SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2009; **2**:83–96
Published online 24 September 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.58

Network risk management using attacker profiling

Ram Dantu[1]*,† Prakash Kolan[2] and João Cangussu[3]

[1]Department of Computer Science and Engineering, University of North Texas, TX, U.S.A.
[2]Department of Computer Science, University of North Texas, TX, U.S.A.
[3]Department of Computer Science, University of Texas at Dallas, TX, U.S.A.

Summary

Risk management refers to the process of making decisions that minimize the effects of vulnerabilities on the network hosts. This can be a difficult task in the context of high-exploit probability and the difficult to identify new exploits and vulnerabilities. For many years, security engineers have performed risk analysis using economic models for the design and operation of risk-prone, technological systems using attack profiles. Based on the type of attacker identified, security administrators can formulate effective risk management policies for a network. We hypothesize that sequence of network actions by an attacker depends on the social behavior (e.g., skill level, tenacity, financial ability). We extended this and formulated a mechanism to estimate the risk level of critical resources that may be compromised based on attacker behavior. This estimation is accomplished using behavior based attack graphs representing all the possible attack paths to all the critical resources. The risk level is computed based on these graphs and are used as a measure of the vulnerability of the resource and forming an effective basis for a system administrator to perform suitable changes to network configuration. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: attack graphs; behavior; risk management

1. Introduction

Enterprise networks are growing larger by the moment. In addition, multiple types of hosts and improperly configured machines spell a formula for a network vulnerability nightmare. Accurate vulnerability analysis requires a deep understanding of both failure and attack modes and their impact on each network component, as well as the knowledge of how components interact during normal and attack modes of operation. For managing the security of a network, security administrators must identify security vulnerabilities on a host and network level, assess the risk associated with the vulnerabilities, and fix them using patches released by the vendors.

Product vendors release frequent patches for their products. This practice seems to be most prevalent in the Microsoft OS. Patching network hosts is a short term solution and targets only a few security vulnerabilities at a time. This process of patching end hosts and their components requires a great deal of human intervention, time, and money. We must also consider new security vulnerabilities that will not be identified until a breach has occurred.

Risk management has to be a top priority to system administrators. Risk management refers to the process of making decisions that minimize the effects of vulnerabilities on the network hosts. However, in the context of high-exploit probability, risk management is a nightmare to plan with. Also, identifying new

*Correspondence to: Ram Dantu, Department of Computer Science and Engineering, University of North Texas, TX, U.S.A.
†E-mail: rdantu@unt.edu

exploits and vulnerabilities is a difficult task. For many years, security engineers have performed risk analysis using economic models for the design and operation of risk-prone, technological systems using attack profiles [1–4,24]. Based on the type of attacker identified, security administrators can formulate effective risk management policies for a network. Simultaneously, a great deal of psychological and criminological research has been devoted to the subject; but security engineers do not actually use these studies. Many articles explain how intruders break into systems [5,6].

Companies like Psynapse, Amenaza, and Esecurity have built products using the behavior of intruders. To the best of our knowledge, no work has been reported on integrating behavior-based profiles with sequences of network actions to compute resource vulnerability. Thus, the overall goal of this research is to estimate the risk associated to a critical resource based on attacker behavior and a set of vulnerabilities that the attacker can exploit. This approach implies a more fine-grained repertoire of risk mitigation strategies tailored to the threat rather than using blanket blocking of network activity as their sole response. This paper combines which vulnerabilities can be exploited with the risk of resources based on the attack behavior. Our work uses the theory from criminology, statistical analysis, behavioral based security, and attack graphs.

The remainder of this paper is organized as follow. Section 2 presents a brief introduction to some background work. Section 3 describes the five steps of the proposed methodology; from the creation of attack profiles and graphs to the optimization of the risk level. The attack profile from Section 3 is derived from a survey described elsewhere [7]. Section 4 provides analyses of how sensitive are the models with respect to errors or noise in the initialization process. Concluding remarks are presented in Section 5.

2. Background

There are several pieces of work that coincide with the research proposed here. Jackson [2] introduces the notion of behavioral assessment to determine the intent behind the attack. The proposed Checkmate intrusion detection system distinguishes legitimate use from misuse through behavior intent. Rowley [8] views risk analysis as involving threat identification, risk assessment, and steps to be taken for risk mitigation. The potential threats are identified and an estimate of the damage each threat could pose is calculated. These papers help understand behavior and risk, but do not try to integrate their analysis with attacker behavior and the sequence of network actions that can be performed by the attacker. We think it is necessary to bring risk mitigation together with attacker behavior, in an effort to considerably reduce the risk to potential systems.

The psychological and criminological research on hacker community defines categories of hackers such as *novices*, *crackers,* and *criminals* based on their intent, skill, and attack proficiency [3,4,9–13]. Rogers [13] proposes categorizations of a hacker community and advises hacker profiles derived from intruder behavior. Yuill *et al*. [4] determine the detection of an ongoing attack by developing a profile of the attacker using the information the attacker reveals during attacks. Kleen [9] presents a framework for analysis of hackers by reviewing existing hacking methods, classifications and profiles, with the goal of better understanding their values, skills, and approaches to hacking.

Another approach followed by researchers was to create a graphing technique to represent network actions [25–27]. These ‘attack graphs’ represent all possible network actions that can be used to compromise a network component, or exploit a given vulnerability [14]. An attack graph is created by places/ nodes, which represent a network action and a connecting sequence of nodes from root to leaf to represent a successful attack, using the different node methods. Attack graphs can also be a way of formalizing the risk for a given network topology and given exploits. Sheyner *et al*. [15] model a network by constructing an attack graph using symbolic-model-checking algorithms. Moore *et al*. [16] document attacks on enterprises in the form of attack trees, where each path from the root to the end node documents how an attacker could realize his desire of exploiting the host and network. In our previous work [17–19], we used Bayesian inference techniques and attack graphs for risk management, in this paper however, we will present a comprehensive view of attack profile generation (using surveys with real people [7]) and adaptive risk computation.

3. Methodology

The use of attack graphs (or attack trees) is increasing in popularity. Its ability to formalize and provide models for representing system security has shown

DOI:10.1002/sec

to be practical for risk management. An attack graph can be created using network topology (as seen in Figure 1) and various vulnerabilities of each host [15,16,20]. The attack graphs represent a sequence of network actions for exploiting each network resource and, ultimately, the whole network. Using attack graphs, we calculate the vulnerability level and susceptibility to a critical resource. Our procedure consists of five steps which we repeatedly execute until an optimal level of security is achieved. Our hypothesis is that a relationship exists between network actions and the behavior attributes of the attackers.

3.1. Step 1: Creation of an Attacker Profile

A typical attacker profile would consist of the given attacker and the available resources: including but not limited to cost, computer and hacking skills, time, attitude, tenacity, perseverance, and motives (such as revenge or reputation). Each profile generated will have unique behavioral attribute values for given resources, meaning that a corporate espionage agent has more money than a ‘script kiddie’ who hacks for fun with little or no money. The difference between attributes will weigh differently when applied to separate profiles. In another example, a corporate insider has more knowledge regarding the enterprise network topology—a valuable resource—whereas external hackers may have only a basic understanding of the topology they are attacking. Once we identify an attacker’s resources, we can assign attribute values to those resources. For example, one way is to assign relative attributes for a profile who has low level of skill (e.g., 0.2), medium level of attitude (e.g., 0.6), and high level of time (e.g., 0.8) (for example, a college student could fit in this profile).

3.2. Step 2: Creation of Attack Graphs

Once we have generated the attacker profiles, we develop attack graphs to aid in our analysis of the risk. For a given network topology as shown in Figure 1, we can derive the sequence of network actions that can be executed to exploit various network host vulnerabilities.

