Exchanging Security Events of flow-based Intrusion Detection Systems at Internet Scale

Jessica Steinberger*†, Anna Sperotto†, Harald Baier* and Aiko Pras†

*da/sec - Biometrics and Internet Security Research Group
University of Applied Sciences Darmstadt
Darmstadt, Germany
Email:{Jessica.Steinberger, Harald.Baier}@h-da.de

†Design and Analysis of Communication Systems (DACS)
University of Twente,
Enschede, The Netherlands
Email:{J.Steinberger, A.Sperotto, A.Pras}@utwente.nl

I. Introduction

In recent years, network-based attacks became one of the top concerns for network infrastructure and service outage [1]. To reduce the impact of network-based attacks (e.g. Distributed Denial of Service (DDoS)) multiple attack detection methods [2] and countermeasures have been proposed [3]. In detection and countermeasures, we observe two growing trends. First, flow-based solutions are becoming more and more popular. Second, collaborative approaches, especially among trusted partners, have proven to be a necessary step to counteract large attacks [4]. However, these collaborative approaches do not take into account to exchange threat information in an interoperable, standardized format.

An *interoperable format* provides accurate, context rich, directed and actionable threat information in a timely manner [5]. In case of an ongoing network-based attack, an interoperable format supports an automatic dissemination of threat information and thus lessens the time to respond [6], [7]. A *standardized format* ensures the availability of detail and context information and reduces manual processing (e.g. normalization, exchanging data between different systems) [6], [7]. In the last years, several exchange formats (e.g., Incident Object Description Exchange Format (IODEF), Intrusion Detection Message Exchange Format (IDMEF), Abuse Reporting Format (ARF) and Extended Abuse Reporting Format (x-arf v0.1 and v0.2) have been published [8]. However, it is still a challenge to find a standardized exchange format that is thoroughly validated and tested in full scale of industry trials. A previous study [9] reported that exchange formats are often unknown for network operators. In addition, none of the exchange formats has been used in conjunction with flow-based data.

To overcome the shortcomings of missing flow-based interoperability, this research presents a new exchange format, which we call Flow-based Event Exchange Format (FLEX). FLEX is placed in high-speed networks that use links with a speed of 10 Gbps and higher [10], and use flow export technologies (e.g. Cisco NetFlow, IPFIX) to identify, track and mitigate malicious traffic [9]. Further, FLEX is intended to facilitate the cooperation among network operators and focus on an automated threat information exchange. The contribution that FLEX brings to the state of the art is that it allows the exchange of threat information based on flow data in a structured and unambiguous manner. In addition, since FLEX messages are disseminated using SMTP, FLEX is easy to deploy and it integrates with existing infrastructure.

