Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

MulVAL Extensions

For Dynamic Asset Protection

Eugen Bacic, Michael Froh and Glen Henderson

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Defence R&D Canada – Ottawa
CONTRACT REPORT
DRDC Ottawa CR 2006-251
April 2006

Canada

MulVAL Extensions

For Dynamic Asset Protection

Eugen Bacic
Cinnabar, a Division of Bell Security Solutions Inc.

Michael Froh
RatworX Inc.

Glen Henderson
Cinnabar, a Division of Bell Security Solutions Inc.

Prepared by:

Bell Security Solutions Inc. (Cinnabar Networks Inc.)
265 Carling Avenue, Suite 200
Ottawa, ON, KIS 2E1

Contract number: W7714-5-3247

Contract Scientific Authority: C. Burrell (613) 993-9963

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Defence R&D Canada – Ottawa

Contract Report

DRDC Ottawa CR 2006-251

April 2006

© Her Majesty the Queen as represented by the Minister of National Defence, 2006

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2006

Abstract

This paper documents research into extensions to the Multihost, Multistage Vulnerability Analysis (MulVAL) framework to support DRDC efforts to develop a feasible abstraction in the area of defensive posture technology. The results presented in this paper demonstrate that the MulVAL model is extensible and can be enhanced to include additional data representation and analysis features to tailor the model to meet the need of the DND defence community. The extensions evaluated in this effort have been shown to be both technically valid given the capabilities of logic-based programming and appropriate given the current model data representations. The primary extensions researched as part of this work are: improved representation of network path constructs and assignment of value to data assets in the model. This paper documents a substantial degree of progress in the development of each of the proposed MulVAL extensions.

Résumé

Le présent document explique en détail les recherches menées sur les extensions du cadre d’analyse de vulnérabilité multiutilisateur et échelonnée (MulVAL) à l’appui des démarches de RDDC qui visent à développer une abstraction réalisable dans le domaine de la technologie de position défensive. Les résultats présentés dans ce document démontrent que le modèle MulVAL est extensible et qu’il peut être élargi pour englober d’autres fonctions de représentation des données et d’analyse afin de le personnaliser pour qu’il réponde aux besoins de la collectivité militaire du MDN. Les extensions évaluées dans le cadre de l’initiative se sont révélées valides sur le plan technique compte tenu des capacités de la programmation à base logique et appropriées compte tenu du modèle courant de représentations des données. Les principales extensions qui ont fait l’objet de recherches dans le cadre du projet sont les suivantes : représentation améliorée des constructions de chemins de réseau et attribution de valeurs aux actifs de données dans le modèle. Ce document fait état du degré élevé de progrès accompli dans le développement de chacune des extensions MulVAL proposées.

This page intentionally left blank.

Executive summary

In 2005, Defence Research and Development Canada (DRDC) initiated an effort to develop a feasible abstraction in the area of defensive posture technology. A defensive posture technology would assess the defensive state of a network based on the network topology and configuration, known vulnerabilities, and operational requirements. The defensive posture abstraction was intended to provide the necessary framework for the creation, based on currently available technology, of a modelling system for dynamic risk management and asset protection.

This effort resulted in the creation of a white paper entitled: *Dynamic Asset Protection & Risk Management Abstraction Study*. Multihost, multistage Vulnerability Analysis (MulVAL), a research project at Princeton University, was identified in this report as a promising area of research. MulVAL is an end-to-end framework and reasoning system that conducts complex attack vector analysis on a network. Extensions to MulVAL that would be beneficial in furthering work at DRDC on dynamic network defensive posture were identified in the white paper.

Further investigation revealed several other projects that to varying degrees meet the requirements of a defensive posture technology. In particular, a commercial product called Skybox Security and an AI-based project called CycSecure were identified as interesting and relatively mature projects, which deserve closer consideration as candidates for a defensive posture technology to meet the needs of the defence community.

The research in this paper was sponsored to examine extensions to the MulVAL data model and to provide an updated critique of how well these extensions satisfy the known requirements of a dynamic asset protection solution. A critique of the Skybox Security and CycSecure solutions, with respect to the requirements of dynamic asset protection, was also deemed in scope for this effort.

The results in this paper demonstrate that the MulVAL model is extensible and can be enhanced to include additional data representation and analysis features to tailor the model to a specific problem environment. Specifically, the extensions which were proposed in the *Dynamic Asset Protection & Risk Management Abstraction Study* have been shown to be both technically valid given the capabilities of logic-based programming, but also appropriate given the current MulVAL model data representations. This paper documents a substantial degree of progress in the development of each of the proposed MulVAL extensions, as detailed below.

a. The network MulVAL extension demonstrates that it is possible to extend the simplistic host access control list, currently defined in the existing MulVAL model, with a more intuitive approach that more appropriately reflects real-network design.

b. The asset value extension demonstrates that it is possible to improve existing predicates to include additional characteristics for improved representation of the environment being modelled. Given the ability to assign a valuation to data, the tools are in place to prioritize the attack paths which are established though a MulVAL analysis.

This paper proposes that additional effort to refine the MulVAL extensions will lead to a complete set of valid and appropriate enhancements to the model. Specifically, it is proposed that the following tasks be undertaken to progress this effort.

a. Further enhance the attack path extension to determine how the different attack difficulty assessments / valuation methods should be combined to determine an overall difficulty value for the attack pattern;

b. Perform more data instantiation for all proposed model extensions to prove the validation of the approach (in the spirit of the work that was done for the network MulVAL extension); and

c. Assess the need for and identify any additional extensions.

Bacic, E., Henderson, G., Froh, M. 2006. MulVAL Extensions for Dynamic Asset Protection. DRDC Ottawa CR 2006-251 Defence R&D Canada – Ottawa.

Sommaire

En 2005, Recherche et développement pour la défense Canada (RDDC) a amorcé une initiative en vue de développer une abstraction réalisable dans le domaine de la technologie de position défensive. Ce type de technologie évaluerait l'état défensif d'un réseau en fonction de la topologie et de la configuration de ce dernier, des vulnérabilités connues et des besoins opérationnels. L'abstraction de position défensive visait à fournir le cadre nécessaire à l'élaboration, en fonction de la technologie actuellement offerte, d'un système de modélisation pour gérer les risques et protéger les biens de manière dynamique.

L'initiative a entraîné la rédaction d'un livre blanc intitulé : *Dynamic Asset Protection & Risk Management Abstraction Study*. Le projet d'analyse des vulnérabilités multiutilisateur et échelonnée (MulVAL), un projet de recherche de l'Université de Princeton, a été désigné dans le rapport comme étant un domaine de recherche prometteur. Le modèle MulVAL est un cadre de bout en bout et un système de raisonnement qui procède à des analyses complexes de vecteurs d'attaque dans un réseau. Les extensions du modèle MulVAL qui permettraient de faire avancer à RDDC les travaux sur la position défensive dynamique dans les réseaux ont été relevées dans le livre blanc.

Des études plus poussées ont révélé l'existence de plusieurs autres projets qui, à des niveaux différents, satisfont aux exigences en matière de technologie de position défensive. Plus particulièrement, un produit du commerce appelé Skybox Security et un projet d'IA appelé CycSecure ont été désignés comme étant des projets intéressants et relativement bien développés dont on devrait davantage tenir compte comme acteurs possibles d'une technologie de position défensive afin de répondre aux besoins de la collectivité militaire.

La recherche menée aux fins ce document a été organisée pour nous permettre d'examiner les extensions au modèle de données MulVAL et de formuler une critique à jour en ce qui a trait à la capacité de ces extensions à satisfaire aux besoins connus en matière de solution de protection de données dynamique. Une critique des solutions Skybox Security et CycSecure, à l'égard des exigences en matière de protection de biens dynamique, a aussi été considérée comme faisant partie du cadre de l'initiative.

Les résultats contenus dans le document démontrent que le modèle MulVAL est extensible et qu'il peut être élargi pour englober d'autres fonctions de représentation de données et d'analyse afin de l'adapter à un environnement de problème spécifique. Plus particulièrement, les extensions proposées dans le document *Dynamic Asset Protection & Risk Management Abstraction Study* se sont révélées valides sur le plan technique compte tenu des capacités de la programmation à base logique, mais aussi appropriées compte tenu du modèle MulVAL courant de représentations de données. Ce document fait état du degré élevé de progrès accompli dans le développement de chacune des extensions MulVAL proposées, comme on l'explique ci-dessous.

a. L’extension de réseau MulVAL montre qu’il est possible d’allonger la liste simplifiée de contrôle d’accès de l’hôte, définie dans le modèle MulVAL existant, selon une approche plus intuitive qui correspond de façon plus appropriée à la conception « réseau réel ».

b. L’extension de la valeur des biens démontre qu’il est possible de développer les prédicats existants pour englober d’autres particularités afin d’améliorer la représentation de l’environnement en train d’être modélisé. Comme il est possible d’attribuer une valeur aux données, des outils permettent d’établir l’ordre de priorité des voies d’attaque établies grâce à une analyse MulVAL.

Dans ce document, on suggère que d’autres démarches visant à améliorer les extensions MulVAL mèneront à un éventail complet d’améliorations valables et appropriées au modèle. Plus particulièrement, on propose que les tâches ci-dessous soient accomplies pour faire avancer le dossier.

a. Améliorer davantage l’extension de voies d’attaque pour déterminer comment les différentes évaluations/méthodes d’évaluation de la difficulté des attaques doivent être regroupées afin d’établir un indice de difficulté global pour le modèle d’attaque.

b. Effectuer davantage d’instanciations de données pour les extensions de modèle proposées afin de prouver la validation de l’approche (dans l’esprit du travail accompli pour l’extension MulVAL de réseau).

c. Évaluer les besoins en matière d’extensions et identifier toutes extensions supplémentaires.

Bacic, E., Henderson, G., Froh, M. 2006. MulVAL Extensions for Dynamic Asset Protection. DRDC Ottawa CR 2006-251. R & D pour la défense Canada – Ottawa.

Table of contents

List of figures

List of tables

This page intentionally left blank.

1. Introduction

1.1 Background

In 2005, Defence Research and Development Canada (DRDC) initiated an effort to develop a feasible abstraction in the area of defensive posture technology [1]. A defensive posture technology would assess the defensive state of a network based on the network topology and configuration, known vulnerabilities, and operational requirements. Such assessments have typically been done via a static threat and risk model; however, as networks and operational requirements become more dynamic, a correspondingly dynamic technology for network monitoring and defensive assessment is needed. Ideally, this defensive posture technology would monitor changes to the network configuration, be aware of newly discovered vulnerabilities, permit asset value to be assigned dynamically based on operational requirements, and identify the possible means by which an attacker could compromise the network's security policy or undermine the operations which the network supports.

The defensive posture abstraction was intended to provide the necessary framework for the creation, based on currently available technology, of a modelling system for dynamic risk management and asset protection. It was determined that a new approach was needed which could model threat propagation in the network via multi-stage attacks.

This effort resulted in the creation of a white paper entitled: *Dynamic Asset Protection & Risk Management Abstraction Study* [2]. A research project at Princeton University called MulVAL was identified in this report as a promising area of research [3], [4], [5], and [6]. Multihost, multistage Vulnerability Analysis (MulVAL) is an end-to-end framework and reasoning system that conducts complex attack vector analysis on a network. The output is a set of attack vectors specific to the network in question. Extensions to MulVAL that would be beneficial in furthering work at DRDC on dynamic network defensive posture were identified in [2].

