Comparison and Improvement of Metrics for Selecting Intrusion Response Measures against DoS Attacks [1]

Marko Jahnke, Christian Thul
Research Institute for Communication, Information Processing, and Ergonomics (FGAN-FKIE)
{jahnke|thul}@fgan.de

Peter Martini
University of Bonn
Institute of Computer Science
Departement IV
martini@cs.uni-bonn.de

Abstract: This contribution presents a comparison of metrics used in different approaches for selecting appropriate intrusion response measures in the case of attacks against computer systems and networks. Most of the work is focused on Denial-of-Service (DoS) attacks.

Besides an overview on the techniques and frameworks known from earlier and recent literature, an alternative approach is presented which models the effects of attacks and according response actions in a dynamic fashion, using directed graphs. Certain properties of the graphs are utilized to quantify different response metrics, closely aligned to the pragmatic view of a network security officer. Subsequently, the different metrics are compared and their advantages and disadvantages are discussed in the light of applicability in real-world networks.

1 Introduction

Attacks against computer systems and networks in their different characteristics are omnipresent and thus not surprising anymore. Almost every network that connects computers has been facing processes of reconnaissance, penetration, stealing or damaging information in the past, with more or less serious subsequent effects.

When an attack has been indicated by a monitoring system, network security officers need to select an appropriate response to the attack carefully. The way how to define this 'appropriateness' heavily depends on the properties and the deployment objective of the network and its components. There is only a small number of approaches selecting response mechanisms automatically; this is mainly caused by too many possibilities to damage a system rather than mitigating the effects of an attack.

This contribution compares earlier and recent approaches which select and apply response measures in order to mitigate effects of the attack. Most of the compared approaches make use of directed graphs in order to store and process structured information for different purposes. Some of them provide a methodology or a framework; one approach has been successfully implemented for a single narrow application scenario.

[1] Published in: A. Alkassar, J. Sieckmann (Eds.): Proc of the GI Sicherheit2008 Conference, Saarbrücken, Germany, Apr. 2008.

The rest of this paper is organized as follows: Section 2 presents several intrusion response metrics from practice and related research work. In section 3, an improved approach for using graphs in order to specify metrics for intrusion response measures is described. Section 4 discusses the major differences of the presented approaches in a table.

2 An Overview of Metrics for Automatic Response Selection

The careful selection of response mechanisms to attacks against computer networks has always been a challenging field of work. Different contributions dealt with cost models in the area of intrusion detection and response. A general taxonomy on existing work in the area of intrusion response – not only concerning automatic selection of responses – has been published by Stakhanova et al. in [SBW07b].

2.1 The Practitioner's View

Network security officers (NSOs) are usually equipped with more or less complex monitoring systems and applications, such as network management systems (NMS), intrusion detection systems (IDS), intrusion prevention systems (IPS) and additional tools like administrator consoles, up to complex threat management systems.

Conventionally, an NSO picks a selection of the available response measures together with the appropriate parameters and triggers it manually, at the console of the penetrated systems or even remotely over the network. When choosing the response measures and their parameters, NSOs often take the following factors into account:

- *Expected Response Success.* Clearly, the most important aspect is the expected success of a measure. Negative side effects (e.g. unwanted partial inavailability) need to be considered here. As long as a reaction does not likely have a positive effect (whatever this means in the according application scenario) on the network, it will not be chosen. This also holds for the response parameters.
- *Expected Response Error-Proneness.* The probability of failing when performing a response measure is also very important. Errors may occur in two different contexts: a) The diagnosis of the monitoring system might be incorrect. b) The application of a response might fail (e.g. due to missing access rights). So, in most cases, the alternative with the lowest severity of possible complications would be selected.
- *Expected Response Durability*. The expected duration of the response effects is probably an aspect that is less important than the other three mentioned above. If two alternative sets of responses promise comparable values for the other aspects, most likely the one with the longer expected durability will be chosen, i.e. the expected time period after which additional actions will become necessary for keeping the system healthy.
- *Expected Response Effort.* Another important aspect is the estimated effort (or costs) that is needed for performing response measures. If two sets of possible responses

have the same expected success, most probably the set will be selected, which is easier to apply.

