SECURITY ANALYSIS OF TCP/IP NETWORKS
An Approach to Automatic Analysis of Network Security Properties

Miroslav Sveda, Ondrej Rysavy, Petr Matousek, Jaroslav Rab and Rudolf Cejka
Faculty of Information Technology, Brno University of Technology, Bozetechova 2, Brno, Czech Republic

Keywords: Intranet Topology, Dynamic Routing, State-based Reachability, Security, Bounded Model Checking, SAT.

Abstract: This paper deals with an approach to security analysis of TCP/IP-based computer networks. The method developed stems from a formal model of network topology with changing link states, and deploys bounded model checking of network security properties supported by SAT-based decision procedure. Its implementation consists of a set of tools that provide automatic analysis of router configurations, network topologies, and states with respect to checked properties. While the paper aims at supporting a real practice, its form strives to be exact enough to explain the principles of the method in more detail.

1 INTRODUCTION

The paper focuses on the area of automatic analysis of a network that consists of L3 devices (hosts, routers, firewalls etc.) connected by links and, optionally, with firewall rules applied on them. Based on the network configuration and considering dynamic behavior of the network, we can ask questions like "Is this network protected against P2P connections?", "What packets can be delivered to the given host?", or "Is this WWW service accessible under every configuration of the network?"

Of course, those questions can be partially answered by scanning and testing tools (*ping*, *nmap*), or vulnerability assessment tools (*Nessus*). However, testing can analyze the network only in immediate state, which means in practice: for a fixed configuration. When the topology is changed, the response of the network can be different. In our work we explore how security and safety properties can be verified under every network configuration using model checking (Clarke et al. 1999). The model checking is a technique that explores all reachable states and verifies if the properties are satisfied over each possible path to those states. Model checking requires specification of a model and properties to be verified. In our case, the model of network consists of hosts, links, routing information and access control lists. After reviewing state of the art in section 2, the specification of network model is laid down in section 3. The next section deals with network security properties that are expressed in the form of modal logics formulas as constraints over states and execution paths. If a property is not satisfied, the model checker generates a counterexample that reveals the state of the network, which violates the property. If the property is proved, it means that the property is valid in every state of the system. Those items are discussed in sections 5 and 6.

This paper primarily focuses on the automatic analysis of network security properties of the network. The challenges addressed in the paper include: (a) automatic generation of the network model using routers configuration files, (b) creating templates for specification of network security properties and (c) a combination of tools that verify given properties over the model by model checking.

2 STATE OF THE ART

Research in the area of network security and vulnerability detection has been conducted since the beginning of the Internet. Many papers concentrate on detection of vulnerabilities of hosts and their protection against the network attack, see e,g, (Tidwell, et al., 2001), (Zakeri, et al., 2005), or (Shahriari and Jalili, 2005). Most work follows the similar scheme: (i) Network is modeled as an entity that includes hosts, connections, user privileges, OS

Sveda M., Rysavy O., Matousek P., Rab J. and Cejka R. (2010).
SECURITY ANALYSIS OF TCP/IP NETWORKS - An Approach to Automatic Analysis of Network Security Properties.
In *Proceedings of the International Conference on Data Communication Networking and Optical Communication Systems*, pages 5-11
DOI: 10.5220/0002838300050011

types, running services, and individual vulnerabilities of hosts; (ii) Host vulnerabilities are revealed by external automatic tools like *Nessus*, or by *OVAL scanner* (Ou, et al., 2005). Then, detected vulnerabilities are expressed in the language of pre-condition and post-condition assertions, or rules; (iii) An important step is to determine threat tucker goal, i.e. security violation (e.g. root access on the web server), and in this case the goal is often expressed by a predicate; (iv) Having these, vulnerability analysis follows: it includes an application of derivation rules based on the initial assumptions (i.e., network configuration) in order to prove a predicate (i.e., security violation) − if the predicate is true, then the deduction path corresponds to the possible attack scenario.

