NETFOX.FRAMEWORK - NETWORK FORENSIC EXTENDABLE ANALYSIS TOOL

Jan Pluskal
Master Degree Programme (2MIT), FIT BUT
E-mail: xplusk03@stud.fit.vutbr.cz

Supervised by: Ondřej Ryšavý
E-mail: rysavy@fit.vutbr.cz

Abstract: Network forensic analysis is widely discussed topic in the last decade thanks to the rapid networking development. As computer networks grow and a new equipment is being connected every second, a crucial need for an efficient network monitoring tool arises. This paper introduces a network forensic platform, called Netfox.Framework (NFX), an open-source, extensible, and a modular analytic software framework, providing a conversation-based approach usable for advanced data-mining in captured communication.

Keywords: Network forensic analysis, Netfox framework, network data-mining

1 INTRODUCTION

Network forensic analysis has become an indispensable security incident investigation technique. Incidents, like stated in ISO/IEC 27035:2011, are changes in an information system, or a network in everyday operations, resulting in a possible security safeguard of policy violation. To prevent these incidents from happening, there are commonly used methods based on a netflow statistic collection to determine a network base line. Anomalies are then changes in the flow that does not fit the base line and are detectable by IDS/IPS systems.

Although there are other kinds of incidents, sophistical attacks or malicious behavior that cannot be detected this way. Forensic investigation of these incidents aims deeper in the communication to obtain evidence from transferred data themselves, and meta-data, which are commonly collected, are no longer sufficient.

2 NETFOX.FRAMEWORK – NETWORK FORENSIC EXTENDABLE ANALYSIS TOOL

We have designed and implemented a network forensic platform, called Netfox.Framework (NFX) that has been developed as an open-source, extensible, and modular analytic software framework, providing a conversation-based approach usable for advanced data-mining in captured communication. The NFX development is driven by the need of providing a robust method to reduce a complexity and time during a development of various specific network forensic applications. Almost every possible forensic investigation use-case requires to reconstruct, at least partially, an application data layer. The NFX reassembles not only TCP/IP traffic, but also other mechanisms like tunneling protocols (in development), necessary to understand bidirectional communication up to application protocol layers.

As mentioned above, the NFX was designed to be as modular as possible with a consideration of future extensibility and maintainability. For better understanding, an abstract design diagram (see Figure 1), describing framework architecture, is provided. The NFX is divided into layers that corresponds by their functionality to an equal layer in the networking stack, or aggregate the functionality of more layers in themselves.

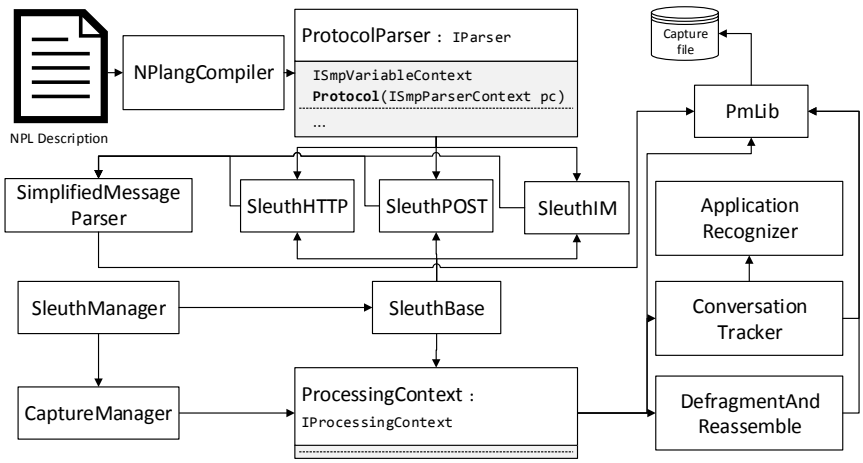

Figure 1: The abstract design diagram of the Netfox.Framework

2.1 PMLIB

The PmLib is a library written in C# by Ing. Vladimír Veselý in collaboration with Bc. Martin Mareš. The module shields upper layers from the manipulation with the PCAP files and provides a general interface for that task. The library supports many capture file formats like Wireshark/TCPDump's LibPcap, Microsoft network monitor's cap – version 2.0, and PCAP-ng's pcap – version 1.0.