Of course, there are more aspects to be considered by NSOs, but these strongly depend on the corresponding deployment scenario.

In many intrusion response systems (e.g. Snort Inline [Sno07]), the reaction itself is coded in the detection signature that has been specified prior to the deployment of the system. Thus, this can simply be viewed as a suggestion of the signature writer. However, in these cases, there is no dynamic on-line estimation of the response involved.

2.2 Early Theoretic Work

In her text book [Den99], Denning stated that cost analysis in the area of IT security – and risk analysis in general – simply cannot be considered an exact science, since in many cases, relevant values cannot be quantified at all. Northcutt describes in [Nor99] the (informal) process of risk analysis in IT systems and defines the value of resources by their *criticality* as well as their *lethality*. Lee et al. [LFM+02] identified different operational costs as metrics for selecting intrusion response measures. Starting with a taxonomy of attacks that have been given by a reference dataset, empirical costs for the attack *damage* and *reactions* have been defined.

2.3 Toth & Kruegel

Toth & Kruegel [TC02] looked at the effects of a reaction in a network model that considers resources (applications/services), users, the network topology and access control mechanisms (firewall rules). In general, for all mentioned model components, a capability value is defined. Also, the respective inter-depedencies of resources have been modeled; *dependency trees* express these relationships. By the decrease of resource capability, the costs of response measures are estimated.

In this approach, the *capability* $c(r)$ reflects the overall ability of a resource r to fulfill its function/duty, whereas the *penalty* is an abstract measure of loss when a resource r is no longer available. The *penalty costs* $p(r)$ need to be re-computed after $c(r)$ was updated according to the proposed depth-first-search (DFS) based update algorithm discussed in their paper:

$$p(r) := (1 - c(r)) \cdot penalty$$

The *overall penalty* p is the sum of all *penalty costs*. The response action with the smallest value of p is chosen for deployment.

2.4 Balepin et al.

Balepin et al. [BMR+03] extended the idea of representing services and their inter-dependencies in a graph for selecting responses through creating a resource type hierarchy, so that every service type has common response measures associated with it. Response sequences need to be optimal for each service node, i.e every response step needs to produce maximum benefit at minimum costs.

The author also proposed the *costs* of priority resources as base metric for response choice. In their *system map*, only priority nodes – representing the important system resources – have a cost value of their own. Cost values are assigned to other nodes based upon the fact, that priority nodes depend on them. The cost values are set by the NSO on creation of the system map. Subsequently, the following values are computed: *Intrusion Damage* (sum of all cost values of the affected nodes), *Response Benefit* (sum of all cost values of the nodes, that are restored to safe state by the response), and *Response Costs* (sum of all cost values of the nodes, that are negatively affected by the response).

In case where the current state of the system after an attack is uncertain, the following matrices are used to aid the choice of responses:

(a) Response Benefits

	Π_1	Π_2	$\cdots$	Π_n
A_1	a_{11}	a_{12}	$\cdots$	a_{1n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_N	a_{N1}	a_{N2}	$\cdots$	a_{Nn}
Q	q_1	q_2	$\cdots$	q_n

(b) Response Risks

	Π_1	Π_2	$\cdots$	Π_n
A_1	r_{11}	r_{12}	$\cdots$	r_{1n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_N	r_{N1}	r_{N2}	$\cdots$	r_{Nn}

In the benefits matrix, Π_i are the possible system states, A_j are the alternative responses, q_i are the probabilities for the system being in state Π_i, $a_{i,j}$ are the response benefits given by $a_{i,j} := -c_j - (-\varepsilon_{ij})^\gamma \cdot B_i$, where B_i is the potential damage of state Π_i, c_j is the response cost of response A_j, ε_{ij} is its effectiveness in state Π_i, and γ is 0, if $\varepsilon_{ij} = 0$ and 1 otherwise. In the risk matrix, the r_{ij} values represent the risk of losing (i.e. making a bad response choice), when choosing response A_j in state Π_i. This risk is defined as $r_{ij} := m_i - a_{ij}$, where $m_i := \max_j a_{ij}$. The selection of the response itself may be based on different criteria. As an example, the *Savage Criterion* avoids high-risk decisions by estimating the efficiency of a response as $W = \min_i \max_j r_{ij}$.