Despite the statement of authors in (Shahriari and Jalili, 2005) that "this model lets automatically verify and prove network safety and vulnerability against the attack," the method of logic deduction and proving requires good knowledge of logics and deductive systems, since the proof is constructive and it is made by human.

In (Ou, et al., 2005), an automatic deduction of network security executed in Prolog is introduced. The authors define reasoning rules that express semantics of different kinds of exploits. The rules are automatically extracted from OVAL scanner and CVE database (Mitre, 2008).

Another approach is an automatic generation of network protection in the form of firewall rules as shown in (Bartal, et al., 1999). The security policy is modeled using Model Definition Language as the first step. Then, the model of a network topology is translated into firewall-specific configurations. These configuration files are loaded into real devices (firewalls).

Ritchey and Ammann in (Ritchey and Ammann, 2000) explain how model checking can be used to analyze network vulnerabilities. They build a network security model of hosts, connections, attackers and exploits to be misused by the attacker. Security properties are described by temporal logics and verified using SMV model checker. However, their goal is different from ours. They verify if hosts on the stable network are vulnerable to attacks. In our case we concentrate on dynamically changing networks and reachability of their nodes.

Our approach is close to the work of G. Xie (Xie, et al., 2005), and J. Burns (Burns et al. 2001). Unlike these works we build a model that includes both static and dynamic behavior, i.e. firewall rules and routing information, see (Matousek, et al,, 2008). In this model, the verification of reachability properties can be made. In comparison to Ritchey's work (Ritchey and Ammann, 2000) we do not focus on hosts vulnerability and their resistance to attacks but on stability of services in dynamic networks. Main contributions of this paper consists of (i) the creation of a network transition system that models dynamic behavior of the network, (ii) the definition of security properties using modal logics, and (iii) the algorithm for verification of specified properties using bounded model checking and SAT-solver.

3 NETWORK MODEL

The aim of this section is to restate a formal model of a network topology that allows specifying a set of attributes for security analysis, which was originally defined formerly and published in (Matousek, et al,, 2008). For the rest of the paper we refer to the example of the network topology as given in Figure 1.

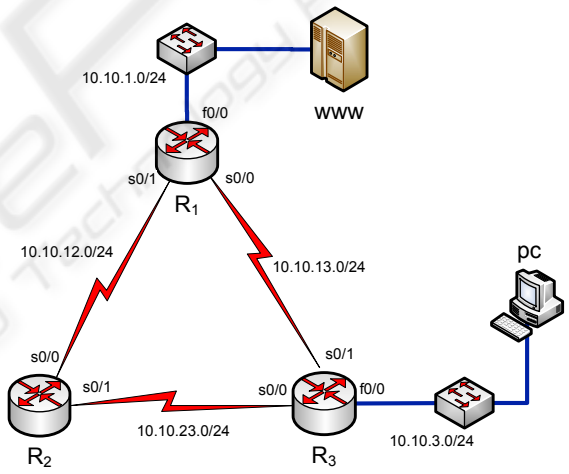

Figure 1: Example of network topology.

The abstract model of the network dealing with routing processes and packet filtering stems from a combination of techniques introduced in (Xie, et al., 2005) and (Christiansen and Fleury, 2004). The network, see Figure 2, is regarded as a directed graph where vertices are routing devices and edges are communication channels that form abstractions of communication links. Each communication link is modeled by a pair of unidirectional communication channels. In real networks there are other network device then routers. However, every end-point device, such as PC or Web server, can be represented using a router with only one interface,

and one outgoing filtering rule representing routing all traffic to default gateway.

Formally, network model, see Figure 2, is a tuple $N = \langle R_N, L_N, F_N \rangle$, where

- R_N is a finite set of network devices,
- $L_N \subseteq R_N \times R_N$ is a finite set of links between routers, such that for every physical link between R_1, R_2 there is a pair of channels $l_{12} = \langle R_1, R_2 \rangle$, $l_{21} = \langle R_2, R_1 \rangle$, and
- $F_N = \{f : P \rightarrow \{true, false\}\}$ is a finite set of filtering predicates and P is a set of all possible packets.

