Network Working Group
Internet-Draft
Intended status: Experimental
Expires: March 14, 2021

S. Rao
S. Nagaraj
Grab
S. Sahib
R. Guest
Salesforce

September 10, 2020

Personal Information Tagging for Logs
draft-irtf-pearg-pitfol-00

Abstract

Software systems typically generate log messages in the course of their operation. These log messages (or 'logs') record events as they happen, thus providing a trail that can be used to understand the state of the system and help with troubleshooting issues. Given that logs try to capture state that is useful for monitoring and debugging, they can contain information that can be used to identify users. Personal data identification and anonymization in logs is crucial to ensure that no personal data is being inadvertently logged and retained which would make the logging system run afoul of laws around storing private information. This document focuses on exploring mechanisms that can be used by a generating or intermediary logging service to specify personal or sensitive data in log message(s), thus allowing a downstream logging server to potentially enforce any redaction or transformation.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 14, 2021.

Table of Contents

1. Introduction

Logs capture the state of a software system in operation, thus providing observability. However, because of the amount of state they capture, they can often contain sensitive user information [link: twitter storing passwords]. Personal data identification and redaction is crucial to make sure that a logging application is not storing and potentially leaking users' private information. There are known precedents that help discover and extract sensitive data, for example, we can define a regular expression or lookup rules that will match a person's name, credit card number, email address and so

on. Besides, there are data dictionary based training models that can analyze logs and predict presence of sensitive data and subsequently redact it. This document proposes an approach and framework for creating logs with personal information tagged, thus marking a step towards privacy aware logging. Once personal information is identified in a log, it has to be appropriately tagged at source. Personal data tagging is especially important in cases where log data is flowing in from disparate sources. In cases where tagging at source is not possible (e.g. log data generated by a legacy application, IoT device, Web server or a Firewall), a centralized logging server can be tasked with making sure the log data is tagged before passing on downstream. Once the logs are tagged, the logging application can use anonymization techniques to redact the fields appropriately. While the proposal described here can be applied to any data deemed sensitive in a log, however this document specifically discusses and illustrates tagging of personal information in logs.

2. Terminology

Personal data: RFC 6973 [RFC6973] defines personal data as "any information relating to an individual who can be identified, directly or indirectly." This typically includes information such as IP addresses, username, email address, financial data, passwords and so on. However, the definition of personal data varies heavily by what other information is available, the jurisdiction of operation and other such factors. Hence, this document does not focus on prescriptively listing what log fields contain personal data but rather on what a tagging mechanism would look like once a logging application has determined which fields it considers to hold personal data.

Structured logging: Most applications generate logs in a unidimensional format that twine together logic status and input data. This makes log output largely free flowing and unstructured without specific delimiters making it hard to segregate personal information from other text in the log. Structured logging refers to a formal arrangement of logs with specific identifiers of personal information and semantic information to enable easy parsing and identification of specific information in the log.

Privacy Sensitivity Level: Sensitivity level defines the degree of sensitivity of a data in log template or schema. Level can be enumerated on a scale 1 to 5 and defined as follows: 1 - Low risk for leaking private information and 5 - Very high risk for leaking private information>

3. Motivation and Use Cases

Most systems like network devices, web servers and application services record information about user activity, transactions, network flows, etc., as log data. Logs are incredibly useful for various purposes such as security monitoring, application debugging, investigations and operational maintenance. In addition, there are use cases of organizations exporting or sharing logs with third party log analyzers for purposes of security incident response, monitoring, business analytics, where logs can be a valuable source of information. In such cases, there are concerns about potential exposure of personal data to unintended systems or recipients.

4. Challenges with Existing Approaches

While methods of detecting personal identifiable information are continuously evolving, most approaches are around use of regular expressions, data or dataset based training models, pattern recognition, checksum matching, building custom logic.

Inconsistent Representation: When applications, services or devices, log personal information, there is no consistency in the representation of the information. For example the name of a user is often logged as either "fullname" (e.g. John Doe) or with "firstname" (John) and "lastname" (Doe).

Context: In most cases, what data is considered personal and sensitive is subjective, provisional and contextual to the data source or the application processing the data, which makes it hard to use automated techniques to identify personal data. Even for a specific domain, it's controversial whether it is possible to definitively say that a piece of data is NOT identifying.

Disparate Types of Personal Data: There are many disparate types of personal data and often require a multitude approaches for detection.

Lack of standards: There are no standards that govern formats of sensitive data making automation difficult for most common use cases.