2.5 ADEPTS

With ADEPTS [WFB+07], a more complex framework for determining automated responses against attacks was proposed. It is based on two types of graphs: A service graph (*S-Graph*), that expresses inter-dependencies between available services, and an attack graph (*I-Graph*), that represents possible attack states and their probabilities. While the S-Graph is used only during the initial creation of the I-Graph, the I-Graph itself is used

for selection of possible response deployment points. The responses are selected based on their effectiveness during previous applications in the past.

For the selection of response deployment points ADEPTS uses the *Compromised Confidence Index (*CCI*)* as its primary metric. The CCI expresses the probability, that the goal represented by the according node in the I-Graph is currently achieved by the attacker. It is initialised with the $alert\_confidence$, provided by the detector that has set up an according alert. This $alert\_confidence$ is moderated by the result of a false-positive estimation in the following way:

$$alert\_confidence := alert\_confidence \cdot (1 - false\_alert\_probability)$$

Then the CCI of all nodes is updated by propagating the values through the graph. Based on the CCI values, the nodes for deployment of the responses are chosen and put into the *response set* with potential candidates for application. [1]

The choice of the responses themself is based upon the *Response Index* RI, which consists of the *Disruptiveness Index* DI and the *Effectiveness Index* EI by $RI := a \cdot EI - b \cdot DI$. DI is set a priori by the NSO, while EI is updated on runtime after deployment of a response. Therefore the system checks if any edge that was affected by the response can still be used to reach a node in the current *response set*. If so this is an indication that this response failed, and its EI is decreased. The amount by which the EI is lowered depends on the edgetype. For edges associated with a logical AND operation (AND edges), the EI is decreased by a fixed value assigned to each edge. For OR and Quorum edges, this fixed value is modified in proportion to the CCI of the nodes. In case a response times out, or is manualy shut down by the NSO, the EI is increased by a predefined percentage, thus reflecting the intuition, that the response has proven to be effective.

The response with the highest RI is chosen for deployment. So ADEPTS also choses responces based upon benefit(effectiveness) and risk(disruptiveness).

2.6 Mirkovic et al.

Two recent papers from Mirkovic et al. [MRF+06], [MHW07] propose an relatively pragmatic way to define metrics for characterizing DoS effects on the user of a network. The authors suggest to use these metrics also for selecting appropriate response measures, but no specific implementation details are given. However, they present a lot of practical measurement results and also discuss ways of implementing measurement methods for simulation environments.

The main metric used for evaluation of DoS impact is the *percentage of failed transactions* (in short pft), within a conversation. A *conversation* is defined as the set of all network packets exchanged between a client and a server with a goal to provide a specific service to the client, at a given time. A *transaction* is defined as the part of a conversation that represents a task, whose completion is meaningful to a user, such as browsing to the next

[1] Alternatively ADEPTS is able to recognize attack sequences that have been applied before. In this case the System immediately evaluates responses attached to that sequence.

link of a website. A transaction can fail due to exceeding the predefined thresholds of one or more of its parameters, such as:

- *One-Way Delay* (e.g. for chat, multimedia traffic, games),
- *Request-Response-Delay* (e.g. for email, web, ftp),
- *Packet Loss* and *Jitter* (e.g. for multimedia traffic).

Using the information about the transactions, different representations are derived for providing further information, such as pft-histogram, an abstract level for the service degradation $DoSLevel := \sum_k pft_k \cdot w_k$, or the severity of the attack, given by $QoSDegrade := \frac{(d-t)}{t}$, where d is the value of the parameter that exceeded its threshold t.

