Network Protocol Reverse Engineering

Literature Survey for “Eavesdropping on the Machines”

Presented 2016-08-05 at DEF CON 24 by Tim Estell and Katea Murray

Prior DEF CON Talks

- DC22 - Jesus Molina presented “Learn how to control every room at a luxury hotel remotely: the dangers of insecure home automation deployment”. In-depth look at a specific protocol, but no repeatable process for NPRE.
- DC 22 – Geoff McDonal presented “Meddle: Framework for Piggy-back Fuzzing and Tool Development”, arguing “Why bother spending time understanding the protocol just to try break it?”. His fuzzing tool sidesteps NPRE?
- DC 22 – Dustin Hoffman and Thomas Kinsey presented “What the Watchers See: Eavesdropping on Municipal Mesh Cameras for Giggles (or Pure Evil)” where they decoded an undocumented protocol (i.e., NPRE). But they didn’t focus on a repeatable process for you to reverse your own protocol.
- DC 23 – Peter Shipely and Ryan Gooler presented “Insteon' False Security And Deceptive Documentation” where they asserted the published protocol documentation from Insteon is incorrect and deceptive. But no generalized process for NPRE.

Summary of Research

Published research for Network Protocol Reverse Engineering (NPRE) has addressed many challenges. The Protocol Informatics Project (PI Project) [Beddoe 2004] accomplished PRE using network traces and two string alignment algorithms, Needleman-Wunsch and Smith-Waterman. A Semi-automated approach [Gopalratnam 2006] uses packets from the protocol of interest and at least one packet with the fields labeled. Gaussian models are used to cluster the field and provide information to the user about field values. They note their algorithm does not scale well with the number of fields being analyzed or the size of the messages.

Some researchers state that the limitation of network traces is a lack of protocol semantics as network traces only contain syntactic information and cannot provide the full protocol grammar [Caballero 2007]. These researchers rely on dynamic binary analysis and data tainting ([Caballero 2007] [Lin 2008] and [Caballero 2009]) or a combination of network traces and access to the binary [Cui 2008].

However, it has been demonstrated [Trifilo 2009] that discovery of the binary features in a protocol and a state machine builder can determine the states and proper transitions from network packet captures alone.

Finally, the research has been reduced to practice for TCP/IP networks through Netzob [Netzob], an open source tool with: vocabulary inference from network traces; semi-autonomous grammar inference; and dynamic analysis through protocol simulation.

Several researchers ([Lin 2008] [Wondracek 2008] and [Trifilo 2009]) implement incremental clustering using tree structures. However each of them requires data normalization or other grooming techniques which are not applicable to all environments. Researchers ([Wondracek 2008] and [Caballero 2009]) have also augmented protocol specifications with semantic information by adding specific running statistical information and a confidence metric.

The approaches do not address resource constrained environments. One potential approach uses Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [Zhang 1996] as a hierarchical clustering method because it will: incrementally and dynamically cluster; execute within given memory and time constraints; classify on a single scan of the data; and handle data with errors or noise. This would require further optimization to prioritize high-level completeness over accuracy as a method of initially capturing all of the message types, then reset the in-memory data structure to prioritize accuracy of individual message types.

Research should begin with a survey of message protocol features to develop a Domain Specific Language (DSL) suitable for compact representation of features. The DSL should extend existing work done by the Open Grid Forum's Data Format Description Language (DFDL) [OGF-DFDL] and by other Data Description Languages (DDL) such as XML Schema [XML Schema]. These sources use human readable text formatted in an XML manner, making them unsuitable for machine processing or automated reasoning. The DSL may extend Abstract Syntax Notation One (ASN.1) encoding rules, which provides for more efficient serialization than the text-based XML format.

Bibliography

[**Beddoe 2004**] Beddoe, M. (2004 August). The Protocol Informatics Project (PI Project). http://www.4tphi.net/~awalters/PI/PI.html.

[**Gopalratnam 2006**] Gopalratnam, K., Basu, S., Dunagan, J., & Wang, H. (2006, June). Automatically extracting fields from unknown network protocols. In First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML06).

[**Cui 2007**] Cui, W., Kennan, J., & Wnag, H. J. (2007, August). Discoverer: Automatic protocol reverse engineering from network traces. In Proceeding of 16th USENIX Security Symposium on USENIX Security Symposium (pp. 1-14).

[**Cui 2008**] Cui, W., Peinado, M., Chen, K., Wang, H. J., & Irun-Briz, L. (2008). Tupni: automatic reverse engineering of inputs formats. In Proceedings of the 15th ACM conference on Computer and Communications Security (CCS '08). ACM.

[**Lin 2008**] Lin, Z., Jian, X., Xu, D., & Zhang, X. (2008). Automatic Protocol Format Reverse Engineering through Context-Aware Monitored Execution. In 15th Symposium on Network and Distributed System Security (NDSS), 2008. Internet Society.

[**Wondracek 2008**] Wondracek, G., Comparetti, P. M., Kruegel, C., & Kirda, E. (2008). Automatic Network Protocol Analysis. In 15th Symposium on Network and Distributed System Security (NDSS), 2008. Internet Society.

[**Caballero 2007**] Caballero, J., Yin, H., Liang, Z., & Song, D. (2007, October). Polyglot: Automatic extraction of protocol message format using dynamic binary analysis. In Proceedings of the 14th ACM Conference on Computer and Communications Security (pp. 317-329). ACM.

[**Caballero 2009**] Caballero, J., Poosankam, P., Kreibich, C., & Song, D. (2009). Dispatcher: enabling active botnet infiltration using automatic protocol reverse-engineering. In Proceeding of the 16th ACM Conference on Computer and Communications Security (CCS '09). ACM.

[**Trifilo 2009**] Trifilo, A., Burschka, S., Biersack, E. (2009 July). Traffic to protocol reverse engineering. In Proceedings of the Computational Intelligence for Security and Defense Applications, 2009 (CISDA 2009). IEEE

[**Zhang 1996**] Zhang, T., Ramakrishnan R., & Livny, M. (1996, June). "BIRCH: An Efficient Data Clustering Method for Very Large Databases, 1996. In ACM SIGMOD Record (Vol. 25, No. 2, pp 103-114). ACM.

[**Netzob**] www.netzob.org. An open source tool for reverse engineering, traffic generation and fuzzing of communication protocols.

[**OGF-DFDL**] http://www.ogf.org/dfdl/. Data Format Description Language (DFDL) is a language for describing text and binary data formats. A DFDL description allows any text or binary data to be read from its native format and to be presented as an instance of an information set. DFDL also allows data to be taken from an instance of an information set and written out to its native format. DFDL achieves this by leveraging W3C XML Schema Definition Language (XSDL) 1.0. It is therefore very easy to use DFDL to convert text and binary data to a corresponding XML document. (Text taken from their web site in January 2014)

[**XML Schema**] http://www.w3.org/standards/xml/schema. An XML Schema is a language for expressing constraints about XML documents. There are several different schema languages in widespread use, but the main ones are Document Type Definitions (DTDs), Relax-NG, Schematron and W3C XSD (XML Schema Definitions). From this page you can find out more about DTDs and W3C XSD, since those are the primary schema languages defined at W3C. (Text taken from their web site in January 2014)