Detection Accuracy: Most of the current PII detections tools employ regular expression based techniques or other pattern recognition techniques to identify the PII data. Due to the very nature of logs, most of the current implementations let administrators to add redaction policies based on 'likelihood' of detection probability categorized as low, medium or high. Defining a low detection scheme causes high false positives and a high detection scheme would cause PII leakage, thereby making a trade off inevitable to organizations.

5. Proposed Model

This section describes a reference model to enable tagging of personal information at source and extends it to include an approach of role or policy based redaction based on personal information annotated at source. The figure below illustrates the proposed model.

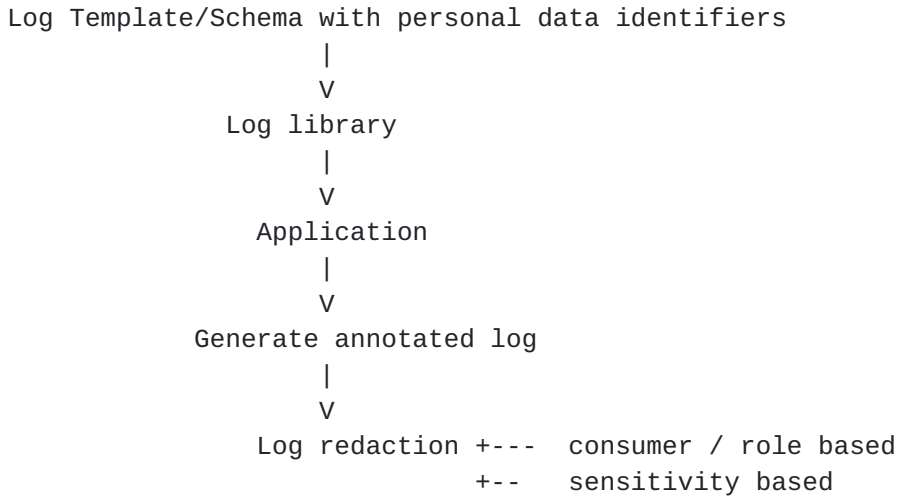

Figure 1: Flow

5.1. Defining the log privacy schema

We propose using structured logging where a log schema or a template defines standardized identifiers for every personal information and each log field is associated with a sensitivity level customized to a use case or log intent.

Note that this is not to be confused with a log severity level (WARN, INFO...) - those are typically defined "dynamically" by the developer while defining the severity of a certain scenario. A privacy sensitivity level is defined statically and is part of a log schema, associated with the log name and data type.

Name	Abstract Data Type	Description	Sensitivity [1-High 5-Normal]
nationalIdentity	String	National IDs issued by sovereign governments. Eg., SSN	1
drivingLicense	String	Driving License number	1
taxIdentity	String	Tax identification numbers	1
credtCardNumber	String	Credit cards	1
bankAccount	String	Bank account number	1
dateOfBirth	Date	Date of Birth	2
personName	String	Person name	1
emailAddress	String	Email	2
phoneNumber	Number	Phone	1
zipCode	Integer	Zip codes	5
ipAddress	ipv4Address	IPv4 or IPv6 Address	4
dateTimeSeconds	dateTimeSeconds	seconds	5
age	Integer	Age	2
ethnicGroup	String	Ethnic group	1
genderIdentity	String	Gender identity	1
macAddress	macAddress	MAC Address	4

Personal Information Identifiers Registry

If an organization already uses structured logging with a log schema, then a privacy sensitivity level can be an additional attribute for the schema.

The privacy sensitivity level for log types is intended to be defined by a centralized effort around privacy preservation in logs. In other words, this mapping might be done by an organization's privacy team (which can include lawyers, engineers and privacy professionals). The intention is that all logs generated by an org should conform to this structured format, which would ease downstream processing of logs for access control and removal of sensitive information.

If the log is being generated by a web server, then two approaches can be taken:

1. Modify log-format for the service: identify the log data type of each piece of log data generated, and tag in generation (examples provided in later section)

2. Add automated tagging in a centralized log aggregator: collect all the logs generated by different services and apply the annotation using the log schema at the aggregator

5.2. Typical Workflow

1. The log privacy schema can be parsed into a structured logging library, that is used by individual developer teams. The intention is for developers to not log arbitrary data i.e. they are asked to identify what is the data type of the state they want to preserve.

2. Any addition to the log schema would have to go through review of the privacy team that came up with the log schema.

3. Once a log is generated, tagged and stored, various kinds of access control techniques can be applied to who can access the logs.

5.3. Log Processing and Access Control

1. Consumer Role Based Access

   A. Once the log is tagged, access to it can be based on a consumer's role and privilege level.

   B. A consumeer role based policy can define what level of sensitivity they can access.

2. Case-based access

   A. If there is a genuine case for which access to sensitive information is needed and granted by the legal department, a cryptographically-signed token (e.g.JWT) can be generated that will allow access to a developer/user to logs of an increased log level. This access can be temporal in nature i.e. the token will only be valid for a certain amount of time.