Figure 2 shows multiple nodes which represent the different network actions that can be executed to exploit vulnerabilities such as ftp.rhosts and remote buffer overflow. Each node in the graph represents a network action and a path from root to leaf represents a successful attack. The graph shows how intruders culminate a sequence of state transitions to successfully achieve an attack. For example, an attack path (see Figure 2) can be a sequence of events such as: overflow *sshd* buffer on host1(H1), overwrite *.rhosts* file on host2(H2) to establish *rsh* trust between H1 and H2, log-in using *rsh* from H1 to H2, and finally, overflow a local buffer on H2 to obtain root privileges. We can derive different attacker profiles based on the network systems in use and the users that access them. It is of common understanding that network systems with sensitive and confidential information (e.g., consumer banks) are the prime targets for unauthorized users such as hackers, spammers etc. In this case, we frequently see attackers with expert knowledge attempting to compromise the security of the networked systems for their personal and organizational gain. However, networked systems with non-critical information (e.g., sports clubs) are seldom attempted to be compromised by expert hackers compared to novices such as script kiddies. Therefore, depending on the network resources and their accessibility and security policies, we can derive attacker profiles and associate behavioral attributes to those profiles.

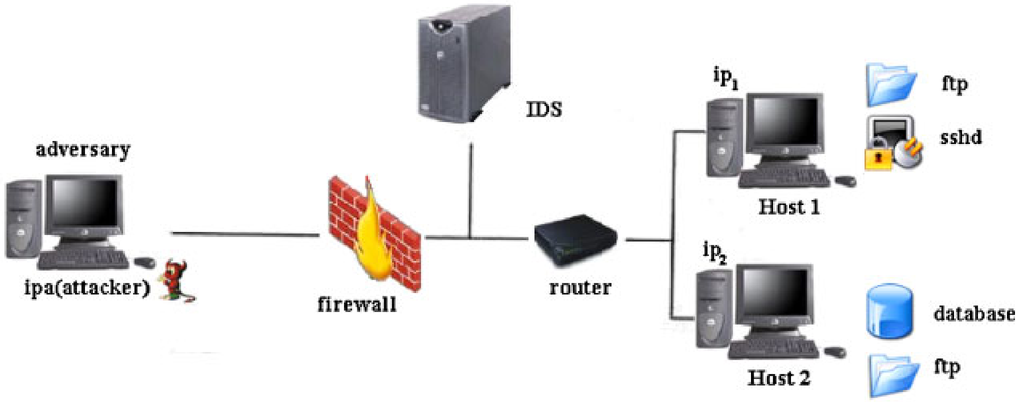


Fig. 1. Example of a network topology.

DOI:10.1002/sec

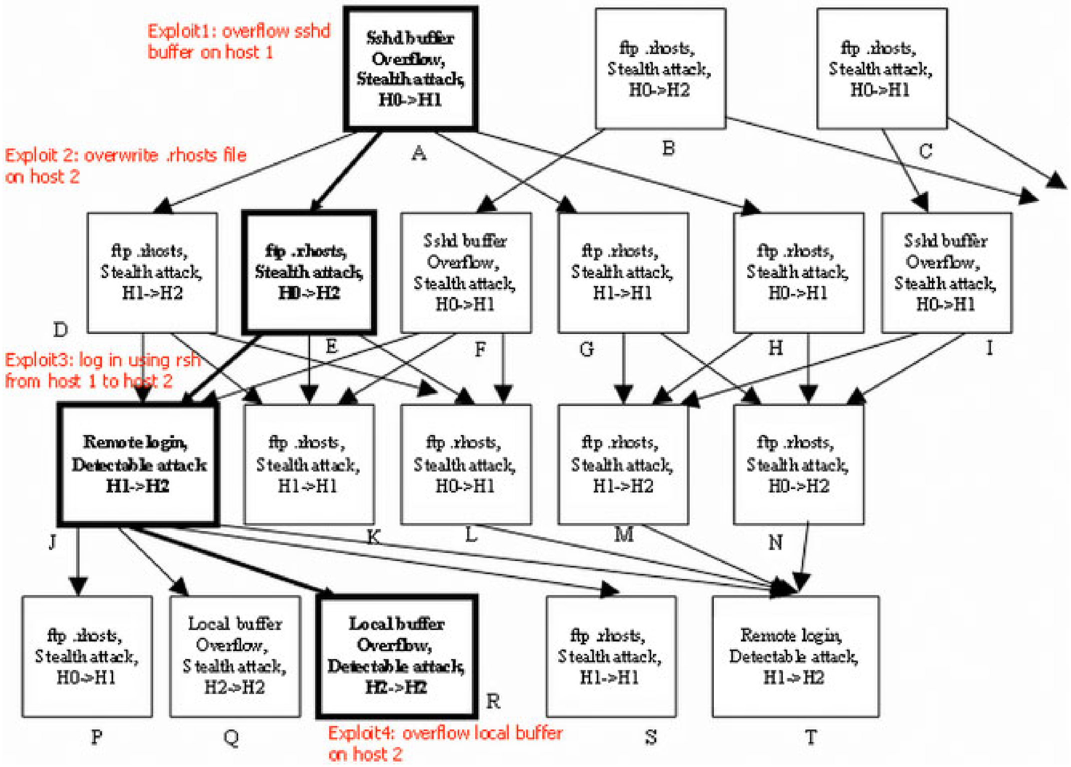

Exploit1: overflow sshd buffer on host 1
Sshd buffer Overflow, Stealth attack, H0->H1
A
ftp .rhosts, Stealth attack, H0->H2
B
ftp .rhosts, Stealth attack, H0->H1
C
Exploit 2: overwrite .rhosts file on host 2
ftp .rhosts, Stealth attack, H1->H2
D
ftp .rhosts, Stealth attack, H0->H2
E
Sshd buffer Overflow, Stealth attack, H0->H1
F
ftp .rhosts, Stealth attack, H1->H1
G
ftp .rhosts, Stealth attack, H0->H1
H
Sshd buffer Overflow, Stealth attack, H0->H1
I
Exploit3: log in using rsh from host 1 to host 2
Remote login, Detectable attack H1->H2
J
ftp .rhosts, Stealth attack, H1->H1
K
ftp .rhosts, Stealth attack, H0->H1
L
ftp .rhosts, Stealth attack, H1->H2
M
ftp .rhosts, Stealth attack, H0->H2
N
ftp .rhosts, Stealth attack, H0->H1
P
Local buffer Overflow, Stealth attack, H2->H2
Q
Local buffer Overflow, Detectable attack, H2->H2
R
Exploit4: overflow local buffer on host 2
ftp .rhosts, Stealth attack, H1->H1
S
Remote login, Detectable attack, H1->H2
T

Fig. 2. An example attack graph showing a successful attack.

With respect to scalability two issues are of concern. The first issue refers to the initialization of the attack graph. Here a survey has been used for such purpose but direct expert opinion can also be used. The construction and initialization of a Bayesian Belief Networks (BBN) for a large enterprise network will clearly demand a considerable effort. However, the effort can be reduced in case the attack graph has a large number of similar subgraphs. In this case the initialization of the sub-graphs can be replicated from the computation of the first sub-graph. This is a reasonable assumption as networks are, in general, organized in a hierarchical way where nodes at same level present similar roles. The second issue refers to the computation of the conditional probabilities for attack graphs from large networks. It is true that the complexity of calculations increases with increase in the size of the attack graph (size of the network). In this type of cases, the methodology described by Castillo *et al.* [21] helps in minimizing the complexity up to a certain extent. In addition, the complexity can be further reduced by iteratively computing the risk probabilities at different levels (e.g., first with respect to gateways, then routers, then maybe by clients connected to it.).

