [11], malware signatures [12], spam [3], *etc.* In order to determine malicious activity, organizations submit their artifacts (files, documents, emails *etc.*) to these services (mostly cloud services) and subsequently get the analysis results back.

Organizations can leverage threat intelligence services in order to defend against modern threats that happen at global scale [13,14]. However, any TI service requires to access the users' *telemetry* (e.g., users' web browsing history, network logs, emails, and *etc.*) to identify the attacks for its desired functionality. In other words, in order to record the malicious behavior and later identify the infections, TI services need the collection and sharing of users' *telemetry*. There are basically two modes for users' *telemetry* consumption (as it is shown in Fig. 1):

1. Users can query the cloud TI services with their local telemetry (Fig. 1a).
2. Users can download the attack artifacts (TI) to integrate with their internal security analytics (Fig. 1b).

One of the main challenges with regards to downloading the whole TI is for the resource constrained organizations as such global threat intelligence is voluminous in nature. Organizations mostly prefer to perform a customized TI lookup through cloud TI services, as oppose to receiving them as feeds, to subsequently perform appropriate actions after identification of an infection.

1.2. Privacy challenges

Sharing user's telemetry could severely compromise the privacy of individuals and organizations. Often users' telemetry contains:

1. Sensitive *Personally Identifiable Information (PII)* such as: email-ids, usernames, addresses, SSNs, credit card numbers, device identifiers (e.g., IP addresses), and *etc.*
2. *Protected Health Information (PHI)* such as: health related websites visited, health records, test reports, and *etc.*
3. Classified and confidential information such as: sensitive documents, trade secrets, and *etc.*

Many organizations are constrained to leveraging such cloud TI services (i.e., performing TI lookups) due to privacy compliance, regional laws, risk of information leakage *etc.*; This results in the lack of global context of the threats [15]. We discuss some of the challenges in current privacy measures next and discuss their shortcomings.

1.3. Current privacy measures

1.3.1. URL Sanitization

Sharing the web browsing data would severely breach the privacy of an individual. In order to prevent such privacy breach, anonymization techniques have been proposed in which parts of the URL is encrypted or certain parameters are truncated [16]. However, such sanitization offers very poor privacy guarantees and is considered very weak. Here are few examples of such weak remedies available in the literature:

(a) *Hashing*: Few services like *Google Safe Browsing* provide an option to query their service using *hashed* URLs. This only provides weak privacy protection. An *adversary* could simply compute a dictionary of hashes for all popular websites and match them against the one *user* provided in his/her query.
(b) *Truncation*: Trimming the *user* parameters beyond "?" in a URL to remove private information could be a mean to protect users' privacy. However, this is a counter productive while matching genuinely malicious domains containing parameters.
(c) *Selective encryption*: Selectively encrypting parts of the URL in order to hide private information also suffers from the same challenges as *truncation*.
(d) *Truncation* & *selective encryption*: Combining the aforementioned approaches do not offer any privacy protection if a *user* has a personal website or a blog like (*mywebsite.blog.com*) in the browsing history.

1.3.2. File matching

Many *Threat Intelligence* services require to match files (i.e., documents, emails, binaries, executables *etc.*) in order to identify malicious activities. Anonymization of sensitive information within such files (to protect their users' privacy) might be counter productive for identification of malicious behavior. For instance, obfuscating an email address in the headers would render futile Spam detection algorithms. Matching files based on their cryptographic signatures (i.e., file digests such as: SHA2, SHA3, MD5) offers high fidelity but low fragility as the adversary could slightly change the file contents to generate a new file hash and evade the detection. Any other advanced static analysis needs the files to be shared with the cloud TI provider in order to determine their disposition.

1.3.3. Attack attribution

Attributions of threat activities to their corresponding threat actors is vital for any subsequent legal actions. In order to identify such attributions, relations between: malicious actors, email addresses, their registered domains, and *etc.* is needed. Often, such relations are mapped as graph databases and made available for investigators [17]. Querying such databases also reveals the privacy of the users. The privacy preserving approaches in which the protection is supposedly achieved through anonymization techniques have been well studied in the literature and mostly been established to be prone to inferential attacks [18–22]. Moreover, other proposed techniques that area based on statistical perturbation are probabilistic in nature and not applicable when the telemetry is precisely needed to be evaluated to confirm that whether a host/user is infected or not. For instance, if a given URL needs to be determined whether it is good or bad then an addition of statistical noise to protect the user privacy could lead to unsuccessful detection verdict.