Workgroup: Network Working Group
Internet-Draft:
draft-richardson-opsawg-pcapng-extras-00
Published: 4 October 2021
Intended Status: Informational
Expires: 7 April 2022

Authors: M. Tuexen, Ed.
Muenster Univ. of Appl. Sciences
F. Risso J. Bongertz
Politecnico di Torino Airbus DS CyberSecurity
G. Combs G. Harris E. Chaudron M. Richardson
Wireshark Red Hat Sandelman

Additional block types for PCAP Next Generation (pcapng) Capture File Format

Abstract

This document contains a number of extensions to the PCAPng file format which are outside of the IETF networking mandate.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction to Additional Block Types

TBD

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Additional Block Types

3.1. systemd Journal Export Block

The systemd Journal Export Block is a lightweight container for systemd Journal Export Format entry data.

One of the primary components of the systemd System and Service Manager is the "Journal", a message logging system that uses arrays of key-value pairs. Journal entries are stored in a database-like file on disk but can be serialized to easily parseable "Journal Export Format" data or to a JSON object. The block described here is limited to Journal Export Format data only.

A systemd Journal Export Block contains a single systemd Journal Export Format entry. Each entry MUST contain a __REALTIME_TIMESTAMP= field. If a timestamp for the block is required it can be derived from this field. Each entry MUST be zero-padded to 32 bits. Although the primary use of this block is intended for importing data from systemd, it could potentially be used to include arbitrary key-value data in a capture file.