3.3. Step 3: Assigning Behavior Attributes to Attack Graph Nodes

We have conducted an online survey to assist with identifying values for attributes such as skill, time, and attitude for people with varied behavior. The goal is to infer a behavior model and construct attack graphs by documenting attack paths that different profiles can execute. A detailed description of the survey is available elsewhere [7]. The survey also helps to understand the attributes associated with attack graphs such as computer and hacking skills, time, attitude, tenacity, cost of attack, and techniques for avoiding detection. In Figure 3, there are two example profiles A and B for three attributes of skill, time, and attitude. A measure of these attribute values gives the amount of risk associated with the profile.

In Figure 3, we show attack graphs that two example profiles (Profile A and Profile B) can execute with their available resources. For example, an attacker with Profile A can exploit the local buffer overflow attack at H2 by executing the sequence of network actions ftp.rhosts stealth attack H1->H2 (with 0.2 skill, 0.3 attitude, and 0.1 time), remote login detectable attack H1->H2 (0.3 skill, 0.7 attitude,

DOI:10.1002/sec

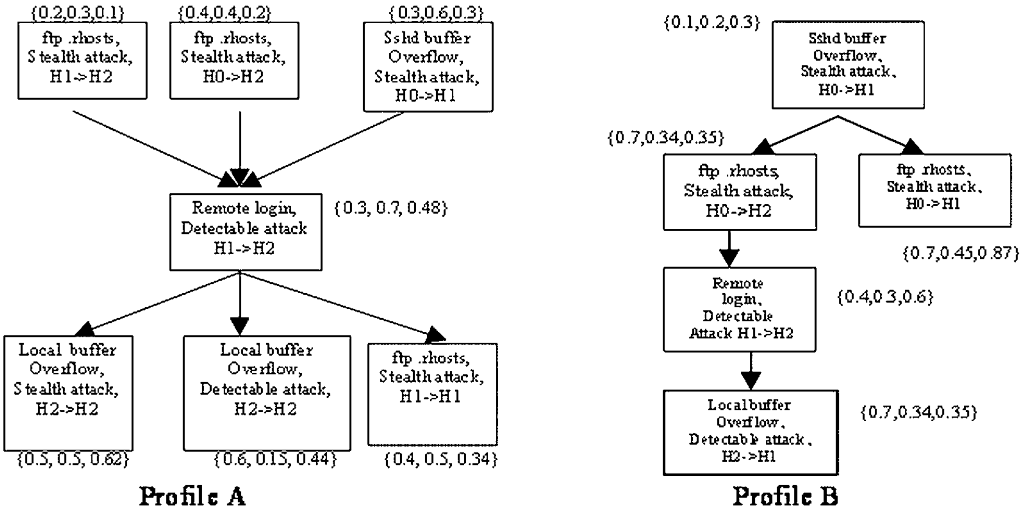

{0.2,0.3,0.1}
ftp .rhosts, Stealth attack, H1->H2
{0.4,0.4,0.2}
ftp .rhosts, Stealth attack, H0->H2
{0.3,0.6,0.3}
Sshd buffer Overflow, Stealth attack, H0->H1
Remote login, Detectable attack H1->H2
{0.3, 0.7, 0.48}
Local buffer Overflow, Stealth attack, H2->H2
{0.5, 0.5, 0.62}
Local buffer Overflow, Detectable attack, H2->H2
{0.6, 0.15, 0.44}
ftp .rhosts, Stealth attack, H1->H1
{0.4, 0.5, 0.34}
Profile A
{0.1,0.2,0.3}
Sshd buffer Overflow, Stealth attack, H0->H1
{0.7,0.34,0.35}
ftp .rhosts, Stealth attack, H0->H2
ftp .rhosts, Stealth attack, H0->H1
{0.7,0.45,0.87}
Remote login, Detectable Attack H1->H2
{0.4,0.3,0.6}
Localbuffer Overflow, Detectable attack, H2->H1
{0.7,0.34,0.35}
Profile B

Fig. 3. Two example attack paths.

and 0.48 time), and the local buffer overflow detectable attack H2->H2 (with 0.6 skill, 0.15 attitude, and 0.44 time). Each of the profile attack graphs represents the attack paths (sequence of network actions to exploit a vulnerability) that the attacker with that profile can execute with his available resources. Deriving attack graphs for different attacker profiles helps us to identify possible attacks by the users of the systems in attack.

3.4. Step 4: Risk Computation

To compute risk, we use the set of paths, attributes, and an attacker profile. Then the risk level for critical resources can be calculated. A Bayesian network-based estimation technique [22] is used to calculate a resource's aggregate risk value which will be attack prone if the risk value exceeds a given threshold.

3.4.1. Inference based on attacker profiles

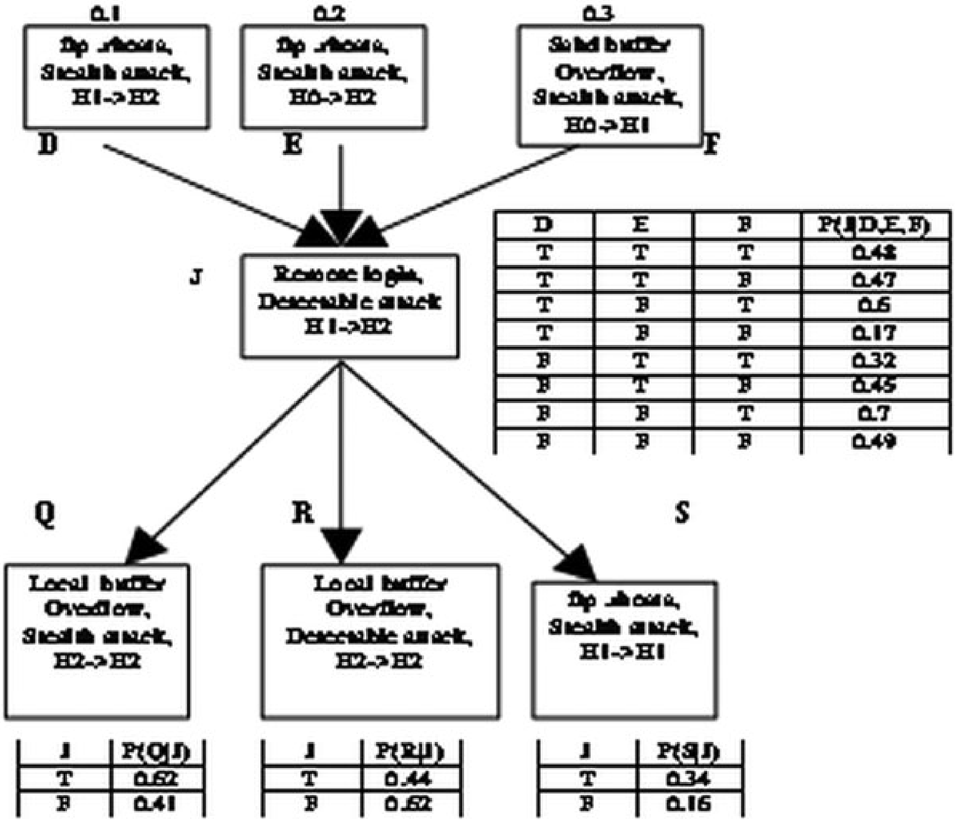
There are many ways to initialize an attack profile such as expert knowledge and past observations. Here, surveys [7] are used in an attempt to build more realistic profiles. The initialization of the profile helps in the assignment of behavior attribute values. For example, consider the attack graph of profile A from Figure 3 for a quantifying variable {attitude} required to exploit the local buffer overflow and the ftp.rhosts attacks (represented by nodes Q, R, and S). The new graph is shown in Figure 4.