A filtering predicate $f(p) \in F_N$ is able to determine whether a packet p is allowed to be send. This function is defined so that it uniformly represents the interpretation of Access Control List (ACL) and routing table information adequate to link l. A simple example is a filter f (p)

$$f(p) = \neg(p.proto = Tcp \wedge p.dstPort = 80)$$

that turns down all web traffic, i.e. TCP packets with destination port 80. Note that dot notation is used to refer to attributes of the current packet. Both ACL and routing information of a network node can be translated to a filtering predicate.

From the ideas mentioned above the following conclusions can be summarized: (1) The model of a network includes specification of hosts, their configurations, network topology and description of vulnerabilities; (2) The list of host vulnerabilities and network threats can be downloaded from open databases, or specified manually; (3) Analysis can be made manually or automatically, based on deductive systems or by model checkers, respectively; (4) Results of the analysis can either show specific vulnerabilities that require intervention of an administrator, generate a new safe configuration for network devices, or prove that the property is valid under every condition of the network.

Many papers in this area deal with static network configuration. If network configuration or topology changes, a model of the network has to be rebuilt. Our approach deals with networks with dynamic behavior. Dynamics is modeled by routing protocols, e.g. RIP or OSPF. Our goal is to automatically verify network security properties in a network model. The network model is constructed with respect to the configuration files of network devices and the network topology.

Geoffrey G.Xie et al. in (Xie, et al., 2005) show that routing information can be added to the static model of the network using additional filtering rules. These filtering rules can be changed as the state of links is changed, so the filtering rules depend on the actual state of the network.

Hence, the introduced formal model can be derived automatically from configuration files of routers. It deals only with IP addressing, and routing and filtering rules. Other parameters and settings are abstracted away for the sake of simplicity.

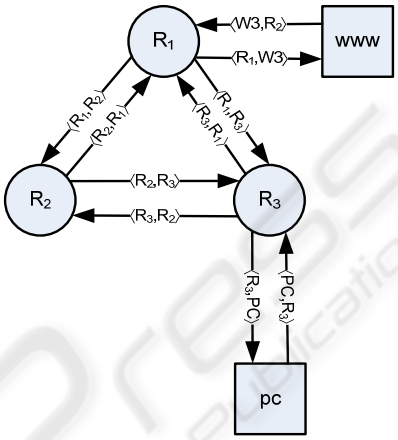

Figure 2: Network model for the example topology.

4 SECURITY PROPERTIES

Description of network security properties is related to the classification of threats and intrusion. There are plenty of different network security problems, such as HTTP attacks, spam, TCP flooding, DoS attacks, Web server misuse, spoofing and sniffing etc.

For instance, Kumar in (Kumar, 1995) offers classification of abstract signatures. Neumann and Parker (Neuman and Parker, 1989) propose nine categories of misuse techniques (external misuse, hardware misuse, masquerading, etc.). Lindquist and Jonsson (Lindqvist and Jonsson, 1997) describe intrusion in two dimensions – intrusion technique and intrusion result. On-line databases of vulnerabilities have their own categories: *CVE* (Common Vulnerabilities and Exposures) (Mitre, 2008) defines many categories like buffer errors, code injection, configuration, credentials, cross-site scripting, etc. Intrusion detection database *Snort* (Snort, 2008) classifies violation rules mostly with respect to applications – *chat, nntp, mysql, pop, icmp, imap, web,* etc.

Our network model deals only with IP addresses and services or ports. Therefore, the analysis does not reflect hardware or OS attacks. We also don't examine the contents of TCP/UDP packets. Our

primary goal is safety or resistance of the network with respect to dynamic behavior of the network. Therefore, our classification includes only basic categories of network security properties. Since it can utilize typical fields from IP, TCP, or UDP headers, namely source/destination IP address and service/port (Matousek, et al,, 2008), it allows to specify wide range of different communications to be analyzed in the network.