Although Mirkovic's paper is not focused on selecting response measures, the authors propose to compare the DoS measurement results before and after deployment of a response in order to determine its value.

2.7 Stakhanova et al.

Recent work by Stakhanova et al. [SBW07] suggests a cost-sensitive model for preemtive intrusion response systems. This model compares the costs of deploying a response to the costs of damage caused by a non-responded attack. Additionally, a methodology for adapting responses in a changed environment through an evaluation of previosly applied response measures is discussed.

The method proposed in this approach uses the very simple metrics *Response Cost* RC and *Damage Cost* DC, that reflect the effect of either the response or the attack on the system and have to be set up by the NSO and updated over time. As in other presented approaches, a high level of expertise is needed to set those metrics to suitabl vealues. For a first response step, the set of applicable measures is selected. This is the set of responses, for which following condition holds:

$$DC \cdot confidenceLevel > RC$$

where the $confidenceLevel$ is the probability, that the attack, that DC belongs to, is actually taking place.

In a second step, the most appropriate element of the applicable measure set is chosen, based on two metrics, namely the *Success Factor* SF and the *Risk Factor* RF. The earlier is the percentage of times, that this response succeeded in the past, whereas the latter represents the negative impact, that this response has on the system and legitimate users.

Intuitivly, the response providing maximum benefit at the lowest risk is chosen. This is done by choosing response r_s with the maximum Expected Value $EV(r_s)$ for the given attack sequence S, given by

$$EV(r_s) := (Pr_{succ}(S) \cdot SF) + (Pr_{risk}(S) \cdot (-RF))$$

$Pr_{succ}(S)$ is the probability that attack-sequence S occurs and $Pr_{risk}(S) = 1 - Pr_{succ}(S)$. The *Success Factor* is adaptive; it is increased by one if the response succeeds in stopping an attack and it is decreased by one if it fails.

Thus, this approach also takes benefit and risk of a response into account for selection of responses.

3 An alternative Graph Model for Response Metrics

Although there have been numerous approaches identified, it became necessary to develop a new approach, due to different reasons: Firstly, there was no approach identified, where all practically relevant metrics are honored. Secondly, in some cases, the value of a response depends only on static values, so that the dynamic network state might not be sufficiently considered. Thirdly, some approaches need to identify the goal of an attacker and his current progress to achieve this goal. These assumptions are too restrictive for our application scenarios.

This section outlines an alternative approach, that makes use of different kinds of directed graphs in order to use certain graph properties for determinig metrics for responses. It is described in detail in our earlier work [JTM07]. However, progress has been made which is presented in this section.

3.1 Resources and Availailabilities

As already suggested by Toth & Kruegel [TC02], this model is based on properties of *resources*. The set of resources is furtheron denoted as $\mathcal{R}$. Resources can either be *service instances* (instances of a service provided by hardware, operating systems, applications or network services) or *users*. The respective sets are furtheron denoted as $\mathcal{S}$ and $\mathcal{U}$ with $\mathcal{R} = \mathcal{S} \cup \mathcal{U}$ and $\mathcal{S} \cap \mathcal{U} = \emptyset$

Concerning availability, we observe different kinds of dependencies between resources. The users depend on applications and services within the network to conduct a certain mission – otherwise the network would be completely useless for them. On the other hand, applications often rely on other applications and services, such as many network communication systems are depending on the availability of directory services and of the network transport service itself.

We assume that every resource $r \in \mathcal{R}$ of a system to be secured has a certain *availability*, expressed as a value $A(r) \in [0, 1]$. E.g., if a router is able to handle only 10% of the traffic it was designed for, its current availability is denoted as 0.1. Intuitively, there is a lower bound for totally inoperable service instances and an upper bound for instances which operate with full capabilities (i.e. operate fully as designed).