From Figure 4, assume each of the nodes to be associated with two states 'yes' or 'no', and the initialized probability values as given in the figure. With this available initial knowledge, we can compute the posterior probability for observed evidence. If an attacker uses the ftp.*rhosts* stealth attack at node D, then we can calculate the probability that the attacker can exploit the local buffer overflow attack at node Q using the computed value of $P(\text{Q}|\text{D}) = 0.48$. That is, given that the attacker has exploited the ftp.rhosts attack at node D, the probability the attacker will exploit the local buffer overflow attack at node Q is equal to 0.48.

In patch management, we can estimate post patch probabilities if we know the exploits actual cause. This can be estimated by calculating the probability of the occurrence of a given cause based on observed evidence. For example, in Figure 4, if we have evidence regarding a local buffer overflow attack at H2 represented by node Q, then we can calculate the probability that the ftp.rhosts attack at node A was exploited using the computed value of $P(\text{D}|\text{Q}) = 0.09300$ that is, given that leaf node Q in Figure 4 has been exploited, the probability that the root node D has been used is 0.09300, a marginal decrease from its prior probability. This analysis was performed for each node in the attack graph, documenting the posterior probabilities of all nodes obtained using our analytical model. We validated the above results using the HUGIN DEMO [23].

HUGIN uses Bayesian inference techniques to create belief networks. However, it initializes values based on the graph structure and given values in the CPT (conditional probability tables.) HUGIN method re-computes probability values for all nodes and keeps them updated. At any given time, the new probability

DOI:10.1002/sec



D	E	F	P(J|D,E,F)
T	T	T	0.48
T	T	F	0.47
T	F	T	0.6
T	F	F	0.17
F	T	T	0.32
F	T	F	0.45
F	F	T	0.7
F	F	F	0.49

J	P(Q|J)
T	0.62
F	0.41

J	P(R|J)
T	0.44
F	0.62

J	P(S|J)
T	0.34
F	0.16

Fig. 4. A simple Bayesian causal graph.

values portray the nodes' current vulnerability level. The HUGIN example only uses boundary values, while our model takes into account any value between 0 and 1. These calculations are derived from the CPT, Bayesian inferences, and conditional probability analysis. The CPT give the probability of the nodes of a graph given their parent nodes. For example, in Figure 4, the probability of node Q given its parent node J is 0.62. To estimate the risk at different nodes of the attack graph, we initialize their probabilities conditioned on the probability values of their parents. We describe in detail a method of initializing node probabilities using a survey [7]. Please note that the values of the attack graph nodes can also be initialized using alternative techniques such as expert knowledge and field observations. Table I shows our model compared to the HUGIN method for deriving probabilities.

With our model, we can periodically update the probability of a successful attack for the attack graph shown in Figure 2. Using the values from the CPT, the graph can be initialized based on the parent child relationships from Figure 2, this data is given in Table II.

Table I. Probability of a node exploit using our analytical model and HUGIN for given evidence of an attack at node Q.

Node	Our analytical model	HUGIN
D	0.093	0.095
E	0.1948	0.196
F	0.297	0.306
J	0.412	0.425
R	0.55	0.543
S	0.17	0.199

Table II. Initialized values using initial values in CPT tables.

Node	Probability	Node	Probability	Node	Probability
A	0.4	B	0.63	C	0.34
D	0.424	E	0.51	F	0.2996
G	0.4560	H	0.356	I	0.4826
J	0.4454	K	0.6472	L	0.5447
M	0.4749	N	0.4165	P	0.3445
Q	0.5035	R	0.5398	S	0.2402
T	0.5112				

DOI:10.1002/sec

Using the values from Table II we can compute the posterior probability values of all the nodes for given evidence at any part of the graph. For example, if it is observed that one or more root nodes have been compromised, the probability that node P will be exploited is given by Table III.

3.4.3. Relating risk, behavior and penetration

As stated earlier, we believe that the sequence of network actions carried out by an attacker relates to a person's behavior. We attempt to derive the relation between vulnerability of a given resource and the level of penetration an attacker can achieve in exploiting the network. Inferring a successful attack, this value can be computed by initializing the probability of each node in each attack path and inferring the posterior probability given evidence at the leaf node.

The probability of the nodes is represented using the CPT, where each node has two states, yes or no. To infer network penetration, we consider an example with five nodes (A, D, J, Q, and R) and three profiles of attack behavior.

The profiles we consider are listed below and the data for each profile have been collect through a survey [7]:

- criminals-People with criminal behavior (e.g., Corporate Insiders—attackers having access to network resources, Corporate Espionage—spies),
- hackers-people with hacking behavior, and
- liberals-people who are liberal minded and believe in open doors (e.g., explorers).

The data for the five nodes are shown in Figure 2 and 5, and Tables IV and V. In Tables IV and V, 'P(U) given V = yes' represents the probability of reaching node U given that node V has been reached, where U can be assigned the values of nodes D and J while V can be assigned the values of nodes A and D. Notice that in Table IV 'P(D) given A = no' should be zero but the fact is that this graph is a sub-graph of a larger attack graph where alternative paths to node D do exist. The same is true for the probability of node Q. The table data are for example use; it would normally be initialized by analysis of statistical data from an interview or a survey [7].

Table III. Probability of Exploitation of Node P.

Exploited nodes	Probability of node P
A	0.3437
B	0.3442
C	0.3445
A & B	0.3436
B & C	0.3442
A & C	0.3437
A & B & C	0.3436

Table IV. Probability of nodes A, D, and J of Figure 6 given their parents.

Probability profile	P(A)	P(D) given A = yes	P(D) given A = no	P(J) given D = yes	P(J) given D = no
Criminals	0.8	0.75	0.82	0.85	0.70
Hackers	0.6	0.7	0.31	0.51	0.46
Liberals	0.4	0.52	0.36	0.48	0.32

For the profiles listed in the CPT, Figure 4, the greatest security risk is that of criminals. They have more knowledge, inside information, and also have the resources and drive to complete a successful attack. Hackers, however, fall into a middle threat level. Their resources and attitude are lower and pose less threat. Liberals possess the least skill, time, and attitude. They pose the least threat to security. The probability of a success attack changes, in some cases dramatically, when comparing different attack profiles.

Figure 5 shows the inference for the network penetration given a successful attack at a leaf node. The posterior probability for each node represents an attacker's ability to compromise the network action represented by them, and from these probabilities the network penetration can be directly inferred. For example, if node R is compromised, then nodes A, D, J, and Q are directly affected. The inferred posterior probabilities of the nodes directly affected by the observed evidence using our analytical model (see Section 3.4.2) are given in Table VI.

The risk values shown in Table VI can be plotted against network penetration (level number of nodes A, D, J, and Q from Figure 5) of the attacker as shown in Figure 6.

Table V. Probability of nodes Q and R from Figure 6 given their parents.

