Journal of Computer Science 6 (7): 716-722, 2010
ISSN 1549-3636

Amelioration of Attack Classifications for Evaluating and Testing Intrusion Detection System

Mohammed Saber, Toumi Bouchentouf, Abdelhamid Benazzi and Mostafa Azizi
Laboratory Mathematics Applied, Treatment of the Signal and Computer Science,
Department of Computer Science, National School of Sciences Applied Oujda,
Mohammed First University,
ENSAO BP 669 Complexes Universities Al Qods Oujda 60000, Morocco

Abstract: Problem statement: The problem of the computer attack system has recently been much studied to improve the evaluation process of the Intrusion Detection Systems (IDS). **Approach:** This study aimed at presenting the principal attacks classifications; especially, the study of classification towards the evaluation for which we suggested some improvements that may allow the generation of a test cases selection about attacks by using the classification tree method. **Results:** The results proposed evaluators to select relevant attack test cases by using the Classification Tree Method (CTM). **Conclusion:** By using the Classification Tree Method (CTM), to the new classification as it was obtained and by applying the CTE tool, we were able to generate some significant and reduced cases test compared to the classification toward the assessment which was studied by Gadelrab.

Key words: IDS, evaluation, Classification Tree Method (CTM) attack classification

INTRODUCTION

The number and complexity of computer attacks against information systems has increased during the recent years. This has caused several problems to the IDS evaluators. So, for a given IDS, how it would behave against of intrusion or attack attempts.

Besides, there is another problem which occurs during an IDS assessment. It is that of attack classification (Kumar and Spafford, 1995) because it is hard to examine exhaustively all attacks. A possible solution of this problem is to use of the class equivalence technique which is used for a software test (Glenford, 1979) in order to reduce the number of test cases. Yet, we notice that some cases, which belong to the same class, stimulate the same software parts in the same conditions and this should produce some equivalent results. This approach has been used to set up the test cases of different attack classes for both testing and evaluating IDS.

In this study, we will adopt Webster's (Merriam-Webster, 2010) suggestion which considers "taxonomy" and "classification" as two synonyms even if the classification is defined as the systematic arrangement inside the groups or the categories according to some established criteria while the taxonomy is defined as the study of the general principles of the scientific classification.

Since the attacks exploit the vulnerabilities of a computer system, several attempts have been carried out to classify the vulnerabilities during the last years. This has led to the building of vulnerability databases such as the Common Vulnerability Exposition (CVE, 2010) of the MITRE or the Open Source Vulnerability Database (OSVDB, 2010).

Several research works have tried to classify the attacks; for instance, (Neumann and Parker, 1989; Kumar and Spafford, 1995; Lindqvist and Jonsson, 1997; Bishop, 1999; Kendall, 1999; Lough *et al*., 2000; Alessandri, 2000; Kevin *et al*., 2004; Hansmann, 2005). However, classifications techniques do not share the same objectives; no full and largely admitted technique of classification has been set up. Besides, a remarkable work has been done in (Gadelrab *et al*., 2007) it is about a classification which takes into account the different suggestions of past classifications.

In this study, we study this last technique of classification and we suggest improving it by reducing the number of generated tests per class. We used here the Classification Tree Method (CTM) (CTE, 2010; Grochtmann and Wegener, 1995) to get an easy and

Corresponding Author: Mohammed Saber, Laboratory Mathematics Applied, Treatment of the Signal and Computer Science, Department of Computer Science, National School of Sciences Applied Oujda, Mohammed First University, ENSAO BP 669 Complexes Universities Al Qods Oujda 60000, Morocco

semi-automatic choice of attack test cases by using the CTE tool which uses the CTM.

This study is composed as follows: Firstly, we give a broad view on the different existing classifications and discuss in details the classification. Then secondly, we present our improvement of the Gadelrab classification while in the third part; we shown on results and set up afterwards follow up discussion between the suggested improvements and the Gadelrab classification.

MATERIALS AND METHODS

Analysis of existing attack classifications:

Presentations of the existing classifications: Though the problem of attack classification has attracted many researchers in the security field, they did not share the same objective. In this study, we will firstly mention the existing taxonomies by describing briefly their principles and objectives while detailed description taxonomies were dealt with in (Lough *et al*., 2000).