5 MODEL-CHECKING OF SECURITY PROPERTIES

5.1 Principles of Implementation

In this subsection, a framework suitable for implementation of a model-checking algorithm based on the interpretation of a dynamic network as a state system with transitions induced by changes of link conditions is described. The employed description language is based on propositional modal logic (Stirling, 1992). For the computation of the model and evaluation of properties in this model, a representation of ACLs and routing rules in the form of filtering predicates F_N for network N is deployed.

We define state predicates that can be interpreted in each state of the model and state functions that can be evaluated in each state of the model. In our case, we use $\text{NetReach}_{\varphi}(R)$ function that determines a set of routers reachable from router R under packet property φ. Evaluating this function in network states s_1, s_2 can give different results because dynamic routing information varies with network topology.

Putting restrictions on the path between two routers enables, e.g., to verify if there exists a path between two routers for Web traffic. This is called *network path under packet property*. It restricts the set of possible paths between routers to those paths where packet property φ is satisfied on every link of the path in network state *s*.

Network reachability under packet property on the network N in network state s, $\text{NetReachs}_{\varphi}^{s}(R)$, is a set of routers reachable from router R for packet satisfying property φ considering network state *s*. It can be computed by a least fixed-point algorithm. We use a language of modal logic to express security properties. Modal logic allows us to reason with validity of packet properties (protocol = TCP, port = 80) in different network states.

A formula of modal language is interpreted in network transition system T_N. For non-modal fragments we need to interpret atomic sentences. For model checking analysis we define a modal model on the network transition system T_N as a pair $M_N = \langle T_N, V_N \rangle$ where V_N is a valuation assigning a subset of T_N's states to each atomic sentence Q. Truth at a state *s* of an arbitrary formula ψ under M_N is inductively defined using Kripke semantics.

5.2 Decision Procedure

This subsection briefly explains how to construct a general decision procedure for modal model in realm of the network transition system T_N. The *small modal property*, ◊ψ, of the logic guarantees the decidability of the procedure that tests the satisfiability of a formula. We adopted bounded model checking (Biere, 2003) that limits state space by reachability diameter. In this case, the reachability diameter equals to the number of links that we consider in the analysis. It means that we are interested in checking whether the given property holds in the given set of network states which are limited by their distance to some initial state, e.g. we accept at most three link failures in the analysis. The problem of performing a full analysis, which means to check network state for any combination of link states (2^n), rises exponentially with the size of the network (number of links). Using bounded model checking with a diameter *k* we can get the most interesting results quickly as we limit the size of the problem for a network with *n* links to $(k \cdot n + 1)$ states that needs to be visited. This is based on practical consideration because for most of the time the network is assumed to be in the normal state, for which we consider that all links in a network are functioning properly and in the case of a failure occurs it affects only a small number of network links. Also depending on a network topology, a larger number of link failures can lead to a situation that some network parts will be inaccessible. If these parts are crucial with respect to network reachability then we can conclude without additional computation that the property cannot be satisfied in this state of the network. One can also find that if even more links fail then reachability is not resumed. From this consideration we may conclude, that for many network properties there exist such states in network transition system, which allows us to reduce a state space that needs to be explored by the variant of the method presented in