Probability profile	P(Q) given J = yes	P(Q) given J = no	P(R) given J = yes	P(R) given J = no
Criminals	0.71	0.83	0.69	0.77
Hackers	0.3	0.57	0.52	0.63
Liberals	0.62	0.41	0.44	0.62

DOI:10.1002/sec

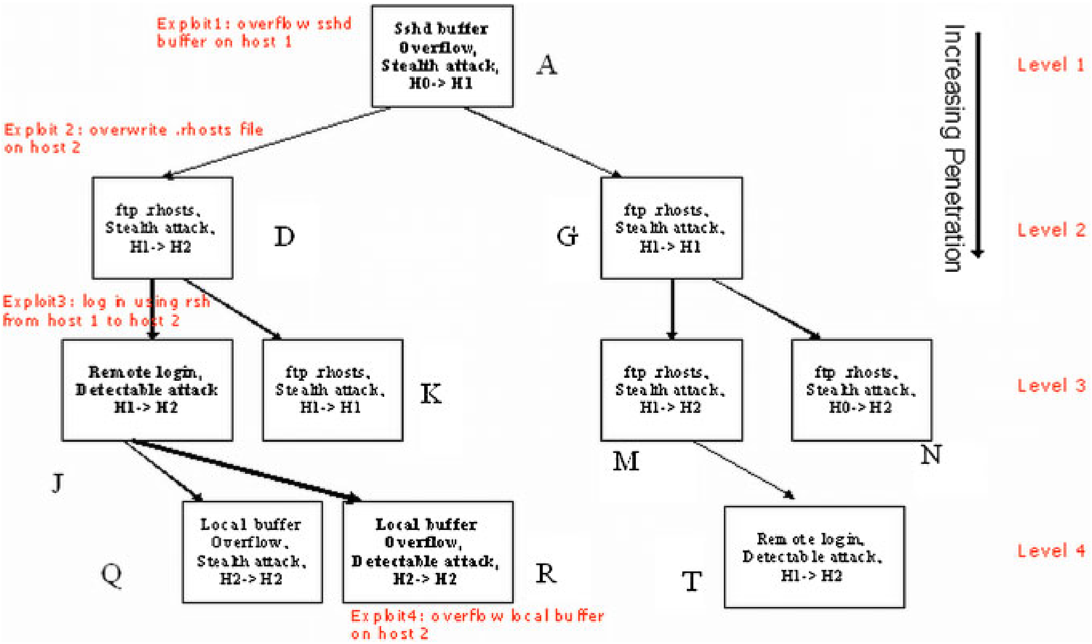

Exploit1: overflow sshd buffer on host 1
Sshd buffer Overflow, Stealth attack, H0-> H1
A
Increasing Penetration
Level 1
Exploit 2: overwrite .rhosts file on host 2
ftp rhosts, Stealth attack, H1-> H2
D
G
ftp rhosts, Stealth attack, H1-> H1
Level 2
Exploit3: log in using rsh from host 1 to host 2
Remote login, Detectable attack H1-> H2
ftp rhosts, Stealth attack, H1-> H1
K
ftp rhosts, Stealth attack, H1-> H2
ftp rhosts, Stealth attack, H0-> H2
Level 3
M
N
J
Q
Local buffer Overflow, Stealth attack, H2-> H2
Local buffer Overflow, Detectable attack, H2-> H2
R
T
Remote login, Detectable attack, H1-> H2
Level 4
Exploit4: overflow local buffer on host 2

Fig. 5. Example of a sub-attack graph extracted from Figure 2.

In Figure 6 the relationship between risk, behavior, and network penetration is given for each profile. The x-axis in Figure 6 represents the 'risk factor' which is computed in the range from 0 (no risk) to 1 (high risk). In real cases, the CPT will consist of a range of values for each profile. The range of probabilities in Tables VII–IX refers to the chances of an attacker been successful according to the profile (Criminal, Hacker, or Liberal). Presented in Tables VII–IX are examples from Tables IV–VI using a range of probabilities instead of single probability; in this case the values of all attributes of the profile are taken into consideration [11]. The values assumed in Table VII–IX are example range values based on the probabilities for the same nodes defined in Figure 4. Loper [11] has observed marked correlation between the behavioral predictor variables such as Hacker skill level and communication content. We have used this correlation information to define probability ranges in which the values of different behavioral attributes extend into. It can be clearly observed that defining probability ranges help us to identify overlapping behavior of different attacker profiles.

Table VI. Bayesian inference for directly affected nodes due to evidence at node R.

Profile	P(A)	P(D)	P(J)	P(Q)
Criminals	0.8002	0.7609	0.7975	0.7343
Hackers	0.5991	0.5416	0.4395	0.4513
Liberals	0.3980	0.4112	0.3102	0.4751

Using Tables VII–IX, we can build a new risk to penetration table for the ranged probabilites as given below in Figure 7.

Figure 8 is a depiction of Figure 7 representing the final relationship between penetration, risk, and behavior.

3.5. Step 5: Optimizing the Risk Level

Risk management is frequently used for mitigating the risk of network resources. This process involves patching exploits or vulnerabilities, changing network configurations (e.g., moving the firewall to another location in the topology, changing the firewall rules), or using intrusion detection systems. For each planned change or action taken to manage risk, our computation scheme (steps outlined in Section 3.1–3.4) can be performed repeatedly to obtain an optimum risk value.

Updating Risk after patching: For example, consider that the host H2 (node Q) has been patched and an attacker cannot exploit the local buffer overflow. This patch will change the attribute values {skill, time, and attitude} of that node which will be greatly reduced. The new example values for the system are shown in Figure 9.

Using Bayesian inference techniques, Equations 1–10, and the new probability for Node Q, we can derive

DOI:10.1002/sec

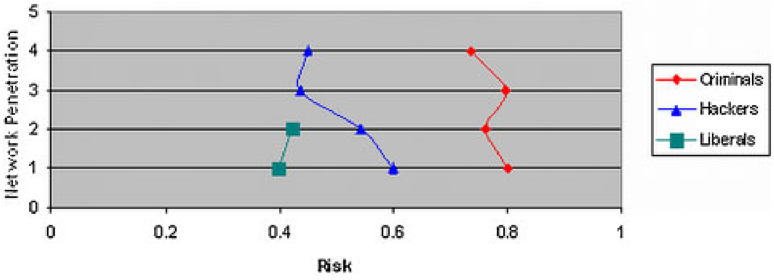


Fig. 6. Relating risk, behavior, and penetration for an attribute of all profiles.

the updated probability that a successful attack was completed by using Node Q. Given that an attacker has exploited the.rhosts vulnerability at node D, the updated probability is 0.0066. Likewise, with the above values of nodes and Equations 11–16, the updated probability that a root node (i.e., node D) was used given that the leaf node (i.e., node Q) was exploited is 0.088. Bayesian inference provides unsupervised learning and identifies the probabilistic links between vulnerabilities and hence it is a good tool for risk management.

Table VII. Given three profiles, range of probabilities of nodes A, D, and J of Figure 5 given their parents.

Profile	P(A)	P(D) given A = yes	P(D) given A = no	P(J) given D = yes	P(J) given D = no
Criminals	0.8/0.92	0.75/0.87	0.82/0.94	0.85/0.97	0.70/0.82
Hackers	0.6/0.75	0.7/0.85	0.31/0.46	0.51/0/0.66	0.46/0.61
Liberals	0.4/0.65	0.52/0.71	0.36/0.56	0.48/0.73	0.32/0.67

Table VIII. For given three profiles, range of probabilities of nodes Q and R of Figure 5, given their parents.

Profile	P(Q) given J = yes	P(Q) given J = no	P(R) given J = yes	P(R) given J = no
Criminals	0.71/0.83	0.83/0.94	0.69/0.82	0.77/0.89
Hackers	0.3/0.45	0.57/0.72	0.52/0.67	0.63/0.78
Liberals	0.62/0.84	0.41/0.63	0.44/0.68	0.62/0.81

Table IX. Inferred probability range of the three profiles for directly affected nodes A, D, J, and Q.