The current availability value of a resource is a result of two indepentent factors: its internal state and the values of other resources it depends on. Thus, we separate the *intrinsic* availability value $A_I(r) \in [0, 1]$ from the *propagated* availability value $A_P(r) \in [0, 1]$, so

that they are statistically independent from each other. We define the resulting availability as

$$A(r) = A_I(r) \cdot A_P(r)$$

to every resource $r \in \mathcal{R}$. Note that $A(r)$ is something that is measurable by a monitoring system, whereas $A_I(r)$ will be changeable, e.g. when implementing a response action.

3.2 The Dependency Graph

A *dependency graph* of a system with the set of resources $\mathcal{R}$ is a directed graph $\hat{G} = (\mathcal{R}, \hat{E})$ with $\hat{E} \subseteq (\mathcal{S} \cup \mathcal{U} \times \mathcal{S})$. $\hat{G}$ contains an edge (r, s) whenever a resource r depends on the service instance s concerning its availability. In other terms, r needs *accessibility* to s. The edges in $\hat{E}$ are labeled with the subjective weight $w(r, s)$ of resource s for r.

When modelling multiple systems in a network, we observe different classes of dependencies between the resources. A set of frequently seen dependency types include *mandatory*, *alternative*, *combined*, and m-*out-of-*n. An additional property of a dependency relationship is the fact that the dependency expresses the need to have either direct or *indirect* access to another resource, i.e. to met this requirement, it is also possible to use other resources as mediators. Fig. 1 depicts an example dependency graph of a typical e-commerce scenario. Note that indirect dependencies are marked with dashed arrows.

The dependency graph is used as an ideal map of the network. It reflects the requirements for a non-derogated state of all components. For a more detailed discussion of dependency types and according examples, please refer to [JTM07].

3.3 The Accessibility Graph

An *accessibility graph* of a system with the set of resources $\mathcal{R}$ is a directed graph $G = (\mathcal{R}, E)$ with $E \subseteq (\mathcal{S} \cup \mathcal{U} \times \mathcal{S})$. G contains an edge (r, s) whenever a resource r has direct access to s. For an edge $(r, s) \in E$, a value $A(s) > 0$ indicates that r has direct accessibility to s. If $(r, s) \notin E$ or $A(s) = 0$, there is no direct access possible from r to s. If there is a path $(r_1, r_2), \ldots, (r_{n-1}, r_n)$ with $(r_i, r_{i+1}) \in E \wedge A(r_i) > 0$ and $r_1 \neq \ldots \neq r_n \in \mathcal{R}$, then r_n is indirectly accessible for r_1.

The nodes $r \in \mathcal{R}$ of the accessibility graph G are labeled with their availability $A(r)$, and the edges $(r, s) \in E$ are labeled with the subjective weight $w(r, s)$ of node s for r (i.e. its relative importantness compared to the other nodes r depends on).

On one hand, the accessibility graph is used as a simplified map of the real-world network to present current status information as given by a monitoring system. On the other hand, it can be used to estimate effects of a response by cloning the current state graph, adjusting availability values (e.g. implementing access control, shutting down or throttling services) and deleting or moving edges (e.g. reconfiguring ports, activating backup services).

