Dynamic Asset Protection & Risk Management Abstraction Study

G. Henderson
Cinnabar Networks Inc.

E. Bacic
Cinnabar Networks Inc.

M. Froh
Cinnabar Networks Inc.

Cinnabar Networks Inc.
265 Carling Ave., Suite 200
Ottawa, ON K1S 2E1

Contract number: W7714-5-3138

Contract Scientific Authority: Julie Lefebvre

Defence R&D Canada – Ottawa

Contract Report

DRDC Ottawa CR 2005-205

November 2005

Dynamic Asset Protection & Risk Management Abstraction Study

Authors:	**G. Henderson, E. Bacic, M. Froh**
Company name and address:	**Cinnabar Networks Inc.** **265 Carling Ave., Suite 200** **Ottawa, ON K1S 2E1**
Contract number:	**W7714-5-3138**
Scientific Authority:	**Dr. Julie Lefebvre, H/NIO** **613-990-7094**
Research Centre:	**DRDC Ottawa**
DRDC Document Number:	**DRDC Ottawa CR-2005-205**
Cinnabar Document Number:	**DRD-5-038**
Client File Number:	**Nil**
Document Date:	**November 7, 2005**
Classification:	**Unclassified**

TABLE OF CONTENTS

1 Introduction **1**
1.1 Problem Space 1
1.2 Scope 1
1.3 Background 2
2 Research Results **4**
2.1 Existing Work 4
2.1.1 Simulation Modelling 4
2.1.2 Rog-O-Matic 5
2.1.3 Counterplanning 6
2.1.4 MulVAL 7
2.1.5 Modelling Concepts Summary 9
2.2 Dynamic Asset Protection Systemic Issues 10
2.2.1 Information Warfare Context 10
2.2.2 Elements of a Successful Model 12
3 Architecture Model **16**
3.1 Rog-O-Matic as the DAP Architecture Model 16
3.1.1 Model Description 16
3.1.2 Model Rationalization 17
3.1.3 Simulated Environment 19
4 DAP Model Abstraction **21**
4.1 Requirements of the DAP Model Abstraction 21
4.2 Evaluation of MulVAL's Model for DAP Use 23
4.2.1 MulVAL Architecture Model 23
4.2.2 How MulVAL Handles DAP Related Elements 25
4.2.3 Critique of MulVAL in Providing DAP Abstraction Requirements 26
4.3 Attack Patterns Within a Layer 29
4.3.1 Determining Asset Impact 32
4.4 Attack Profiles Within a Layer 34
4.4.1 Social Layer 34
4.4.2 Logical Domain 34
4.4.3 Physical Layer 35
4.5 Attack Profiles Between Layers 35
4.6 Dynamic Aspects of the Abstraction 38
5 Abstraction Implementation Issues **39**
5.1 Future Research 39

5.2 Next Steps 40
6 References 41
Annex A: Applicable Standards 43

1 INTRODUCTION

This paper presents a study of a dynamic risk management model for asset protection. The intent of this effort is to define and develop a feasible abstraction in the area of defensive posture technology. This abstraction will provide the groundwork and required elements, based on generally available technology, so as to provide the necessary framework for the creation of a modeling system for dynamic risk management and asset protection.

Within this research, the most pressing question will be how to quantify values pertaining to each asset, safeguard, and threat in such a way as to provide input to the abstraction and, hence, to the a future usable model. Furthermore, how much of that quantified data needs to be defined or enhanced in order to bring forward adequate data for the model to be effective and dynamic.

1.1 PROBLEM SPACE

It has been recognized that existing methods for performing risk management, such as penetration testing or threat and risk assessments, suffer from the following limitations:

1) Their effectiveness as a decision making tool decreases over time as changes are observed in the:
    a) Target environment (systems, services, data); or
    b) Threat environment (new vulnerabilities, new attack techniques);
2) There is substantial effort to repeat the analysis in order to keep an accurate representation o f the target environment;
3) Their ability to determine the overall impact, in terms of threat propagation, of an of a security incident is limited; and
4) They cannot be used in hypothetical analyses to assess the impact of potential changes to the internal or external environment.

As such, a new approach to risk management is needed. This new approach must be able to monitor a dynamic environment and provide a more comprehensive view of the defensive posture presented by an organization, including the impact of vulnerability exploitation as it pertains to the propagation of a threat.

1.2 SCOPE

This work will determine and define a feasible abstraction for dynamic asset protection (DAP) and risk management. This investigation includes the following areas as a focus for research in defining this model.

1) This work leverages existing research into Information Warfare (IW) and Computer Network Defence (CND) Situational Awareness (SA). As such, this work takes a view that is primarily based on the military operational environment. Nevertheless, the model put forward in this paper can be equally applied to non-military and non-governmental environments.
2) The model put forward by this paper includes elements that have been previously identified, in existing DRDC sponsored research, as relevant in the area of dynamic risk management. These elements include:

a) The unified view of threat/vulnerability/asset triplet;
b) The dynamic protection of assets; and
c) The use of automation.

3) This paper will define a way forward in terms of potential future work and future research. Furthermore, the goal of the white paper will be to define a viable abstraction and also delineate what is presently feasible with today's technology.

1.3 BACKGROUND

As a prelude to this paper, a summary of some of the significant points derived from existing research, sponsored by DRDC, is provided.

1) For a model to successfully assess the risk posture presented by an organization, it must ensure that the safeguards that are in place to protect against potential attack are in proportion to the value of the assets being protected. A low value asset will not need to be protected to the same degree as a higher value one. The CIA construct (confidentiality, integrity, availability) used to define the statement of sensitivity to determine the value of assets is a potential mechanism by which asset value may be modeled.

2) Threats must be modeled from two perspectives:
   a) The impact to assets under attack; and
   b) The impact in terms of propagation of the threat further into the target environment.

   The second perspective maps closely the view a penetration tester will take when analyzing the security posture of a system or network. Successful vulnerability exploitation will give the tester more information about the target environment, more privileges to launch additional attacks or new routes of attack for compromising the environment. In effect, each successful exploit ratchets the level of infiltration until, ultimately, the system is compromised. Any model that attempts to model attack patterns against a target environment must be able to perform this decision-making action to use the knowledge that has been gained through successful exploits against the defensive mechanisms that have been put in place to protect the target environment.

3) There is a high degree of variation in terms of the capabilities and motivations of an attacker. It is possible for an attacker to have full, partial or no knowledge of the target environment. Similarly, an attacker will be motivated by specific characteristics:
   a) The desire to reach a known target as opposed to performing a general reconnaissance of the network environment; and
   b) The desire for speed rather than stealth in performing the attacks.

4) Additionally, following the work of Lefebvre and Bacic [Dap04], the following observations have been used as a starting point to drive the research into an appropriate abstraction for dynamic asset protection:
    a) McCumber's [McCum] notion that any given safeguard can have technical, procedural and people/HR aspects must be considered in defining the characteristics of the target and threat environment of a DAP model.
    b) There is a clear distinction between inherent risk and risk reduced by safeguards.
    c) That safeguards and vulnerabilities are opposite sides of the same coin (that is, safeguards are in direct response to a given vulnerability).
    d) Time must be part of any viable solution.
    e) There are logical layers including: Social, Application, System/Network, Data Link[1], and Physical.
    f) Defences should be a balanced ratio between vulnerability and asset value.

The remainder of this paper is divided into the following sections:

Section 2:Research Results: This section provides a summary of some existing research in the area of dynamic asset protection, specifically including any models and implementations that have been created to assist the risk management task. From this existing work, a set of desirable characteristics for any successful DAP model is defined.

Section 3:Architecture Model: This section provides a high-level description of a suggested model for dynamic asset protection. This section focused on the operational characteristics of the model in terms of interaction with the environment, data flow and information sources.

Section 4:DAP Model Abstraction: This section provides a more detailed description of the data model abstraction for capturing the nature of the target environment in terms of assets, threats and safeguards.

Section 5:Table 8: Dynamic Aspects of the DAP Model Abstraction

Abstraction Implementation Issues***:*** This section points to future research that is needed to more completely define a viable DAP model with the intent to develop a workable prototype. Both general research suggestions and specific next steps are identified.

[1] Note that this was called the Logical Layer in Lefebvre and Bacic [LefBac] but renamed here to prevent confusion with the revised Logical Layer.

2 RESEARCH RESULTS

2.1 EXISTING WORK

As part of the mandate for this investigation into an appropriate model for dynamic asset protection a review of existing work in this area was performed. It was determined that existing research into the following general categories provided useful input to the development of the DAP model:

1) Simulation;
2) Automation into solving exploration problems;
3) State modeling of IT attack and defense; and
4) Detection, analysis and response to risks in IT infrastructure.

The following sections describe the most relevant research initiatives that touch on these topics. These research projects, which range from theoretical modeling approaches to actual deployed solutions, were evaluated from the perspective of identifying common modeling characteristics and constraints.

2.1.1 SIMULATION MODELLING

A summary of existing research that may provide insight into the development of a model for Dynamic Asset Protection may properly begin with simulation studies. That is, there have been efforts to model the attacks on and defence of information assets through classical simulation methodologies. It is recognized that IW poses a challenging problem space for simulation activities. Elements which factor into the problem space, including the involvement of human activities, highly interconnected network architectures and a large number of potential software vulnerabilities, define a highly complex environment. The search space, that is, the number of possible paths through the simulated environment, is very large for such a complex environment, yet the elimination or omission of paths may miss a significant result.

As is true for any simulation effort that must model a complex environment, a balance between performance and specificity must be achieved. It must be possible to create, test and analyze the model within a reasonable time constraints to generate a usable tool for IW purposes. Performance improvements can be achieved by simplifying the model through data set aggregation, categorization and inheritance. There is concern; however, that the value of the information gained from such a simplified model decreases with the corresponding reduction in granularity over the modeled elements.

Additionally, classical simulation practices do not easily allow for some of the significant aspects of the environment to be modeled such as:

1) How to model the impact of events that occur;
2) How to model a systemic view of the protection mechanisms in the environment;
3) Dynamic changes in the IT community (new vulnerabilities, attack methods, etc)

One such simulation study was put forward in *Simulating Cyber Attacks, Defenses and Consequences*[i]. This paper describes a model that categorized classes of threats, attack mechanisms and safeguards. Linkages were created within the model to associate threats

with attacks and attacks with defenses. Additional associations were added to factor in consequences (in terms of impact on integrity, availability, access and disclosure), incident handling (in terms of prevention, detection, and response) and the capabilities of the attacker.

In addition to these data components, the simulation was governed by metrics which drove the behaviour of the modeled components including time values (for attack, detection and response), effectiveness of safeguards to protect against attacks and attacker/defender skill levels. The metrics were implemented as a set of statistical functions enhanced with stochastic behaviour, similar to war-gaming models.

Based on the model and programmed characteristics, a sample simulation run would randomly chose a feasible (non-looping) attack path from the source to the destination. Feasible attack paths are based on the capabilities of the attacker and any strategic attack information that is provided as input, such as the password on a certain system. For each run, time is the critical factor that determines a winning or losing scenario. If the safeguards are able to prevent, detect or respond to the attacker, or if the maximum time is exceeded, the scenario is lost from the perspective of the attacker.