Profile	P(A)	P(D)	P(J)	P(Q)
Criminals	0.8/0.92	0.76/0.874	0.7975/0.974	0.734/0.835
Hackers	0.68/0.75	0.57/0.75	0.5/0.61	0.43/0.55
Liberals	0.398/0.649	0.411/0.655	0.310/0.672	0.475/0.771

Inferring Network Penetration after patching: After patching the local overflow vulnerability at node Q, the level of penetration per profile might be affected. In this example criminals would have the same level of penetration, but for a hacker and liberal, their level will be affected. Table X shows the updated levels of penetration given the new values for node Q.

Re-computing the posterior probabilities of the other nodes after the patch leads to different values for the hacker and liberal profiles as shown in Table XI.

The posterior probability for hackers and liberals has been reduced, observed from Table XI, thus reducing the risk associated with them. Using Table XI, post patch probabilities and the Figure 5 network penetration data, a new risk to penetration graph can be formulated, shown in Figure 10.

It is demonstrated that a difference is observed for the profile hacker. Comparing Figure 10 with Figure 6, the hacker is limited to level 4 before a patch at node Q and to level 3 after a patch. In this example the network penetration of a criminal is unaffected, as an insider a criminal will have access even after patching. However, to affect a criminal the network topology and rules for accessing it would have to be altered. For example, a system administrator could implement a policy rule advocating an extra line of defence may limit the access to the network by the Criminals and, thereby, limit how deeply they can penetrate the network and, thus,

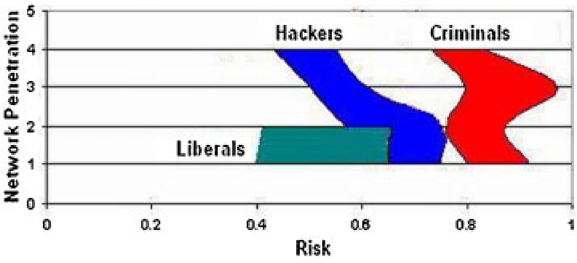


Fig. 7. Relation between risk and network penetration.

DOI:10.1002/sec

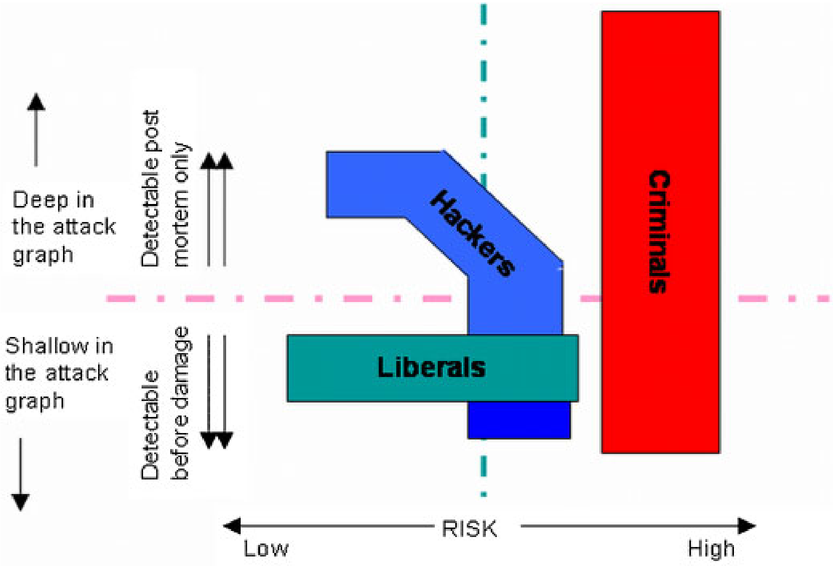
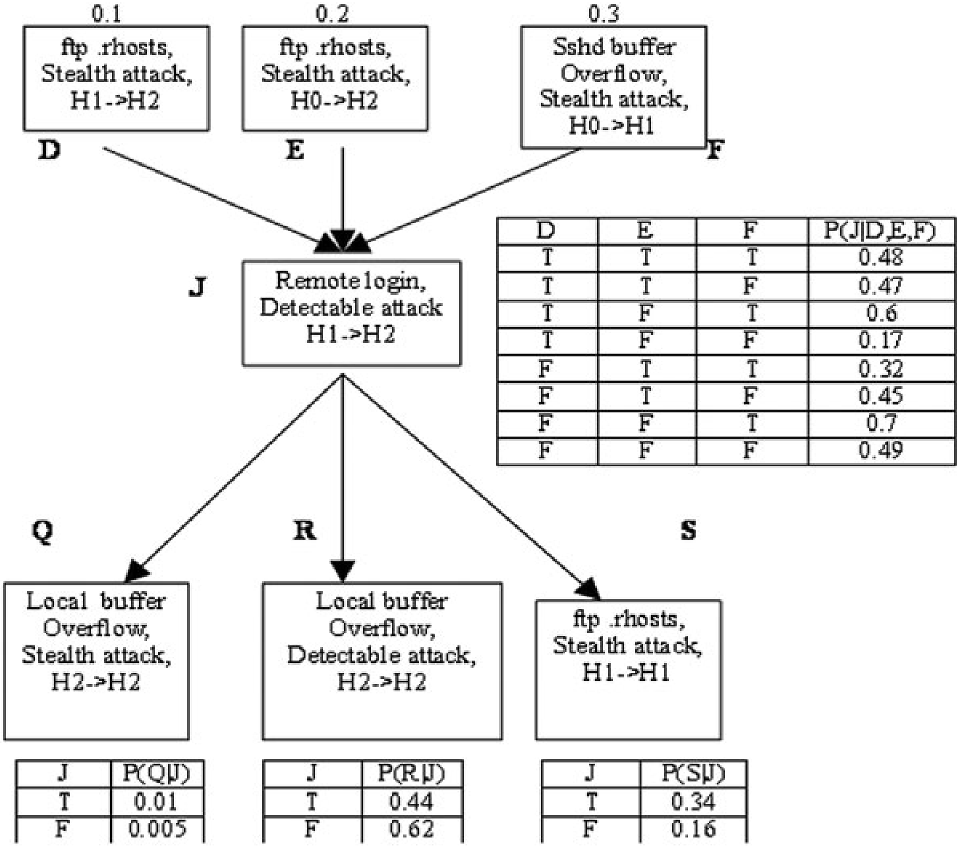


Fig. 8. Relation between risk, behavior, and network penetration.


0.1
ftp .rhosts, Stealth attack, H1->H2
D
0.2
ftp .rhosts, Stealth attack, H0->H2
E
0.3
Sshd buffer Overflow, Stealth attack, H0->H1
F
J
Remote login, Detectable attack H1->H2
Q
Local buffer Overflow, Stealth attack, H2->H2
R
Local buffer Overflow, Detectable attack, H2->H2
S
ftp .rhosts, Stealth attack, H1->H1

D	E	F	P(J|D,E,F)
T	T	T	0.48
T	T	F	0.47
T	F	T	0.6
T	F	F	0.17
F	T	T	0.32
F	T	F	0.45
F	F	T	0.7
F	F	F	0.49

J	P(Q|J)
T	0.01
F	0.005

J	P(R|J)
T	0.44
F	0.62

J	P(S|J)
T	0.34
F	0.16

Fig. 9. Modified attack graph of Figure 5 after patching the local buffer overflow vulnerability.

DOI:10.1002/sec

Table X. Probability of node Q (Figure 5) given its parents after patching local over flow vulnerability for the two profiles that are affected by the patching process.

Probability profile	*P*(Q) given J = yes	*P*(Q) given J = no
Criminals	0.71	0.83
Hackers	0.19	0.24
Liberals	0.01	0.005

reduce risk presented by individuals who fit this profile.