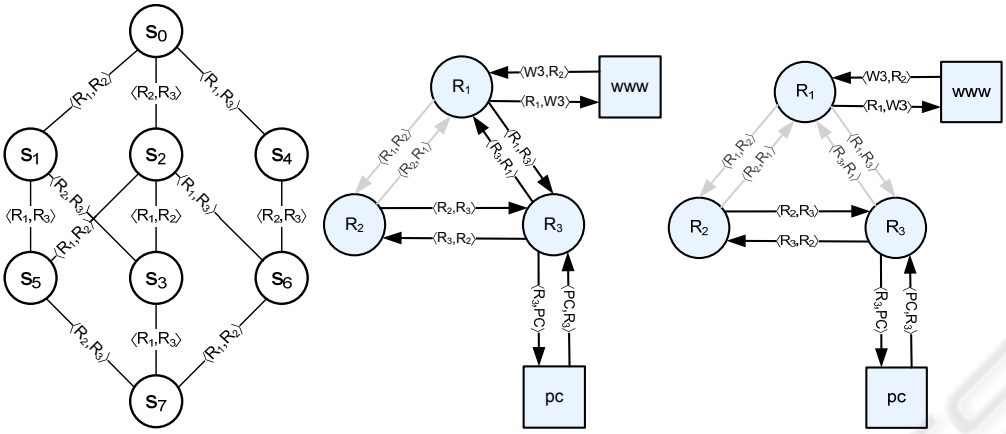

Figure 3: Example of (a) network transition system, network model in (b) state s1, and (c) state s5.

this paper that would provide an exhaustive analysis of network reachability under packet property.

In the following, the translation of the model-checking problem to SAT based problem is exposed.

For a modal-model M_N and a modal-formula ψ, the bound k is given by the total number of links that may change their state. As shown above, expression $NetReachs^s_\varphi(R)$ can be evaluated in every state s. In this case, the SAT-based decision procedure evaluates propositional formula $[[M_N, \psi]]_k$ that consists of a propositional representation of transition system T_N and network property ψ with standard modal interpretation:

$$[[M_N, \psi]]_k \triangleq [[M_N]]_k \wedge [[\psi]]_k.$$

5.3 Example

Assume the network model drawn in Figure 2, network transition system shown in Figure 3 and routing rules that generate filtering predicates. Then the decision procedure that evaluates property ψ as defined in this section works as follow. Using Kleene's Algorithm (see e.g. Gross and Yellen, 2004, p.66), the reachability matrix for the network at any given state is computed. Each cell of the reachability matrix defines a formula consisting of filtering predicates. The formula specifies overall filtering predicate for the whole path. Therefore, the proposition

$$WWW \in NetReach_\varphi(PC)$$

is translated to

$$(f_{1,3} \vee (f_{1,2} \wedge f_{2,3})) \wedge \varphi.$$

The network transition system is translated into the following propositional formula:

$$M \triangleq s_7 \xrightarrow{\downarrow(R_1, R_2)} s_6 \wedge s_7 \xrightarrow{\downarrow(R_1, R_3)} s_3 \wedge \ldots s_1 \xrightarrow{\downarrow(R_1, R_2)} s_0 \wedge s_4 \xrightarrow{\downarrow(R_1, R_2)} s_0,$$

where e.g. $\downarrow(R_1, R_2)$ means that the connection from R1 to R2 changed its state from *connected* to *disconnected*; hence, it is no longer available.

SAT solver takes a boolean proposition and attempts to find a valuation of the boolean variables such that the formula is satisfied. To allow application of SAT solver, the predicates need to be represented as boolean variables.

This example cannot be run without exploiting at least a prototype model checker implemented as a SAT-based decision procedure, and without converters or handwork that can convert a concrete network model and a verified property expressed as a modal formula into the input formats required by the employed model checker. The other possibility is development of special tools briefly described in the next section.

This demonstration example is discussed in more detail in (Cejka et al., 2008).

6 IMPLEMENTATION

The first step of implementation phase, currently realized, consisted in transforming the methods mentioned above into the prototypes of ensuing software tools. These prototypes enabled us to perform case studies in order to measure and analyze attributes of the proposed verification method. To shorten the design phases as much as possible the

free software libraries implementing verification procedures were exploited including high-level programming techniques because the short run-time and a small footprint of the tools were not the primary concerns at that phase. Instead, the rapid development of tools allowing us to carry out experiments with the methods proposed was main apprehension. The resulting set of tools can provide automatic analysis of router configurations, network topologies and states with respect to checked properties.