While providing interesting results, classical simulation suffers from the significant limitation of being dependant on stochastic processes. These elements in the model, which are necessary to drive the simulation in the absence of real metrics, result in an inaccurate tool for decision-making purposes. One example of this is attack path selection. While the model determines this randomly, a true attacker will be driven to specific attack paths based on experience and goals (a particular attack may be governed by the need for speed, stealth or least resistance). Additionally, the static nature of simulation models and the level of categorization that must be achieved to create the simulation make this approach of limited usefulness.

2.1.2 ROG-O-MATIC

Rog-O-Matic [Rog84] is an expert system[2] that was developed to explore an environment that was defined by a set of specific rules. This environment was the computer game Rogue[3]. The goal of the Rog-O-Matic effort was to create an expert system with the ability to solve an exploration problem. Exploration problems can be defined by the following characteristics:

1) The environment to be explored can be seen as an undirected planar graph,
2) It is possible to define a starting point, or node, within this graph from which the exploration activities can be initiated,
3) There must be a mechanism by which it is possible to observe connected nodes and transition between nodes.

The game, Rogue, was seen to include all necessary elements to be considered as an exploration task. The game generates terrain to be explored, the player is provided with a starting point within this environment and the game's model allows the explorer to move about the environment from node to node. Exploration activities must include the ability to react to adversaries that attempt to prevent the explorer from reaching further into the environment.

[2] Note that in this context, an expert system is a set of heuristics collected for specialized decision making as opposed to more recent views on machine learning and artificial intelligence which to not factor into the Rog-O-Matic model.
[3] For details regarding Rogue, please refer to http://www.wichman.org/roguehistory.html (A brief history of Rogue)

Rog-O-Matic was developed as an expert system in that it includes production rules for making decisions on the best course of action for a given scenario. It also draws upon algorithmic knowledge for such solution elements such as optimal path calculation. However, Rog-O-Matic differs from traditional expert systems in that it has the ability to work within a dynamic environment, for example the randomly generated terrain and adversaries. More importantly, the system was designed to operate in spite of limited information, recording and integrating knowledge about the environment as it is discovered.

A statement of the success of Rog-O-Matic can be made when its performance is compared with human players. During a period of evaluation against a small group of human players, Rog-O-Matic did not achieve the best overall score but did succeed in gaining the highest median score of any player.

Within the context of the DAP model problem space, network attack and defence can similarly be viewed as an exploration task. Interconnections between systems and networks can generate the equivalent of terrain to be explored. An attacker can be modeled as originating at any point in this environment, both internally or externally. Successfully exploiting vulnerabilities in the environment will allow the attacker to detect new targets of opportunity, furthering the exploration activities. A useful component to the exploration task model is that the overall goal of the activities can be dynamically set, for example, to completely traverse the target environment or to locate the most direct path to a particular target. Additionally, the dynamic nature of the Rog-O-Matic solution is a characteristic which will aid in solving any exploration problem and should be present in the DAP model. This dynamic nature for the DAP model will allow the solution to adapt to items such as:

1) The presence of new safeguards
2) The discovery of new vulnerabilities
3) The determination of new methods by which attacks can be executed
4) Changes in the time-sensitivity of information assets

The primary focus of the Rog-O-Matic production rules is to detect, respond and react to adversaries. Within the DAP model, adversaries can be equated to safeguards, that is, environmental elements that attempt to prevent exploration. This focus on safeguards as one of the core model components is in accord with the McCumber Cube model.

As a final note, the concept of Rog-O-Matic being able to operate in the absence of information about the target environment is a useful element which will have a bearing in the discussion of the dual Computer Network Attack/Defence nature of the DAP model (see section *2.2.2.3:Dual Computer Network Attack/Defence Nature)*

2.1.3 COUNTERPLANNING

Counterplanning activities are defensive tactics designed to foil attack plans. Research into counterplanning as it applies to Computer Network Defence (CND) activities is exemplified in the paper *Counterplanning Deceptions to Foil Cyber-Attack Plans* [Rowe]. In essence, it is suggested that the use of ploys, a counterplanning method, to deceive an attacker and provide sufficient resistance to attack so as to render targets of opportunity seemingly unreachable. Ploys can include the deletion, addition or alteration of the facts that are presented to an attacker to invalidate the attacker's immediate or ultimate goals.

The author describes a tool created for the purpose of identifying the most cost-effective set of ploys to apply to a network environment so as to achieve a desired defensive posture. This tool, MECOUNTER, utilizes machine learning to anticipate attack plans that would be used by an attacker and where ploys should be implemented. It is notable that selection, placement and presentation of appropriate ploys are all relevant of the defensive posture to ensure that the attacker is thwarted without realizing the deceptive tactics are being employed.

The MECOUNTER model reduces complex attack patterns into individual actions that must be sequentially followed to execute a specific attack plan; the example given as part of the research paper was the installation of a rootkit tool. The ultimate goal of counterplanning is to select and place the most effective ploys into the environment so as to disrupt the actions that constitute the attack plans. The modelling approach examines the impact of each applicable ploy at each stage in the attack plan under a number of conditions such as different start and end nodes. From this analysis it is possible to create a Markov state model with state transition probabilities and the expected time to the goal state. An inference process examines the relative amount of 'damage' each ploy causes to the attacker, for example, how many states must the attacker re-visit to repair the damage caused by the ploy or has the ploy resulted in total invalidation of the execution of the attack plan.

Given this list of potentially useful ploys, a decision theory process is used to identify the most effective ploys given such constraints as:

1) Use of too many ploys increases the probability that an attacker will detect that deception tactics are being used;
2) Not all ploys will succeed and the attack will proceed in spite of the defensive tactics; and
3) Ploys may impact legitimate system use.

The MECOUNTER model factors these elements into an expression of the expected benefit of a specific ploy at a specific point in the attack plan.

While counterplanning provides a view on attack and defence strategies using well known modelling and decision-making theory, there remains questions as to the validity of the approach to create a model that can be used in an IW context. Counterplanning is focussed on specific attack methodologies and each attack plan and associated defence strategy must be modelled separately. There is benefit to the approach that was taken as part of this model that creates categorized attack plans, especially since these attack plans will not change as frequently as the list of known software vulnerabilities. However, the effort needed to create these attack plans is substantial and the existence of alternate attack plans, particularly in highly interconnected environments, will likely lead to significant results being missed.

2.1.4 MULVAL

The team that has developed Multi-host, Multistage Vulnerability Analysis, a network security analyzer, takes a more pragmatic view of vulnerability analysis tools [MulVal]. MulVAL has been designed to determine the impact that software vulnerabilities have on a target network. To ensure that this tool is a valid solution, the following features have been of significant importance during the development:

1) The tool should use formal vulnerability specifications from the IT community; and
2) The tool should be able to scale to large networks without performance degradation.

To achieve these factors, the MulVAL solution uses information from the following input sources, in an industry standard format, for its analysis:

1) An Open Vulnerability Assessment Language (OVAL) scanner for vulnerability detection and host configuration
2) Host Access Control Lists (HACL) for network and interconnectivity configuration

These inputs are converted to Prolog-like expressions for inclusion into the MulVAL world model and this model is enhanced with additional environmental expressions to define: user capabilities, access rights and interaction between all elements in the model. After defining a complete world model, MulVAL proceeds to apply reasoning rules to develop attack trees that correspond to complete specifications of means by which a system can be compromised. The reasoning engine is able to assess the degree to which the network environment can be compromised through propagation of attacks, once an exploit has been successfully applied. Compromise propagation is examined in terms of network service exploitation and multi-hop network access.

MulVAL includes the ability to perform what-if analysis to assess the robustness of the network environment in the face of a new vulnerability or threat. Once the world model has been created via the input sources, simulated software vulnerabilities can be introduced to the model to determine the impact of successful exploitation of these vulnerabilities. Alternatively, MulVAL can document the impact should a privileged user be compromised.

MulVAL includes an OVAL scanner as part of the solution space and this scanner is deployed to each of the systems in the target environment. In this way, each scanner is able to operate in parallel gathering information about the local system and reporting this information back to the MulVAL reasoning engine. The engine itself is able to perform analyses on networks with thousands of hosts in less than a minute, using typical hardware (Pentium 4/2.8Ghz, 512 Meg RAM).

The MulVAL model provides a large portion of the functionality needed by the DAP model. Specifically, the need to leverage industry standards source, like OVAL, and the ability to scale to large networks are seen as critical success factors for the DAP model. Recognized limitations to MulVAL which will be significant factors in the development of the DAP model include the following.

1) There is no representation of the role of safeguards in resisting attack.
2) There is no representation for the exploits whose consequences impact the confidentiality, integrity or availability of information assets. The focus of the attacks that are analyzed by MulVAL are privilege escalation attacks or denial of service attacks.
3) There is no representation of the time-sensitivity as it pertains to the confidentiality, integrity or availability of information assets.

The fact that the MulVAL model has been implemented and has been seen to provide valuable vulnerability analysis is confirmation that the intent and goals of the DAP model are feasible and valid.

2.1.5 MODELLING CONCEPTS SUMMARY

Based on the research listed above it is believed that the following elements must be considered for the architectural design of any DAP model. The selection of these specific elements from previously developed models can be attributed to the following rationale:

1) The modelled element captures a significant aspect of the DAP asset/threat/safeguard environment; or
2) The modelled element lends itself to the creation of a viable tool, enhancing the tool ability to remain up to date, to scale or to perform within desired operational limits.

With this selection rationale in mind, the following elements are seen as desired characteristics for the DAP model.

Automated collection of information: Given the dynamic nature of the problem, a DAP model will be more successful as it is able to obtain information directly from the target environment. This includes an active scanning capability and the ability to gain and use up to date information regarding IT vulnerabilities and their consequences.

Scalability: As the environment to be modelled may be large, the ability to perform information collection and analysis must be able to scale. The results of the analysis must be sufficiently fast to allow for a real-time response within a window of acceptability. If the DAP model cannot meet these requirements that threat posed by undetected vulnerabilities will negatively impact the security posture of the network environment.

Use of Standards: As has been identified, the ability to have the DAP model update automatically and integrate new vulnerabilities in the detection and analysis process will assist in ensure that the model provides an accurate representation of the security posture of the environment. In order to include the automatic update component, the model will draw upon industry information sources. There has been a trend to expressing this information in an industry-sanctioned format for ease of communication between vendors. These standards should be leveraged by the DAP model component to easily interpret this information as well as leveraging existing tools that use these data formats. A description of applicable standards is provided in Annex A: Applicable Standards.