Further investigation revealed several other projects that to varying degrees meet the requirements of a defensive posture technology. In particular, a commercial product called Skybox Security and an AI-based project called CycSecure were identified as interesting and relatively mature projects, which deserve closer consideration as candidates for a defensive posture technology to meet the needs of DND.

1.2 Scope

The proposed research will result in new MulVAL data model extensions in Datalog, an updated critique of how well MulVAL plus the extensions satisfies the requirements of dynamic asset protection, and a summary of areas requiring future research.

A secondary objective is to critique Skybox Security and CycSecure with respect to the requirements of dynamic asset protection.

1.3 Report Structure

Section 2 contains the conclusions from the critique of the alternative DAP models: Skybox and CycSecure. The full critique is contained in Annex A.

Section 3 contains the MulVAL model extensions in each of the following areas: Networking, Asset Sensitivity, and Attack Paths. For each of these extensions, the sub-section contains:

a. The Objective of the extension;

b. The Extension definition in Datalog syntax;

c. The proposed source of any Datalog facts needed for the extension;

d. A discussion on the design rationale behind the extension definition; and

e. A critique of how well the defined extension meets the objective.

Section 4 contains observations on the general MulVAL model. These observations were made while working with the existing MulVAL code.

Section 5 contains observations on the information found within the National Vulnerability Database (NVD), a prime source for MulVAL related facts on vulnerabilities. The observations include a brief analysis of the Common Vulnerability Scoring System (CVSS), which is included for most of the listed vulnerabilities.

Section 6 contains conclusions of the MulVAL extension work, as well as identifying topics for future research.

Section 7 contains the references used within this report.

Annex A contains the full critique of MulVAL, Skybox, and CycSecure against the DAP requirements.

Annex B contains detailed listings of the MulVAL Network Extension including some modelling outputs.

Annex C contains detailed listings of the MulVAL Asset Extension.

Annex D contains detailed listings of the MulVAL Attack Weight Extension.

2. Alternative Model Critique

Prior to working on the MulVAL model extensions, two other projects that appeared to meet the requirements of a defensive posture technology were examined:

a. A commercial product called Skybox Security [7], [8], [9], [10], [11], and [12]; and

b. An Artificial-Intelligence (AI) based project called CycSecure [13], [14], and [15].

An overall examination of these two models, and MulVAL, was performed. The results of the analysis are contained in Annex A – MulVAL, Skybox, and CycSecure Critique. The presentation of the analysis is done in two large tables:

a. Annex A – Table 1 – Tool Set Architecture provides a comparative view of all three tools with regard to: information available, network model collection, vulnerability knowledge, information analysis, information presentation, performance of information collection, performance of information analysis, and scalability; and

b. Annex A – Table 2 – Critique of Toolsets in Providing DAP Abstraction Requirements. This table has identical rows to the MulVAL critique in [2]. The table has three columns for each of MulVAL, Skybox, and CycSecure. The MulVAL critique is largely that provided in [2] with some minor updates. The Skybox and CycSecure critiques are new.

The following are conclusions regarding the three tool sets available to implement DAP:

a. MulVAL and CycSecure provide the most comprehensive attack path modelling since they both include host-based scanners. Of the two CycSecure may provide a more comprehensive attack path model since its ontology appears more refined; however, this hypothesis is not supported by the CycSecure references [13], [14], and [15]. In order to model most real-world attack paths, a tool needs to model internal host privilege escalation vulnerabilities that are tied to user accounts and file access permissions. Skybox does not include this depth of internal system information collection. Therefore, attack path reasoning can only be a subset of the attack paths found using MulVAL or CycSecure reasoning.

b. MulVAL is much better at detecting accidental mis-configurations since it has an inside view of the host and uses Open Vulnerability and Assessment Language (OVAL), which defines typical configuration errors. Skybox can only base their analysis on what they see from the outside and would likely miss many configuration issues. CycSecure should be capable of detecting mis-configurations, but this is cited as future work [15].

c. Skybox is a “glue” product that will correlate and analyse other information sources in a more meaningful manner than the originals. The degree of the product’s analysis capabilities is unclear but likely contained in the View Dictionary. Depending on how the analysis and View Dictionary are implemented, the product may have a brittle architecture, which requires constant tweaking/updating from the company to accommodate new vulnerabilities and how they relate to existing ones. The fact that Skybox presents other product information makes it a likely target for large players in the vulnerability scanning market to easily incorporate Skybox functionality. This is a risky business model. Skybox needs to provide significant value-add in analyzing correlated data from these various information sources in order to survive; but this doesn’t appear to be a strong capability yet.

d. CycSecure is a proof of concept to prove that Cyc is of use in real-world problems. CycSecure was trialled for six months in the US STRATCOM CERT, but it is unclear how extensively its Sentinels were deployed. CycSecure is not yet a viable product since its web pages do not provide any marketing, sales, technical, support, or contact information!

e. We only have hard performance and scalability data for MulVAL. CycSecure appears to be scalable and reason on attack plans in a reasonable timeframe. Skybox has no indications of scalability. To handle really large networks, we will likely have to use abstraction techniques to decompose large networks into smaller elements for sub-analysis. The Cyc knowledge base and reasoning engine already have techniques for handling this. MulVAL could likely be extended to include these types of techniques. It is unclear whether Skybox can apply abstractions in its analysis ability.

f. All of the tools essentially provide a snapshot in time analysis. All can be configured for periodic information collection and analysis. None of the products are geared towards truly dynamic environments, such as tactical military networks, which tend to favour a DAP-O-Matic[1] approach. None of the products incorporates feedback into the model to refine its scanning/analysis methods (for example, altering information collection techniques based on previous scan data).

After this tool analysis, MulVAL is still a good candidate for implementing DAP. Therefore, the remainder of this report provides the modelling extensions performed in Datalog according to the existing MulVAL data model.

[1] The term DAP-O-Matic was adopted in [2], although this term has been changed to Mole-O-Matic.

3. MulVAL Extensions

The approach taken in developing MulVAL extensions was to follow the following principle: minimize changes to the existing MulVAL model if possible. This reduces the work in folding these extensions back into the MulVAL source

The MulVAL extensions identified in [2] were reviewed and a priority of extension implementation was developed based on:

a. DAP requirements;

b. Preliminary design discussion; and

c. Likelihood of successful implementation in a given timeframe.

The priority of MulVAL extension implementation is given in Table 1 below. A sub-section is presented for each of the three MulVAL model extensions examined.

Table 1. *MulVAL Extension Implementation Priority (Highest to Lowest)*

DAP REQUIREMENT	DESIGN DISCUSSION	LIKELIHOOD OF IMPLEMENTATION	IMPLEMENTATION STATUS
Networking Element • Describe computer network resources as interdependent elements. • Describe security safeguards as attributes of computer network resources. • Map threat events onto computer network resources with vulnerability safeguard attributes. • Generally decompose a large network into smaller networks.	• More closely matches actual network configurations. • Explicitly models network safeguards (firewall/router/VPN) as hosts that can have vulnerabilities and be part of attack paths.	• High for following reasons: • The authors have previous work experience in the area. • Easy extension due to understanding of what needs to be modelled. • Many possible automated info sources (SNMP managers, firewall outputs). • Easy MulVAL integration since can derive existing `hacl/4` rules.	• Done • See Section 3.2

Table 1. *MulVAL Extension Implementation Priority (Highest to Lowest)*

DAP REQUIREMENT	DESIGN DISCUSSION	LIKELIHOOD OF IMPLEMENTATION	IMPLEMENTATION STATUS
Asset Value			
• Map mission-required IT services onto computer network resources. • Combine computer network resources into IT service offerings that support a required confidentiality, integrity, and availability Quality of Service (QoS). • Ref [1] added Accountability. • Relate threat events to safeguard effectiveness and vulnerabilities.	• Source of asset value will have to be manually created. • MulVAL currently doesn't handle service modelling to the extent we need it to: no binding to data. • Consider an asset impact extension to show more granularity on consequences. • Consider extending to CVSS C/I/A model (richer than currently used NVD model).	• Moderate for following reasons: • Good understanding of Confidentiality requirement. • Bad understanding of Integrity requirement. • Moderate understanding of Availability requirement. • Bad understanding of Accountability requirement.	• Started • See Section 3.3
Safeguard Model			
• Describe security safeguards as attributes of computer network resources. • Extension to model *NIX execute privilege (can attacker execute vulnerable program?). • Extension to model *NIX user/group/world privileges. • Extension to model *NIX and/or Windows ACLs. • Map threat events onto computer network resources with vulnerability and safeguard attributes. • Relate threat events to safeguard effectiveness and vulnerabilities.	• Princeton has already modelled Windows environment including ACLs [6] (and much more). Is it extendable into their USENIX paper MulVAL model? • Currently only model preventative safeguards. Can we extend to include detection, containment and recovery safeguards? • Need to model network safeguards explicitly (see Section 3.2). • Need to model safeguard effectiveness. Current model has binary effectiveness, or lack of vulnerability implies perfect safeguard.	• Implementing non-binary Attack Weights likely a big impact on existing model and attack tree visualization. • Prolog truth output (true/false) to queries now a probability function based on attack path safeguard effectiveness. • Consider CVSS as an available online data source for vulnerability difficulty, or safeguard effectiveness. This information can be queried through an automated process, without the intervention from an analyst. • Modelling other than preventative safeguards is embryonic.	• Network safeguard modelling done, see Section 3.2 • Started examining modelling attack weights in a non-binary manner. See Section 3.3.5

3.1 General Extension Approach

In order to deal with information dynamically we need to consider the following:

a. Asset value;

b. Attack path;

c. Value as a function of time; and

d. Dynamism of attacks.

There are a number of ways of proceeding, but the most logical way would be to leverage the power of Prolog and predicates. This can be achieved by actually storing all information relating to a specific predicate within the predicate itself.

This means that we define asset value, asset classification, etc. as predicates:

```
        assetValue(Asset, Value)
```

or possibly as a set of values:

```
        assetValue(Asset, ConfidentialityValue, IntegrityValue,
AvailabilityValue).
```

Similarly, we define each asset as having a specific classification:

```
        assetClassification(Asset, Classification, Caveat)
```

so that we can state predicates such as:

```
        assetClassification(fileXYZ, secret, ceo).
```

This would then allow us to quickly determine which assets are of a given classification:

```
        assetClassification(AssetList, secret, _)
```

or we can find what a specific asset has as its classification and caveat:

```
        assetClassification(fileXYZ, Classification, Caveat)
```

or we can find all assets with CEO:

```
        assetClassification(AssetList, _, ceo).
```

The flexibility this offers us is that we can use this in a codified manner within the various predicates that determine attack paths. Based on other external inputs we can modify the behaviour of the system dynamically since the predicates can be restated as required.

This ability to change the associations between a given asset and its associated values is crucial to allowing for temporal attack path determination. For example, should an asset be deemed no longer classified, it can be downgraded by means of a simple function which will perform the following actions

a. Take an asset as an argument;

b. Store the current classification;

c. Retract the asset; and

d. Add the asset to the database with a classification level one lower than its previous value.

For example, if we are only interested in attacks against Secret assets, then the attack paths to the provided asset immediately can be eliminated from the search. This dynamism is something that has historically been very difficult to do.