Evaluation of the project will be based on the outputs from the experiments with the computer tools developed in the previous phase. The experiments are considered to be an inevitable part of the project. The evaluation can be split into the following steps:

1. Capability of the proposed methods will be demonstrated.
2. The comparison of methods based on simulation with methods based on verification in the domain of network analysis will be given.
3. Analysis performed on case studies will reveal how the methods can be applied in real conditions.

Note that several different methods may be suitable for modeling and analysis of the environment and properties in the domain of interest. Most often, the combination of several methods leads to better results. The emphasis of the project's research is put on the formal verification methods, but other methods are certainly worthwhile to explore as well. The other methods may be orthogonal with formal verification, or they may support the formal methods.

In particular, monitoring may provide a fruitful data for classification and definition of security-related properties based on the real traffic. Simulation serves as a useful tool to specify and replay possible dangerous scenarios found by the formal verification. Therefore, simulators and monitors are seen as supporting tools for the network-wide analysis. Their study brings added-value insight with respect to the main research topic of the project. We plan to determine relative positions of all these methods and tools in the next evaluation report.

7 CONCLUSIONS

In this paper, we demonstrate the problem of automatic security analysis of TCP/IP based computer networks. The presented verification method aims at validating network design against the absence of security and configuration flaws. The network model allows describing effects of static and dynamic routing and access control lists configured on the network devices. The verification technique based on the bounded model checking supported by SAT-based decision procedure counts for varying link conditions' checks whether a given property holds in the network model. It was shown that the method is able to deal with various classes of properties, namely availability, safety and security. In all cases, a language of modal logic was used to express the property formally while serving as an input to the model checking algorithm. Although not validated, we believe the application of this technique is feasible for a large class of network models and properties.

It was shown that bounded model checking is a useful method in this area. The developed experimental tools provided reasonable data convincing us that the method is applicable in practice. Nevertheless, the experiments with real-size models are still in progress and further analysis is required to fully evaluate the method. There are also various possible extensions to the method. First, the specification language is rather minimalist and it is challenging to show whether all important security properties can be specified in it. In case of negative answer the further work should be focused on refining classification of properties and proposing an adequate extension of the language and the verification procedure. Second, a lot can be done in the area of optimization of the method. It requires deeper understanding of the relation of dynamic routing protocols behavior to the network transition system. Finally, for conducting practical experiments it is necessary to implement reliable and effective tools that would improve and extend the current experimental tools that need to be sometimes manually supported.

We believe that currently appearing papers in some sense inspired by, or at least referencing and/or developing our approach to modeling of dynamic networks for security analysis, see e.g. (Jeffrey and Samak, 2009), (Bera, Ghosh and Dasgupta, 2009), (Bera, Ghosh and Dasgupta, 2009a) or (Holloway, 2009) can demonstrate usefulness of the presented conception.

ACKNOWLEDGEMENTS

This project has been carried out with a financial support from the Czech Republic state budget through the CEZ MMT project no.

MSM0021630528: *Security-Oriented Research in Information Technology*, by the Grant Agency of the Czech Republic through the grant no. GACR 102/08/1429: *Safety and Security of Networked Embedded System Applications*, and by the Brno University of Technology, Faculty of Information Technology through the specific research grant no. FIT-10-S-1: *Secured, Reliable and Adaptive Computer Systems*. Also, the first co-author was supported by the grant no. FR-TI1/037 of Ministry of Industry and Trade: *Automatic Attack Processing*.

REFERENCES

Bartal, Y., Mayer, A.J., Nissim, K., Wool, A., 1999. Firmato: A Novel Firewall Management Toolkit. In *IEEE Symposium on Security and Privacy*, pages 17–31.

Bera, P., Ghosh, S.K., Dasgupta, Pallab, 2009. Fault Analysis of Security Policy Implementations in Enterprise Networks. In the *First International Conference on Networks & Communications*, IEEE Comp.Soc., pages 240-245.