Bishop (1999) taxonomy is only about the vulnerabilities and not about the attacks. It takes into consideration the followings: Gap-nature, phase of vulnerability introduction (e.g., during the conception or implementation stage), exploitation area (i.e., how to exploit), effect area (what is affected), minimum number of necessary elements of the exploitation of this vulnerability and its identification source (the broadcasting site or list where the vulnerability was published).

Kumar and Spafford (1995) has suggested an attack classification according to four outline attributes or of the attack signature: Existence, sequence, space and duration.

There is another interesting work, which is that of Lindqvist and Jonsson (1997) who have enlarged Neumann and Parker (1989) taxonomy. The latter take into account just one dimension which is the technique while Lindqvist and Jonsson (1997) add the result as an extra-dimension. This classification is considered as one of the carried out experiences by some internal users (students of a computer science class) in order to improve the IDS detection abilities which use the filter by form-identification (pattern matching).

Weber, however, has presented a taxonomy based on three dimensions which are the required privileged level so as to lead the assault, the attacker's used means (e.g., a software bug exploitation) as well as the wished effect (e.g., ill-service).

The DARPA taxonomy (Lippmann *et al*., 2000a) and (Lippmann *et al*., 2000b) is in fact a reduced version of Weber's it consider but the attack effect as the only dimension. The attacks are divided into five parts: "distant toward local" (or R2L for remote to local), "user toward super-user" (or 2UR for User to Root), "Sounder" (scan) and "corrupt service" (Kendall, 1999) and (Lippmann *et al*., 2000a). We can see that this classification considers some different abstract levels, which raise some problems; mainly, the resulting class mutual existence.

Hansmann (2005) taxonomy takes into consideration four dimensions that are related to the attacks: The vector or the type (i.e., the means used by the attacker to come to its ends, such as viruses, bugs, ill-service, the target (e.g., the operating system, network protocol), the attack effects as well as the exploited vulnerability.

Moreover, there is another significant work (Kevin *et al*., 2004) which comes with a taxonomy that has a defensive view. The purpose was to give some information to help administrators so as to defend their systems. The attacks were, then, classified in terms of their manifestations as they were seen by some Host-based Intrusion System Detection (HIDS). The four dimensions of this taxonomy are:

- External signs: They are about call systems which appear after the attack execution, but which never turn up in normal operations
- Minimal sequence: It is the smallest sequence, but it never appears in normal operations
- Sleeping sequence: It is a sequence which partially corresponds to a sub-sequence of normal operations
- Normal sequence: It is a sequence in the attack which cannot be distinguished from non-intrusive activities

The taxonomy of (Alessandri, 2000) was elaborated so as to analyze some IDS patterns. Instead of directly categorizing the attacks, it wholly classifies all the activities that might be pertinent to the IDS. An analytical assessment was, then, set up to determine the IDS detection capacities towards a particular class of attacks. The corresponding pattern to this classification makes the difference between dynamic characteristics of an IDS observable activity and static ones. The static activities are divided into the characteristics that are related to both interface-objects and those which are bound to the attack's affected or corrupt objects. The dynamic characteristics are developed according to three criteria:

- Communication characteristics (e.g., mono directional, bidirectional)

- Invocation method (e.g., creation, deleting reading)
- Other additional attributes which are qualified as minors (e.g., the attack may come from several origins or it may contain some repetitive events). As for the attack itself, it is described according to five criteria:
  - The interface object
  - The affected object
  - Communication
  - The invocation method
  - Other minor attributes

Therefore, this taxonomy involves twenty-five interface-objects, ten affected objects, three related characteristics to communication, five invocation methods as well as four additional minor attributes.

In the next part, we tried to analyze these different classifications between them and those which mainly have a pertinent value for both testing and evaluating IDS.

Analysis of existing classifications: In (Gadelrab *et al*., 2007) a debate was carried out on these different taxonomies and which we can summarize here by adding our own deductions and notices. The following facts came out: Each classification was developed for a certain goal; for instance, understanding the vulnerabilities so as to reinforce the corrective and defensive measures, apprehending the attack processes as well as the attacker's behavior.

