Developing a Taxonomy of Message Formats from Network Protocols

Annie Li
Yijing.Li@tufts.edu

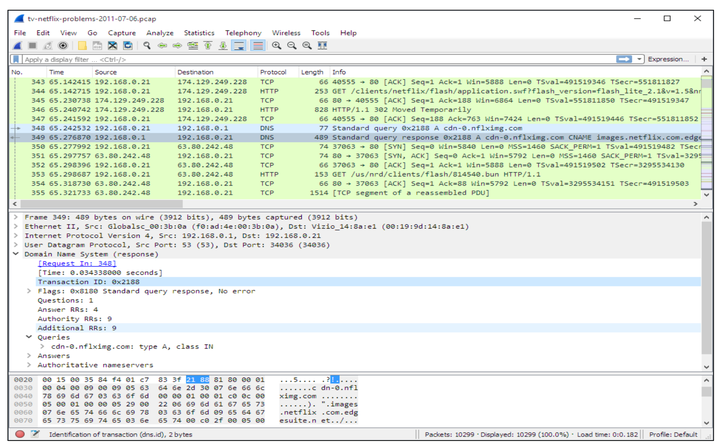
Jared Chandler
Jared.Chandler@tufts.edu

Kathleen Fisher
Kathleen.Fisher@tufts.edu

Research Motivation & Goals

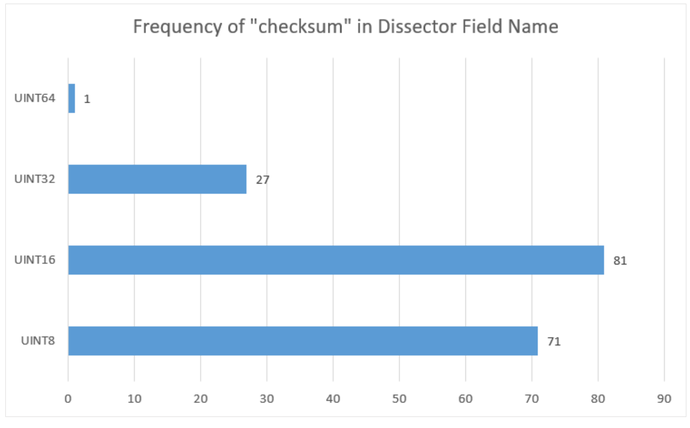
When cybersecurity analysts encounter a new or unknown network protocol it can require substantial human effort to reverse engineer the specification from a network capture. While there has been work on automating portions of this task, we seek to develop an automatic reverse engineering approach guided by common design patterns in extant network protocols. As part of this effort we seek to create a taxonomy of network protocols and the design patterns they exhibit. Our goal for this taxonomy is to create abstract categories of protocols serving as a roadmap for our reverse engineering efforts.

Taxonomy Approach: Wireshark Dissectors

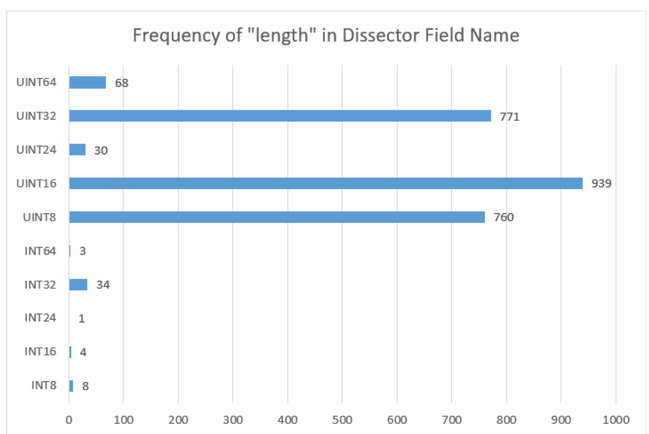


We use protocol glossary data extracted from the Wireshark network capture and dissection tool as the basis for our taxonomy construction. Wireshark is widely used opensource packet inspection tool used to capture and explore network traffic. Wireshark includes over 800 protocol dissectors which parse network messages into the fields according to the protocol specification. This makes it a useful source of protocol specification data across the protocol ecosystem. Wireshark allows dissector field data to be exported as a data glossary containing information about over 180,000 unique protocol fields. We parse this glossary format and load it into a SQL database for exploration and statistical analysis.

Frequency of Field Data Types Across Protocols



Different data types are used across protocols for the semantic type "checksum." The most frequently used data type is UINT16, which appears 45% of the time. The frequency of UINT32 and UINT64 is significantly lower than UINT16 and UINT8. There are no instances of UINT24 "checksum" or any INT "checksum".



Different data types are used across protocols for the semantic type "length." The most frequently used data type is UINT16, which appears 35.87% of the time. The frequency of UINT24 and UINT64 is significantly lower than the other three UINT types. The frequency of INT type is minimal compared to that of the UINT type.

Correlation between Checksum and Length Data Types within Wireshark Protocol Dissectors

	"checksum"			
"length"	UINT8	UINT16	UINT32	Total
UINT8	128	95	69	292
UINT16	49	112	40	201
UINT32	19	28	24	71
Total	196	235	133	564

An area of interest for our research are the relationships between field types within a protocol dissector specification. For protocol specifications that have fields with length and checksum in their names, we compare the data types used for each field respectively. In some cases, a dissector has multiple fields which match our criteria. We note that pairs of length and checksum fields occur most frequently for datatypes UINT8 and UINT16. We hypothesize that this may be a reflection of machine register size constraints.

Going From Names to Semantic Categories

Dissector field name	Dissector field description
data_checksum	Data Checksum
length_checksum	Length Checksum
prev_packet_checksum	Checksum of prev. packet
cryptoCheckSum	cryptoCheckSum
message_md5_checksum	Message MD5 Checksum
lm_cksum	Lm Cksum
ctl_cksum	ITDM Control Message Checksum

One challenge is assigning meaningful semantic categories to protocol fields. Wireshark dissectors are written by various members of the open-source community. As a result, semantically meaningful information is written in different ways by different people. An example is the "checksum" semantic field type. Authors of dissectors label field names and the corresponding human readable descriptions with many variants of the word "checksum" as shown in the table above. Our future work will include developing methods to infer and decide membership on such semantic categories.