

Workgroup: Network Working Group
Internet-Draft:
draft-dulaunoy-hashlookup-format-03
Published: 23 June 2022
Intended Status: Informational
Expires: 25 December 2022
Authors: A. Dulaunoy J-L. Huynen
 CIRCL CIRCL
 hashlookup format

Abstract

This document describes the hashlookup output format used to express meta information of hash values seen in databases of known files. The output description includes a common semantic. The hashlookup format is used by public and private digital forensics investigations services.

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 25 December 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

- [1. Introduction](#)
 - [1.1. Conventions and Terminology](#)
- [2. Format](#)
 - [2.1. Overview](#)
 - [2.2. Rational](#)
 - [2.3. Fields Format](#)
 - [2.3.1. Cryptographic hashing](#)
 - [2.3.2. Fuzzy hashing \(Context Triggered Piecewise Hashing\)](#)
 - [2.3.3. Additional fields](#)
 - [2.3.4. Relationships fields](#)
 - [2.4. Sample hashlookup output](#)
 - [2.4.1. Binary file](#)
 - [2.4.2. Binary file - package](#)
- [3. Implementation](#)
- [4. Security Considerations](#)
- [5. Acknowledgements](#)
- [6. References](#)
- [7. Normative References](#)
- [8. Informative References](#)
- [Authors' Addresses](#)

1. Introduction

Digital forensics is a critical field in information security and especially incident response. Providing intelligence about known set of files is crucial to avoid wasting efforts while conducting digital investigations. hashlookup format provides a common output format for diverse known databases of file hashes. Those databases are, for example, the NIST National Software Reference Library (NSRL) or Known File Filter (KFF) lists used in digital forensics software.

1.1. Conventions and Terminology

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](#)].

2. Format

2.1. Overview

The hashlookup format follows the JSON [[RFC8259](#)] format. The intent of this output to be easily parsed by machines or generated by software in stream mode. Each JSON object is expressed on a single line to be processed by the client line-by-line. Examples of JSON output are presented below.

2.2. Rational

The main goal of the hashlookup format is to share common fields and to easily combine results from different sources. There are different reference library which are used in different uses-cases such as:

- *Conducting digital forensic investigation and trace the origin of released software distribution
- *Tracing the origin and especially the overlap between software distribution to conduct copyright assessment
- *Improve and support cyber supply chain and its risk management

2.3. Fields Format

As there is a wide variety of sources with various granularities of information available, the hashlookup format has been made quite lax regarding the mandatory fields. The only condition is to have at least one cryptographic hash or fuzzy hashing value **MUST** be present in an hashlookup JSON object.

The following sections define the JSON fields which are permissible. The values in the key-value pairs are strings.

2.3.1. Cryptographic hashing

The cryptographic hashing value **MUST** be a JSON string. The string represents the hashed value of the file represented. The string **MUST** be the hexadecimal representation of the hash in upper case.

- *MD5
- *SHA-1
- *SHA-256
- *SHA-512

2.3.2. Fuzzy hashing (Context Triggered Piecewise Hashing)

The fuzzy hashing value **MUST** be a JSON string. The string represents the hashed value of the file represented.

- *TLSH
- *SSDEEP

2.3.3. Additional fields

Additional fields **MAY** be present to describe additional metadata from the file. The value **MUST** be a JSON string.

- *FileName: Filename of the hashed file (NSRL)
- *FileSize: FileSize of the hashed file (NSRL)

- *CRC: CRC of the hashed file (NSRL)
- *SpecialCode: Special code of the hashed file (NSRL) as described in [[NSRL-RDS](#)]
- *OpSystemCode: OpSystemCode of the hashed file (NSRL) as described in [[NSRL-RDS](#)]
- *ProductCode: ProductCode of the hashed file (NSRL) as described in [[NSRL-RDS](#)]
- *PackageName: Package Name of the hashed file as seen in metadata of Debian package format, RPM or similar package managers (CIRCL)
- *PackageMaintainer: Package maintainer of the hashed file as seen in metadata of the Debian package format, RPM or similar package managers (CIRCL)
- *PackageSection: Package section of the hashed file as seen in the metadata of the Debian package format, RPM or similar package managers (CIRCL)
- *PackageVersion: Package version of the hashed file as seen in the metadata of the Debian package format, RPM or similar package managers (CIRCL)
- *KnownMalicious: List of source considering the hashed file as being malicious (CIRCL)
- *tar:gname: Group name used to create the Tar archive
- *tar:uname: User name used to create the Tar archive
- *source: Source of the hashed file
- *db: Db where the hashed file come from (if it's an import of an existing dataset)
- *insert-timestamp: When the hashed file has been inserted in the hashlooup database
- *mimetype: Guessed mimetype of the file (CIRCL)
- *nsrl-sha256: Specify if the file SHA-256 comes from the original NSRL SHA-1 to SHA-256 list

2.3.4. Relationships fields

Two **OPTIONAL** fields parents and children **MAY** be present to represent the relationships with other hashlookup objects. The parent or children field **MUST** be a JSON array. The value is a JSON string representing one the hashing algorithms. The SHA-1 is the **RECOMMENDED** algorithm for the relationship. Other algorithms **MAY** be used if SHA-1 is not available.