It is possible to create biases in the attack paths by simply adding in a few predicates that leverage classification and asset value. In fact, it could be argued that asset value should be tightly tied to classification.

It would appear that the best option, therefore, is to create more predicates. This appears to be generally true for everything that must be extended within the MulVAL model. The reason for this view is that it becomes possible to query the rules efficiently to determine, for example, all assets of a particular classification or of a particular value. The other reason to prefer predicate representation is that it allows the model to remain simple rather than creating the complexity of passing vectors and variables. Leveraging the power of Prolog actually lessens the amount of custom coding that is required.

Thus, instead of extending a given predicate with more variables we amend the actual code to handle new predicates such as:

```
assetValue(Asset, Confidentiality, Integrity, Availability,
bias)
assetClass(Asset, Classification)
```

This allows the definition of each asset uniquely as a name-value pair within Datalog:

```
assetClass(someAsset, secret)
assetClass(someOtherAsset, secret)
assetClass(yetAnother, topsecret)
```

This allows any routine simply do the following to get back every asset that is classified Secret, namely someAsset and someOtherAsset.

```
assetClass(Assets, secret)
```

It also means that predicates focus on the assets as opposed to trying to pass parameters around unnecessarily.

In this way, Prolog handles all the actual data storage and retrieval and manipulation. Also, if an asset's classification changes, for example, from Secret to Unclassified, it only requires the following:

```
assetClass(yetAnother, unclassified)
```

This representation is simple and intuitive to the reader. Also, manipulating the predicates is straightforward as with the function to perform a downgrade, for example, to declassify an existing asset:

```
makeUnclassified(Asset) :-
      Old   = ..[Asset, _], retract (Old),
      New = ..[Asset, unclassified], assert (New).
```

By performing dynamic downgrades, or by inserting new clauses into Prolog's database, it is possible to alter dynamically the behaviour of MulVAL. It is significant to note that Datalog permits dynamic manipulation of its clause database.[2]

For the ranking of attack trees, an association would need to be created between the asset and the device upon which the asset resides. In order to examine paths that have high value assets, a bias in MulVAL would need to be established to examine paths to assets of a particular value before others. This should be easily done by having the appropriate functions request:

```
assetValue (Asset, secret)
```

to get a full list of all secret assets and then:

```
assetResides (Asset, Machine)
```

In other words, the machine classification is a function of the assets that reside on it. The predicate to list all machines with a specified type of data is:

```
machineClassification(Classification, Machine) :-
    assetValue(Asset, Classification,),
    assetResides(Asset, Machine).
```

Obtaining the list of machines that are classified as secret would be done through a call similar to:

```
machineClassification(secret, Machine).
```

Alternatively, obtaining the classification of a specific machine (e.g. “target system”) would be done through a call similar to

```
machineClassification(Classification, targetsystem).
```

[2] Specifically, the Datalog implementation currently used by the MulVAL implementation (XSB) supports this dynamic manipulation of the clause database.
http://xsb.sourceforge.net/manual1/node103.html

3.2 Network MulVAL Extension

3.2.1 Objective

The objective of this MulVAL extension is to derive `hacl/4` Datalog rules in a more intuitive manner that reflects real-network design principles. The extension should be capable of modelling the network elements shown in Figure 1:

a. Hosts belonging to a subnet;

b. Multi-homed hosts which belong to two, or more, sub-nets;

c. Routers and firewalls are a special case of multi-homed hosts which also perform a routing function between sub-nets;

d. Modelling of simple routes of two subnets connected by a single router; and

e. Modelling of transitive routes of two subnets connected by two, or more, routers, and one, or more, transiting sub-nets.

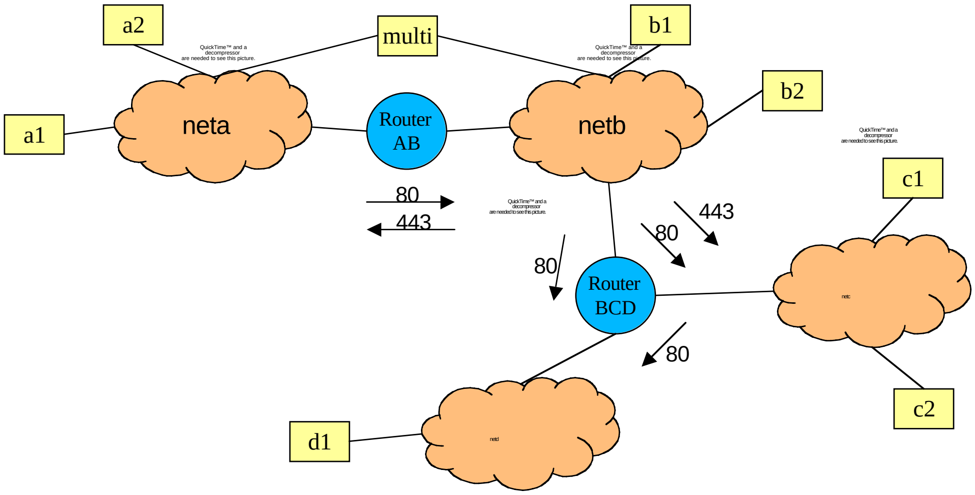
a2
multi
b1
QuickTime™ and a decompressor are needed to see this picture.
QuickTime™ and a decompressor are needed to see this picture.
b2
a1
neta
Router AB
netb
QuickTime™ and a decompressor are needed to see this picture.
c1
80
443
QuickTime™ and a decompressor are needed to see this picture.
443
80
80
Router BCD
netc
80
c2
d1
netd

Figure 1. *Complex Network Modelling*

3.2.2 Extension Definition

The complete extension definition is contained in Annex B – MulVAL Network Extension. The key extension components are described here.

Complex networks can be modelled using two primitive Datalog predicates: hosts are attached to a network:

```
hostNet(Host, Net).

routeEntry(Router, Initnet, Targetnet, Protocol, Port).
```

The above primitives define which subnets hosts are attached to and defined routes between subnets at routing points.

From these two primitive Datalog elements the following two derived Datalog primitives are defined:

```
route(InitNet, TargetNet, Protocol, Port).

hacl(InitHost, TargetHost, Protocol, Port).
```

The first derived primitive is looking for valid routes amongst all routers. The second derives `hacl/4` rules based on `hostNet/2` and `route/4`. The derivation of these two primitives is defined using the following four interaction_rules, two to derive `route/4`, and two to derive `hacl/4` rules.[3]

```
/****** Route Section *******/

interaction_rule(
   (route(InitNet, TargetNet, Protocol, Port) :-
   routeEntry(Router, InitNet, TargetNet, Protocol, Port),
   hostNet(Router,InitNet),
   hostNet(Router,TargetNet)),
   'Direct route between subnets through an intermediate
router').

interaction_rule(
   (route(InitNet, TargetNet, Protocol, Port) :-
   route(InitNet, TransitNet, Protocol, Port),
   route(TransitNet, TargetNet, Protocol, Port)),
   'Transitive routing through an intermediate network').

/****** HACL Section *******/

interaction_rule(
   (hacl(InitHost, TargetHost, Protocol, Port) :-
   hostNet(InitHost, InitNet),
   hostNet(TargetHost, TargetNet),
   InitNet \= TargetNet,
   route(InitNet, TargetNet, Protocol, Port)),
   'Hosts can only communicate between networks through a
valid route.').

interaction_rule(
```

[3] Commonly, port information from network devices (e.g. firewalls) is presented as ranges. A useful expansion of the work presented in this paper would be to determine a method by which port ranges can be expressed within the constraints of the logic-based program syntax.

```
    (hacl(InitHost, TargetHost, _, _) :-
     hostNet(InitHost, CommonNet),
     hostNet(TargetHost, CommonNet)),
    'Hosts on same network have no communication
  restrictions.').
```

3.2.3 Data Source

With the network extension defined above, a data source needs to be identified for the identified primitive Datalog predicates, namely:

a. `hostNet/2` – this information can be derived from a number of automated sources including: host-based scanners such as MulVAL's OVAL scanner, DNS records, DHCP leases, ARP listening or ping sweeps of a network; and

b. `routeEntry/5`—the most reliable source of information can be derived from routers and firewalls, or from any management system for these devices. Examples might include Simple Network Management Protocol (SNMP) consoles, proprietary device management consoles, and processes running on the devices themselves. Less reliable routing information can be derived from network mapping tools such as nmap.

3.2.4 Design Rationale

The existing MulVAL `hacl/4` network model is limited in the complexity of networks that it can model. It is particularly difficult to use wild cards in defining `hacl/4` primitives since these often won't reflect actual network connectivity. This means one would typically have to define individual `hacl/4` primitives for all host-to-host network connectivity, which is a tedious exercise.

The primitive Datalog predicate design ensures that data can be automatically derived from several potential sources.

The primitive and derived Datalog predicates are much more intuitive and model real network designs than `hacl/4` rules do.

Since routers and firewalls are special cases of multi-homed hosts, they can also have inherent vulnerabilities and network services running on them. This explicitly models filtering routers and firewalls as safeguards.

Typical router and firewall configurations can be defined including situations with multiple interfaces. For example, a single firewall can have Internet, de-militarized zone (DMZ), and Intranet interfaces with differing `routeEntry/5` definitions that reflect a real policy.

The `routeEntry/5` primitives define individual one-way data flow policies. This method of representing network connectivity in terms of source, destination, protocol, and port and using this construct to define and drive valid path routes is very much in accordance with existing firewall practices, most notably, the iptables[4] packet filtering framework. While iptables allows the definition of policy for inbound, forwarding and output packet flow, MulVAL, and the proposed extension, takes a purely outbound view of the network path. That is, the `hacl/4` rules determine what is accessible from a specific source to a specific destination.

In the first `hacl/4` interaction rule, `InitNet \= TargetNet` was added for efficiency. In the case where InitHost and TargetHost are on the same subnet (the case handled by the second hacl/4 rule) this condition makes evaluation much faster.

3.2.5 Extension Critique

The network extension successfully derived `hacl/4` Datalog primitives when integrated with MulVAL. This provided a simple and elegant manner in which the extension was integrated back into the MulVAL model. Tests on the network shown in Figure 1 show that the network could be defined using 14 `hostNet/2` and 6 `routeEntry/5` network extension primitives while it derived 131 `hacl/4` primitives (the source Datalog facts and derived `hacl/4` primitives are contained in Annex B). A review of the derived `hacl/4` primitives in Annex B shows that routes to all of the interfaces of multi-homed systems are found. An example of finding routes to different interfaces of multi-homed systems is:

a. `hacl(a1, multi, _, _)` on the netA interface of multi; and

b. `hacl(a1, multi, tcp, 80)` on the netB interface of multi, using the route through routerAB.

One concern with the currently defined extension is that `hacl/4` rules that refer to local machine do not distinguish between interfaces on that system:

```
hacl(router, router, _, _)
```

In this example, there are valid paths between all interfaces on the router, a loss of granularity which may expose more paths that actually exist. An amendment to the extension would include interface to the `hacl/4` rule, using a primitive declaration which would only have significance for multi-homed systems.

In complex networks, the savings in data input are substantial and eliminate any potential for logic errors in manually deriving `hacl/4` primitives assumed in the existing MulVAL model.