Feedback: As the goal of the DAP model is to determine the security posture of the target environment from both an attack and defence perspective, the successful model will implement logic functions which mimic an attacker's reasoning. As such, the model will institute a feedback mechanism that, at its simplest, alternates between vulnerability exploitation and planning the next stage in the attack. Information that is gained from an exploited vulnerability provides information to the logic function for aiding in furthering the attack. The planning function dictates what next vulnerability should be exploited to further the attack. This cycle of attack/analyze allows for greater flexibility in defining the parameters around the DAP model (starting point, previous known information, privileges) since the logic function can be called upon given any state of the model.

Attacker Goals: It should be possible to assign intent to an attacker in terms of speed versus stealth and general network traversal (reconnaissance) versus a directed attack against a specific asset,

Attacker Capabilities: It should similarly be possible to assign a level of competence to an attacker such that certain attack profiles are beyond the attacker's capability set capability set.

What-if analysis: The model must be able to accept hypothetical scenarios to improve the decision-making response based on perceived threat. These scenarios may include:

- A compromised internal user (administrator) versus an external attacker;
- New vulnerability which renders a system susceptible to attack; and
- The impact of adding safeguards to the model.

Learning Component: The logic function must be able to adapt over time. This can be achieved by either having the model adapt its decision-making rules based on experience or updating the rules through a manual re-structuring/re-stating of the rules by an expert in the field. Having the logic function which can be adapted is not learning in the strictest sense, however, it has the advantage of ensuring that the DAP model accurately reflects the knowledge and behaviour of a real attacker. The machine learning option is difficult to achieve at this time given the lack of metrics with which to tune the logic functions.

Save State: It should be possible to save the model state for archival and trend analysis purposes.

These model characteristics, taken from existing model implementations in the problem space, are seen as providing useful guidance in the development of a successful DAP model.

2.2 DYNAMIC ASSET PROTECTION SYSTEMIC ISSUES

This section provides a discussion of the perceived requirements that will drive the definition of an architectural model of the DAP solution. These requirements have been collected based on previous work on Computer Network Defence Situational Awareness [Cndsa], project discussions and research into existing models.

2.2.1 INFORMATION WARFARE CONTEXT

Previous work in the field of Computer Network Defence (CND) Situational Awareness (SA) has shown the value of a robustly networked force. Through the timely sharing of accurate information, mission objectives can be more effectively achieved through enhanced:

1) Collaboration;
2) Self-synchronization;
3) Sustainability; and
4) Decision-making response time.

However, leveraging a computer network (CN) to gain strategic and tactical advantage in a military scenario requires that force commanders retain a position of information superiority. In this context, information superiority can be taken to mean the ability to acquire, exploit and disseminate an uninterrupted flow of information while denying an adversary's ability to do the same [DND98, DoD00]. Existing research in the area of situational awareness has attempted to map SA actions to the traditional military command and control cycle, resulting in the following actions:

- **Observe**: the environmental sensing function that gathers raw information from the target environment;

- **Orient**: the analysis function that assesses the risk/impact associated with information that is observed in the target environment;
- **Decide**: the function which determines the course of action (COA) based on mission goals and the core results from the orientation phase; and
- **Act**: Selection and implementation of a specific COA.

It is interesting to note that two of the modelling approaches documented in the previous section map to this OODA/SA cycle.

Model	Rog-O-Matic	MulVal	Dynamic Nature
Observe	Uses a sensory interface to extract information from the environment and introduce it to the world model	Uses the distributed deployment of an OVAL scanner to collect information from the target environment.	High
Orient	Uses knowledge sources (e.g. object recognition) and algorithmic sources (e.g. path calculation) to define the world state and provide a context for analyzing information from the environment.	Combines the observations with pre-defined policies (access control rules, privileges, etc) to generate a comprehensive world model	Medium
Decide	Uses production rules (coded conditional statements) to identify the optimal course of action	Uses Prolog predicates to establish potential avenues of attack against the modelled environment	Low
Act	Uses an effector module to implement the appropriate actions.	Presents the valid attack profiles (no direct action taken)	Low

Table 1: OODA/SA Cycle and existing models

One interesting fact in this relation between these models and the OODA/SA cycle is that there is a decreasing dynamic nature as one progresses through the cycle. Using a military example, the possible scenarios that can be observed during a reconnaissance action are numerous. Interpretation of this information in the world environment, in terms of risks posed by specific units and the impact of terrain on those units, with have some variation, but the analysis of the information will be grounded in known characteristics of the observed elements. The development of response will fall along well-defined military guidelines and strategies.

The CN environment has an equivalent interpretation. A periodic scan of the CN environment may detect a specific exploitable vulnerability (the observe function). The consequences of such an exploitation may have on the CN environment can be determined (the orient function). Using a known set of attack strategies, this exploitation and its consequences can be leveraged to launch a successful attack (decide, act). However, the frequency at which new

vulnerabilities are discovered in the target environment is much higher than the frequency at which new attack strategies are developed.

2.2.2 ELEMENTS OF A SUCCESSFUL MODEL

Given the previous discussion of the OODA/SA model, it can be concluded that:

a) There will be a dynamic component to each of the modelled elements; and

b) There frequency at which dynamic behaviour is seen will decrease as one progresses through the observe, orient and decide stages of the cycle.

This section provides a discussion of this dynamic nature for each element to be modelled.

2.2.2.1 MODEL ELEMENTS

Based on the existing work on Situational Awareness [Cndsa], specific elements have been identified that will drive the definition and dynamic aspects of a valid DAP solution. These elements are described below. It is notable that all elements described have a dynamic characteristic that must be reflected in the model.

Assets: Within a CN environment, these are elements to be protected. Within the context of the Computer Network Attack/Defence (CNA/CND) view, assets will have varying levels of sensitivity and this sensitivity will change over time. The successful DAP model must be able to capture this information and have it made relevant in the analysis and decision phase of the model operation.

Vulnerabilities: Within a CN environment, these are potential avenues of attack. Each vulnerability can be characterized by pre-conditions and post-conditions. Post-conditions may impact assets directly (e.g. denial-of-service) or result in additional capability on the part of the attack (e.g. privilege escalation). It is significant to note that vulnerabilities are one area where the correlation between the model and the environment cannot be ensured. Vulnerabilities may exist in the environment, but until this vulnerability is discovered, reported and introduced into the model, it will not be reflected in the analysis. For this reason, there is a dynamic aspect to how vulnerabilities must be represented in the successful model.

Safeguards: Within a CN environment, a safeguard presents resistance to an attacker. The main purpose of a safeguard is to present new pre-conditions for exploiting a vulnerability or neutralize the effectiveness of pre-conditions gained by an attack to prevent their use in an attack. As safeguards can be added, removed or moved, the successful model must be able to dynamically represent safeguards. This is particularly Important for a model that will attempt to rationalize or optimize the placement of safeguards.

Additionally, any model that attempts to determine the security posture of a CN from the perspective of an attacker will also require:

1) An element to define the set of characteristics that identify the capability of the attacker; and

2) A set of decision rules to define the manner by which an attack is perpetuated against the CN environment.

2.2.2.2 DYNAMIC NATURE

If we equate the following functions from the OODA/SA cycle to their model counterparts, we obtain the following mapping:

Observe = elements that can be discovered about the target environment

Orient = pre-populated knowledge regarding the modeled environment

Decide/Act = the production rules through which the simulated attack is progressed

From this mapping, is possible to gain an initial insight into where the dynamic nature must be incorporated into any successful DAP model.

Dynamic Nature of:	Impact of the Model at each of the following functions		
	Observe	**Orient**	**Decide/Act**
Assets	New service added New data asset added	Updated sensitivity for asset (possibly time-based)	Indirectly, the presence of a more sensitive target asset may bias the attack to reach this goal
Vulnerabilities	New vulnerable service discovered New vulnerability discovered (for existing services) New exploit discovered (different consequence)		
Safeguards		New/Removed Safeguards	
Attacker		Level of ability Motivation	Indirectly, motivation (stealth, speed, specific targets) will bias the attack to match the attacker's goal.
Rules			New attack strategy

Table 2: Dynamic Nature of Model Elements

This paper uses the elements that have been detailed in this section as a basis for defining the operational architecture in which a DAP model will operate and the data model abstraction that will represent the elements analyzed within the within the context of that architecture.

2.2.2.3 DUAL COMPUTER NETWORK ATTACK/DEFENCE NATURE

As a final point to this section, a desirable quality for a DAP model would be the ability to perform an analysis of the security posture of a CN from not only the perspective of the attacker, but also that of the defender. Different qualities of the attack/defence scenario can be viewed through the definition of the following characteristics:

	Knowledge of…					
Viewpoint	Assets & Associated Value	CN Environment Connectivity	CN Environment Safeguards	CN Environment Vulnerabilities	General Vulnerabilities & Exploits	**Attacker's Intent** (the degree to which this is a targeted attack with a specific goal)
Attack (Insider)	High	High	Moderate	Moderate	Moderate	Moderate
Attack (External)	Low	Low	Low	Low	Moderate	Low
Attack (Directed)	Moderate	Low	Low	Low	High	High
CND (Defence)	High	High	High	Low	Low	Unknown

Table 3: CNA/CND Perspectives

The values given in this table represent an approximation of the level of knowledge and proficiency for each class of attacker/defender in the CN environment. The actual values for a given organization may vary. The point to be taken from this summary is that there is more than one perspective in defining the security posture presented by an organization. Two examples are discussed in depth:

1) A general hacker will likely not be seeking specific targets to attack within a CN environment (assets or services). As such this will be an undirected attack with a no specific goal in mind. The attacker will therefore start with a little knowledge about the nature of the information assets that are held within the CN environment. Specific knowledge about the networks, safeguards and potential vulnerabilities will also be

initially low, but this knowledge will rapidly increase as an attack proceeds since the attacker will draw upon a body of personal knowledge of vulnerabilities and methods by which they can be exploited. The scenario described can be expressed by the attacker in terms of "how far can I penetrate into this environment and what assets can I gain access to during the course of an attack"

2) The opposing viewpoint, the defence perspective, is expressed with an asset-centric focus. The appropriate question from a defence posture is "given a valuable resource, what is the most vulnerable path by which this asset can be compromised". It is notable that these terms are, by necessity, vague since their meaning will depend heavily on the nature of the scenario:

   - How has the asset's value been defined? (e.g. sensitive information, mission critical service)
   - What is the most vulnerable path? (e.g. fastest route to the asset, fewest safeguards, stealthiest attack)
   - What does it mean to compromise an asset? (e.g. disclosure, denial of service)

In short, the chosen DAP model must be sufficiently flexible to accommodate the likely interpretations of the attack patterns from both the attack and defence perspectives.

3 ARCHITECTURE MODEL

Prior to presenting an the proposed DAP model, it is important to note that the model itself is seen to have two independent, but compatible, components:

1) The **architecture model**, which defined the operational space in which the analysis can be performed including such elements as interaction with the environment and monitoring of the dynamic elements of the model; and
2) The **data model abstraction**, which encapsulates the significance and interrelationships between elements in the environment under study and performs the analysis of the security posture presented by the elements in the model.

It is the position of this paper that the architecture and data abstraction be modelled through extensions to existing and proven models in this field of study: Rog-O-Matic and MulVAL, respectively.