The result is that the identified attributes in a certain classification are not always pertinent to another which has a different objective. The most important attributes are:

- Attack type: Virus, bugs, Trojan, corrupt service
- Attack detection technique: Statistic approach, filtering, motive identification
- Attack signature: Observed motive (pattern) or sequence of observed motives
- Tool used by the attacker: Toolkit, script, user's order
- Attack target: Exploitation system, network protocol application service
- Attack result: Illicit modification or information divulging, corrupt service
- Attack targeted access: Super-user access, normal-user access
- Attack pre-conditions: Existence of some particular versions of a certain software
- Attack exploited vulnerability: Memory disorder, bad choice of passwords, bad configuration
- The attack objective: Financial gain, terrorism, self-satisfaction
- Locating the attack origin: Internal, external
- Attack violated or targeted security side: Confidentiality, integrity, availability

With these attributes, we have all the attack attributes; but why do not these classifications give good results to the IDS test and evaluation? In other words, what are the weaknesses of these classifications.

Following the analysis of major existing works, we notice that most classifications mix the assault attributes and the attacker's ones. This type of approach often ignores or hides certain important characteristics of attacks, as they are seen by the IDS or the administrators system. The existing taxonomies are not really adapted to the IDS evaluation. The reasons may be summarized in the following points:

- Most classifications mix the attacker's attributes with those of the attack; so, the resulting attributes are less pertinent for the IDS test and evaluation
- The definition of certain attributes is a bit ambiguous or even incoherent and thus, hard to determine an efficient classification which may produce several test-cases allowing the facilitation of IDS assessment
- The number of the resulting classes is sometimes higher, which makes the IDS test and evaluation too complex and less efficient
- These classifications are, unfortunately, not followed by any layout of test-cases selection and generation

After having presented and analyzed the existing classifications, we intend to discuss, in the next part of this study, the classification suggested by Gadelrab *et al*. (2007).

Classification oriented towards the assessment:
Presentation of Gadelrab's classification: The purpose of this classification, which is based on the formerly presented attributes in the last part of this study, is to eliminate the ambiguous attributes or those which are not pertinent for both testing and evaluating IDS.

This classification lies on five dimensions, as indicated in Fig. 1. These dimensions are selected in terms of covering the sources, attack targets and attacks manifestations, enough and necessary information for the IDS test. These dimensions are:

- Source: Indicates the point where the attack was launched

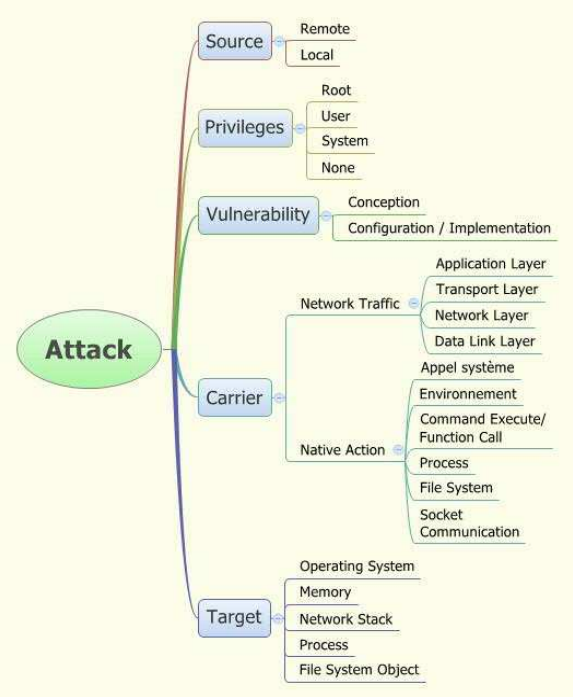

Fig. 1: Classification towards the evaluation

- Resulted privilege: Distinguishes four privilege classes aimed by the attacker. The "Root" class and the "User" respectively mean that the attacker has successfully got the "root/administrator" or "user" access. The "Class System" which allows the process execution with the "privilege Systems". The "No Class" covers the attacks which need no privilege access to the system like the scans
- Vulnerability: From the evaluator's point of view, it is interesting to target the most pertinent test system, well-prepare the test platform, express the relation between the attacks and the exploited vulnerabilities. This would, particularly, not only help to choose (during the test period) the assaults which might exploit these vulnerabilities (and which are ranked and available in some vulnerability standardized data bases), but also to identify the system gaps for any eventual correction
- Porter or means by which the attack was carried out: It may come from the network traffic or from a action that was directly executed on the target-machine and which does not appear on the network interface
- Target: It may be the memory, the operating system, the network pile, the file system or a process

By implementing the (CTM) method, we generate 1920 cases test for the classification of (Gadelrab *et al*., 2007) whereas Alessandri's classification (Alessandri, 2004) generates 3500 test cases.