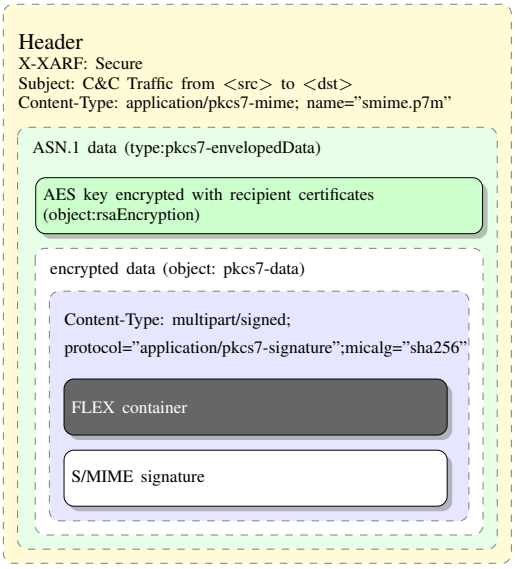

Fig. 1. Components of a FLEX message

II. FLEX

The Flow-based Event Exchange Format (FLEX) is based on the x-arf specification draft v0.2 `X-XARF:SECURE` (henceforth referred to as x-xarf). In contrast to x-xarf, FLEX uses a generic template system. This generic template system is described by an abstract syntax denoted using the language of Abstract Syntax Notation (ASN.1). Both, the generic template and the abstract syntax of FLEX prevent all ambiguities when being interpreted and handled by an automatic mitigation and response system. Further, FLEX ensures the interoperability with different flow-based export technologies as input source and makes use of both signature and encryption methods to prevent unauthorized access to the security event message at the application layer. FLEX consists of a mail header and an enveloped-data content type. The enveloped-data content type consists of an encrypted content of a signed multipart MIME message and encrypted content-encryption keys for one or more recipients [11]. The enveloped-data content consists of two parts: The first part contains the FLEX container that is signed. The second part conveys the detached signature Cryptographic Message Syntax (CMS) SignedData object. Figure 1 visualizes the components of a FLEX message.

The FLEX container is composed of data arranged in vector form and represented by $\mathcal{F} = \{s_0, \ldots, s_{n_0}, f_0, \ldots, f_{n_1}, d_0, \ldots, d_{n_2}, c_0, \ldots, c_{n_3}\}$, where s represents the settings, f the flow fields of the flow export technology, d additional information provided by the detection engine and c security event related flow data. In addition, the FLEX container uses the Octet Encoding Rules (OER) of ASN.1. In contrast to other encoding rules (e.g., Basic Encoding Rules (BER), Canonical Encoding Rules (CER), Distinguished Encoding Rules (DER)), OER produces a

compact octet-oriented encoding and increases the encoding/decoding speed. Listing 1 shows the ASN.1 structure of a FLEX message and can be read: a FLEX message is defined as a FLEXRecord that consists of a structure (SEQUENCE) with four components: settings, flowfields, detection and correlation. These components are called identifiers. In addition, these identifiers are described by a type. For example, settings denotes data of type Settings and correlation, denotes a list (SEQUENCE OF) of data which are all of type CorrelatedFlows. Besides an identifier and a type, the FLEXRecord contains tags. A tag is a number between square brackets before a type. In order to properly decode a FLEX message and remove all ambiguities a FLEX messages uses explicitly tags.

The first vector stores setting information $(s_0, \ldots, s_{n_0})$ on how to interpret the transmitted data. The settings vector contains a unique message identifier, a description of the event type and the type and version of the flow data. The data representation of the settings vector is described in Listing 2. The flow field types of a flow format are stored in the vector $(f_0, \ldots, f_{n_1})$. The quantity of the flow field types depends on the ASN.1 choice element used to enter the flow data. Thus, FLEX uses a generic template system that provides the capabilities to exchange several flow-based security events. This generic template is shown in Listing 3.

```
FLEXRecord ::= [APPLICATION 0] SEQUENCE {
settings      [0] Settings,
flowFields    [1] FlowFieldTypes,
detection     [2] Detection,
correlation   [3] SEQUENCE OF CorrelatedFlows DEFAULT {}
}
```

Listing 1. ASN.1 structure of a FLEX message

```
Settings ::= [APPLICATION 1] SEQUENCE {
id        [0] INTEGER,
eventType [1] ENUMERATED{CuC(0), DDoS(1)},
type      [2] ENUMERATED{netflow(0), ipfix(1)},
version   [3] ENUMERATED{two(0), four(1), five(2), nine(3)}
}
```

Listing 2. Settings

```
FlowFieldTypes ::= CHOICE {
netflow5 [0] NetFlow5,
netflow9 [1] NetFlow9,
ipfix    [2] Ipfix
}
```