This section provides a description of the proposed architectural model. This section provides a rationale for the selection of this model as it relates to the identified required model elements and characteristics of a successful model.

3.1 ROG-O-MATIC AS THE DAP ARCHITECTURE MODEL

Based on the review of existing models that have successfully been implemented in the problem space and the guiding objectives of the DAP effort, it is the contention in this paper that the best architectural model is a derivation of the Rog-O-Matic work.

3.1.1 MODEL DESCRIPTION

A closer examination of such a model produces the following diagram:

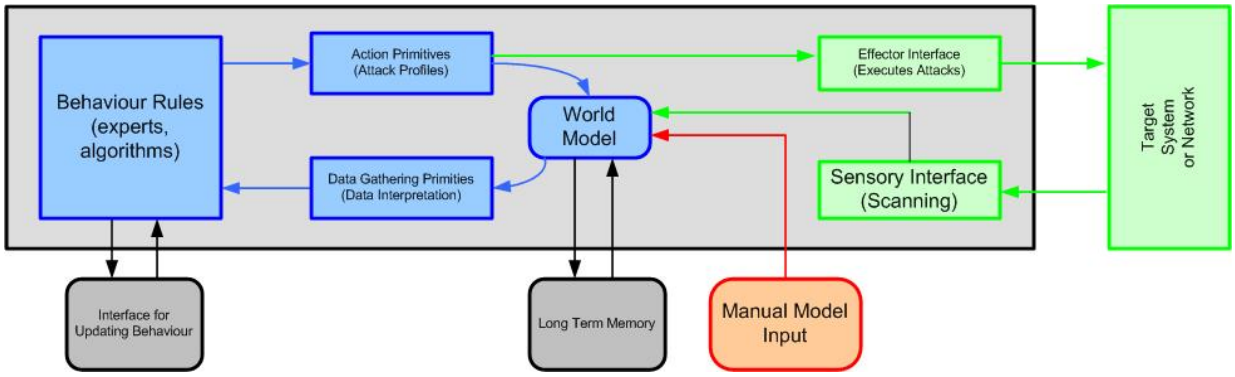

Figure 1: Rog-O-Matic in an DAP Context

The rightmost model elements comprise the externally facing interface to the target environment. This includes the scanning element that performs the observing function and the effector element that would instantiate attack profiles into specific systemic or network actions to detect or exploit vulnerabilities in the target environment. The scanning interface has not only the ability to observe the target environment but also the ability to introduce this information to the world model, making the appropriate data transformations to present this information in a form that can be used by the logic function.

The leftmost model elements define a feedback loop, which was identified as a desirable characteristic of a successful DAP model. The core logic of this feedback loop, known as "production rules" in the Rog-O-Matic model, builds upon the work presented in section *2.1.5:Modelling Concepts Summary.* In simple terms, however, it can be seen that the feedback loop will:

1) Read and interpret the information in the world model;
2) Apply the production rules to determine the next course of action in furthering the attack against the system;
3) Express the next action in terms of an attack profile; and
4) Simultaneously updating the world model and sending the appropriate commands to the effector interface to perform the attack actions on the target environment. Note that the result of these actions must be collected by the sensory interface for incorporation into the world model.

Additionally, there are 3 interfaces to the DAP model to support the operation of the model:

1) An interface to the production rules to allow for:
    a. The addition of new vulnerabilities;
    b. The addition of new exploits for existing vulnerabilities; and
    c. Modification of the decision-making rules to instantiate a new attack technique
2) An interface to the world model to allow saving/restoring of the world model state; and
3) An interface to the world model to allow information to be manually entered (model elements which cannot be observed directly and are not instantiated in the production rules)

In effect, the DAP solution can be viewed as a composite of two models, the derived Rog-O-Matic work to interact with the environment and to capture the dynamic aspects of the environment and the data model abstractions, which is detailed in the next section, to perform the COA decision making functions. In this context, it is useful to view the modified Rog-O-Matic approach as a ***controller*** function:

1) Executing and observing exploitation of vulnerabilities;
2) Presenting observations to the production rules which encapsulate the data model abstractions;
3) Formulating attacks based on the decisions made by the data model abstraction; and
4) Ensuring that information (target environment based or threat environment based) used by the data model abstraction is up to date.

The success of this two-model approach will depend on the ability to ensure that there is an adequate bridge between the two models in term of information exchange.

3.1.2 MODEL RATIONALIZATION

Based on the requirements and desirable characteristics defined earlier in this document, the architecture model can be evaluated to illustrate why this approach will lead to a successful DAP implementation. Note that some elements are more appropriately captured in the data

model abstraction as they imply information by which elements are modelled as data abstractions, which is described in the following section.

Characteristic	Rationale
Automated collection of information	This model support the automated collection information in terms of: 1) A sensory interface that accepts information from the environment. This information can be collected as an appropriately scheduled process, depending on the need to detect changes in the environment in real time or with a reasonable delay. 2) Update interfaces that can pull vulnerability information (in terms of detection or impact) from industry sources. In this way, changes in the target environment or threat environment can be automatically incorporated into the model with the resulting change in security posture derived through the data model abstraction analysis
Scalability	The sensor interface can be implemented as a collector of information rather than an active scanning process. In this way, the need to perform sensor functions can be distributed across many systems, thus allowing the solution to scale to any sized network.
Use of Standards	It is expected that the sensor and update interfaces will communicate using industry supported standards and tool which have been designed to utilize these standards.
Feedback	As previously described, this model supports the needed feedback model to closely match the attacker's behaviour to leverage successful exploits and use this information to further attacks against the target environment.
What-if analysis	The separation of environment interaction and the data model abstraction allows the execution of hypothetical analyses by forwarding potential threat/safeguard information to the data model abstraction.
Learning Component	Apart from any machine learning capabilities build into the data model abstraction; this model has the ability to update the production rules in the data model abstraction to reflect new attack strategies or techniques.
Save State	The world state of this model can saved and restored. Note that the concept of state is held in this model rather than in the data model abstraction centralizing the save state function to this model only.

3.1.3 SIMULATED ENVIRONMENT

It is significant to note that the feedback loop between the world model and the productions rules is not dependant on an actual instantiation of the target environment. It is possible to manually represent the target environment in the world model by adding this information directly to the world model. Under these conditions, the sensory and effector modules would not be used.

This approach has the following disadvantages:

1) No response information from the target environment would be integrated into the world model. For example, a test to see if a patch has been applied to a specific binary would not be possible since access to the binary is not available;
2) There is no mechanism to ensure that the world model is kept synchronized with the target environment; and
3) The effort to fully define the world model would be onerous for a large target environment.

However, the ability to use a simulated environment by populating the world model directly has advantages in that:

1) It provides a mechanism whereby the production rules can be developed and tested; and
2) It would allow the DAP model to be used in a situation where direct action against the environment is not permitted.

It is the position of this paper that the initial work in developing a DAP model should focus on the production rules and feedback loop behaviour, since these elements comprise the core value of this effort and also hold areas with the most uncertainty. The development of an initial DAP prototype would be well served by using the simulated environment for the world model rather than populating the world model through the scanning function.

An additional benefit to having the ability to pre-populate the world model is that it allows the DAP model to perform an analysis under a variety of scenarios, based on the level of advanced knowledge of the target environment. Three typical cases are provided:

- Zero Knowledge: The attacker starts with no knowledge of the target environment. This would be equivalent to an undirected attack, for example, a generic hacker that is looking for potential targets. In this case, the world model would not be populated with any information regarding the target environment. As the attack proceeds, the world model would be updated based on the observed results from the attack profiles. This provides an assessment of the security posture presented by the target environment to the general community.
- Partial Knowledge: In this case, the attacker begins with a certain amount of previous knowledge that it not attainable through attack profiles. For example, the attacker may be provided with access to a low privilege account. The method by which this information is gained may include methods which are not in scope of the DAP model (e.g. dumpster diving).

- Full Knowledge: This is the worst-case scenario, namely, an system administrator is compromised. The world model will be populated with all knowledge and access rights granted to that administrator. It is expected that the DAP model would, correctly, show a high impact in terms of network and system penetration under this scenario.

It is notable that using the ability to pre-populate the world model to represent "pre-existing knowledge" on the part of the attacker requires the connection to the target environment to identify the true consequences of attacks of this nature. Pre-populating the world model in this case is merely a method by which the model can be brought to a specific state, rather than a means by which the model can run in a purely simulated environment.

4 DAP MODEL ABSTRACTION

Work has been done by DRDC in the area of dynamic risk and asset protection [Dap04]. This past work has shown that a focus for a viable abstraction must be:

1. Based on the triple: threat (vulnerability) – safeguard – asset;
2. Able to be Implemented utilizing existing or foreseen technology;
3. Address concerns regarding the safeguarding / protection of assets dynamically; and
4. Quantifying security aspects such as threats, safeguards, and assets so as to be usable within an automated system.

By focusing on metrics that presently exist within networks, the desire is to monitor network and system behaviour in terms of threats, risks, and safeguards and how they pertain to assets. These associations should allow us to define an abstraction. This abstraction would be the basis for any future dynamic asset protection and risk management system.

The ultimate goal of the abstraction is to provide the groundwork and requirement elements, based on generally available technology, so as to provide the necessary framework for the creation of a modeling system for dynamic risk management and asset protection.

4.1 REQUIREMENTS OF THE DAP MODEL ABSTRACTION

Lefebvre et al [Cndsa] describe a number of modelling requirements in order to achieve CNDSA. These requirements can be used to help define the DAP Model Abstraction. The following table provides these requirements with an interpretation for DAP.

Requirement	DAP Use
Map mission-required IT Services onto computer network resources	IT Services are defined as the predominant assets required by CND SA and include both information and the services to process that information. For DAP, this will mean a specific set of data and services running on specific network hosts, with any required connectivity.

Requirement	DAP Use
Combine computer network resources into IT Service offerings that support a required confidentiality, integrity, and availability Quality of Service (QoS)	Lefebvre et al [Cndsa] use QoS as the language between a Mission Commander and the Network Commander. The C/I/A QoS attributed to an IT Service will map in some fashion to the underlying network resources defined in the previous requirement. This CND SA requirement is based on the need for a Network Commander to construct effective IT Services using a constellation of network resources. The requirement to construct IT Services from network resources is not seen as a DAP goal at this time. Initially, DAP will take a static set of C/I/A QoS attributes affiliated with an IT Service as static data. Lefebvre and Bacic [Dap04] also included Accountability as a separate asset value for assets. Therefore, the requirement is to define asset value in terms of C/I/A/A.
Describe computer network resources as interdependent elements	A computer network is modelled as an interconnected set of hosts, data, and applications. DAP requires a model with this interconnection in order to reason on attack patterns and profiles resulting in asset injury.
Describe vulnerabilities as attributes of computer network resources	Vulnerabilities are negative attributes of a system, which allow some form of exploitable threat.
Describe security safeguards as attributes of computer network resources	Safeguards are positive attributes of a system, which protect against specific vulnerabilities.
Map threat events onto computer network resources with vulnerability and safeguard attributes	A threat event is an attempt at exploiting a system vulnerability. The likelihood of success will depend on whether the system has the vulnerability and any mitigating safeguards that can prevent, detect, contain, or recover from the threat event. We call threat events “attack patterns” in this paper and they are viewed as atomic elements in the DAP analysis.
Relate threat events to safeguard effectiveness and vulnerabilities	This is a key element in the DAP model. Modelling safeguard effectiveness is a key goal in DAP in order to effectively protect assets. An important part of this requirement is modelling the consequence of a threat event on associated assets.
Show physical and logical network connectivity (as possible attack ingress paths) as graphs of nodes and links	The various elements in a network are interconnected in some fashion. The nature of the interconnectivity requires modelling so that attack paths can be modelled.
Map sequences of threat events into threat vectors applied to the computer network resource connectivity	It must be possible to model sequences of likely or plausible threat events. We call threat vectors “attack profiles” in this paper.