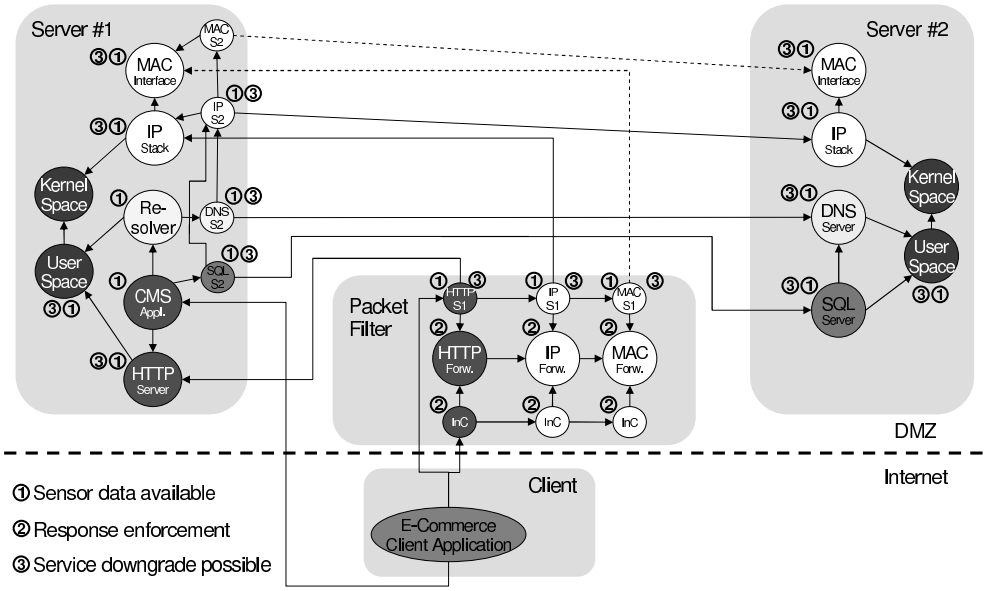

Figure 1: Example Dependency Graph of an e-commerce scenario with two DMZ hosts, different services and a packet filter for access control.

3.4 Determining the Resource and Overall Availability

Each time the monitoring system indicates a changed availability of a resource, the availability of resources which directly or indirectly depend on the changed one, need to update their values immediately. For doing so, the types of dependencies of each node need to be honored. For the above mentioned dependency types, we have suggested a number of numerical operations that reflect the types (see above).

To quantify the overall availability of the network in the light of supporting users when conducting a mission, a corresponding definition is needed. Intuitively, defining the overall availability as the availability of the service instances that are immediately needed by the users is useful. So we define the overall availability as

$$A(G) := \frac{\sum_{u \in \mathcal{U}} A(u) \cdot m(u)}{\sum_{u \in \mathcal{U}} m(u)}$$

where $m(u) \in [0, 1]$ is the *relative importance* of the user $u \in \mathcal{U}$ for the common mission, that needs either to be defined beforehand or to be determined adaptively. This reflects the *user & mission based approach* for our metrics.

For each availability changed at some vertex in the graph, the values need to be propagated to all affected resources in the network. For achieving this, an algorithm is needed, that captures all affected nodes, terminates even in presence of cycles – which cannot be ultimatively precluded – and yields stable results. A possibility to fulfil most of the requirements is based on an inverse BFS in a directed graph. This is comparable to algorithms used in software reliability analysis (e.g. [YA02], [YCA04]). This algorithm traverses the accessibility graph, beginning at the nodes with changed availabilities and updates the values of their children. This behaviour guarantees the completeness and the termination. Cyclic dependencies are ignored during the traversal process.

3.5 Metrics Definitions

To be able to assess properties of a reaction, an accessibility graph for the system state prior and after the application needs to be generated. Assuming that G is the graph we obtain before the reaction, and G' after a response Θ, the *success* of the reaction can be intuitively defined as the change of availability after the reaction against an attack:

$$\delta_1(G, G') := A(G') - A(G)$$

Obviously, this metric may also have negative values, since a wrong selection of response measures might also damage the network rather than having a positive effect.

As opposite to our first approach as described in [JTM07], our revised *costs* metric honors both the effort for implementing the response and potential intermediately lowered availabilities (which – to our best knowledge – other approaches do not consider explicitly). We define it as

$$\delta_2(G, G') := \sum_{i=0}^{k} T(\theta_i) \cdot (1 - A(G_i))$$

where $\theta_0, \ldots \theta_k$ are the elementary actions to form response Θ, G_i is the resulting graph after the application of step θ_i, and $T(\theta_i)$ is the time needed for the application of step θ_i. Using this definition, the most appropriate response is the one that promises the highest increase of availability while simultaneously implies the lowest cost value as defined above. Thus, the practical aspects of the response selection process are well considered so far.

Two additional metrics, *durability* and *error-proneness* are currently under development. The first one might be modeled as time ranges after which a response related graph modification is automatically redrawn. The latter might be expressed as an appropriate graph distance measure between the current accessibility graph and the graph from the last known stable system state.