Classification analysis towards the evaluation: While the other classifications take into account the assault and the attacker, Gadelrab *et al*. (2007) classification takes into consideration just the attack itself. By bringing a better clarity, this classification takes into account not only some observable characteristics of the attack (Alessandri, 2004; Kevin *et al*., 2004), but also some operational aspects which remain primordial for the IDS test and evaluation.

We can notice that in the classification towards the evaluation, there is a redundancy of some attributes like those between the operating system and the memory, between the operating system and the file system and between the operating system and the process. Any attack cannot reach the memory or the process; for instance, without going through the operating system.

An IDS which is based; for example, on the scenario detection method (James, 1980; Cuppens and Ortalo, 2000; Steven *et al*., 2002; Michel and Me, 2001) should have a precise network pile service or protocol, but this classification has brought no accuracy concerning the assaulted service or protocols.

To find a solution to the deduced defaults, we suggest in the next part of this study, an amelioration of the oriented taxonomy towards Gadelrab's evaluation.

Improvement of the evaluation classification:

Presentation of amelioration's classification: The classification which we suggest is based on the same principles as that of (Gadelrab *et al*., 2007). The resulting classes as well as the classification process must respect, as much as possible, the satisfaction characteristics studied in (Lindqvist and Jonsson, 1997; Alessandri, 2000; Hansmann, 2005) which are:

- Fullness (i.e., exhaustiveness): A categorization outline should take into account all the possible attacks (either known or unknown)
- Extensibility: When some new attacks appear the categorization outline should allow classifying them their classification
- Criteria clarity: The classification outline and rules should be well-established in a way that an attack can be classified by taking just one class from every dimension

- Repetitiveness: The reimplementation of the classification process must always produce the same results; in other words, if we repeat the followed stages for a certain attack classification, we must always put it in the same category
- Conformity with the standards and resulting terminologies; mainly, with vulnerability data bases and dictionaries like CVE (2010) and OSVDB (2010) which are nowadays widely used
- Mutual Exclusion: be certain that an attack does not come from two different categories. Therefore, a dimension will have only but mutually exclusive classes.

To improve the classification process of (Gadelrab *et al*., 2007) and avoiding the previously mentioned drawbacks, we suggest the use of the following attributes as presented in Fig. 2:

- Source: Indicates the point where the attack was launched. It has got two classes: local and distant
- Resulted privilege: Distinguishes four privilege classes aimed by the attacker. The "Root" class and the "User" respectively mean that the attacker has successfully got the "root/administrator" or "user" access. The "Class System" which allows the process execution with the "privilege Systems". The "No Class" covers the attacks which need no privilege access to the system like the identification attacks; for instance, the scan type
- Vulnerability: Expresses the relation between the attacks and the exploited vulnerabilities; this would particularly help to choose the attacks which might exploit these vulnerabilities. In this case, we notice two classes: Either the "Conception" class which gathers all the weaknesses during the conception stage, or the "configuration/implementation" class which represent all the errors during the configuration period of an application system or implementation, network service.
- Used Means of a launched attack: It may be the network traffic i.e., all the traffic which is generated by the network pile of different layers TCP/IP, or of a directly executed action on the target machinelike; for instance, an order execution, or a script/ program or a socket execution
- Target: It may be either the operating system (memory, the file system or a process) or the network (application, network server)

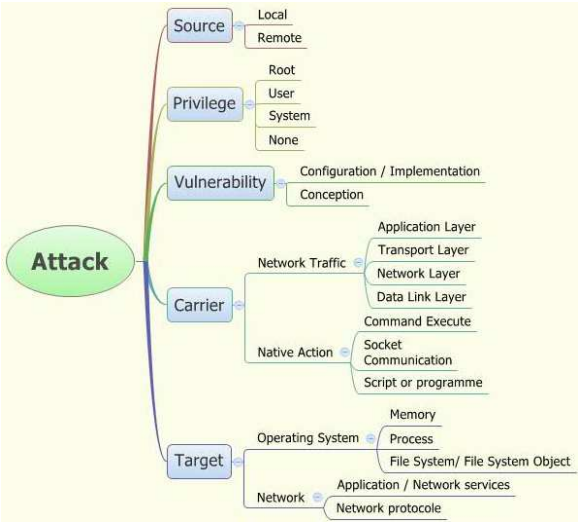

Fig. 2: Amelioration of the evaluation classification

In the following results and discussion, we use this classification to present a simple approach of test-case selection. For this, we suggest using a method which is based on the classification tree (CTM for Classification Tree Method); then, we will present a simple example of the implementation of this approach.