Requirement	DAP Use
Map areas of responsibility onto IT Services and computer network resources	This requirement is needed in organization environments where different elements of the network being modelled are controlled and managed by different entities (for example, coalition military deployments). This requirement is not needed during the initial research into a DAP abstraction, but will become necessary if DAP is extended to handle these environments.
Generally decompose a large computer network into smaller computer networks	This is seen as a required element in order to handle large network complexity.
Support geographic representations of computer network physical components	Military commanders need a geographic representation in order to understand how battlefield actions will physically impact their network. For DAP, we need some aspect of modelling the physical layer in order to determine physically initiated ingress attacks; although modelling geographic representations is probably not needed.
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	Layered abstraction is seen as a useful means of decomposing networks. This may be a DAP requirement in order to handle large networks or coalition networks.

Table 4: DAP Abstraction Requirements

4.2 EVALUATION OF MULVAL'S MODEL FOR DAP USE

Ou et al [MulVAL] present MulVAL (Multi-host, multi-stage Vulnerability Analysis), a logic-based network security analyser which models many of the features sought in DAP.

4.2.1 MULVAL ARCHITECTURE MODEL

MulVAL models elements in Datalog, a subset of Prolog. The model elements are recorded as Datalog facts. MulVAL requires all Datalog facts to be defined prior to performing any analysis. Missing or incorrect facts will result in a misleading analysis of the system being modelled.

The following table shows the elements modelled by MulVAL and their Datalog fact statements sorted by the DAP layer in which they belong.

MulVAL Model Element	Datalog Fact	DAP Layer
Threat Agent Intention	malicious(Principal)	Social
Services that run on Hosts	networkService(Host, Program, Protocol, Port, Account)	Application

MulVAL Model Element	Datalog Fact	DAP Layer
Vulnerabilities to Services running on Hosts	vulExists(Host, CVE_id, Program)	Application
Client Programs that run on Hosts	clientProgram(Host, Program, RunAccount) setuidProgram(Host, Program, OwnerAccount)	Application
Vulnerabilities to Client Programs running on Hosts	vulExists(Host, CVE_id, Program)	Application
Consequences of Vulnerabilities	vulProperty(CVE_id, ExploitRange, Consequence) ExploitRange = local or remote Consequence = confidentiality loss, integrity loss, denial of service, or privilege escalation	Application
Starting location of attacks	Derived from malicious() and hasAccount()	System
Hosts	Implied in other Datalog facts	System
Accounts on Hosts	Implied in other Datalog facts	System
Principals having Accounts on Hosts	hasAccount(Principal, Host, Account)	System
Paths on Hosts	filePath(Host, Owner, Path)	System
Remotely mounted Paths on Hosts	nfsExport(Host, Path, Access, Client) nfsMounted(Client, ClientPath, Server, ServerPath)	System
Data residing in Paths on Hosts	dataBind(Data, Host, Path)	System
Access Control Rights on Paths on Hosts		System
Policies where Principals can access Data	allow(Principal, Access, Data)	System
Connectivity between Hosts	hacl(Host, Host, Protocol, Port)	Network

Table 5: MulVAL Datalog Facts

How the MulVAL Datalog facts interrelate is recorded as Datalog reasoning rules that are shown in the following table.

MulVAL Model Element	Datalog Rule
Remote service exploitation resulting in privilege escalation using vulnerable services.	execCode(Attacker, Host, Priv) :- vulExists(Host, CVE_id, Program), vulProperty(CVE_id, remoteExploit, privEscalation), networkService(Host, Program, Protocol, Port, Priv), netAccess(Attacker, Host, Protocol, Port), malicious(Attacker)
Remote client exploitation	execCode(Attacker, Host, Priv) :-

MulVAL Model Element	Datalog Rule
resulting in privilege escalation using vulnerable client programs.	vulExists(Host, CVE_id, Program), vulProperty(CVE_id, remoteExploit, privEscalation), clientProgram(Host, Program, Priv), malicious(Attacker)
Local client exploitation resulting in privilege escalation using vulnerable client programs.	execCode(Attacker, Host, Owner) :- vulExists(Host, CVE_id, Program), vulProperty(CVE_id, localExploit, privEscalation), setuidProgram(Host, Program, Owner), execCode(Attacker, Host, SomePriv), malicious(Attacker)
Local user exploitation resulting in privilege escalation using Trojan programs.	execCode(Attacker, Host, Owner) :- accessFile(Attacker, Host, write, Path), filePath(Host, Owner, Path), malicious(Attacker)
Local file access exploitation.	accessFile(Principal, Host, Access, Path) :- execCode(Principal, Host, Owner), filePath(Host, Owner, Path)
Remote file access exploitation using NFS.	accessFile(Principal, Host, Access, Path) :- malicious(Principal), execCode(Principal, Client, root), nfsExport(Server, Path, Access, Client), hacl(Client, Server, rpc, 100003)
Multi-hop network access.	netAccess(Principal, TargetHost, Protocol, Port) :- execCode(Principal, InitiatingHost, Priv) hacl(InitiatingHost, TargetHost, Protocol, Port)
Policy Violations.	policyViolation(Principal, Access, Data) :- access(Principal, Access, Data), not allow(Principal, Access, Data)

Table 6: MulVAL Datalog Reasoning Rules

4.2.2 HOW MULVAL HANDLES DAP RELATED ELEMENTS

MulVAL handles DAP related elements as defined below:

1) Threat Agent intention is modelled explicitly using the malicious() Datalog fact. Other threat agent behaviours are not modelled (for example, requirement for stealth, targeting of specific asset types);
2) Threat Agent capability and opportunity are not modelled;
3) Vulnerabilities are modelled explicitly using the vulExists() Datalog fact;
4) Vulnerabilities are tied to a specific program on a specific host. Operating system kernel vulnerabilities are handled by treating the kernel as a service running as root and a setuid program owned by root;
5) Threats are modelled implicitly as the existence of a vulnerability;

6) Threat likelihood is implicitly modelled as binary, if a vulnerability exists it is assumed exploitable provided an attacker can access the required host/service/program.
7) Some preventative safeguards are modelled implicitly in the following Datalog facts:
    a. Network firewall, host firewall, filtering routers, and virtual private networks are modelled using hacl() facts,
    b. Host access controls are included in filePath() and accessFile() facts. Although, a simplification is used which assumes that only the owner of a local file path can access any data in that path. That is, group, world, and access control list (ACL) access privileges are not modelled,
    c. Network file access (NFS specifically) are included in the nfsExport() and nfsMountTable() facts. These specifically provide read/write access privilege information to the network mounted data, and
    d. User accounts on systems is explicitly modelled using hasAccount() facts. Principals that have certain account access are then implied to have access to that account's data and programs;
8) Other preventative safeguards (for example, data encryption, authentication, application access controls, and database access controls) are not modelled;
9) Preventative safeguards that are implicitly modelled are assumed to correctly implement their configuration. That is, vulnerabilities in safeguards are not modelled;
10) Detection, containment, and recovery safeguards are not modelled;
11) Procedural and human elements to safeguards are not modelled;
12) Assets are modelled implicitly using the dataBind() Datalog fact;
13) Asset sensitivity is not modelled;
14) Consequences to assets are modelled explicitly using the vulProperties() Datalog fact. Consequences are limited to: confidentiality loss, integrity loss, loss of availability, and privilege escalation; and
15) System data access policy is explicitly modelled using the allow() Datalog fact. Policy violations can be modelled using the reasoning rule policyViolation(). If no policy is defined, then MulVAL will report all entities and what they have access to.

MulVAL handles attack profiles with reasoning rules on system facts. The reasoning rules look for local and remote privilege escalation vulnerabilities accessible through network paths. It is interesting to note that without the privilege escalation consequences relating to vulnerabilities, MulVAL would loose its attack profile reasoning ability. Attack profiles are reasoned to start from any account on any system where a principal has been identified as malicious().

MulVAL takes a decent approach at modelling systems and their vulnerabilities. The simplifications taken are a reasonable set in order to create a useful network-modelling tool.

4.2.3 CRITIQUE OF MULVAL IN PROVIDING DAP ABSTRACTION REQUIREMENTS

The following table provides a critique of MulVAL in providing DAP Abstraction requirements as defined in **Error! Reference source not found.**.

Requirement	MulVAL Critique	MulVAL Extension Needed
Map mission-required IT Services onto computer network resources	• MulVAL needs to be extended to handle asset sensitivity or value. The source of information for these Datalog facts will have to be manually created.	• Asset Value
Combine computer network resources into IT Service offerings that support a required confidentiality, integrity, and availability Quality of Service (QoS)	• MulVAL does not focus on the modelling of IT services. • The Asset Value MulVAL extension must provide C/I/A values. • [dap] adds accountability to C/I/A	• Asset Value to model C/I/A/A
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).	• Networking Element
Describe vulnerabilities as attributes of computer network resources	• MulVAL adequately describes vulnerabilities as attributes of programs. • Operating system kernels are modelled as a setuidProgram() and a networkService(). • MulVAL does not model human behaviour as a vulnerability in social engineering attacks.	• Social Layer vulnerabilities
Describe security safeguards as attributes of computer network resources	• MulVAL implies only some prevention safeguards using hacl(), filePath(), accessFile(), nfsExport(), and nfsMountTable() 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. • MulVAL needs an extension that models network element safeguards explicitly. • MulVAL needs an extension that models other types of safeguards more explicitly (that is, detection, containment, and recovery) • MulVAL does not model human behaviour as safeguards.	• Network Element to include network safeguards explicitly • Safeguard Model • Access Control extension to programs • Access Control extension to world, group and ACLs • Social Layer safeguards