[4] http://www.netfilter.org/

The network extension as defined must have consistent `hostNet/2` and `routeEntry/5` primitives defined to properly map routers and firewalls. That is, the data source must ensure that a `hostNet/2` primitive is defined for each router/firewall subnet interface to ensure that `route/4` is properly derived from `routeEntry/5`. We could loosen the first `route/4` definition by removing the `hostNet/2` requirements of the routers. However, one would not derive `hacl/4` rules for attack paths to the routers themselves, which is limiting. A more elegant design would remove the requirement for explicitly defining `hostNet/2` primitives for any device with a `routeEntry/5` and the model would derive these.

The network extension as defined doesn't handle port forwarding at a firewall. The port forwarding could be handled by extending the `routeEntry/5` to:

```
routeEntry(initNet, targetNet, targetHost, protocol, port)
```

where typical routing entries would have "Any" under `targetHost`. This would handle the port forwarding but one would have to be careful in defining the rules (for example, don't define two port 80 forwards).

The network extension as defined doesn't handle Network Address Translation Port (NATP), or port redirection. This might also be handled by a similar extension to:

```
routeEntry(initNet, incomingPort, targetNet, targetHost,
protocol, outgoingPort)
```

where `incomingPort` = `outgoingPort` in most cases (for example, 80 and 80 for http). With this rule one could define incoming port 80 becomes outgoing port 8080. Our discussion to date is that this extension might be overcomplicating the model right now for a feature that may not be heavily used in an Enterprise environment (from our estimation).

The network extension as defined doesn't handle source IP blocking, which is typical firewall functionality. We started down a similar `routeEntry/5` extension of adding an "`initHost`" parameter but quickly decided this was flawed thinking. It is very difficult to build blocking rules with inclusive rule types. For example, if one host were blocked from any outgoing connections, one would have to explicitly define allowed rules for all remaining hosts on the subnet – clearly not a good model design. We think some form of `notAllowed` parameter similar to `routeEntry/5` and use with negation to build `hacl/4` rules. We have not explored this any further.

The network extension as defined implies a change to `networkServiceInfo/5` to include a `_net` parameter that the service is listening on. Without defining this extension to `networkServiceInfo/5`, the MulVAL logic will assume that any service is listening on all interfaces. Although this may be the default configuration of many services, it becomes particularly important at important firewall boundaries where firewall management services would only be listening on inner, trusted subnets. This is evident in the Annex B listing where it appears that multiple `hacl/4` primitives are derived between the same initiator and target systems. In these cases, the listing explicitly shows routes to two different interfaces on the target system. Therefore, consideration should be given to extend the `hacl/4` primitive to also show the target interface. An example of these two explicit routes from Annex B are:

a. `hacl(a1,multi, _, _)` on the netA interface of multi; and

b. `hacl(a1,multi,tcp,80)` on the netB interface of multi, routed through routerAB.

3.3 Asset MulVAL Extension

3.3.1 Objective

The objective of a MulVAL Asset Extension is to provide basic tracking of asset sensitivity. The current MulVAL model [4] does not track asset sensitivity at all, but can define authorized policies of who can access what data. It is notable that the USENIX paper where the MulVAL model was presented included a statement to indicate that the omission of asset value from the model was a function of the lack of source data, not inherent limitation of the model itself.

"Currently we do not have exploit rules for vulnerabilities whose exploit consequence is confidentiality loss or integrity loss. The ICAT database does not provide precise information as to what confidential information may be leaked to an attacker and what information on the system may be modified by an attacker."

Tracking asset sensitivity will allow sorting of attack trees derived by MulVAL to be sorted in an order of criticality, a feature that is more crucial to defending networks. An ordered list of attack paths against an organization’s most sensitive assets is needed.

An initial point is to establish the categories for asset valuation that will be targeted by this extension. From an asset perspective there are two main categories: data and processes. Each of these categories can be assessed in terms of the need for: confidentiality, integrity and availability.

When addressing the availability issue, it is important to note that service availability and quality of service concepts are closely linked. For a critical application that must provide a certain level of throughput with a minimum amount of latency (e.g. a voice over IP application), a degradation of throughput, or reduction in quality of service is operationally equivalent to loss of availability.

3.3.2 Extension Definition

An initial view of the categories of assets in the context of data and processes is presented in the following table:

***Table 2**. Asset Categories*

	DATA	PROCESSES
CONFIDENTIALITY	Data elements can be classified using a set of sensitivity levels. This classification follows the data as it exists at various points along the network Significance: High	When data access must be done through a specific process, confidentiality of that process may be relevant. Significance: Low
INTEGRITY	There may be a need to require proof against tampering for a data element. Significance: Moderate	There may be a need to require proof against tampering for a process. Significance: Moderate
AVAILABILITY	When a process is fed by the presence of data, the process may be faulted through the lack of availability of the data. Significance: Low	As per the CNDSA paper, network requirements can be expressed in terms of the need to access data and the processes that act on that data. Significance: High

This table presents each of the asset valuations in terms of its significance to the MulVAL extension effort. Significance is determined by:

a. The degree to which the asset valuation is commonly associated with the asset categories;

b. The ease by which the current MulVAL model can include the asset valuation element; and

c. The degree to which the means by which the asset valuation is determined is compatible with the MulVAL approach.

To focus this investigation, the most significant asset valuation mechanisms have been addressed, specifically,

a. Application of confidentiality and integrity levels to data; and

b. Application of availability needs of services.

There are two options for associating C/I values to data: the existing predicate *dataBind* can be extended or a new predicate can be established to specify a confidentiality/integrity levels for a data set. Both potential predicates are presented below:

```
dataBind(Data, Host, Path, Confidentiality, Integrity)

dataValue(Data, Confidentiality, Integrity)
```

For the first option, the data valuation levels would be associated with the data that is located at the specified host/path. The second option specifies that the entire data set identified by data would be given a single classification level. This classification level could potentially apply to many paths on hosts as per the *dataBind* primitives that define the content of the data set.

To address the availability component of information asset value, the ideal scenario would link the impact of existing vulnerabilities to the required level of service that must be provided by applications in the target environment. These levels of impact can be associated with an operational service to indicate that the service must be able to provide a level of availability using either an existing primitive or a new one, as detailed below:

```
networkServiceInfo(Host, Program, Protocol, Port, User,
Availability)

available(Host, Program, Availability)
```

Note that the *vulProperty* predicate which defines the impact of vulnerabilities would have to be extended to include the C/I/A impact for each vulnerability in the target environment.

```
vulProperty(Cveid, Range, Consequence, Confidentiality,
Integrity, Availability)
```

3.3.3 Data Source

Unfortunately, data sensitivity is typically something that cannot be derived from automated sources. The derivation of what is sensitive is largely a manual exercise that focuses specifically on data confidentiality, integrity and availability. Ref [1] includes Accountability as another measure of sensitivity.

The eventual intent is that military commanders can define their computer network needs in terms of high-level network services. This can then be overlaid on a network to determine what particular assets (data, hardware, software, and network links) are crucial to providing the required services.

From the analysis perspective, C/I/A values to be populated in the *vulProperty* predicate can be obtained from the published CVSS vector available through the National Vulnerability Database. The NVD CVSS score (see Section 4) includes C/I/A based metric that measures the impact a successful exploit a given vulnerability will have on the target system. C/I/A values are expressed in terms of their impact on a scale from: None (no impact), Partial (some impact) to Full (significant impact). As an example, the CVSS impact ranking as it applies to availability would have the following interpretation.

None: No impact on availability.

Partial: Considerable lag in or interruptions in resource availability. For example, a network-based flood attack that reduces available bandwidth to a web server farm to such an extent that only a small number of connections successfully complete.

Complete: Total shutdown of the affected resource. The attacker can render the resource completely unavailable.

When populating the Datalog predicates to associate C/I/A requirement on data and services in the target environment, there must be commonality between the values used to express these requirements and the values available through the CVSS vector.

3.3.4 Design Rationale

With the intent of the chosen asset valuation extensions in mind, the selection of the location to present value data must be in accord with the elements to which the value applies.

Confidentiality and integrity values must be associated with data elements at a level of granularity that is suitable for the environment that is being modelled. The presentation of two options for creating this association provides sufficient flexibility to achieve this needed level of granularity.

For example, a web server application will have information in various directories, each of which may have different protection requirements:

Table 3. *Example dataValue entries for a web application*

DATATYPE	LOCATION	CONF	INTEG	PREDICATE
Executables	/usr/bin	None	Partial	dataBind(webdata, host, /usr/bin, none, partial)
Data	/var/www	High	Partial	dataBind(webdata, host, /var/www, high, partial)
Configuration	/etc/www	Partial	Partial	dataBind(webdata, host, /etc/www, partial, partial)
Log	/var/www/log	None	Full	dataBind(webdata, host, /var/www/log, none. full)

Should the second primitive be used, that is, *dataValue* which sets the confidentiality and integrity values for a complete data set, a sample primitive would be:

```
dataValue(webdata, high, full)
```

Note that an attempt to determine the confidentiality of the data at a specific host/path location would always return the high water mark confidentiality for the entire data set.

```
hostPathSensitivity(Conf, Host, Path) :-
   dataValue(Data, Conf, Integ)
   dataBind(Data, Host, Path)
```

The result of using either of these approaches is that any analysis that presents an attack path that touches on data can include, as part of the output, a statement of the confidentiality and/or integrity impact of the result.

For example, a *policyViolation* analysis that generates an attack tree for cases where an attacker can access data will include an identifier for the data that has been accessed in violation of the policy. The results from this analysis can be ranked based on the C/I values to identify those attacks that are more critical from an asset valuation perspective. Note that the valuation ranking can be merged in with the logic for visualizing the attack trees or included in the analysis output to allow sorting/presentation of this ranked attack list using external tools.

Similarly, analyses which identify vulnerabilities (specifically of the denial of service type) which can be effectively used against the environment can be ranked based on the relative importance, operationally speaking, of the service.

3.3.5 Extension Critique

The enhanced data and network service primitives can effectively provide a means by which to rank the relative importance of attacks against the assets to which these primitive apply. The fact that that C/I/A impact information relating to vulnerabilities can be obtained from the NVD implies that the attack information can be automatically generated. However, population of asset valuation data as it applies to data and processes will remain a manual process. This manual process may limit the scalability of the solution once a large number of data sets and services are in the target environment.

The NVD provides a limited set of rankings for C/I/A information and the assignment of these rankings to a specific vulnerability is, to a large extent, subjective. In the light of the GoC/GSP, it is not clear what the implication of “Partial” integrity impact would be for a specific data element or service. Similarly, the available range of impact for availability issues would benefit from expansion to include additional impact levels. For example, the following set of impacts for a denial of service attack could be instituted as each level would have a different implication in terms of downtime and time to recover.

```
level 0: the attack compromises the system/network stack,
rendering the entire system unavailable until it is
restarted.

level 1: the attack compromises an application, requiring
the service to be restarted.

level 2: the attack does not fault the application, but a
reduction in service is incurred.

level 3: the attack does not fault the application and there
is a minimal impact to the level of service provided by the
service.

level 4: No availability impact.
```

In this scenario, levels 0 to 1 would correspond to the NVD's interpretation of a full impact for an availability attack. Levels 2-3 would correspond to the NVD's interpretation of partial impact for an availability attack. The use, however, of additional impact rankings would allow the MulVAL model to reflect more detail in the consequences of such an attack. There is still the need, however, for an external source that can provide this impact information. For example, the NVD does not include any information about availability attacks that can target a service to bring down the network stack on a target box. Such an attack would have a significant impact to all services hosted on that target.