3.6 Experimental Validation

For validating our dependency structures, we implemented an experimental e-commerce setup with a sample web shop application in our lab. Several active probing tests for

measuring the quality of services with respect to success and delay of transactions have been installed, as indicated by the marks (1) in Fig. 1. At different locations – marked with (2) – we have been able to implement responses, e.g. by adjusting access control rules or by reconfiguring services. By downgrading the availability of certain services (marked with (3)), we extracted the respective dependencies and their weights by fitting a linear regression model to the normalized availability values.

We noticed that most of the affected resource dependencies have piecewise linear characteristics. In cases, where internal timeout mechanisms are involved inside a service instance $r \in \mathcal{S}$, we observed a lower bound $a_{min}(r, s)$ for $(r, s) \in E$, such that $A(s) < a_{min}(r, s) \Rightarrow A(r) = c$. If $c = 0$, then the timeout renders the service unavailable, whereas if $c > 0$, a fallback information (cached value) is used, which might lead to other serious conseqences if the information is outdated.

Finally, the extracted weights have been transferred to a software tool for estimating the overall availability values. Currently, we are examining different responses in order to compare the measured values with estimation results.

4 Comparison of the Approaches

In the following table, the identified response selection approaches are compared with respect to availability as the main security objective and to the best of our knowledge. Firstly, they are examined concerning the four practically relevant metrics from sect. 2.1. After this, properties of the data structures algorithms and implementation maturity are compared to each other.

5 Conclusion and Further Work

This contribution has presented various examples for metrics which are used for determining the most appropriate response measure to detected attacks against computer systems and networks, both from practice and research. An improved approach for specifying metrics using directed graphs has been proposed. This approach was subsequently compared with existing metrices.

Many presented approaches deploy multiple metrics concurrently for expressing properties of response measures. In most cases, response effectiveness and risk are considered the most important factors which is in fact not surprising, since these correspond to common best practices. Some of the approaches honor additional metrics, e.g. the deployment costs for response measures. Their main difference is the way how the metric values are actually determined. Some approaches are solely relying on experiences from the past (e.g. previous applications of a response), whereas only two contributions incorporate actually measured values (e.g. degree of service quality).

[1] A set of pre-determined dependency weights (e.g. in software distribution package databases) might be used instead of re-evaluating services before deployment.

Criteria	[TC02]	[BMR+03]	[WFB+07]	[MHW07]	[SBW07]	[JTM07]
Response Effectiveness/Success	(Abstract) cabapilities	Response Benefit	Effectiveness index (EI)	Percentage of failed transactions (PFT)	Success factor (SF)	Availability metric
Response Risk/Damage	Penalties for damaged resources (by the response)	Response costs for non-available priority resources	Disruptiveness index (DI)	(not applicable)	Response and damage costs	(Revised) response cost metric, error-proneness metric (t.b.d.)
Response Deployment Costs/Effort	None	None	None	(not applicable)	None	Response cost metric
Response Durability	None	None	None	(not applicable)	None	Durability mectric (t.b.d.)
Model Dynamics	Resource capabilities	Intrusion damage	Probability of attack states, compromised confidence, response effectiveness	User-perceived resource availability	Probab. of attacks, success factors	Resource accessibilities, response effects
Usage of graph structures	Resource dependencies	Resource dependencies	Attack states and transitions, network map	(not applicable)	Attack states and transitions	Resource dependencies, accessibilities, response effects
Principle of update algorithms	DFS updates capability	(not applicable)	BFS updates compromise confidence	(not applicable)	(not applicable)	BFS updates availability
Implementation maturity	Concept, data structures, algorithm implem., exper. results	Concept, data structures, algorithm implem.	Fully implemented and validated	Concept, implementation, exper. results	Concept, data structures, algorithm implem., exper. results	Concept, data structures, algorithm implem., exper. results
Approach universality	General computer networks	General computer networks	Static network scenario	General computer networks	Static network scenario	General computer networks
Required a-priori knowledge	Network resource map, response effects	Network resource map, attack effects, response effects	Network resource map, vulnerabilities, attack steps and goals, attack state traversal probab., response effects	User view access to services	Attack state probabilities, risk values, damage costs, resource costs	Network resource map incl. dependency weights[1], response effects, response deployment costs

For the proposed improved graph based metrics – which is work-in-progress – some practical advantages have been revealed. Firstly, the graph structures meet the intuition of

a network security administrator and thus may provide additional inside information on propagated attack effects in the network. Secondly, the data structures may be easily adjusted by human experts, maybe with additional support of network management systems in order to reflect changes in the structure of the network resources.