Requirement	MulVAL Critique	MulVAL Extension Needed
Map threat events onto computer network resources with vulnerability and safeguard attributes	• MulVAL models vulnerabilities explicitly. • MulVAL implies that if a vulnerability exists, it can be exploited (that is, a threat event) provided an attacker can gain access to the vulnerability. Network safeguards and host access controls are implicitly modelled. • MulVAL models attacker intent using malicious() but does not model other attacker intentions (for example, detection avoidance)	• Network Element • Safeguard Model • Social Layer modelling of attacker intention
Relate threat events to safeguard effectiveness and vulnerabilities	• MulVAL models prevention safeguards implicitly and with binary effectiveness (that is, non-effective or totally effective). • The MulVAL extension to model Safeguards needs to model safeguard effectiveness against specific threat events. • MulVAL explicitly models asset impact in a binary fashion without taking into account safeguard effectiveness.	• Safeguard Model to include safeguard effectiveness • Asset Impact extension to show more granularity on consequences
Show physical and logical network connectivity (as possible attack ingress paths) as graphs of nodes and links	• MulVAL does not model the Physical Layer. • MulVAL models the Logical domain well. • MulVAL may benefit from explicitly modelling some Data Link Layer technologies that imply physical access as ingress paths to the Logical domain. For example, wireless access point proximity, or physical access to a LAN connection)	• Physical Model to model ingress access into logical domain using various Data Link and System access methods
Map sequences of threat events into threat vectors applied to the computer network resource connectivity	• MulVAL models attack profiles well.	• None Required
Map areas of responsibility onto IT Services and computer network resources	• Organizational responsibility is not a model requirement at this time.	• None Required
Generally decompose a large computer network into smaller computer networks	• MulVAL does not explicitly handle network decomposition. It assumes a flat network with all hosts and their interconnectivity defined for its reasoning. • MulVAL can model groups of entities as a class provided their attributes are identical (for example, multiple legitimate users or hosts). • MulVAL implicitly decomposes networking infrastructure into a set of hacl() Datalog facts that defines TCP/IP connectivity between any two hosts. Therefore, network makeup (firewalls, routers, links, LANs, etc) is assumed to be embodied in the hacl() rules.	• Possibly allow the Network Element extension to output resultant hacl() facts? • Is higher-level decomposition needed? For example, at IT Service application level?

Requirement	MulVAL Critique	MulVAL Extension Needed
Support geographic representations of computer network physical components	• MulVAL does not model the physical domain. • Geographic modelling is not seen as a requirement at this time.	• Physical model to model ingress paths but not necessarily geographic representations
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	• MulVAL implicitly abstracts the network infrastructure into a set of TCP/IP hacl() facts. • MulVAL does not model IT Services. • It is not clear if DAP requires modelling abstraction at thit time.	• Unknown at this time.

Table 7: Critique of MulVAL in Providing DAP Abstraction Requirements

It is interesting to note that MulVAL models considerable detail in host system configurations, yet the authors have chosen to limit their modelling to show only multi-staged logical attacks. MulVAL's use of hacl(), execCode(), and accessFile() effectively define all the actions necessary to model all types of logical attacks. A natural conclusion from this approach is that the DAP distinction between Application, System/Network, and Logical can effectively be combined into a single architectural layer. This simplifies our inter-layer interactions as shown in a Section *4.5:Attack Profiles Between Layers..*

MulVAL, with suitable extensions, provides a solid basis for the DAP Model Abstraction for the following reasons:

1) MulVAL performs its Datalog analysis in near-real time even when thousands of hosts are modelled. This high performance makes it a suitable candidate to form the basis for the DAP model, even if that model becomes much more complicated;
2) MulVAL has excellent modelling of multi-staged attacks over networks and locally on hosts; and
3) MulVAL has good modelling of data location and access within hosts. With extensions, it can provide an even more accurate model.

4.3 ATTACK PATTERNS WITHIN A LAYER

An “attack pattern” is defined as an atomic element of attack where a threat agent attempts to exploit a single vulnerability to cause some consequence to a particular asset. The degree of success in the attack pattern is mitigated by any intervening safeguard effectiveness.

As shown in the following figure, Lefebvre and Bacic [Dap04] derived this atomic relation between three elements, which defines a single step Attack Pattern, namely: Vulnerability – Safeguard – Asset. A threat event will attempt to exploit a specific vulnerability with some specific event properties (for example, consequence of attack) and some probability of success.

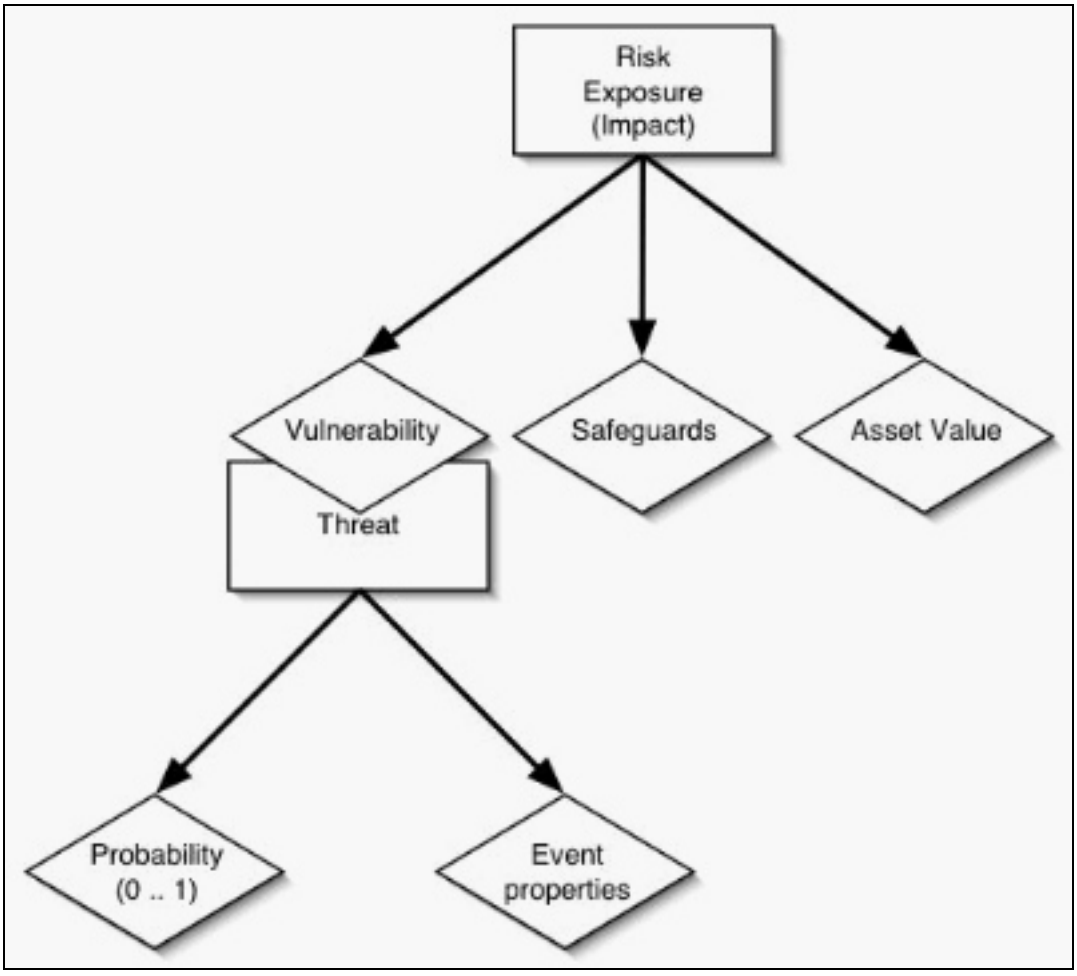

Figure 2 – Risk Model Diagram

Further analysis of the model in Figure 2 is shown in the following figure and described below:

1) In order to determine threat probability, one must take into account threat agents and their capabilities, opportunity, and intentions;
2) Probability of a successful attack is based on the following two elements:
    a. Probability of Attack Success is a function of threat agent capability and the required capability derived from any preventative safeguards used, and
    b. Probability of Attempt is a function of: the probability of attack success, the threat agent intention, any deterrence (detection) safeguards used, and the type of asset; and
3) Impact (or injury to assets) is a function of: the consequence of the attack (derived largely from the vulnerability being exploited), any injury reduction safeguards used (such as containment or recovery safeguards), and the value of the asset.

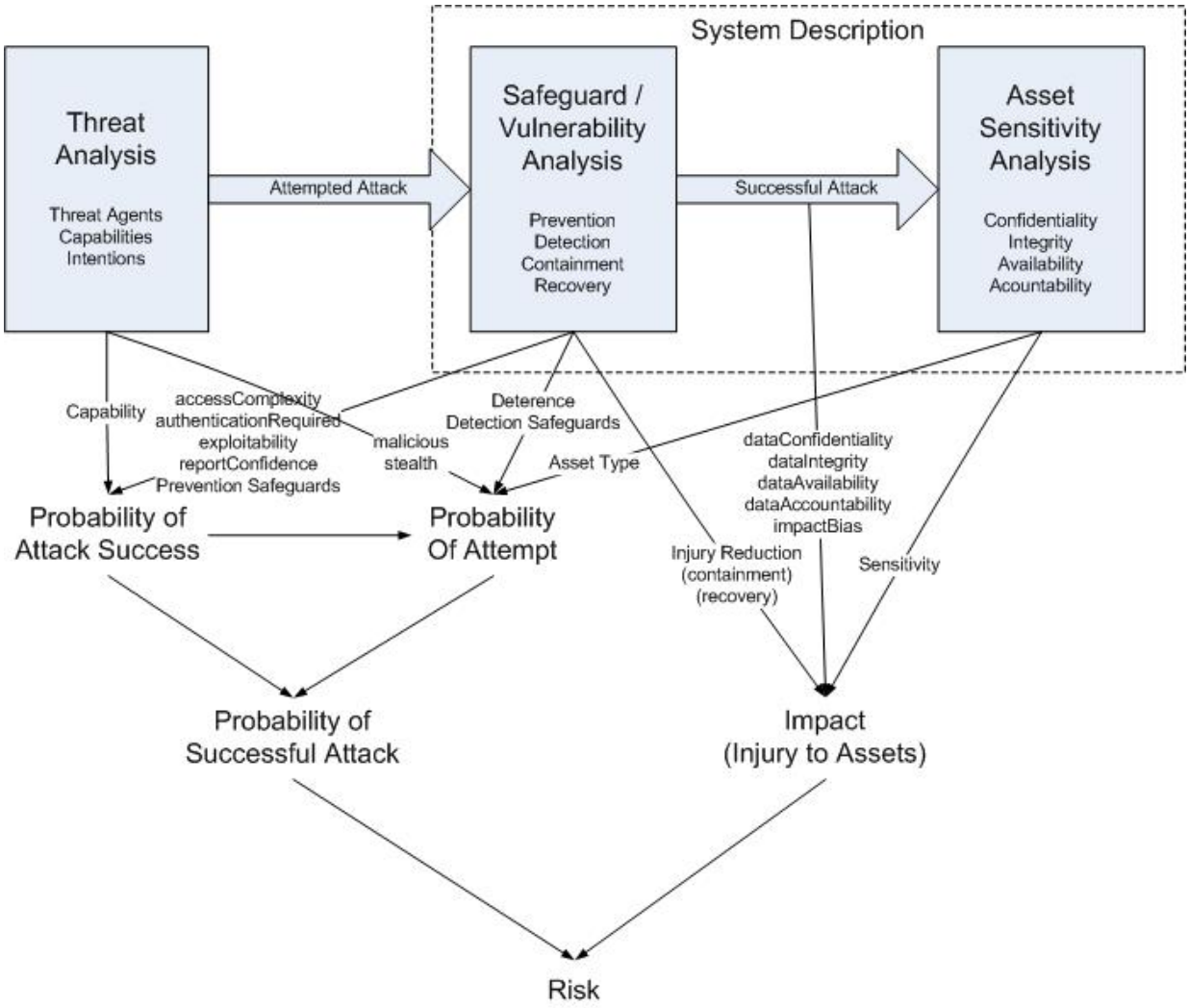

Figure 3 – Alternate View of Risk Modelling

Lefebvre et al [Cndsa] note that vulnerability and safeguards are negative and positive attributes of a system, while assets are predominantly data IT services which will map data and data processing to certain elements in a system or network.

