Advanced Science and Technology Letters
Vol.29 (ASEA 2013), pp.252-257
http://dx.doi.org/10.14257/astl.2013.29.53

isDSL– A Domain Specific Language for Intrusion Signature Declaration

Kanin Chotvorrarak and Yachai Limpiyakorn,

Department of Computer Engineering, Chulalongkorn University,
Bangkok 10330, Thailand
Kanin.C@student.chula.ac.th, Yachai.L@chula.ac.th

Abstract. This paper presents a signature-based network intrusion detection system. The intrusions are detected using their signatures defined as a set of rules contained in DSL scripts. A domain specific language is created, called isDSL, of which the syntax aligns with the TCP/ IP stack and the encoding of GA chromosomes. Genetic algorithm is the technique used for searching malicious states on network traffics. The attack is defined as a combination of properties and values that could be across the packets, matches with the conditions defined in a particular rule described in DSL scripts. The presented approach is promising for several reasons. A domain specific language is a declarative approach for describing intrusion signatures that could spread across network packets. Additionally, the rules are expressive that could tune out false positives. Moreover, the use of genetic algorithm would reduce the number of packet combinations being investigated for signs of the intrusions.

Keywords: domain specific language, intrusion detection system, genetic algorithm, network security.

1 Introduction

Intrusion detection is a type of security management system for computers and networks. In computer security, it is one of the important technologies for identifying possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). An intrusion detection system, IDS, is a device or software application that gathers and analyzes information from various areas within a computer or a network to assess the signs of intrusions defined as attempts to compromise the confidentiality, integrity and availability, or to bypass the security mechanisms of the network system. Various commercial products in the market include Snort, STAT, and Real Secure, etc.

Network intrusion detection has been an active field of research for a long time. In 1980, Anderson introduced the concept of intrusion detection, and defined a threat from the unauthorized access [1]. Denning published “An Intrusion Detection Model” in 1987, presenting intrusion detection methods which include profiles, anomalies and rules [2]. During 1990s, the Internet growth had led to the major concern of network security. To maintain the security of computer networks and the integrity of the user data, some aspects or conditions must be verified as an intrusion detection task. These

ISSN: 2287-1233 ASTL

conditions, specified in terms of properties of the network traffic, will form a specification of a desired or a malicious state of the network.

A Domain Specific Language, DSL, is a specification language dedicated to a particular problem domain. It is expressive and enabling an easy description of intrusion signatures that spread across several network packets. This paper thus presents a signature-based intrusion detection system using genetic algorithm for searching the desired intrusions against the rules defined in a DSL script written with the domain specific language created in this work.

2 Background

2.1 Intrusion Detection System (IDS)

Intrusion detection systems can be characterized based on Location and Detection methods. The former can be further classified into two types: 1) Network intrusion detection, and 2) Host intrusion detection. Whereas the latter can be majorly classified into two types: 1) signature-based detection, and 2) anomaly-based detection [3]. With signature-based detection, intrusions are detected using their signatures, i.e. particular properties of network packets used by the intrusion. On the other hand, in anomaly based detection, the system models the normal behavior of the network using statistical methods and/or data mining approaches. The anomaly detection will monitor network behaviors, and if it is considered anomalous according to the network model, then there is a great probability of an attack. The network intrusion detection system, NIDS, developed in this work is a type of signature-based method.

2.2 Domain Specific Language (DSL)

DSL is the high level language focusing only on must-have ability. It is used to define syntax and semantics for the abstraction of the problem domain, for the following purposes: efficiency, easy to understand, and reduce complexity of the language. The domain expert like security professional in network security domain can understand and maintain the DSL code without much programming skill [4]. Moreover, there are a lot of libraries to support the development of DSL in several programming languages, such as .Net parser generator so called “Irony” in C# language, and “ANTLR” in Java language.

Some intrusion detection systems, like Snort and Bro [5], provide custom languages to describe the intrusion signatures, but they are usually scripting languages, based mostly on pattern matching and regular expressions. Since these languages do not use a declarative approach, thus making them less expressive. In literature, the NeMODe system [6] is an example of network intrusion detection systems that provides a declarative and expressive domain specific language for describing intrusion signatures that could spread across network packets. Simply stating constraints over network packets in the script, the desired intrusions will be recognized by Constraint Programming paradigm used as the backend detection mechanism.

3 Design of isDSL

The design of the Domain Specific Language for intrusion signature declaration, isDSL, is on the basis of TCP/ IP protocol, integrated with the searching algorithm, GA [7]. The following subsections describe the influences of TCP/ IP and GA that affect the design of isDSL.

3.1 Transmission Control Protocol/ Internet Protocol (TCP/IP)

The software that manages the packets of information in a computer network is called Transmission Control Protocol/Internet Protocol, TCP/ IP. This software has become the universal standard for information exchange on the Internet. TCP creates the packets and reassembles them into the original message. IP handles packet addressing and ensures that they are transmitted across multiple networks and computers to the correct destination. A lot of vulnerabilities of TCP/ IP protocol have been reported, and network intruders use these flaws to make various network attacks.