Although there have been numerous approaches identified, and some of them have even proved their applicability in specific deployment environments, research is still far from delivering a general solution for selecting appropriate responses against attacks. Currently, we conduct quantitative comparisons of the graph based approach with the DoS measurement metrics in e-commerce setups and mobile adhoc networks (MANETs). Our future work includes graph based definitions of the two pending metrics 'error-proneness' and 'durability', examination of ways to extend the approach to other security objectives than availability, and real-world deployment experiments in different scenarios.

References

[SBW07b] N. Stakhanova, S. Basu, and J. Wong. A Taxonomy of Intrusion Response Systems. International Journal of Information and Computer Security Vol. 1(1), pp. 169–184, 2007.

[Sno07] The Snort Inline Project Team. Snort Inline Project Homepage. Online accessible at `http://snort-inline.sourceforge.net/`, accessed 2007.

[Den99] D. Denning. Information Warfare and Security. Addison-Wesley, 1999.

[Nor99] S. Northcutt. Intrusion Detection: An Analyst's Handbook. New Riders Publishing, 1999.

[LFM+02] W. Lee, W. Fan, M. Miller, and S. Stolfo. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. Journal of Computer Security Vol. 10, pp. 5–22, 2002.

[TC02] T. Toth and C. Kruegel. Evaluating the impact of automated intrusion response mechanisms. In: Proc. of the 18th Computer Security Applications Conference (ACSAC'02), pp. 301–310, Las Vegas, NV, USA, 2002.

[BMR+03] I. Balepin, S. Maltsev, J. Rowe, and K. Levitt. Using specification-based intrusion detection for automated response. In: Proc. of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003), 2003.

[WFB+07] Y. Wu, B. Foo, Y. Mao, S. Bagchi, and E. Spafford. Automated adaptive intrusion containment in systems of interacting services. Computer Networks Vol. 51(5), pp. 1334–1360, 2007.

[MRF+06] J. Mirkovic, P. Reiher, S. Fahmy, R. Thomas, A. Hussain, S. Schwab, and C. Ko. Measuring Denial of Service. In: Proceedings of the 2nd ACM workshop on Quality of protection, 2006

[MHW07] J. Mirkovic, A. Hussain, B. Wilson, S. Fahmy, P. Reiher, R. Thomas, W. Yao, and S. Schwab. Towards User-Centric Metrics for Denial-Of-Service Measurement. In: Proc. of the Workshop on Experimental Computer Science, June 2007

[SBW07] N. Stakhanova, S. Basu, and J. Wong. A Cost-Sensitive Model for Preemtive Intrusion Response Systems. In: Proc of the The IEEE International Conference on Advanced Information Networking and Applications, Niagara Falls, Canada, 2007.

[JTM07] M. Jahnke, C. Thul, and P. Martini. Graph based Metrics for Intrusion Response in Computer Networks. In: Proc. of the 3rd LCN Workshop on Network Security. Held in conjunction with the 32nd conference on IEEE Local Computer Network Conference, Dublin, Ireland, 2007.

[YA02] S. Yacoub and H. Ammar. A Methodology for Architectural-Level Reliability Risk Analysis. IEEE Transactions on Software Engineering, Vol. 28(6), 2002.

[YCA04] S. Yacoub, B. Cucik, and H. Ammar. A Scenario-Based Reliability Analysis Approach for Component-Based Software. IEEE Transactions on Reliability, Vol. 53(4), 2004.