Vulnerability describes a weakness in a system that can be exploited. Vulnerability is the dual to some threat agent attempting the exploit. Therefore, we could classify the vulnerability/exploit relationship using either exploit or vulnerability identification. Other terms used to describe this relationship between exploits and vulnerabilities are attacks and threat events. In general, vulnerabilities are discovered prior to the implementation of working exploit code, so this element of the tuple will be tracked using knowledge of known vulnerabilities.

Safeguards are positive attributes of a system that are applied to block entirely, or mitigate the effect of, inherent vulnerabilities in a system/network. McCumber [McCum] describes safeguards and countermeasures as having technical, procedural, and human factors. Bacic

& Lefebvre [Dap04] introduce the notion of safeguard efficacy but only for the McCumber technical aspect of safeguards. We have tried to further the work on safeguard efficacy using the notion of resistance to attack, however, this still remains a tough problem. Good examples of resistance abound at the Physical Layer given the material nature of safeguards in this layer; however, logical safeguards do not seem to have this easy classification of resistance. We note that the view that safeguards are positive attributes of systems works well by incorporating typical system functionality, which is not often thought of as security safeguards (for example, file access controls, application protocols, and correctness of implementation).

Safeguards and assets are also related in that safeguards will help to block or mitigate the effects of the Attack/Exploit/Vulnerability. Safeguards can be broadly classed as preventing, detecting, containing, or recovering from an attack. Safeguards also apply to an asset's confidentiality, integrity, availability, and accountability attributes differently.

The current MulVAL model needs to better model the probability of successful attacks. MulVAL currently models any plausible attack pattern with a probability of 1. The Common Vulnerability Scoring System (CVSS) is a new initiative tied to CVE (implying it can be correlated to vulnerability information) which provides more information on scoring the severity of a vulnerability and its consequences. Although CVSS is not yet widely adopted, it holds promise as an open repository of this type of information. MulVAL could be extended to model the following attack pattern elements:

1. The difficulty to exploit a vulnerability could be modelled using the following CVSS facts. This would produce a more realistic Probability of Attack Success by weeding out non-capable threat agents:
    a. accessComplexity(high, low),
    b. authenticationRequired(yes, no); and
2. Threat agent capability would then have to be modelled on a scale that would correlate to accessComplexity() and authenticationRequired(). For example, a cross product could lead to a 4-level skill/knowledge capability scale (that is, low-no, low-yes, high-no, high-yes), which could be used to remove attack patterns based on lack of capability; and
3. The Probability of Attack Success for capable threat agents can be reduced based on the following CVSS facts. Note that the CVSS RemediationLevel() fact was not employed since MulVAL measures the actual presence of vulnerabilities and this fact relates to degrees of fix available.
    a. exploitability(unproven, proof-of-concept, functional); and
    b. reportConfidence(unconfirmed, uncorroborated, confirmed).

4.3.1 DETERMINING ASSET IMPACT

MulVAL does not model asset value. For DAP, MulVAL needs to be extended to model asset value of information using Datalog facts such as:

1) dataConfidentiality(Data, ConfidentialityLevel). The GSP [Gsp] provides the QoS value range for ConfidentialityLevel based on injury tests as: unclassified, ProtectedA, ProtectedB, ProtectedC, Confidential, Secret, TopSecret;

2) dataIntegrity(Data, IntegrityLevel). The QoS range of values for IntegrityLevel lacks definition so is limited to 0 and 1;

3) dataAvailability(Data, AvailabilityLevel). The QoS range of AvailabilityLevel could be defined by a single uptime parameter (for example, five 9's or 99.999% available, which is 30 seconds downtime per month). However, this type of figure represents an average, which may not provide insight into tolerable individual outages for lower value ranges. For example, A 95% availability implies up to 1.5 days of outage during a month; however, outages greater than 4 hours may not be tolerable. Mean Time to Repair (MTTR) has also been used to assess availability requirements. These types of parameters also define a threshold that a customer can tolerate. However, no metrics were found which define how much extra outage time correlates to amount of injury. For example, does the threshold define the maximum injury with lesser outages implying partial impact, or does the threshold define the point at which impact starts and builds during the outage? Further definition is required in order to determine:

    a. Use of MTTR or availability, or both as means of defining an availability threshold, and

    b. Whether impact starts occurring before or after this threshold, and how partial impact is determined; and

4) dataAccountability(Data, AccountabilityLevel). The AccountabilityLevel parameter is not well understood. It will not be initially modelled in DAP.

Consideration should be given to extending MulVAL to define IT Service availability. One approach is to consider information availability through its corresponding application program availability. Further definition is required.

MulVAL currently only models consequences of an attack in a binary fashion using the National Vulnerability Database[4] (NVD). NVD has five impact types:

1) Allows denial of service;

2) Allows unauthorized disclosure of information;

3) Allows unauthorized modification;

4) Provides unauthorized access to user accounts (privilege escalation outcome); and

5) Provides unauthorized access to administrator account (privilege escalation outcome.

MulVAL is dependent on the privilege escalation outcomes to provide attack profile modelling. With regard to asset consequence, CVSS has a similar set of criteria, but more granular for impacts to asset C/I/A. Therefore, the following CVSS elements should be incorporated into MulVAL for DAP modelling in addition to the two NVD privilege escalation impacts:

1) confidentialtyImpact(none, partial, complete);

2) integrityImpact(none, partial, complete);

3) availabilityImpact(none, partial, complete); and

4) impactBias(normal, confidentiality, integrity, availability). Note that this fact shows the weighting between the C/I/A elements of the asset impacted.

[4] The National Vulnerability Database can be accessed at: http://nvd.nist.gov/

Note that no rating scheme has been found for the Accountability value of assets.

Note that NVD provides a vulnerabilitySeverity(high, high&medium, medium, low) fact but is not entirely clear how this information is derived. The semantics seem to imply impact or injury.

4.4 ATTACK PROFILES WITHIN A LAYER

Bacic & Lefebvre [Dap04] indicated that heuristics could be applied to the system model so that a threat entered at a given point can be moved throughout the model to realize downstream threats and what that would do to the security of the network. Attack Profiles is the term used to describe sequences of Attack Patterns from a given starting point.

Attackers can sequence Attack Profiles within an architectural layer, typically the logical domain. Attack profiles may also be constructed between layers which is handled in Section 0 below.

4.4.1 SOCIAL LAYER

MulVAL only models an attacker intention as malicious(). It is suggested that MulVAL be extended to include:

1) Social Layer Vulnerabilities. These would be humans interacting with the networked system that present vulnerabilities. The types of vulnerabilities would be susceptibility to human vulnerabilities such as: phishing or coercion. Note that malicious insiders can be modelled now using MulVAL. The modelling of this aspect of the social layer requires further research;
2) Social Layer Safeguards. These would be humans interacting with the networked system that provide safeguard capabilities. McCumber [McCum] asserts that all safeguards have a social element. The modelling of this aspect of the social layer requires further research; and
3) Social Layer Modelling of Attackers. MulVAL currently models attackers using the malicious() Datalog fact. Other attacker intentions might also be modelled. Of particular note would be whether the attacker is trying to remain undetected using a Datalog fact like stealth(Principal). This could be used to reason about whether detection safeguards would deter such attackers.

4.4.2 LOGICAL DOMAIN

The MulVAL model provides a very good starting point for modelling complex attack profiles within the logical domain. The hacl() function describes basic network reachability, while execCode() defines methods of escalating privileges on hosts, and accessFile() defines what host resources are available.

MulVAL assumes that hacl() is populated using automated output from network access control devices such as routers, host firewalls, and network firewalls. This information could also be collected using network mapping tools such as nmap. In general, there is a huge amount of network connectivity amongst hosts in all but the most trivial networks. Therefore, this information must be automatically provided to be effective.

The MulVAL execCode() reasoning rules derive where an attacker can escalate privileges either locally or remotely. Escalation vulnerabilities are essential to building attack profiles.

MulVAL is only able to achieve its multi-stage attack recognition due to the fact that privilege escalation data ties to CVE is identified in NVD.

Once an attacker has access to a host with certain account privileges, they will then have access to data and programs as defined in the MulVAL accessFile() reasoning rule. The nature of that access will determine the consequences to the data/programs.

It is suggested that the following extensions to the existing MulVAL logical domain reasoning:

1) Network Elements. This would explicitly model network elements such as routers and firewalls as hosts, which could contain vulnerabilities. The incorporation of this modelling element would impact the existing hacl() facts which are an abstraction of the network infrastructure. One approach might be to abstract network infrastructure into a separate model where network elements are explicitly modelled and the abstraction outputs hacl() results into an existing MulVAL analysis. Further definition is required;
2) Safeguard Model. This would explicitly model safeguard efficacy relative to vulnerabilities. No significant work in this area was found. It remains an area of research;
3) Access Control Extension to Programs. This would explicitly model execution access control rights to Programs in the existing MulVAL Datalog facts. This would more closely model execution access rights in preventing specific account holders access to programs. Further definition of this extension is required; and
4) Access Control Extension to World, Group, and ACLs. This would explicitly model other existing access control rights to more accurately reflect read and write access to data. This would allow impacts on data to be limited to C/I/A based on read/write access. Further definition of this extension is needed.

4.4.3 PHYSICAL LAYER

It is feasible for DAP to model physical environments in order to define physical attack paths with associated vulnerabilities and safeguards, although this is left for further study. Risk tools that model physical attack paths have similar models for physical access reachability.

It is suggested that the following extensions be added to the existing MulVAL logical domain reasoning:

1) Physical Model. This would explicitly model systems and communications paths in the physical world including vulnerabilities and safeguards. The intent is to model ingress access into the logical domain. Various Data Link Layer technologies should be considered. The intent of the physical modelling is to strictly assess the probability of successfully accessing the Logical Layer. Further research is required.

4.5 ATTACK PROFILES BETWEEN LAYERS

The DAP model defines a number of architecture layers defined in Bacic & Lefebvre [Dap04] including:

1) Social, which includes the human users of the system/network.
2) Application, which includes user visible stuff like client and server applications, except for operating system elements.