The TCP/IP stack is a complete set of networking protocols. Each layer contains a set of properties as illustrated in Fig. 1. The syntax of isDSL is designed based on the layered structure of TCP/ IP stack. Only 3 layers are used in this work, excluding the Application layer. Fig. 2 illustrates the properties within the 3 layers of TCP/IP stack that are used as tokens in the syntax of isDSL. Example properties within each layer include: source and destination MAC address contained in Ethernet Frame header, source and destination IP address contained in IP datagram, flags of syn, ack, psh, fin, etc. contained in TCP packets.

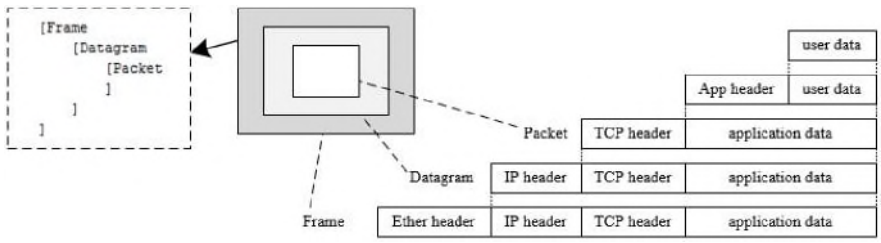

Fig. 1. Design of isDSL based on layered structure of TCP/IP stack.

```
<Frame_prop> ::= "srcMac" | "desMac"
<Datagram_prop> ::= "srcIP" | "desIP" | "TTL" | "length"
<TCP_prop> ::= "srcPort" | "desPort" | <TCP_flags> |  "winSize" | "seqNum" | "ackNum"
<TCP_flag> :: = "ns" | "crw" | "ece" | "urg" | "ack" | "psh" | "rst" | "syn" | "fin"
<UDP_prop> ::= "srcPort" | "desPort" | "dnsID"
```

Fig. 2. isDSL tokens aligned with properties within TCP/IP packets.

3.2 Chromosome Encoding

In this work, GA is used for generating several chromosomes of packet combination based on a single signature description. And with these populated chromosomes, the desired intrusions would be detected. To search for the intrusions matching with the signatures defined in isDSL scripts, the logical operators *equal* "=" and *not equal* "! =" are used for the comparisons in the rule conditions. The GA search is implemented as the back end engine that works with a set of rule structures, resulting from the parsing of isDSL scripts. Each GA chromosome is encoded with the indexing number of the packets contained in the buffer, which is the working space storing a number of packets being examined.

The process of GA search is illustrated in Fig. 3. Considering the rule of "Attack type1" with three arguments, a, b, c, it indicates that three packets will be simultaneously investigated. Therefore, the chromosome is encoded with the indexing number of three packets inside the buffer, i.e. packets with the index numbered 2, 1, 4. Example of initial packet combination as one of the initial population members is shown in Fig. 4. The hash table is used to provide the direct access to the properties and values residing the packets being indexed. As a result, the comparison whether the properties match with the conditions defined in the rule can be performed concurrently at one time. The score of the number of matching conditions will then be used for the evaluation of fitness function. The absolute matching signals a great probability of the intrusion. Next, the probabilistic selection will be applied for keeping some chromosomes into the next population generation. The rest will be thrown for generating new packet combination via the GA operators, cross over and mutation. It is expected that the use of Genetic algorithm would optimize the number of packet combinations to be compared for intrusion detection.

Some intrusion signatures declared as the rules contained in isDSL scripts are presented in Fig. 5, namely ARP (Address Resolution Protocol) spoofing and Syn flooding.

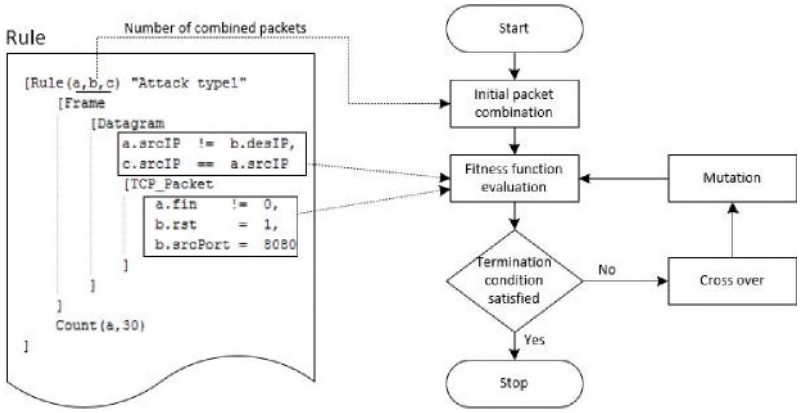

Fig. 3. GA process of generating packet combinations for comparison against rules.