Bera, P., Ghosh, S.K., Dasgupta, Pallab, 2009a. Formal Verification of Security Policy Implementations in Enterprise Networks. In *LNCS 5905*, Springer Berlin / Heidelberg, pages 117-131.

Biere, A., Cinnatti, A., Clarke, E., Strichman, O., Zhu, Y., 2003. *Bounded model checking. Advances in Computers*, Advances in Computers, Academic Press.

Burns, J., et al., 2001. Automatic management of network security policy. In *DARPA Information Survivability Conference and Exposition*, pages 1012–1026.

Cejka, R., Matoušek, P., Rab J., Rysavy, O., Sveda, M., 2008. *A Formal Approach to Network Security Analysis*. Technical Report FIT, Brno University of Technology, Brno, CZ.

Christiansen, M., Fleury, E., 2004. An Interval Decision Diagram Based Firewall. In *3rd International Conference on Networking (ICN'04)*. IEEE, pages 1–6.

Clarke, E.M., Grumberg, O., Peled, D.A., 1999. *Model Checking*. MIT Press.

Gross, J.L., Yellen, J., (editors), 2004. *Handbook of Graph Theory*. CRC Press.

Holloway, E.M., 2009. *Self Organized Multi Agent Swarms (SOMAS) for Network Security*. Master's Thesis, Air Force Inst of Tech Wright-Patterson AFB OH School of Engineering and Management.

Jeffrey, A., Samak, T., 2009. Model Checking Firewall Policy Configurations. In *IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 60-67, 2009.

Kumar, S., 1995. *Classification and Detection of Computer Intrusions*. PhD Thesis, Purdue, IN.

Lindqvist, U., Jonsson, E., 1997. How to Systematically Classify Computer Security Intrusions. In *IEEE Symposium on Security and Privacy*, Washington DC.

Matousek, P., Rab, J., Rysavy, O., Sveda, M., 2008. A formal model for network-wide security analysis. In *15th IEEE Symposium and Workshop on ECBS*, 2008.

Mitre, 2008. *Common Vulnerabilities and Exposures Database*. Available on
http://cve.mitre.org/; accessed on Feb 2008.

Neumann, P.G., Parker, D.B., 1989. A Summary of Computer Misuse Techniques. In *Proc. 12th National Computer Security Conference*, pages 396–407.

Ou, X., Govindavajhala, S., Appel, A.W., 2005. MulVAL: A logic-based network security analyzer. In *Proc. of the 14th USENIX Security Symposium*, Baltimore.

Ritchey, R.W., Ammann, P., 2000. Using model checking to analyze network vulnerabilities. In *IEEE Symposium on Security and Privacy*, Washington, USA.

Shahriari, H.R., Jalili, R., 2005. Modeling and Analyzing Network Vulnerabilities via a Logic-Based Approach. In *2nd Int. Symposium of Telecommunications*, pages 13–18.

Snort, 2008. Snort network intrusion and prevention system. Available from http://www.snort.org/; accessed on Feb 2008.

Stirling, C., 1992. *Modal and temporal logics*. pages 477–563. Oxford University Press, Inc., New York, NY, USA.

Tidwell, T., Larson R., Fitch K., Hale J., 2001. Modeling Internet attacks. In *Proc. of the IEEE Workshop on Information Assurance and Security*, West Point, NY.

Xie, G.G., Zhan, J., Maltz, D.A., Zhang, H., Greenberg, A.G., Hjalmtysson, G., Rexford, J., 2005. On static reachability analysis of ip networks. In *INFOCOM*, pages 2170–2183.

Zakeri, R., Shahriari, H.R., Jalili, R., Sadoddin, R. , 2005. Modeling TCP/IP Networks Topology for Network Vulnerability Analysis. In *2nd Int. Symposium of Telecommunications*, pages 653–658.