4. Sensitivity Analysis

Since the results for risk management are dependent on the initial values specified for the nodes, it is important to understand the impact of changes or errors in the input values on the outcome of the model. Let us take for example Figure 4 and the associated probabilities $P(Q|D) = 0.48$ and $P(D|Q) = 0.093$ which are dependent on the initial values of *P*(E) and *P*(F). What if the values of *P*(E) or *P*(F) are slightly different? These differences may either be due to some errors during the initialization or even some noise in the data sets used to derive the profiles.

A sensitivity analysis is used here to provide some answers for the question above. Notice that the goal of the sensitivity analysis is to obtain a better understanding of the model under variations of its input value; sensitivity analysis is not directly used for network risk management. Again, using Figure 4, we compute the percentage of change in the values of *P*(Q|D) and *P*(D|Q) as the values for *P*(E) and *P*(F) vary from 80 to 120%, that is, from −20 to 20%.

As can be seen from Figure 11, as the values of *P*(F) and *P*(E) are altered by changes ranging from −20% to 20% the values of *P*(Q|D) have a maximum change of only 1.25%. It can also be observed from Figure 11 that changes in *P*(E) have a larger impact on *P*(Q|D) than changes in *P*(F) as the slope for *P*(E) is larger when *P*(F) is kept constant. In general, it can be concluded that even a 20% error in the initialization of both values for *P*(E) and *P*(F) will not incur a large error in the computation of *P*(Q|D).

The computation of *P*(Q|D) shows the sensitivity for a top-down computation. To evaluate the results in a bottom-up sequence, we consider the computation of *P*(D|Q) as given by Equation 11. The results are plotted in Figure 12. As it can be observed, *P*(D|Q) is much more sensitive than *P*(Q|D) as the maximum change perceived in this value reaches almost 28%. The impact of *P*(E) is even more notable in this case while the impact of changes in *P*(F) is almost negligible when compared to changes in *P*(E). In summary, a change of *X*% in *P*(E) will incur a 1.25 *X*% change in *P*(Q|D).

The sensitivity analysis has shown that the model is not too sensitive and can properly accommodate for variations in the input values within the range of ±20% without major changes in the results. The

Table XI. Updated node values due to evidence at node R after patching local over flow vulnerability at node Q.

Profile	Before patching				After patching			
	P(A)	*P*(D)	*P*(J)	*P*(Q)	*P*(A)	*P*(D)	*P*(J)	*P*(Q)
Criminals	0.8002	0.76	0.797	0.734	0.8002	0.760	0.797	0.734
Hackers	0.599	0.541	0.439	0.451	0.599	0.541	0.436	0.218
Liberals	0.398	0.411	0.310	0.475	0.398	0.411	0.310	0.022

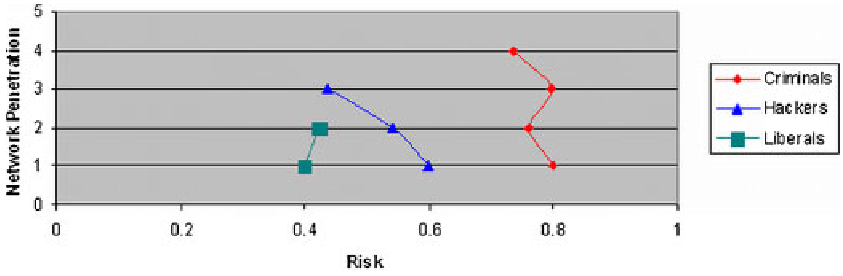


Fig. 10. Network penetration of all the profiles after patching the local overflow vulnerability.

DOI:10.1002/sec

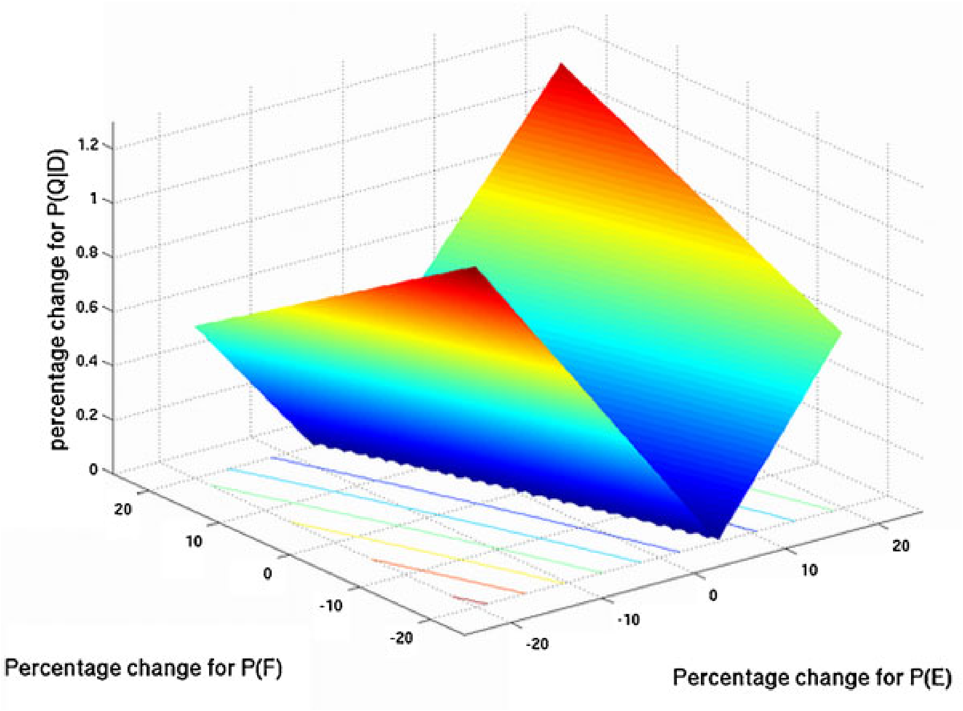


Fig. 11. Sensitivity of *P*(Q|D) for changes to *P*(F) and *P*(E).

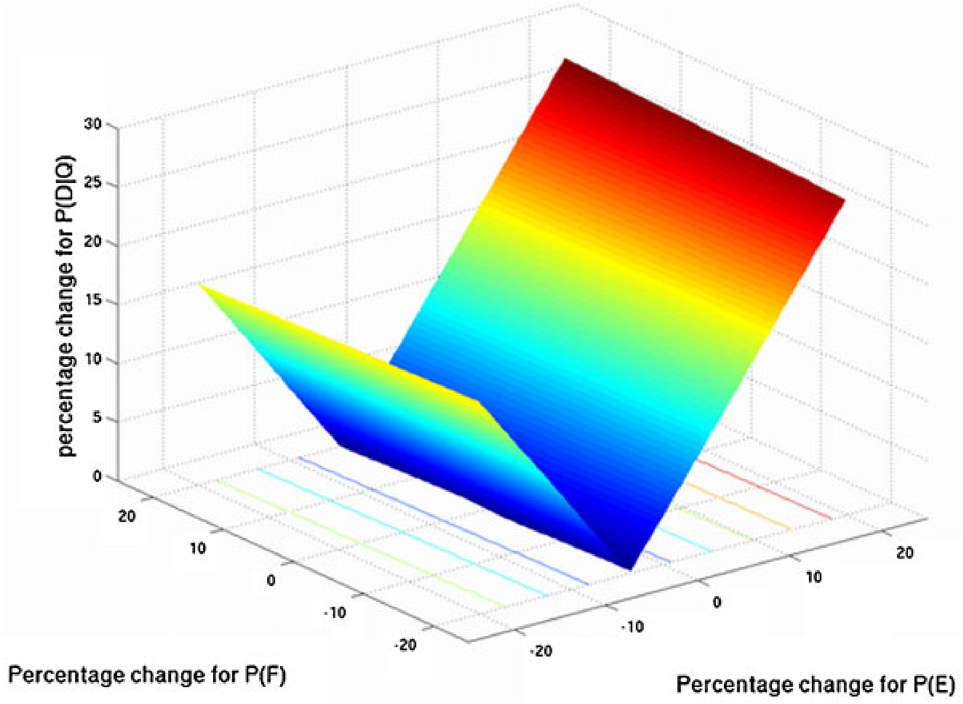


Fig. 12. Sensitivity of *P*(D|Q) for *P*(F) and *P*(E).