Once asset valuation is in place in the MulVAL model, it would become possible to drive policy analyses which will match malicious users and their clearance level to classified information. For example, a primitive to assign a user a clearance level could be expressed as follows.

```
clearance(user1, secret)
```

In this scenario, it would be possible to produce attack trees whereby a malicious user gains access to data that is beyond that user's clearance level. Note that this attack does not require the exploitation (or existence) of any specific service vulnerabilities. That is, a user can access a local file through allowed network policy, but the user's clearance does not meet the classification level of the data, a security policy violation can be detected.

It is notable that C/I issues can only be described in the MulVAL model in terms of violation of the access control policy. There is no representation in the model for protecting data in transit. However, availability can be considered in the context of a multi-hop transmission path. That is, availability issues must consider the network environment in which data is being transmitted. A merging of the Network MulVAL extension that defines firewalls/routers as network devices with this discussion of availability can potentially address part of this concern through the addition of an availability element to the *routeEntry* predicate. An investigation into availability issues in terms of vulnerable network devices, single point of failure and dynamic routing is recommended.

As a final point, it is important to note that attacks that impact availability are focussed on software vulnerabilities rather than network-level attacks. This is another recommended area of investigation as the need to recognize the presence of network attacks (such as flooding attacks) means that environment specific threshold values must be established to differentiate between normal traffic and an attack.

Ranking asset valuation using a small range of values on an absolute scale can be problematic. For example, using (High, Medium, Low) values is not sufficient to model the Government of Canada security policy for both classified and designated confidential assets. The classified (Top Secret, Secret, Confidential) and designated (Protected A, B, and C) scales are based on injury to the national interest and individuals/corporations, respectively. However, when one combines the scales in a single absolute value range, highly classified national interest material (for example, Top Secret Special Access) typically eclipses all other values. For example, a viable absolute scale of asset value for (TS/SA, TS, S, C, PC, PB, PA, Unclass) might be (1.0, 0.9, 0.7, 0.4, 0.3, 0.2, 0.1, 0.001). This skewing towards highly classified assets can then bury reasonably likely attacks to lower valued assets that should be addressed. Using a relative asset value scale may be necessary that is specific to the system being modelled. In these cases, MulVAL models of systems with differing assets are not comparable from an asset valuation or attack path injury perspective.

3.4 Attack Path MulVAL Extension

3.4.1 Objective

The objective of this MulVAL extension is to develop a method of prioritizing attack path generation so that MulVAL users can concentrate on remediation of the most important potential attacks, that is, those with the greatest impact to the organization.

Organizational impact, or risk, is primarily a function of two elements:

a. *Injury to Asset Value based on any successful attack*. Determining asset value was dealt with in Section 3.3 above. However, the injury a specific attack causes to an asset is only partially tracked in MulVAL with the consequence parameter in `vulExists/3` predicate. This currently specifies only one of confidentiality, integrity, or availability consequences (in addition to privilege escalation). Additional information is available from CVSS scores to provide a more granular derivation of injury to asset value; and

b. *The likelihood of an attack being successful*. Attack likelihood is a complex function of the probability of an attempted attack, the difficulty in mounting the attack, and the capability of the attacker. Determining the likelihood of attack initiation is also complex and fuzzy based on such elements as attacker motivation, potential injury, probability of detection, and consequences to the attacker. Determining attack success likelihood is not possible in MulVAL given the lack of creditable sources of data to all the composite factors. However, MulVAL could be usefully extended to approximate likelihood by tracking creditable data sources that we can access. Therefore, the objective is to approximate attack likelihood by tracking the difficulty in mounting the attack by using CVSS scores.

Additionally, this MulVAL extension requires the ability to order attack trees in a manner that makes organizational impact readily evident.

3.4.2 Extension Definition

A suitable MulVAL extension could not be developed during the project timeframe.

3.4.3 Data Source

The originally proposed data sources for modelling asset injury [2] were the following CVSS base scores:

a. Confidentiality (None, Partial, and Complete);

b. Integrity (None, Partial, Complete);

c. Availability (None, Partial, Complete); and

d. Bias (Normal, Confidentiality, Integrity, and Availability).

The NVD CVSS analysis in Section 4 below, shows that these values are tracked for tracked for nearly all NVD tracked vulnerabilities. The bias score is of little use since >99% of all values are normal (that is, confidentiality, integrity and availability are equally biased in injury). The values of C/I/A tend to be clustered into the following groupings:

a. Complete injury to all of C/I/A;

b. Partial injury to all of C/I/A;

c. Complete injury to just C;

d. Complete injury to just I; and

e. Complete injury to just A.

The originally proposed data sources for modelling the likelihood of successful attacks [2] were the following CVSS scores:

a. accessComplexity (High, Low) from the base CVSS score;

b. authenticationRequired (Required, Not Required) from the base CVSS score;

c. exploitability (Unproven, Proof-of-concept, Functional) from the temporal CVSS score; and

d. reportConfidence (Unconfirmed, Uncorroborated, Confirmed) from the temporal CVSS score.

The NVD CVSS analysis in Section 4 below shows that temporal CVSS scores are not currently tracked in NVD although the database has the capability to carry this data in its schema.

Additionally, analysis of the values recorded for the two CVSS base scores show that they are heavily biased to a single data value, which decreases the element's usefulness in determining attack pattern likelihood:

a. Greater than 97% of vulnerabilities have an accessComplexity of *Low*; and

b. Greater than 99% of vulnerabilities have an authenticationRequired of *Not Required*.

Therefore, from a creditable data source perspective, there is only data available that allows a more granular determination of asset injury in a MulVAL extension. All the data sources for determining attack likelihood either don't exist or provide no discernable value.

3.4.4 Design Rationale

MulVAL models attack paths as sequences of attack steps that may have local asset consequences but mainly lead to other attack steps. The question of when one stops the attack sequence to cause injury to the final set of assets resident at the final attack step remains. From an attacker perspective, it depends on their motivation:

a. If system exploration is the motive, then the attack paths will not end until all aspects of the system are compromised; or

b. If organizational injury is the motive, then the attack will terminate when the attacker has achieved their objective of impacting some targeted asset.

From a defender perspective, the attacks of greatest concern are those that cause the greatest injury to the organizational assets. Although this can be aligned with the second attacker motive above, it may not be based on the attacker and defender having differing views of organizational injury.

So a MulVAL attack pattern is made up of a number of steps, which ultimately lead to injury to a valued asset. In order to prioritize all the attack paths one must find final step vulnerabilities in viable attack patterns that lead to unacceptable injury. These attack patterns can be further prioritized by determining the difficulty, or path resistance, in completing all the steps in the attack pattern.

It is not clear at this time how the different steps should be combined to determine an overall difficulty value for the attack pattern. Possible mathematics include: summation of step difficulty, product of step difficulty, or high/low water marks.

Further research is needed in the type of factors that might influence the determination of attack step difficulty, which might include:

a. *Attack complexity*. Here we try to rate the complexity of mounting the attack step. Although NVD currently tracks this CVSS base score, the data present is of limited value being mainly low. Note that CVSS is considering adding an intermediate value which may make this CVSS score more useful in determining MulVAL attack step difficulty;

b. *Availability of exploit code*. Although CVSS temporal scores track this notion, this data is not populated in the NVD. There is a danger that the public knowledge of available exploit code does not reflect secretly held exploit code developed within the hacker or Information Operations adversary fields;

c. *More detail on what type of authentication, if any, is required*. CVSS base scores currently have an authentication-required value, although most of the vulnerabilities listed do not require authentication. Furthermore, in the cases where authentication is required, MulVAL already tracks prior attack steps which provide privilege escalation to a useable account in the current attack step;

d. *Attack Path Length*. The overall number of steps and complexity in sequencing an attack pattern may provide an element of difficulty for human attackers; however, automated scripting of attack patterns and automated attack tools ameliorate this difficulty; and

e. *Attack Execution Time*. Some attack steps may take a determinate amount of time to complete based on the resistance of underlying safeguards. Two examples are: cryptanalysis and password cracking.

Attack execution time seems to be related to underlying preventative safeguards.

3.4.5 Extension Critique

A suitable MulVAL extension could not be developed during the project timeframe.

4. MulVAL Model Observations

While working closely with the definition and composition of MulVAL model elements, some observations were made with regards to model behaviour which may warrant further investigation. These observations are listed below.

1. There is no examination of the scenario where a vulnerable network service is operating under a program that is, itself, identified as being setuid enabled.

2. Under conditions where a program supports both TCP and UDP protocols (e.g. DNS), this single program will be associated with both network services. If a vulnerability exists within that program, but only applies to one of these protocols, MulVAL will incorrectly report that the program is vulnerable though either protocol.

5. NVD CVSS Observations

The US National Institute of Science and Technology (NIST) maintains the NVD on-line database of vulnerabilities. This database ties together vulnerability data from a number of sources in a standardized schema.

The current MulVAL data model relies on the exploit range (local or remote) and the privilege escalation consequence data that were previously stored on the ICAT database, but are now stored in NVD. Note that ICAT data has been integrated into NVD and ICAT format data output will be deprecated sometime in the future.

The Asset and Attack Weight MulVAL Extensions also rely on the CVSS values contained in NVD. As noted in the DAP Report [2], CVSS provides the following:

a. The CVSS base scores Confidentiality, Integrity, and Availability provide an indication of the degree of impact (None, Partial, and Complete) a vulnerability has. The Bias information indicates the general impact weighting between these three asset values (Normal, or C/I/A biased); and

b. The CVSS base scores for Access Complexity (High, Low) and authentication required (Yes, No) could be used to eliminate attack success by non-capable threat agents. Also the CVSS temporal scores for exploitability (unproven, proof-of-concept, and functional), and the report confidence (unconfirmed, uncorroborated, and confirmed) could be used to determine the probability of success of an attack.

Since the NVD and its CVSS scores are critical data sources for MulVAL extensions, a quick analysis was performed on the NVD CVSS scores[5]. The results are as follows:

a. The NVD currently contains 15,828 vulnerabilities from 2002 to present. Of these, CVSS scores were provided for all but 234 entries which were blank;

b. The NVD currently only contains CVSS base scores. The NVD website notes that CVSS temporal scores can also be tracked, but no temporal data exists. The NVD contains the entire CVSS base score vector in the format "(AV:R/AC:L/Au:NR/C:N/I:N/A:C/B:N)", which corresponds to: Access Vector (Local or Remote), Access Complexity (Low or High), Authentication Required (Not Required or Required), Confidentiality (None, Partial, or Complete), Integrity (None, Partial, or Complete), Availability (None, Partial, or Complete), and Bias (Normal, Confidentiality, Integrity, or Availability);

c. Two CVSS scores were heavily biased to a single data value that their use in MulVAL extension is questionable: over 99% of all vulnerabilities did not require authentication and had a normal impact bias;

[5] The NVD database was downloaded and analyzed on 15 March, 2006.

d. Additionally, the Access Complexity score was heavily biased towards a single answer: over 97% of all vulnerabilities have a low access complexity;

e. 25% of vulnerabilities are listed as local and the remaining 75% can be initiated remotely;

f. As shown in pivot Table 4, for most of the vulnerabilities in NVD (that is, those with low complexity and no authentication required), the asset value impacts are not evenly distributed but rather clustered with the majority of vulnerabilities (83%) having either: Complete impact in C/I/A (15%), only C impact (12%), only I impact (10%), only A impact (16%), or partial impact in C/I/A (30%).