RESULTS AND DISCUSSION

Result obtained: In this part, we propose evaluators to select relevant attack test cases by using the Classification Tree Method (CTM), which was developed by Grochtmann and Wegener (1995). It was applied in testing systems in various domains and we apply it to the security-testing domain. But first, let us describe the method itself.

By means of the CTM, the input domain of a test object is regarded under various aspects or dimensions that are assessed according to their relevance for the test. For each aspect, disjoint and complete classifications are formed. The stepwise partition of the input domain by means of classifications is represented graphically in the form of a tree.

To construct test cases, a grid is drawn below the tree. The columns of the grid result from vertical lines that correspond to the leaves of the classification tree. A tester can construct a test case by selecting a single leaf class of each higher-level branch of the classification. Each row of the grid indicates a distinct category of test cases. Because not all test cases are legal or valid, the tester should eliminate the invalid ones. This can be done by the definition of constraints or generation rules in the Classification Tree Editor (CTE) tool.

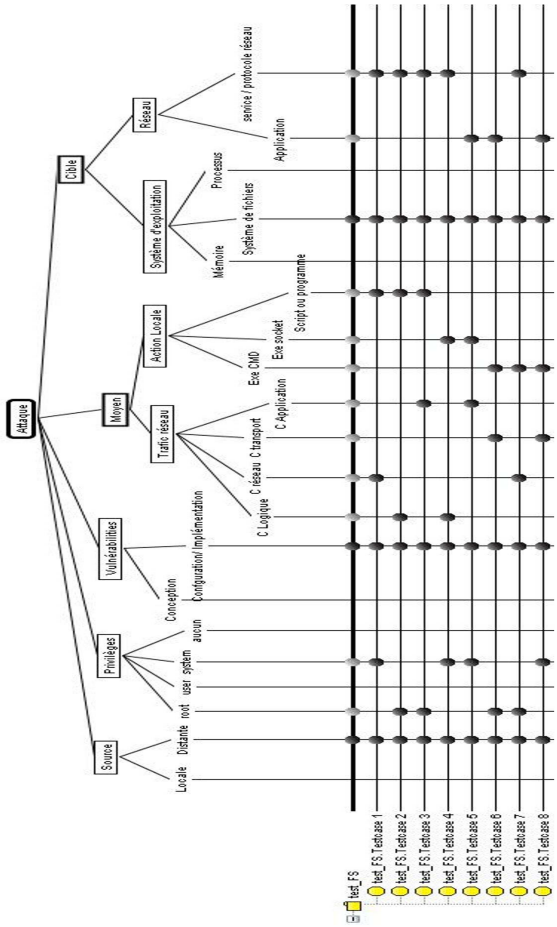

Fig. 3: Example of classification tree and test cases generated by CTE

More precisely, the CTE offers a simple and powerful formalism for the constraints expression by combining some rules which include some sub-ones. Between brackets (under a predicting form), some inter-propositional connectors such as and (*), or (+), no (NOT).

For example, the following rule: (Remote * (root + system) * Vulnerability configuration * Traffic network * File system) will result in 8 test case categories, which represent remote attacks that provide root or system access by exploiting configuration vulnerabilities and that could be observed in network traffic, targeting the system file (Fig. 3), whereas by applying classification the evaluation of Gadelrab one obtains 16.

CONCLUSION

After having studied the main attack classification toward Gadelrab's evaluation by specifying their dimensions and their attributes, we tried to analyze in this study the classification toward Gadelrab's evaluation by suggesting some amelioration of the target-dimension as well as its attributes.

By using the Classification Tree Method (CTM), to the new classification as it was obtained and by applying the CTE tool, we were able to generate some significant and reduced cases test compared to the classification toward the assessment which was studied by Gadelrab *et al*. (2007).

An interesting point would be to see to what extent the discoveries might be represented by the amelioration toward the classification, like for example, the implementation on the metasploits.

REFERENCES

Alessandri, D., 2000. Using rule-based activity descriptions to evaluate intrusion detection systems. Lecturer Notes Comput. Sci., 1907: 183-196.

Alessandri, D., 2004. Attack-class-based analysis of intrusion detection systems. Ph.D. Thesis, School of Computing Science, University of Newcastle upon Tyne, Newcastle upon Tyne, UK.