The PmLib also provides unified interface to access parsed representation of frames stored in capture files. Frames are parsed to upper layers (L2, L3 and L4) using PacketDotNet 0.13 dissectors supporting Ethernet, LinuxSLL, PPP, PPPoE, IP, UDP, TCP, etc ... The module supports pcap file indexing so after one time opening there is no need to parse a whole file again and data needed for processing in the NFX are preserved in an index file.

2.2 CONVERSATIONTRACKER WITH APPLICATIONRECOGNITION

The conversation tracker separates frames according to their affiliation to a conversation. The conversation is understood as a pair of collections, gathering up and down flow frames ordered by timestamp. Frames are separated into the flows based on an equivalence relation defined on their IP Endpoints and an IP protocol type. The conversation is also the smallest data processing unit.

The ApplicationRecognizer was requested by Law enforcement units to provide advance identification of an application protocol contained in the conversation. It is safe to assume that one conversation uses only one application layer protocol. Based on that assumption, several application protocol recognizers will be provided to suit actual needs of the Netfox.Framework usage. The ApplicationRecognizer module is designed to easily support an addition of programmer's custom recognizers as a plug-in modules. Currently we are supporting a port based recognition, and are experimenting with a Statistical Protocol IDentification (SPID)[1].

2.3 DEFRAGMENTANDREASSEMBLE

The DefragmentAndReassemble (DaR) is the most important module in the Netfox.Framework. Without this module it would not be possible to reconstruct any data, because data are segmented into more frames, which could be even fragmented on routers along the way to a destination side. There is also a possibility that some frames get lost, because the IP networks are based on the best effort delivery principle. In the case of UDP the loss is definite, but by using TCP, the frames, which are not delivered to the destination, have to be re-transmitted.

The DaR is used to reorder, filter, and group frames into ordered lists storing frame numbers. This

information is contained in PDU objects holding also reassembly process statistics. If some frames are missing, this anomaly is detected by a mismatching calculated next sequence number in the TCP frame. They are substituted with virtual frames, provided by the PmLib. Every virtual frame has a unique frame number and carries an information about missing data length to be used as a stuffing space, when the PmLib is called to retrieve the data from selected frames. Concerning UDP, missing frames cannot be detected on the L4. The full module and the algorithm description will be provided in the Jan Pluskal's Diploma thesis [2].

2.4 SLEUTHS

The Sleuths are top layer components that are aggregating the information extracted of an intercepted communication on a network, parsed by the SimplifiedMessageParser. The main functionality is to give a semantic meaning to the syntactically parsed data. Therefore one specific application protocol has to correspond to one Sleuth, which understands the protocol semantic and creates the added value by exportation the forensically significant data to the XML export log for a future analysis.

2.5 SIMPLIFIEDMESSAGEPARSER

Acts as an interpret of the compiled network protocol description in a Microsoft network parsing language, provided by the NPlangCompiler written by Ing. Ondřej Ryšavý, Ph.D. The module using the PDU object, recovered by the DaR, provides the data to the generated parser. After an application message is successfully parsed, the control is returned to the Sleuth to process the obtained data.

2.6 SLEUTHMANAGER

The SleuthManager (SM) controls data processing so that the application programmer can use the framework without understanding its full functionality and is shielded from the inner implementation. This lets the SM to control the entire data-mining and increases a computing resource utilization, because the SM has actual information and can prioritize longer running sleuths to simpler ones.

3 CONCLUSION

The current version of the NFX represents a fully working proof of concept supporting common network and transport protocols, providing the data-mining functionality for selected application protocols, namely the HTTP, IMAP, SMTP, POP3, OSCAR, XMPP, YMSG and MSN. The future development of the NFX is focused on providing a scalable solution that can be deployed in an distributed environment to handle big forensic data. Also, the current research is aiming on an intelligent application data classification and processing. These intelligent methods will be implemented as plugins to the framework performing specific analytic functions. Based on preliminary experiments with real world data, we also identified the need of robust methods for processing lower-layer protocols including a data decapsulation from a tunneled traffic.

ACKNOWLEDGEMENT

This work was partly supported by project – Modern Tools for Detection and Mitigation of Cyber Criminality on the New Generation Internet, MV, VG20102015022.

REFERENCES

[1] Erik Hjelmvik: The SPID Algorithm - Statistical Protocol IDentification, Gävle, Sweden. October 2008

[2] Jan Pluskal: Framework for Captured Network Communication Processing, diploma thesis, Brno, FIT VUT v Brně, 2014