Table 4. *Pivot table of C/I/A impacts*

CONF	INT	AVAIL	AV:L	AV:R	GRAND TOTAL
C:C	***I:C***	***A:C***	33.01%	9.14%	14.80%
		A:N	1.40%	1.14%	1.20%
		A:P	0.25%	0.29%	0.28%
	I:N	***A:C***	0.20%	0.31%	0.29%
		A:N	9.15%	12.77%	11.91%
	I:P	***A:C***	0.00%	0.03%	0.03%
		A:P	2.10%	1.50%	1.64%
C:N	***I:C***	***A:C***	0.53%	0.33%	0.38%
		A:N	9.57%	10.66%	10.40%
	I:N	***A:C***	9.66%	17.55%	15.68%
		A:P	0.42%	1.42%	1.18%
	I:P	***A:N***	0.81%	5.48%	4.38%
		A:P	0.06%	0.10%	0.09%
C:P	***I:C***	***A:C***	0.08%	0.07%	0.07%
		A:P	2.46%	2.45%	2.46%
	I:N	***A:N***	1.04%	1.90%	1.69%
		A:P	0.03%	0.06%	0.05%
	I:P	***A:C***	1.34%	3.77%	3.19%
		A:N	0.08%	0.49%	0.39%
		A:P	27.80%	30.53%	29.88%
Grand Total			100.00%	100.00%	100.00%

6. Conclusion

This paper has demonstrated that the MulVAL model is extensible and can be enhanced to include additional data representation and analysis features to tailor the model to a specific problem environment. Specifically, the extensions which were proposed in the *Dynamic Asset Protection & Risk Management Abstraction Study* have been shown to be both technically valid given the capabilities of logic-based programming, but also appropriate given the current MulVAL model data representations. This paper documented a substantial degree of progress in the development of each of the proposed MulVAL extensions, as detailed below.

a. The network MulVAL extension demonstrated that it is possible to extend the simplistic host access control list, currently defined in the existing MulVAL model, with a more intuitive approach that more appropriately reflects real-network design. The ability to represent hosts on one or more subnets, network devices that gate access between subnets, and routes between subnets provides a great deal of flexibility for the creation of new assertions to capture the true network and dataflow configuration.

b. The asset value extension demonstrated that it is possible to extend existing predicates to include additional characteristics for improved representation of the environment being modelled. By drawing upon industry accepted practices for the definition of data and service value, a view of the model can be made which is more consistent with established data valuation practices. The impact of vulnerabilities in the target environment can be expressed in a context that is more easily understood by security professionals. Given the ability to assign a valuation to data, the tools are in place to prioritize the attack paths which are established though a MulVAL analysis.

c. This paper provided not only a theoretical evaluation of potential extensions for the MulVAL model, but also the design and implementation of actual predicates that instantiate the extensions as defined. Testing was undertaken to prove the actual operation of some of the proposed model extensions.

This paper proposes that additional effort to refine the MulVAL extensions will lead to a complete set of valid and appropriate enhancements to the model. Specifically, it is proposed that the following tasks be undertaken to progress this effort.

a. Further enhance the attack path extension to determine how the different attack difficulty assessments / valuation methods should be combined to determine an overall difficulty value for the attack pattern;

b. Perform more data instantiation for all proposed model extensions to prove the validation of the approach (in the spirit of the work that was done for the network MulVAL extension); and

c. Assess the need for and identify any additional extensions.

As a final point, it is recommended that the work on MulVAL extensions be shared with the MulVAL development team so as to establish a useful dialogue for the further development of extensions and returning feedback to the model originators. Also, any work to be done on model extension should be made in the context of industry best practices and security community resources (e.g. the NVD). This will assist in making the MulVAL analysis relevant for security professionals and accurate for military applications.

7. References

1. Eugen Bacic, Julie Lefebvre, *Dynamic Asset Protection*, October 2005

2. Cinnabar Networks Inc., *Dynamic Asset Protection & Risk Management Abstraction Study*, CSE-5-403, 3 October 2005.

3. Xinming Ou, Sudhakar Govindavajhala, Andrew W. Appel, *Policy-based Multihost Multistage Vulnerability Analysis*, Princeton University, TR-718-04, Dec04.
ftp://ftp.cs.princeton.edu/techreports/2004/718.pdf

4. Xinming Ou, Sudhakar Govindavajhala, Andrew W. Appel, "MulVAL: A Logic-based Network Security Analyzer," Princeton University, *14th USENIX Security Symposium*, Baltimore, Aug05.
http://www.cs.princeton.edu/~appel/papers/mulval.pdf

5. Sudhakar Govindavajhala, *Status of the MulVAL Project*, Princeton University, 31May05.
http://www.cs.princeton.edu/~sudhakar/papers/mulval_preSummer2005.pdf

6. Sudhakar Govindavajhala, Andrew W. Appel, *Windows Access Control Demystified*, Princeton University, 31Jan06.
http://www.cs.princeton.edu/~sudhakar/papers/winval.pdf

7. Skybox View Suite Overview.
http://www.skyboxsecurity.com/data_sheets/Skybox_View_Suite_July 2005.pdf

8. Skybox Secure Datasheet.
http://www.skyboxsecurity.com/data_sheets/Skybox_Secure_July 2005.pdf

9. Skybox Assure Datasheet.
http://www.skyboxsecurity.com/data_sheets/Skybox_Assure_July 2005.pdf

10. Skybox Editions Overview.
http://www.skyboxsecurity.com/data_sheets/editions overview aug 2005.pdf

11. Skybox Spec Sheet.
http://www.skyboxsecurity.com/data_sheets/Skybox View Spec Sheet by Edition Oct 2005.pdf

12. Skybox Deployment Overview.
http://www.skyboxsecurity.com/data_sheets/Skybox_Deployment_Overview Sept 2005.pdf

13. Overview of CycSecure.
http://www.cyc.com/cyc/applications/cycsecure

14. CycSecure Features.
http://www.cyc.com/cyc/applications/features

15. Shepard et al, “A Knowledge-Based Approach to Network Security: Applying Cyc in the Domain of Network Risk Assessment,” *Proceedings of the 17th Innovative Applications of Artificial Intelligence Conference, Menlo Park, CA,* Jul05. http://www.aaai.org/Library/IAAI/2005/iaai05-016.php

This page intentionally left blank.

Annex A – MulVAL, Skybox, and CycSecure Critique

The presentation of the analysis is done in two large tables:

a. Annex A – Table 1 – Tool Set Architecture provides a comparative view of all three tools with regard to: information available, network model collection, vulnerability knowledge, information analysis, information presentation, performance of information collection, performance of information analysis, and scalability; and

b. Annex A – Table 2 – Critique of Toolsets in Providing DAP Abstraction Requirements. This table has identical rows to the MulVAL critique in [2]. The table has three columns for each of MulVAL, Skybox, and CycSecure. The MulVAL critique is largely that provided in [2] with some minor updates. The Skybox and CycSecure critiques are new.

The two tables are presented in landscape format due to the size of their columns.

Table 5. Tool Set Architectures

	MULVAL	SKYBOX	CYCSECURE
Information Available	• Excellent technical information • Refereed paper [4]	• Limited technical information • Marketing Material only [7], [8], [9], [10], [11], and [12]	• Good technical information • Refereed paper [15]
Architecture – Network Model Collection	• All information represented as Datalog facts • Uses Mitre reference OVAL host-based scanner with custom code to translate to Datalog [4] • Policy is manually input and assumed to be small • Hacl/4 appears to be manually input and assumes a fairly static network configuration	• Imports all information from external sources except for the View Dictionary, which comes from Skybox • Supported scanners are predominantly support network-based and not host-based vulnerability scanning • Asset classification can be imported from other [manually generated] asset information sources • Network configuration can be imported from firewalls, routers and load balancers	• Built in network based vulnerability scanner [14] • Uses host-based scanners called Sentinels which are polled from central server [15] • Sentinels gather information on software, hardware, and machine status.
Architecture – Vulnerability Knowledge	• OVAL vulnerability notifications • Datalog parsed from OVAL schema	• Skybox provides the View Dictionary, which contains vulnerability information • No information on sources	• CERT and BugTraq information sources • Requires trained ontologist to adapt knowledge base

Table 5. Tool Set Architectures

	MULVAL	SKYBOX	CYCSECURE
Architecture – Information Analysis	• Prolog logic reasoning based on Datalog facts and Datalog reasoning rules which searches for attack paths	• Attack path analysis reasoning not provided	• Uses Cyc common sense knowledge base and reasoning system [15] • Uses a CERT and BugTraq populated Computing domain knowledge base • Uses a Sentinel (and network scanner?) populated network model knowledge base • Uses Cyc inference engine to reason across 3 knowledge bases
Architecture – Information Presentation	• Command line text output with tracing of attack logic [4] • No GUI output capability	• GUI presentation [8] • Simple attack path diagrams [8]	• Dynamic HTML user interface that is textual in common language [15]
Performance – Information Collection	• Near real-time collection • Collection performed on hosts, therefore parallel operation for arbitrary network size • 236 seconds for a Red Hat Linux 9 host running on a PIII 730MHz processor with 128MB RAM. [4]	• Collection done outside Skybox and imported • Most supported vulnerability scanners are network based, therefore, scan time would increase with network size • No timing information provided	• Collection performed using a non-disruptive network scanner [14] and Sentinels [15] • Network scan time would increase with network size • Multiple Sentinel scanning could be done concurrently • No timing information provided

Table 5. Tool Set Architectures

	MULVAL	SKYBOX	CYCSECURE
Performance – Information Analysis	• Near real-time reasoning using Windows 2.8GHz PC over large networks [4] • 0.22 seconds – 200 hosts • 0.75 seconds – 400 hosts • 3.85 seconds – 1000 hosts • 15.8 seconds – 2000 hosts	• No timing information provided	• "[Attack] Plan generation takes on the order of minutes (or, in the case of very complex plans, a few hours)" [15] • No timing context given (that is, model size, server hardware, etc.)
Scalability	• Large network capable • Appears to have an exponential rise in analysis time	• Scalability information not provided	• Inference engine uses multi-bindings to treat similar instances of vulnerabilities as a class in attack analysis • "dynamic multi-bindings enables [attack] planning times to remain relatively independent of network size." [15] • Large network capable

The following table provides a critique of three potential technologies for providing DAP Abstraction requirements:

Table 6. Critique of Toolsets in Providing DAP Abstraction Requirements

DAP REQUIREMENT	MULVAL CRITIQUE	SKYBOX CRITIQUE	CYCSECURE CRITIQUE
Map mission-required IT Services onto computer network resources	• None of the tools models mission-required IT services explicitly, where an IT service implies data as well as any required processing of that data.		
	• MulVAL models data assets but without explicit value	• Skybox models data assets in monetary value, C/I/A, or risk scale (high/med/low)	• CycSecure does not appear to model data assets [15]
Combine computer network resources into IT Service offerings that support a required confidentiality, integrity, and availability Quality of Service (QoS)	• MulVAL implies that data is the asset needing protecting as evidenced in the policy Datalog fact. • MulVAL does not model asset sensitivity or value. Therefore, all data assets are of equal implied value • MulVAL does model IT services as assets (at least not in their policy facts). However, MulVAL reasoning does identify C/I/A impacts to data, which might be extendable to IT Services.	• Skybox appears to model assets from a C/I/A perspective through the importation of externally defined asset classification data sources. • Asset valuation is implied through impact rules using monetary, risk scale (high – medium – low), C/I/A, or regulatory compliance. • The above marketing literature statement is unclear whether several scales may be used in each of the C/I/A areas.	• CycSecure appears to model asset value since it alludes to identifying attacks “having the most serious overall consequences.” [14] • Calculating consequences is determining impact on assets and this implies knowing asset values. • Conflicting information with [15].
Describe computer network resources as interdependent elements	• MulVAL sufficiently handles the interdependency of hosts, programs, services, data, and network connectivity. • MulVAL does not model network elements explicitly, which can also be sources of vulnerability (for example, firewalls and routers).	• Skybox models the interdependencies of network elements such as: filtering routers, firewalls, hosts, and servers. • It is unclear whether Skybox can explicitly model logical elements internal to a host or server (for example, different processes)	• CycSecure models the interdependency of network elements. [15] • CycSecure models logical elements internal to hosts and servers using Sentinel host-based scanning. [15]

Table 6. Critique of Toolsets in Providing DAP Abstraction Requirements

<table>
<tr><th>DAP REQUIREMENT</th><th>MULVAL CRITIQUE</th><th>SKYBOX CRITIQUE</th><th>CYCSECURE CRITIQUE</th></tr>
<tr><td>Describe vulnerabilities as attributes of computer network resources</td><td>• MulVAL adequately describes vulnerabilities as attributes of software programs.
• Operating system kernel vulnerabilities are modelled as a setuidProgram/3 and a networkServiceInfo/5.
• MulVAL does not model human behaviour as vulnerability in social engineering attacks.</td><td>• Skybox relies on external tools to provide vulnerability information. Skybox supports market leading network vulnerability scanners, which all have limited, if any, host based scanning capabilities.
• Given the dependency on these external tools, Skybox appears to model network visible vulnerabilities, and may be able to model aspects of internal host vulnerabilities.</td><td>• CycSecure collects and models vulnerabilities.
• Ontology includes: 354 classes of software faults, 683 classes of vulnerability, and 12409 computer programs.</td></tr>
<tr><td rowspan="3">Describe security safeguards as attributes of computer network resources</td><td colspan="3">• None of the models appear to handle detection, containment, or recovery safeguards.
• None of the models appear to deal with human behaviour as safeguards.</td></tr>
<tr><td>• MulVAL implies only some prevention safeguards using hacl/4, filePath, accessFile/4, nfsExport/4, and nfsMountTable/5 Datalog facts.
• MulVAL explicitly models data file access rights but not program access controls.
• MulVAL only models owner access controls to data and not world, group, or ACL access controls.</td><td>• It is unclear how or if Skybox models application safeguards.
• With importing data from host-based scanners, or from network management tools, Skybox may be able to model some forms of application safeguards.</td><td>• It is unclear how or if CycSecure models application safeguards.</td></tr>
<tr><td>• MulVAL models all network connectivity in hacl/4 rules, which imply filtering router and firewall configurations.
• MulVAL needs an extension that models network element safeguards explicitly.</td><td>• Skybox models network filtering routers and firewalls explicitly by importing their configurations.</td><td>• It is assumed that Sentinels need to run on routers and firewalls in order to gather network connectivity safeguards as implied in the statement “Sentinels … run on each machine on the target network” [15]</td></tr>
</table>

Table 6. Critique of Toolsets in Providing DAP Abstraction Requirements

DAP REQUIREMENT	MULVAL CRITIQUE	SKYBOX CRITIQUE	CYCSECURE CRITIQUE
Map threat events onto computer network resources with vulnerability and safeguard attributes	• All of the tools explicitly model vulnerabilities as threat events, or steps in an attack. • All of the tools appear to model safeguards as the lack of vulnerabilities in attack paths. • None of the models appear to model threat agent capability		
	• MulVAL models attacker intent using malicious/1 but does not model other attacker motivations (for example, detection avoidance) • Attack paths only start from malicious/1 entities (that is, attackers)	• Skybox models Threat Origins as “profiles of all potential sources of attack” • It is not clear what aspects of threat origins are modelled, or how this information is generated. • It is not clear if Skybox’s attack scenarios are limited to only those originating from known threat origins (versus all possible attack paths).	• CycSecure does not appear to model threat agents explicitly. • It is unclear how CycSecure determines the origins of network attacks, yet an attack plan can be built using the term “hypothetical hacker” [15]
	• MulVAL models the probability of vulnerability exploitation as 1 provided an attacker could gain access to the vulnerability.	• Skybox appears to calculate probabilities of attacks in the statement “The Attack Simulation Engine computes the likelihood of attacks.” • It is not clear whether this likelihood applies to individual vulnerabilities being exploited, or to entire attack paths. • It is not clear how these probabilities are calculated.	• CycSecure appears to model the probability of vulnerability exploitation as 1 provided an attacker could gain access to the vulnerability. • As above, it is not clear how CycSecure determines the starting location of attacks.
Relate threat events to safeguard effectiveness and vulnerabilities	• All of the tools model preventative safeguards only, and generally as a lack of vulnerabilities. • None of the tools seems to model safeguards effectiveness in the face of specific threat events.		

Table 6. Critique of Toolsets in Providing DAP Abstraction Requirements

DAP REQUIREMENT	MULVAL CRITIQUE	SKYBOX CRITIQUE	CYCSECURE CRITIQUE
Show physical and logical network connectivity (as possible attack ingress paths) as graphs of nodes and links	• None of the tools models the Physical Layer. • All of the tools model the Logical Layer.		
	• The MulVAL logical model is fairly complete since it uses a host-based scanner.	• The Skybox logical model is limited by the 3rd party firewall and router configurations, and network vulnerability scanner output.	• The CycSecure logical model is fairly complete since it uses a host-based scanner. • It appears that CycSecure models similar host information to MulVAL ("…programs installed or running … privileges the running programs have … what users are logged into … parameters of critical operations system files") [15].
Map sequences of threat events into threat vectors applied to the computer network resource connectivity	• MulVAL includes a host based vulnerability scanner based on the open source OVAL scanner that outputs Datalog facts according to the MulVAL data model. • MulVAL models attack profiles well, including complex attacks involving both remotely exploitable and local privilege escalation vulnerabilities.	• Skybox has no integral method of determining vulnerabilities but can import network and host based vulnerability scanning data. • It is unclear from Skybox marketing information whether privilege escalation vulnerabilities found by host-based scanners can be integrated into multi-stage attacks, since this requires a uniformity of data model, which may or may not be present in Skybox. The Skybox vulnerability dictionary may carry translation information from supported scanners into a common data model. • Therefore, it is unclear whether Skybox can model attack profiles to the extent that MulVAL can.	• CycSecure includes custom host-based scanners, which are polled from a central server. • CycSecure uses the network model, computing domain (vulnerability), and common sense knowledge bases to reason on attack plans. • The CycSecure ontology is much more complex than MulVAL's so it may have better reasoning on attack paths; however, more information is needed to confirm this hypothesis.
Map areas of responsibility onto IT	• Organizational responsibility is not a model requirement at this time.		

Table 6. Critique of Toolsets in Providing DAP Abstraction Requirements

<table>
<tr><th>DAP REQUIREMENT</th><th>MULVAL CRITIQUE</th><th>SKYBOX CRITIQUE</th><th>CYCSECURE CRITIQUE</th></tr>
<tr><td>Services and computer network resources</td><td>• MulVAL does not model responsibility.</td><td>• Skybox appears to model business units (“[Risk] metrics are consolidated for individual Business Units.”), based on server/application ownership.</td><td>• CycSecure does not model responsibility.</td></tr>
<tr><td rowspan="3">Generally decompose a large computer network into smaller computer networks</td><td colspan="3">• None of the tools explicitly handle network decomposition.</td></tr>
<tr><td colspan="2">• It is assumed that MulVAL and Skybox can model groups of entities as a class in order to simplify the network model, provided their attributes are identical (for example, multiple legitimate users or hosts).</td><td>• CycSecure explicitly models all network elements (for which it has installed Sentinels.
• Classing of elements in attack plan generation is handled dynamically using multi-bindings [15].</td></tr>
<tr><td>• MulVAL implicitly decomposes networking infrastructure into a set of hacl/4 Datalog facts that defines TCP/IP connectivity between any two hosts. Therefore, network makeup (firewalls, routers, links, LANs, etc) is embodied in the hacl/4 rules.</td><td>• Skybox explicitly models networking infrastructure elements.</td><td>• CycSecure appears to model networking infrastructure elements.</td></tr>
<tr><td>Support geographic representations of computer network physical components</td><td colspan="3">• None of the tools models the physical domain.
• Geographic modelling is not seen as a requirement at this time.</td></tr>
<tr><td>Support layered abstraction based on service definitions in order to support coalition networks, joint task force networks, and externally provided computer network services such as Internet Service Providers (ISPs) and satellite providers</td><td colspan="3">• It is not clear if DAP requires modelling abstraction at this time.
• None of the tools appear to support layered model abstractions.</td></tr>
</table>

Annex B – MulVAL Network Extension

Net_rules_schema.P listing:

```
% Copyright (C) 2006 Defence R&D Canada
% MulVAL interaction rules for mapping networks.
% Author : Mike Froh

/*******************************************************/
/****           Primitive predicates               *****/
/*******************************************************/

primitive(hostNet(_host, _net)).
explain(hostNet(Host, Network), Text)
  :- fmt_write_string(Text,
  "The host %S is attached to network %S.",
  args(Host,Network)).

primitive(routeEntry(_router, _initnet, _targetnet, _protocol,
_port)).
explain(routeEntry(Router, InitNet, TargetNet, Protocol, Port), Text)
  :- fmt_write_string(Text,
  "Router %S has a rule which allows communication using protocol %S
to
   destination port %S from subnet %S to subnet %S.",
  args(Router, Protocol, Port, InitNet, TargetNet)).

/*******************************************************/
/****             Derived predicates               *****/
/*******************************************************/

derived(route(_initnet, _targetnet, _protocol, _port)).
explain(route(InitNet, TargetNet, Protocol, Port), Text)
  :- fmt_write_string(Text,
  "A route exists using protocol %S from subnet %S to subnet %S
   to destination port %S.",
  args(Protocol, InitNet, TargetNet, Port)).

derived(hacl(_ihost, _thost, _protocol, _port)).
explain(hacl(InitHost, TargetHost, Protocol, Port), Text) :-
       fmt_write_string(Text,
   "Host %S can initiate %S communications to Host %S on port %S.",
   args(InitHost, Protocol, Port, TargetHost)).
```