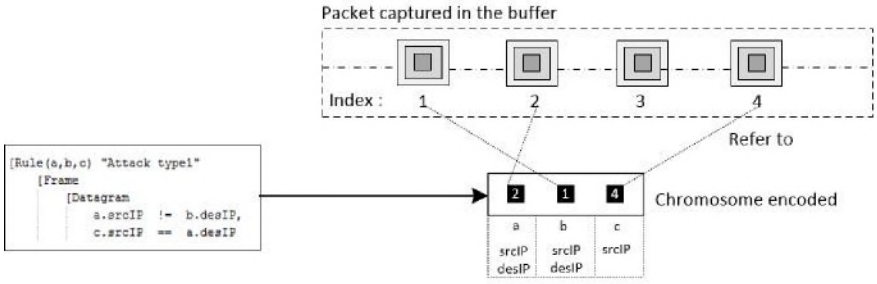

Fig. 4. Mapping between rule and chromosome encoding.

```
[Rule (a,b,c) "ARP spoofing"
    [Frame
        a.desMac = FF:FF:FF:FF:FF:FF,
        b.srcMac != c.SrcMac,
        [Datagram
            a.desIP = b.srcIP, a.desIP = c.srcIP,
            b.srcIP != c.srcIP   ]]]
```

```
[Rule (a) "Syn flooding"
    [TCP_Packet
        a.syn = 1
    ]
    Count(a,100)
]
```

Fig. 5. isDSL rules of ARP spoofing rule and Syn flooding.

4 Implementation

The prototype system (Fig. 6) consists of three main components: 1) NIDS engine, 2) isDSL parser, and 3) Traffic monitoring engine, as described in the following subsections.

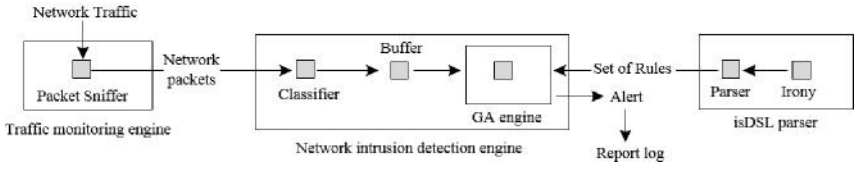

Fig. 6. Architecture of implemented prototype.

4.1 Network Intrusion Detection Engine

The network intrusion detection engine is considered as the backend module responsible for detecting the intrusions, and if found, it will alert and log the incident. Classifier is the component in charge of filtering out those which are not TCP/IP packets. Only TCP/IP packets will then be stored in the buffer to generate the chromosome population for further matching with the rules. Genetic algorithm is used for generating various combinations of packets that would form the malicious chromosome.

4.2 isDSL Parser

The Irony [8] .Net platform parser generator is used to create the syntax and grammars of isDSL in this research. Irony is an open source software library with a complete set of libraries and tool for language implementation.

4.3 Traffic Monitoring Engine

The traffic monitoring engine is responsible for sniffing the network traffic in order to capture the packets and output them to the network intrusion detection engine for further processing. The Pcap.Net library is used for implementing this module.

5 Conclusion

Several protocols manage and control the transmission of information across the networks, including TCP/IP. Despite of its popularity, there exist a number of serious security flaws reported on the design of TCP/IP protocol suite. The security attacks, such as IP spoofing, Syn flooding, and others, can be described in the state and condition for properties among packets. This research has developed the domain specific language called isDSL for describing the rules containing the relation of properties in single or across packets. Network security professionals can easily and expressively define a set of rules describing signs of intrusions. Genetic algorithm is used to reduce the number of packet combinations generated for matching with the rules. Future work includes the evaluation of the proposed approach measured by the detection rate defined as the number of intrusion instances detected by the system (True Positive) divided by the total number of intrusion instances present in the test set.

References

1. Peddisetty, N.R.: State-of-the-art intrusion detection: Technology, challenges, and evaluation, Institutionen för systemteknik (2005)
2. Patel, P., Langin, C, Yu, F., Rahimi, S.: Network Intrusion Detection Types and Computation. International Journal of Computer Science and Information Security, Vol. 10(1) (2012)
3. Jaiganesh,V., Mangayarkarasi, S., Sumathi, P.: Intrusion Detection Systems: A Survey and Analysis of Classification Techniques. International Journal of Advanced Research in Computer and Communication Engineering Vol. 2 (2013)
4. Fowler, M.: Domain-Specific Languages. Addison-Wesley Professional (2010)
5. Paxson, V.: Bro: A System for Detecting Network Intruders in Real-Time. Proceedings of the 7th USENIX Security Symposium, San Antonio, TX (1998)
6. Salgueiro, P.D., Diaz, D., Brito, I., Abreu, S.: Using Constraints for Intrusion Detection: The NeMODe System. PADL, 115--129 (2011)
7. Alander, J.T.: An indexed bibliography of genetic algorithms: Years1957-1993. Art of CAD Ltd. Vaasa, Finland (1994)
8. Irony - .NET Language Implementation Kit, http://irony.codeplex.com