Figure 1 shows the format of the Journal Export Block.

```
                       1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
0 |                    Block Type = 0x00000009                    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
4 |                      Block Total Length                       |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
8 /                                                               /
  /                         Journal Entry                         /
  /              variable length, padded to 32 bits               /
  /                                                               /
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                      Block Total Length                       |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: systemd Journal Export Block Format

The systemd Journal Export Block has the following fields:

*Block Type: The block type of the Journal Export Block is 9.

*Block Total Length: total size of this block, as described in [I-D.tuexen-opsawg-pcapng], section "Section Blocks".

*Journal Entry: A journal entry as described in the Journal Export Format documentation. Entries consist of a series of field names followed by text or binary field data. Common field names can be found in the systemd.journal-fields documentation. The __REALTIME_TIMESTAMP= field MUST be present and valid as described above. Entries are not guaranteed to be a multiple of four octets and must be zero-padded. This allows the length of the entry to be determined by finding the last non-zero octet in the Journal Entry data. An entry may contain an entry separator (trailing newline) as described in the Journal Export Format specification

3.2. Alternative Packet Blocks (experimental)

Can some other packet blocks (besides the ones described in the previous paragraphs) be useful?

3.3. Compression Block (experimental)

The Compression Block is optional. A file can contain an arbitrary number of these blocks. A Compression Block, as the name says, is used to store compressed data. Its format is shown in Figure 2.

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 0 |                        Block Type = ?                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 4 |                      Block Total Length                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 8 |  Compr. Type  |                                               /
   +-+-+-+-+-+-+-+-+                                               /
   /                                                               /
   /                       Compressed Data                         /
   /                                                               /
   /  variable length, octet-aligned and padded to end on a 32-bit /
   /                         boundary                              /
   /                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Block Total Length                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Compression Block Format

The fields have the following meaning:

- Block Type: The block type of the Compression Block is not yet assigned.

- Block Total Length: total size of this block, as described in [I-D.tuexen-opsawg-pcapng], section "Section Blocks".

- Compression Type (8 bits): an unsigned value that specifies the compression algorithm. Possible values for this field are 0 (uncompressed), 1 (Lempel-Ziv), 2 (Gzip), other?? Probably some kind of dumb and fast compression algorithm could be effective with some types of traffic (for example web), but which?

- Compressed Data: data of this block. Once decompressed, it is made of other blocks.

3.4. Encryption Block (experimental)

The Encryption Block is optional. A file can contain an arbitrary number of these blocks. An Encryption Block is used to store encrypted data. Its format is shown in Figure 3.

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 0 |                        Block Type = ?                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 4 |                      Block Total Length                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 8 |   Encr. Type  |                                               /
   +-+-+-+-+-+-+-+-+                                               /
   /                                                               /
   /                       Encrypted Data                          /
   /                                                               /
   /                 variable length, octet-aligned                /
   /                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Block Total Length                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Encryption Block Format

The fields have the following meaning:

- Block Type: The block type of the Encryption Block is not yet assigned.

- Block Total Length: total size of this block, as described in [I-D.tuexen-opsawg-pcapng], section "Section Blocks".

- Encryption Type (8 bits): an unsigned value that specifies the encryption algorithm. Possible values for this field are ??? (TODO) NOTE: this block should probably contain other fields, depending on the encryption algorithm. To be defined precisely.

- Encrypted Data: data of this block. Once decrypted, it originates other blocks.

3.5. Fixed Length Block (experimental)

The Fixed Length Block is optional. A file can contain an arbitrary number of these blocks. A Fixed Length Block can be used to optimize the access to the file. Its format is shown in Figure 4. A Fixed Length Block stores records with constant size. It contains a set of Blocks (normally Enhanced Packet Blocks or Simple Packet Blocks), of which it specifies the size. Knowing this size a priori helps to scan the file and to load some portions of it without truncating a block, and is particularly useful with cell-based networks like ATM.

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 0 |                        Block Type = ?                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 4 |                      Block Total Length                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 8 |          Cell Size            |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               /
   /                                                               /
   /                        Fixed Size Data                        /
   /                                                               /
   /                 variable length, octet-aligned                /
   /                                                               /
   /                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Block Total Length                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Fixed Length Block Format

The fields have the following meaning:

- Block Type: The block type of the Fixed Length Block is not yet assigned.
- Block Total Length: total size of this block, as described in [I-D.tuexen-opsawg-pcapng], section "Section Blocks".
- Cell size (16 bits): an unsigned value that indicates the size of the blocks contained in the data field.
- Fixed Size Data: data of this block.

3.6. Directory Block (experimental)

If present, this block contains the following information:

- number of indexed packets (N)
- table with position and length of any indexed packet (N entries)

A directory block MUST be followed by at least N packets, otherwise it MUST be considered invalid. It can be used to efficiently load portions of the file to memory and to support operations on memory mapped files. This block can be added by tools like network analyzers as a consequence of file processing.

3.7. Traffic Statistics and Monitoring Blocks (experimental)

One or more blocks could be defined to contain network statistics or traffic monitoring information. They could be use to store data collected from RMON or Netflow probes, or from other network monitoring tools.

3.8. Event/Security Block (experimental)

This block could be used to store events. Events could contain generic information (for example network load over 50%, server down...) or security alerts. An event could be:

*skipped, if the application doesn't know how to do with it

*processed independently by the packets. In other words, the applications skips the packets and processes only the alerts

*processed in relation to packets: for example, a security tool could load only the packets of the file that are near a security alert; a monitoring tool could skip the packets captured while the server was down.

4. Security Considerations

TBD.

5. IANA Considerations

TBD.

[Open issue: decide whether the block types, option types, NRB Record types, etc. should be IANA registries. And if so, what the IANA policy for each should be (see RFC 5226)]

6. Contributors

Loris Degioanni and Gianluca Varenni were coauthoring this document before it was submitted to the IETF.

7. Acknowledgments

The authors wish to thank Anders Broman, Ulf Lamping, Richard Sharpe and many others for their invaluable comments.

8. References

8.1. Normative References

[I-D.tuexen-opsawg-pcapng]
Tuexen, M., Risso, F., Bongertz, J., Combs, G., Harris, G., Chaudron, E., and M. C. Richardson, "PCAP Next Generation (pcapng) Capture File Format", Work in Progress, Internet-Draft, draft-tuexen-opsawg-pcapng-03, 23 June 2021, <https://www.ietf.org/archive/id/draft-tuexen-opsawg-pcapng-03.txt>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[LINKTYPES] The Tcpdump Group, "the tcpdump.org link-layer header types registry", <http://www.tcpdump.org/linktypes.html>.

Authors' Addresses

Michael Tuexen (editor)
Muenster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
Germany

Email: tuexen@fh-muenster.de

Fulvio Risso
Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Torino
Italy

Email: fulvio.risso@polito.it

Jasper Bongertz
Airbus Defence and Space CyberSecurity
Kanzlei 63c
40667 Meerbusch
Germany

Email: jasper@packet-foo.com

Gerald Combs
Wireshark Foundation
339 Madson Pl
Davis, CA 95618
United States of America

Email: gerald@wireshark.org

Guy Harris

Email: gharris@sonic.net

Eelco Chaudron
Red Hat
De Entree 238
1101 EE Amsterdam
Netherlands

Email: eelco@redhat.com

Michael C. Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/