DOI:10.1002/sec

analysis has also showed that P(Q|D) is more sensitive than P(D|Q). However this is due more to the absolute values of the probabilities. That is, since values for P(Q|D) are around 0.48, percentage changes in the probability are less compared to percentage changes for P(D|Q) which values are around 0.093. In summary, the sensitivity analysis has allowed for an evaluation of the model showing its conformance to the expected behavior.

5. Conclusion

We *hypothesize* that there is a relation between sequence of network actions and attacker behavior, and that this relationship can be used for network risk analysis and management. Risk management requires that administrators quickly identify the seriousness of potential attacks by various types of attackers and install patches that effectively limit the level of penetration such attackers achieve. This paper describes a *five-step model* of detection and risk estimation that uses attack graphs and attacker behaviors. The creation of attack graphs, and through the use of Bayesian estimation, show which network vulnerabilities will probably lead to a successful network compromise. Finally, through the use of this model, we suggest that minimizing the risk to the network can be achieved by patching priority locations which can be revealed through optimizing the before and after risk probabilities (Table XI). Future work includes applying our method to real-world network configurations and testing the methodology on data collected during past attacks.

References

1. Desmond J. Checkmate IDS tries to anticipate hackers actions. www.esecurityplanet.com/prodser [12 June], 2003.
2. Jackson G, Checkmate intrusion protection system: evolution or revolution. *Psynapse Technologies*, January 1, 2003, available at http://cnscenter.future.co.kr/resource/security/ids/whitepaper.pdf
3. Richarte G. Modern intrusion practices. *CORE Security Technologies*, http://www1.corest.com/common/showdoc.php?idx=360&idxseccion=13&idxmen [July] 2003.
4. Yuill J, Wu SF, Gong F, Ming-Yuh H. Intrusion detection for an on-going attack. *RAID symposium*, 1999.
5. Chandler A. Changing definition of hackers in popular discourse. *International Journal of Sociology and Law* 1996; **24**(2): 229–252.
6. Jasanoff S. A sociology of hackers. *The Sociological Review* 1998; **46**(4): 757–780.
7. Dantu R, Kolan P. Survey of behavior profiles. http://secnet.csci.unt.edu/risk/, 2006.
8. Rowley I. Managing in an uncertain world: risk analysis and the bottom line. *IEEE Colloquium on Systems Engineering Contribution to Increased Profitability*, October 1989.
9. Kleen L. Malicious hackers: a framework for analysis and case study. *Ph.D. Thesis*, Air Force Institute of Technology, Ohio, 2001.
10. Foreward, by Bred Powell. Know Your Enemy Motives: The Motives and Psychology of the Black-hat Community. Honeynet Project (Whitepaper) http://www.honeynet.org/papers/motives/index.html
11. Loper K. The criminology of computer hackers: a qualitative and quantitative analysis Ph.D. Thesis, Michigan State University, 2000.
12. Rogers M. Running head: theories of crime and hacking. *MS Thesis*, University of Manitoba, 2003.
13. Rogers M. The psychology of hackers: A new taxonomy. Paper presented at the RSA World Security Conference, San Jose, California, Feb 1999.
14. Scheiner B. Attack trees: modeling security threats. *Dr. Dobb's Journal*, December 1999 http://www.ddj.com/184411129.
15. Sheyner O, Joshua HJ, Jha S, Lippmann R, Wing JM. Automated generation and analysis of attack graphs. *IEEE Symposium on Security and Privacy*, 2002.
16. Moore AP, Ellison RJ, Linger RC. Attack modeling for information security and survivability. *Technical Note*, CMU/SE1-2001-TN-001, March 2001.
17. Dantu R. Intrusion detection using attacker profiles. *Work in Progress, 19th Annual Computer Security Applications Conference(ACSAC)*, Las Vegas, 2003.
18. Dantu R, Loper K, Kolan P. Risk management using behavior based attack graphs. *IEEE International Conference on Information Technology (ITCC)*, April 2004.
19. Dantu R, Kolan P. Risk management using behavior based Bayesian networks. *Springer Verlag, Lecture Notes in Computer Science (LNCS)*, 2005.
20. Swiler LP, Phillips C, Ellis D, Chakerian S. Computer-attack graph generation tool. *IEEE Symposium on Security and Privacy*, 2001.
21. Castillo E, Gutierrez JM, Hadi AS. Sensitivity analysis in discrete Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernatics* 1997; **27**(4): 412–423.
22. WINBUGS—http://www.mrc-bsu.cam.ac.uk/bugs
23. HUGIN DEMO—http://www.HUGIN.com/
24. Ou X, Govindavajhala S, Appel AW. MulVAL: a logic-based network security analyzer. *14th Usenix Security Symposium*, August 2005.
25. Noel S, Jajodia S. Understanding complex network attack graphs through clustered adjacency matrices. *Annual Computer Security Applications Conference*, December 2005.
26. Noel S, Robertson E, Jajodia S. Correlating intrusion events and building attack scenarios through attack graph distances. *Proceedings of the 20th Annual Computer Security Applications Conference*, 2004.
27. Sarda1 K, Wijesekera D, Jajodia S. Implementing consistency checking in correlating attacks. *Lecture Notes in Computer Science*, 2004.

Authors' Biographies

Ram Dantu has 20 years of experience in the networking industry, where he worked for Cisco, Nortel, Alcatel, and Fujitsu and was responsible for advanced technology products from concept to delivery. For the last 5 years, he has been researching on the prevention of DoS and spam attacks

DOI:10.1002/sec

in VoIP networks. He is currently an assistant professor in the Department of Computer Science and Engineering, University of North Texas (UNT). He is the founding director of the Network Security Laboratory (NSL) at UNT, the objective of which is to study the problems and issues related to next-generation networks. Prior to UNT, he was a technology director at Netrake, where he was the architect of the redundancy mechanism for VoIP firewalls. He has cochaired three workshops on VoIP security. His additional experience includes being a technical director in IPMobile (acquired by Cisco), where he was instrumental in the wireless/IP product concept, architecture, design, and delivery. His research focus is on detecting spam, network security and next-generation networks. In addition to more then 70 research papers, he has authored several Requests For Comments (RFCs) related to Multiprotocol Label Switching (MPLS), SS7 over IP, and routing. Due to his innovative work, Cisco and Alcatel were granted a total of 12 patents; another eight are pending.

Prakash Kolan has 5 years of research experience in VoIP and network security. Currently, he works for Niksun Inc, a network security and surveillance company, as a researcher for Media Solutions. He has a Ph.D. degree from University of North Texas (UNT). He has authored several papers on voice spam filtering techniques, vulnerability analysis, and network risk management.

João Cangussu received the B.S. degree in computer science from the Federal University of Mato Grosso do Sul, Brazil, in 1990, the M.S. degree in computer science from the University of Sao Paulo, Sao Carlos, Brazil, in 1993, and the Ph.D. degree in computer science from Purdue University in 2002. He is currently an assistant professor in the Department of Computer Science, University of Texas, Dallas. His research interests include software process modeling and control, software testing, adaptive systems, and network security. He is a member of the IEEE and the ACM.

DOI:10.1002/sec