Listing 3. Flow field types

Next, information of a detection engine is stored in the vector $(d_0, \ldots, d_{n_2})$. The detection engine provides additional data such as severity, impact, priority, NAT, reliability, correlated data flow sets and an observation ID. The data representation of the detection engine vector is presented in Listing 4. The last vector $(c_0, \ldots, c_{n_3})$ provides optional data and is composed of unique identification numbers of the correlated flow sets and an identification number of the observation point. The data representation of flows related to this security event is shown in Listing 5. Finally, this FLEX message is placed within the FLEX container that is signed. The encrypted data content consists of both, the FLEX container and the detached signature.

```
Detection ::= [APPLICATION 2] SEQUENCE {
severity      [0] INTEGER,
impact        [1] INTEGER,
priority      [2] ENUMERATED{high(0), medium(1), low(2)},
nat           [3] ENUMERATED{true(0), false(1), na(3)},
observationID [4] INTEGER
}
```

Listing 4. Additional data of the detection engine

```
CorrelatedFlows ::= SEQUENCE {
flowID        [0] INTEGER,
observationID [1] INTEGER
}
```

Listing 5. Correlated flows of a FLEX message

The main advantage of FLEX over existing exchange formats lies in the generic template system that provides extensibility and machine readability to support automatic processing of security events. In addition, FLEX integrates with the existing infrastructure using SMTP and thus is easy to deploy. Further, FLEX constitutes an viable and more structured alternative to share threat information based on flow data.

ACKNOWLEDGEMENTS

The work has been funded by the German Federal Ministry of Education and Research (#03FH005PB2), CASED and by EU FP7 Flamingo (ICT-318488).

REFERENCES

[1] D. Anstee, C. Chui, J. Escobar, and G. Sockrider, "Worldwide Infrastructure Security Report," Arbor Networks Inc., Tech. Rep. X, Jan. 2015, http://www.arbornetworks.com/research/infrastructure-security-report.

[2] G. Münz and G. Carle, "Real-time Analysis of Flow Data for Network Attack Detection," in *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, May 2007, pp. 100–108.

[3] H. Beitollahi and G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," *Computer Communications*, vol. 35, no. 11, pp. 1312–1332, 2012.

[4] J. François, I. Aib, and R. Boutaba, "Firecol: A collaborative protection network for the detection of flooding DDoS attacks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1828–1841, Dec 2012.

[5] TM Forum, "Sharing Threat Intelligence to Mitigate Cyber Attacks," http://www.tmforum.org/browse.aspx?linkid=51490&docid=19968, 2013.

[6] B. Hartman, D. Marting, D. Moreau, K. Moriarty, E. Schwartz, and P. M. Tan, "Breaking Down Barriers to Collaboration in the Fight Against Advanced Threats," http://www.emc.com/collateral/industry-overview/11652-h9084-aptbdb-brf-0212-online.pdf, 2012.

[7] K. Moriarty, "Transforming Expectations For Threat-Intelligence Sharing," http://www.emc.com/collateral/emc-perspective/h12175-transf-expect-for-threat-intell-sharing.pdf, 2013.

[8] J. Steinberger, A. Sperotto, M. Golling, H. Baier, and A. Pras, "How to Exchange Security Events? Overview and Evaluation of Formats and Protocols," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, May 2015, to appear.

[9] J. Steinberger, L. Schehlmann, S. Abt, and H. Baier, "Anomaly Detection and Mitigation at Internet Scale: A Survey," in *Emerging Management Mechanisms for the Future Internet*, ser. Lecture Notes in Computer Science, G. Doyen, M. Waldburger, P. Čeleda, A. Sperotto, and B. Stiller, Eds. Springer Berlin Heidelberg, 2013, vol. 7943, pp. 49–60. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38998-6_7

[10] M. Golling, R. Hofstede, and R. Koch, "Towards multi-layered intrusion detection in high-speed networks," in *Cyber Conflict (CyCon 2014), 2014 6th International Conference On*, June 2014, pp. 191–206.

[11] R. Housley, "Cryptographic Message Syntax (CMS)," RFC 5652 (Standard), Internet Engineering Task Force, Sep. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5652.txt