Bishop, M., 1999. Vulnerabilities analysis. Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID'99), West Lafayette, Indiana, USA., pp: 1-14.

CTE, 2010. Systematic-testing. Classification Tree Editor. http://www.systematic-testing.com

CVE, 2010. Mitre's Common Vulnerability and Exposure. http://cve.mitre.org/

Cuppens, F. and R. Ortalo, 2000. LAMBDA: A language to model a database for detection of attacks. Lecturer Notes Comput. Sci., 1907: 197-216.

Gadelrab, M.S., A.A. El Kalam and Y. Deswarte, 2007. Defining categories to select representative attack test-cases. Proceedings of the 2007 ACM workshop on Quality of protection, (QP'07), Alexandria, Virginia, USA., pp: 40-42.

Glenford, J.M., 1979. The Art of Software Testing. 1st Edn., John Wiley and Sons, ISBN: 10: 0471043281, pp: 192.

Grochtmann, M. and J. Wegener, 1995. Test case design using classification trees and the classification-tree editor CTE. Proceedings of the 8th International Software Quality Week, (QW'95), San Francisco, USA., pp: 1-11.

Hansmann, S., 2005. A Taxonomy of network and computer attacks. Comput. Secur., 24: 31-43.

James, P.A., 1980. Computer security threat monitoring and surveillance. Technical Report, James P. Anderson Company, Fort Washington, Pennsylvania. http://csrc.nist.gov/publications/history/ande80.pdf

Kendall, K., 1999. A database of computer attacks for the evaluation of intrusion detection systems. Proceedings DARPA Information Survivability Conference and Exposition (DISCEX), MIT Press, pp: 12-26. http://dspace.mit.edu/handle/1721.1/9459?show=full

Kevin, S.K., R.A. Maxion and K.M.C. Tan, 2004. A defense-centric taxonomy based on attack manifestations. Proceedings of the 2004 International Conference on Dependable Systems and Networks, June 28-July 1, IEEE Computer Society, Washington DC., USA., pp: 102-111. http://portal2.acm.org/citation.cfm?id=1009726&dl=ACM&coll=ACM

Kumar, S. and E. Spafford, 1995. A pattern matching model for misuse intrusion detection. Comput. Secur., 14: 28-28. http://www.ingentaconnect.com/content/els/01674048/1995/00000014/00000001/art96999

Lindqvist, U. and E. Jonsson, 1997. How to systematically classify computer security intrusions. Proceedings of IEEE Symposium on Security and Privacy, May 4-7, IEEE Computer Society, Washington DC., USA., pp: 154-163. http://portal.acm.org/citation.cfm?id=882493.884387

Lippmann, R., D. Fried, I. Graf, J. Haines and K. Kendall *et al*., 2000a. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2000), Los Alamitos, CA., USA., pp: 12-26.

Lippmann, R., J.W. Haines, D.J. Fried, J. Korba and K. Das, 2000b. Analysis and results of the 1999 DARPA off-line intrusion detection evaluation. Lecture Notes Comput. Sci., 1907: 162-182.

Lough, D.L., N.J. Davis, C.E. Nunnally, E.A. Brown and M.T. Jones *et al*., 2001. A taxonomy of computer attacks with applications to wireless networks. Ph.D. Dissertation, Department of Electrical and Computer Engineering. http://scholar.lib.vt.edu/theses/available/etd-04252001-234145/

Merriam-Webster, 2010. Dictionary and thesaurus online. http://www.merriam-webster.com

Michel, C. and L. Me, 2001. Adele: An attack description language for knowledge-based intrusion detection. Proceedings of the 16th International Conference on Information Security: Trusted Information: The New Decade Challenge, June 11-13, ACM Press, Paris, France, pp: 353-365. http://portal.acm.org/citation.cfm?id=510791

Neumann, P.G. and D.B. Parker, 1989. A summary of computer misuse techniques. Proceedings of the 12th National Computer Security Conference, Oct. 10-13, Baltimore, MD., USA., pp: 396-407. https://www.cerias.purdue.edu/apps/reports_and_papers/view/810

OSVDB, 2010. Open Source Vulnerability database. http://osvdb.org

Steven, T.E., G.V. Richard and A. Kemmerer, 2002. Statl: An attack language for state-based intrusion detection. J. Comput. Secur., 10: 71-103.