Net_rules.P listing:

```
% Copyright (C) 2006 Defence R&D Canada

% MulVAL interaction rules for mapping networks.
% Author : Mike Froh

:- table route/4.
:- table hacl/4.

/****** Route Section *******/

interaction_rule(
   (route(InitNet, TargetNet, Protocol, Port) :-
       routeEntry(Router, InitNet, TargetNet, Protocol, Port),
       hostNet(Router,InitNet),
       hostNet(Router,TargetNet)),
   'Direct route between subnets through an intermediate router').

interaction_rule(
   (route(InitNet, TargetNet, Protocol, Port) :-
       route(InitNet, TransitNet, Protocol, Port),
       route(TransitNet, TargetNet, Protocol, Port)),
   'Transitive routing through an intermediate network').

/****** HACL Section *******/

interaction_rule(
   (hacl(InitHost, TargetHost, Protocol, Port) :-
       hostNet(InitHost, InitNet),
       hostNet(TargetHost, TargetNet),
       InitNet \= TargetNet,
       route(InitNet, TargetNet, Protocol, Port)),
   'Hosts can only communicate between networks through a valid
route.').

interaction_rule(
  (hacl(InitHost, TargetHost, _, _) :-
       hostNet(InitHost, CommonNet),
       hostNet(TargetHost, CommonNet)),
  'Hosts on same network have no communication restrictions.').

/*  I was thinking of automatically deriving the hostNet predicates
    for routers/firewalls from the routeEntry statements.  But this
    means we need to define a primitive hostNet and a derived
    hostNetprime and render all hostNet to hostNetprime while
    deriving router/firewall hostNetprime from routeEntry predicates.

    This will not work as currently defined!

interaction_rule(
  (hostNet(Router,Net) :-
       routeEntry(Router, Net, _, _, _);
       routeEntry(Router, _, Net, _, _)),
  'Routers are hosts on their network interfaces.').

*/
```

net_test.P listing showing the network in :

```
/*** Net A ***/
hostNet(a1, neta).
hostNet(a2, neta).
hostNet(multi, neta).

/*** Net B ***/
hostNet(b1, netb).
hostNet(b2, netb).
hostNet(multi, netb).

/*** Net C ***/
hostNet(c1, netc).
hostNet(c2, netc).

/*** Net D ***/
hostNet(d1, netd).

/*** Router between nets A & B ***/
hostNet(routerAB, neta).
hostNet(routerAB, netb).
routeEntry(routerAB, neta, netb, tcp, 80).
routeEntry(routerAB, netb, neta, tcp, 433).

/*** Router between nets B & C & D ***/
hostNet(routerBCD, netb).
hostNet(routerBCD, netc).
hostNet(routerBCD, netd).
routeEntry(routerBCD, netb, netc, tcp, 80).
routeEntry(routerBCD, netb, netc, tcp, 433).
routeEntry(routerBCD, netb, netd, tcp, 80).
routeEntry(routerBCD, netc, netd, tcp, 80).

/*** test entering a derived predicate ***/
/*  derived route predicate was ignored in mulVAL
route(neta,netd,udp,500).
*/
```

The hacl/4 for the network in Figure 1 were derived using the following XSB command:

```
stdout2file(visualize(hacl(Init,Target,Protocol,Port),text),’attack.tx
t’).
```

The output file was pretty printed using the following bash command line:

```
cat attack.txt | grep hacl | sed 's/.*\(hacl(.*)\).*/\1/' | sed
's/_h[0-9]*/Any/g' | sort >attack_pp.txt
```

The 131 hacl/4 rules derived were as follows:

```
hacl(a1,a1,Any,Any)
hacl(a1,a2,Any,Any)
hacl(a1,b1,tcp,80)
hacl(a1,b2,tcp,80)
hacl(a1,c1,tcp,80)
hacl(a1,c2,tcp,80)
hacl(a1,d1,tcp,80)
hacl(a1,multi,Any,Any)
hacl(a1,multi,tcp,80)
hacl(a1,routerAB,Any,Any)
hacl(a1,routerAB,tcp,80)
hacl(a1,routerBCD,tcp,80)
hacl(a2,a1,Any,Any)
hacl(a2,a2,Any,Any)
hacl(a2,b1,tcp,80)
hacl(a2,b2,tcp,80)
hacl(a2,c1,tcp,80)
hacl(a2,c2,tcp,80)
hacl(a2,d1,tcp,80)
hacl(a2,multi,Any,Any)
hacl(a2,multi,tcp,80)
hacl(a2,routerAB,Any,Any)
hacl(a2,routerAB,tcp,80)
hacl(a2,routerBCD,tcp,80)
hacl(b1,a1,tcp,433)
hacl(b1,a2,tcp,433)
hacl(b1,b1,Any,Any)
hacl(b1,b2,Any,Any)
hacl(b1,c1,tcp,433)
hacl(b1,c1,tcp,80)
hacl(b1,c2,tcp,433)
hacl(b1,c2,tcp,80)
hacl(b1,d1,tcp,80)
hacl(b1,multi,Any,Any)
hacl(b1,multi,tcp,433)
hacl(b1,routerAB,Any,Any)
hacl(b1,routerAB,tcp,433)
hacl(b1,routerBCD,Any,Any)
hacl(b1,routerBCD,tcp,433)
hacl(b1,routerBCD,tcp,80)
hacl(b2,a1,tcp,433)
hacl(b2,a2,tcp,433)
hacl(b2,b1,Any,Any)
hacl(b2,b2,Any,Any)
hacl(b2,c1,tcp,433)
hacl(b2,c1,tcp,80)
hacl(b2,c2,tcp,433)
hacl(b2,c2,tcp,80)
hacl(b2,d1,tcp,80)
hacl(b2,multi,Any,Any)
hacl(b2,multi,tcp,433)
hacl(b2,routerAB,Any,Any)
hacl(b2,routerAB,tcp,433)
hacl(b2,routerBCD,Any,Any)
hacl(b2,routerBCD,tcp,433)
hacl(b2,routerBCD,tcp,80)
hacl(c1,c1,Any,Any)
hacl(c1,c2,Any,Any)
hacl(c1,d1,tcp,80)
hacl(c1,routerBCD,Any,Any)
hacl(c1,routerBCD,tcp,80)
hacl(c2,c1,Any,Any)
hacl(c2,c2,Any,Any)
hacl(c2,d1,tcp,80)
hacl(c2,routerBCD,Any,Any)
hacl(c2,routerBCD,tcp,80)
hacl(d1,d1,Any,Any)
hacl(d1,routerBCD,Any,Any)
hacl(multi,a1,Any,Any)
hacl(multi,a1,tcp,433)
hacl(multi,a2,Any,Any)
hacl(multi,a2,tcp,433)
hacl(multi,b1,Any,Any)
hacl(multi,b1,tcp,80)
hacl(multi,b2,Any,Any)
hacl(multi,b2,tcp,80)
hacl(multi,c1,tcp,433)
hacl(multi,c1,tcp,80)
hacl(multi,c2,tcp,433)
hacl(multi,c2,tcp,80)
hacl(multi,d1,tcp,80)
hacl(multi,multi,Any,Any)
hacl(multi,multi,tcp,433)
hacl(multi,multi,tcp,80)
hacl(multi,routerAB,Any,Any)
hacl(multi,routerAB,tcp,433)
hacl(multi,routerAB,tcp,80)
hacl(multi,routerBCD,Any,Any)
hacl(multi,routerBCD,tcp,433)
hacl(multi,routerBCD,tcp,80)
hacl(routerAB,a1,Any,Any)
hacl(routerAB,a1,tcp,433)
hacl(routerAB,a2,Any,Any)
hacl(routerAB,a2,tcp,433)
hacl(routerAB,b1,Any,Any)
hacl(routerAB,b1,tcp,80)
hacl(routerAB,b2,Any,Any)
hacl(routerAB,b2,tcp,80)
hacl(routerAB,c1,tcp,433)
hacl(routerAB,c1,tcp,80)
hacl(routerAB,c2,tcp,433)
hacl(routerAB,c2,tcp,80)
hacl(routerAB,d1,tcp,80)
hacl(routerAB,multi,Any,Any)
hacl(routerAB,multi,tcp,433)
hacl(routerAB,multi,tcp,80)
hacl(routerAB,routerAB,Any,Any)
hacl(routerAB,routerAB,tcp,433)
hacl(routerAB,routerAB,tcp,80)
hacl(routerAB,routerBCD,Any,Any)
hacl(routerAB,routerBCD,tcp,433)
hacl(routerAB,routerBCD,tcp,80)
hacl(routerBCD,a1,tcp,433)
hacl(routerBCD,a2,tcp,433)
hacl(routerBCD,b1,Any,Any)
hacl(routerBCD,b2,Any,Any)
hacl(routerBCD,c1,Any,Any)
hacl(routerBCD,c1,tcp,433)
hacl(routerBCD,c1,tcp,80)
hacl(routerBCD,c2,Any,Any)
hacl(routerBCD,c2,tcp,433)
hacl(routerBCD,c2,tcp,80)
hacl(routerBCD,d1,Any,Any)
hacl(routerBCD,d1,tcp,80)
hacl(routerBCD,multi,Any,Any)
hacl(routerBCD,multi,tcp,433)
hacl(routerBCD,routerAB,Any,Any)
hacl(routerBCD,routerAB,tcp,433)
hacl(routerBCD,routerBCD,Any,Any)
hacl(routerBCD,routerBCD,tcp,433)
hacl(routerBCD,routerBCD,tcp,80)
```

List of symbols/abbreviations/acronyms/initialisms

AI	Artificial Intelligence
CVSS	Common Vulnerability Scoring System
DAP	Dynamic Asset Protection
dmz	demilitarized zone
DND	Department of National Defence
DRDC	Defence Research & Development Canada
DRDC	Defence Research and Development Canada
ICAT	
MulVAL	Multihost, multistage Vulnerability Analysis
NATP	Network Address Translation, Port
NIST	National Institute of Science and Technology
NVD	National Vulnerability Database
OVAL	Open Vulnerability and Assessment Language
QoS	Quality of Service
SNMP	Simple Network Management Protocol
SYN	[TCP] SYNchronization packet
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Distribution List

DRDC Ottawa CR 2006-251

Internal Distribution

3	Author
1	Library

Total Internal Copies: 4

Total Copies: 4

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Bell Security Solutions Inc.
333 Preston
Ottawa, ON K1S 5N4

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

MulVAL Extensions for Dynamic Asset Protection (U)

4. AUTHORS (Last name, first name, middle initial)

Bacic, Eugen ; Froh, Michael ; Henderson, Glen

5. DATE OF PUBLICATION (month and year of publication of document)

April 2006

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

51

6b. NO. OF REFS (total cited in document)

15

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DEFENCE R&D CANADA - OTTAWA
3701 Carling Avenue, Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

15bo03

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

W7714-5-3247

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Ottawa CR 2006-251

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Distribution limited to defence departments and defence contractors; further distribution only as approved
(   ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
(   ) Distribution limited to government departments and agencies; further distribution only as approved
(   ) Distribution limited to defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This paper documents research into extensions to the Multihost, Multistage Vulnerability Analysis (MulVAL) framework to support DRDC efforts to develop a feasible abstraction in the area of defensive posture technology. The results presented in this paper demonstrate that the MulVAL model is extensible and can be enhanced to include additional data representation and analysis features to tailor the model to meet the need of the DND defence community. The extensions evaluated in this effort have been shown to be both technically valid given the capabilities of logic-based programming and appropriate given the current model data representations. The primary extensions researched as part of this work are: improved representation of network path constructs and assignment of value to data assets in the model. This paper documents a substantial degree of progress in the development of each of the proposed MulVAL extensions.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Network Security, Security Analysis, Deductive Reasoning, Datalog

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

www.drdc-rddc.gc.ca