2.4. Sample hashlookup output

2.4.1. Binary file

```
{
  "CRC32": "B4DD44A4",
  "FileName": "./bin/ls",
  "FileSize": "110080",
  "MD5": "945FEDB3A3C290D69F075F997E5320FF",
  "OpSystemCode": {
    "MfgCode": "1006",
    "OpSystemCode": "362",
    "OpSystemName": "TBD",
    "OpSystemVersion": "none"
  },
  "ProductCode": {
    "ApplicationType": "Operating System",
    "Language": "English",
    "MfgCode": "534",
    "OpSystemCode": "599",
    "ProductCode": "163568",
    "ProductName": "Vinux ",
    "ProductVersion": "5.1"
  },
  "SHA-1": "5848386F77B4C60319C68B69C4594E29959381A2",
  "SHA-256": "08AC13B08BFE4407E0F0C2E12E7F5B1B5E77EB817349A5EA1D836E83CD",
  "SpecialCode": "",
  "parents": [
    {
      "FileSize": "1090622",
      "MD5": "10A2318BE86F38A6ED113E16AABAA76B",
      "PackageDescription": "GNU core utilities\n This package contains",
      "PackageMaintainer": "Ubuntu Developers <ubuntu-devel-discuss@list",
      "PackageName": "coreutils",
      "PackageSection": "utils",
      "PackageVersion": "8.21-1ubuntu5.4",
      "SHA-1": "F335B669CCB7BA8A2FC8FAF315B1B4BFF9D4217F",
      "SHA-256": "07995A739DAEBD60297F0E9C2B44DFAB0C735A0FE08FACC097ECE0"
    }
  ]
}
```

2.4.2. Binary file - package

```
{"FileSize": "1090622", "MD5": "10A2318BE86F38A6ED113E16AABAA76B", "Pack
```

3. Implementation

A public hashlookup service [[HASHLOOKUP-SERVICE](#)] is provided by CIRCL and accessible as a ReST HTTP API. A software back-end implementation which produces a hashlookup format output is available [[HASHLOOKUP-SERVER](#)]. Commercial implementation such as [[METALOOKUP-SERVICE](#)] provides a compatible interface with the hashlookup format. The hashlookup project [[HASHLOOKUP-IO](#)] provides an umbrella for all the related open source projects using hashlookup format.

4. Security Considerations

hashlookup results events might contain sensitive or confidential information. Adequate access control and encrypted transport layer shall be implemented to ensure the confidentiality of the hashlookup results.

hashlookup results don't imply a specific assumption concerning the maliciousness or non-maliciousness of a file. hashlookup only provides the information about the presence of a file in a specific set, known source or database.

5. Acknowledgements

The authors wish to thank all the users of the CIRCL hashlookup services for their feedback.

6. References

7. 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>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.

8. Informative References

- [HASHLOOKUP-IO] hashlookup.io, "hashlookup project - Open source tools and standards to lookup known files", <<https://www.hashlookup.io/>>.
- [HASHLOOKUP-SERVER] Dulaunoy, A., "hashlookup-server is a minimal and fast open source server (ReST/API) to lookup quickly

hash value from large dataset.", <<https://github.com/adulau/hashlookup-server>>.

[HASHLOOKUP-SERVICE] CIRCL.LU, "CIRCL hash lookup is a public API to lookup hash values against known database of files.", <<https://www.circl.lu/services/hashlookup/>>.

[METALOOKUP-SERVICE] miwakeru.com, M. S. O. B., "Metallookup is a large database of hash values and meta information about published software. Metallookup provides a fast-lookup API to quickly check forensic evidences.", <<https://www.metallookup.com/>>.

[NSRL-RDS] NIST.gov, "Data Formats of the NSRL Reference Data Set (RDS) Distribution", <<https://www.nist.gov/system/files/data-formats-of-the-nsrl-reference-data-set-16.pdf>>.

Authors' Addresses

Alexandre Dulaunoy
Computer Incident Response Center Luxembourg
16, bd d'Avranches
L-L-1160 Luxembourg
Luxembourg

Phone: [+352 247 88444](tel:+35224788444)
Email: alexandre.dulaunoy@circl.lu

Jean-Louis Huynen
Computer Incident Response Center Luxembourg
16, bd d'Avranches
L-L-1160 Luxembourg
Luxembourg

Phone: [+352 247 88444](tel:+35224788444)
Email: jean-louis.huynen@circl.lu