   B. A transaction ID can also be propagated automatically throughout the request processing, to correlate different

logs related to a single request. Note that the notion of a "request" can vary based on what the application is doing. The idea is to have a single unifying ID to tie a particular action. If this is done, then the temporary token can be restricted to a particular request ID.

3. Redaction Techniques

A. Given that the log is tagged, an organization might choose to redact the more sensitive logs i.e. ones above a certain sensitivity level, ones of a certain log type.

B. More sophisticated approaches can be developed i.e. completely redact log types username and email, but obfuscate IP address so that a rough location can be garnered from the log record. In this way, techniques such as differential privacy can be used in tandem to have privacy guarantees for logs while still providing usefulness to developers.

6. Examples

An example based on RFC 3164 Log format

Normal Log Ouput

```
<120> Nov 16 16:00:00 10.0.1.11 ABCDEFG: [AF@0 event="AF-Authority
failure" violation="A-Not authorized to object" actual_type="AF-A"
jrn_seq="1001363" timestamp="20120418163258988000"
job_name="QPADEV000B" user_name="XYZZY" job_number="256937"
err_user="TESTFORAF" ip_addr="10.0.1.21" port="55875"
action="Undefined(x00)" val_job="QPADEV000B" val_user="XYZZY"
val_jobno="256937" object="TEST" object_library="CUS9242"
object_type="*FILE" pgm_name="" pgm_libr="" workstation=""]
```

Log Output with Personal Information Tagging

```
<120> Apr 18 16:32:58 10.0.1.11 QAUDJRN: [AF@0 event="AF-Authority
failure" violation="A-Not authorized to object" actual_type="AF-A"
jrn_seq="1001363" timestamp="20120418163258988000"
job_name="QPADEV000B" {personName="XYZZY" pii_sensitivity_level=1}
job_number="256937" {emailAddress="xyz@foo.com"
pii_sensitivity_level=2] [ip_addr="10.0.1.21"
pii_sensitivity_level=4] port="55875" action="Undefined(x00)"
val_job="QPADEV000B" val_jobno="256937" object="TEST"
object_library="CUS9242" object_type="*FILE" pgm_name="" pgm_libr=""
workstation=""]
```

7. IANA Considerations

IANA can consider defining a new central respository for Personal Information name and identifier registries to used in logging personal information. The personal identifier registry would enumerate namee and identifiers as described in Section 5.1.

8. Security Considerations

It is anticipated that developers will want additional log data types for capturing application logic, and might abuse an existing log type instead of going through the process of adding a new one. In such a case, the log would be incorrectly tagged. This can be mitigated by having stronger typing for the log data types i.e. restricting address to a certain string length instead of storing arbitrary length.

Encouraging developers to think carefully about what kind of data they're logging is a good practice and will lead to fewer incidents of private data being inadvertently logged. An organization might choose to have an unstructured log type for letting developers log data that truly do not fit anywhere else. This is still better than not having structured privacy-aware logging, because the potential privacy leakage is isolated to one particular field and its use can be monitored.

Having a mapping from log data type to privacy sensitivity will need continuous effort by a privacy team, which might be expensive for an organization.

Log data is often collated, propagated, transformed, loaded into different formats or data models for purposes of analytics, troubleshooting and visualization. In such cases, it is necessary and critical to ensure that personal information tagging and annotations is preserved and forwarded across format transformations.

If the privacy marking or classification changes for a log, for historical logs, the change of privacy classification is applied on subsequent access of the log.

TODO: In case of logs that are not tagged or marked with personal information, an out-of-band mechanism to communicate log template or schema with personal data identifiers can be considered. Such a mechansim can also be used to notify changes to privacy tagging or classification.

9. Acknowledgements

The authors would like to thank everyone who provided helpful comments at the mic at IETF 106 during the PEARG session. Thanks also to Joe Salowey for thoughts on aspects of log transformations, change of privacy classifications, models for privacy marking.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164, DOI 10.17487/RFC3164, August 2001, <https://www.rfc-editor.org/info/rfc3164>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

Authors' Addresses

Sandeep Rao
Grab
Bangalore
India

Email: sandeeprao.ietf@gmail.com

Santhosh C N
Grab

Email: santoshcn1@gmail.com

Shivan Sahib
Salesforce

Email: shivankaulsahib@gmail.com

Ryan Guest
Salesforce

Email: rguest@salesforce.com