3) System/Network, which includes firewalls, intelligent network solutions, and basic host operating system elements.
4) Data Link, which includes elements of particular Data Link types. For example, wireless LAN, BaseT Ethernet, and satellite.
5) Physical, which includes physical access to host systems, console and terminal access, physical access to Ethernet cabling, and proximity access to a wireless LAN.

Attack profiles can only move between layers as shown by the Original DAP Layer Model in Figure 4. These inter-layer movements must be incorporated into the architecture model. In general, the logical domain (that is, anything you can do remotely) is represented by the Application, System/Network, and Logical layers, and there is freedom of movement between these layers. The physical domain is represented by the Physical layer, which can only be an ingress point to the logical domain (for example, gaining access to a LAN cable or host terminal). Once an attacker moves into the logical domain, they cannot return to the physical. The Social layer represents human interaction with the system. Social engineering can be a lucrative attack vector. It is proposed that the Physical, System and Application layers can all move an attack into the Social Layer since these layers can interface directly with humans (for example, a phone call or fake memo, a password harvesting login screen, and a phishing email). The intent of social engineering attack patterns is to gain useful system information, or to entice the system users to perform some function on behalf of the attacker. Therefore, attacks can move from the Social layer to either the Application or System layer which represents this information flow and/or user actions.

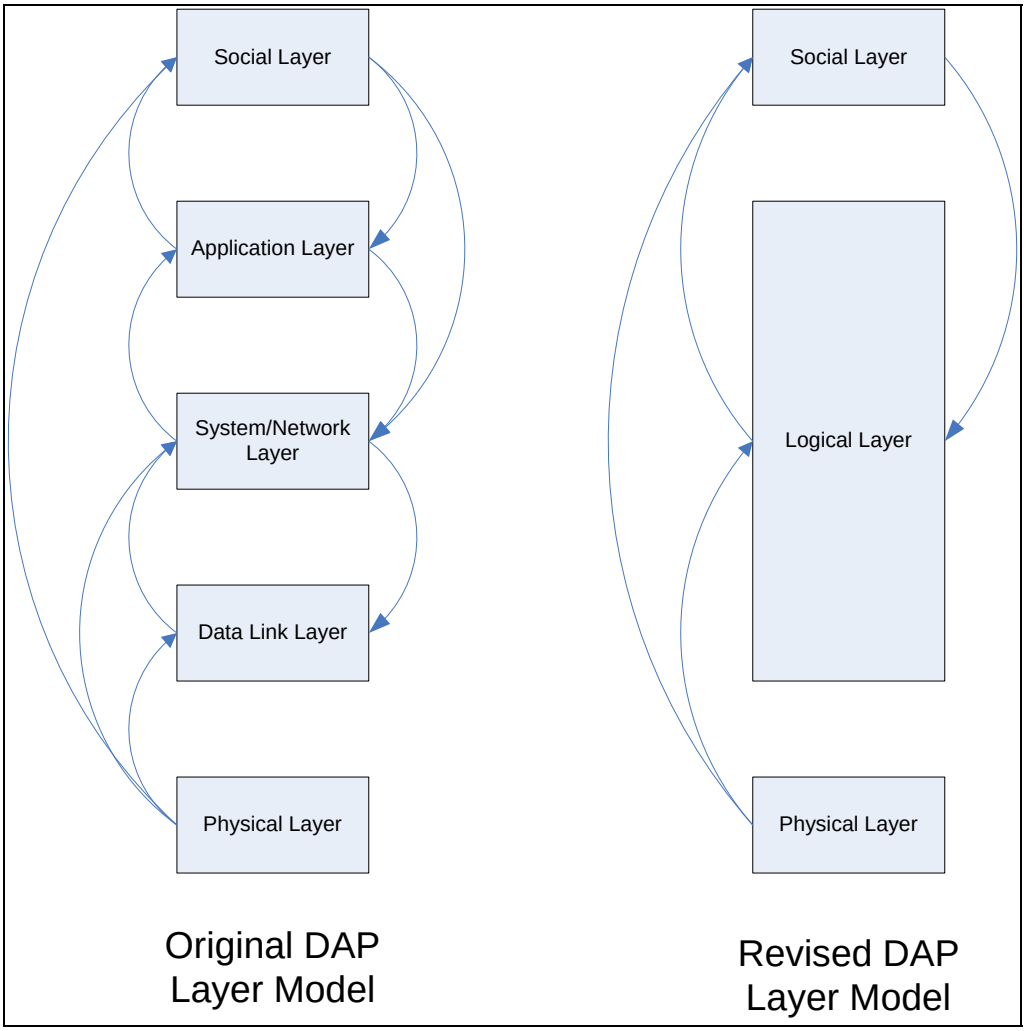

Figure 4- DAP Architecture Model Layers

The MulVAL system model leads to a simplification of the DAP architecture layers by reducing the Application, System/Network, and Logical layers to a single Logical layer as noted in the Revised DAP Layer Model. The interaction between these elements is defined in a MulVAL like system model. This seems a reasonable approach since the Logical, System/Network, and Application layers all reside within host systems as user processes, kernel processes, or device drivers, all of which can be accessed directly with enough privilege through logical means.

The DAP Logical layer provided for specific logical networking technologies such as wireless, Ethernet, satellite, etc. An attacker accessing these Data Link types can be viewed similarly to the Physical layer; they are ingress points only to the logical domain. Using any of these Logical layer ingress points may require an attacker to have certain capabilities including: physical proximity, specialized equipment such as a satellite receiver, resources, and knowledge.

4.6 DYNAMIC ASPECTS OF THE ABSTRACTION

The DAP model abstraction presented has a number of dynamic aspects as shown in the following table.

Model Element Changes	Rate of Change	Amount of Change
New vulnerabilities found	Daily	10-20 new vulnerabilities per day in CVE
New exploit code available - new exploitability() vulnerability fact	Daily	same rate as new vulnerability rate
Changes in safeguard resistance/effectiveness: • weaking of existing safeguards • ratcheting up safeguards during dynamic attacks		
Asset value changes (new idea from CND SA paper)	Strategic: as Operations change Tactical: hourly depending on battlefield conditions	Strategic: minimal Tactical: significant?
IT system changes over time (more rapidly in tactical military case)	Strategic: monthly Tactical: daily	Strategic: large Tactical: large
Changes in threat agent capabilities over time (slow moving except for acquisition of exploit code which gives quantum jump in capability)		
Mounting of attacks	Real-time	Variable
Access Policy Changes	Monthly	small

Table 8: Dynamic Aspects of the DAP Model Abstraction

5 ABSTRACTION IMPLEMENTATION ISSUES

5.1 FUTURE RESEARCH

The following are the identified MulVAL Extensions needed in order of most to least needed to fully implement the DAP model abstraction:

- Asset Value modelling including C/I/A/A values.
- Asset Impact Extension to show more granularity on consequences.
- Networking Elements including explicit network safeguards.
- Social Layer modelling of attacker intention.
- Safeguard Model including safeguard effectiveness.
- Access Control Extensions to model program execution access, and world, group and ACL access controls.
- Physical Model to show ingress access into the logical domain possibly including key Data Link Layer technologies.
- Possible Network Element variation to model networks abstractly and output hacl() statements.
- Social Layer vulnerabilities and safeguards.

Of these, the following are identified as requiring significant further research:

- Asset Integrity valuation.
- Asset Accountability valuation.
- Safeguard Effectiveness.
- Social Layer modelling.
- Physical Layer modelling.

Conversely, the following are identified as DAP areas which are close to being implementable and need further definition:

- Asset Confidentiality valuation modelling using the GSP.
- Asset Availability valuation modelling using uptime and MTTR.
- Asset impact extenstion to show more granularity on consequences through the inclusion of CVSS confidentialityImpact(), integrityImpact(), availabilityImpact(), and impactBias() factors.
- Asset impact extension to show more granularity on consequences by providing a range of attack pattern probabilities through the inclusion of CVSS accessComplexity(), authenticationRequired(), exploitability(), and reportConfidence() factors. Although CVSS data does not currently exist for all CVE vulnerabilities, it can be simulated for a subset of vulnerabilities in a test environment.
- Network Element MulVAL modelling including modelling firewalls and routers as systems.

- Social Layer modelling of attacker intention with the inclusion of a stealth() Datalog fact.
- Access Control Extensions to model program execution access, and world, group and ACL access rights using extensions to existing MulVAL Datalog facts.

5.2 NEXT STEPS

The following next steps are considered by the authors as the most useful steps to further developing the DAP model:

1. Contact MulVAL authors for information exchange. A discussion on intended DAP directions may provide some insight on modelling details;
2. Implement MulVAL in a DRDC test environment to prove the existing MulVAL modelling technology;
3. Implement the top priority implementable MulVAL logic extensions for DAP (that is, Asset C/I/A, Asset impact extension, Network element modelling, social layer attacker intention modelling, and access control extensions).
4. Implement a prototype DAP-O-Matic architecture with MulVAL model abstraction. The initial model should likely use a simulated world model to test the integration of MulVAL and DAP-O-Matic models.

6 REFERENCES

[Ati04] Michael Atighetchi, Partha Pal, Franklin Webber, Richard Schantz, Christopher Jones, and Joseph Loyall, *Adaptive Cyberdefense for Survival and Intrusion Tolerance, IEEE Internet Computing, November/December 2004*

[Cndsa] Julie Lefebvre, Marc Gregoire, Luc Beaudoin, Michael Froh, *Computer Network Defence Situational Awareness Information requirements*, Defence Research and Development Canada, 2005

[Cohen] Fred Cohen*, Simulating Cyber Attacks, Defenses and Consequences*, March 1999

[Dap04] In reference to subject matter discussions relating to dynamic asset protection which occurred between Julie Lefebvre (DRDC) and Eugen Bacic (Cinnabar Networks, Inc.), including the following topics:

Dynamic Asset Protection, November 23, 2004
Dealing with Vulnerabilities, Dec. 10, 2004
The Basis of the McCumber Model, Jan 31, 2005
The McCumber Cube, Feb 14, 2005
Risk vs Threat vs Exposure, March 1, 2005

[Dnd98] Department of National Defence (DND), *CF Information Operations*, B-GG-005-004/AF-010, National Defence, 15 Apr 1998

[DoD00] Department of Defence (DoD*), Joint Vision 2020*, Washington, D.C., Jun 2000

[Gsp] Treasury Board Secretariat, *Government Security Policy*, Feb 1, 2003

[Mca04] McAfee Research, Attribute-based Access Control*, Improved Decentralized Security for Distributed Coalition Environments*, Data Sheet, 2004

[McCum] John McCumber, *Assessing and Managing Security Risk in IT Systems: A Structured Methodology*, Auerbach Publications, New York, 2004.

[MulVal] Xinming Ou, Sudhakar Govindavajhala, Andrew W. Appel, *MulVAL: A Logic-based Network Security Analyzer*, 14th Usenix Security Symposium, August 2005

[Nin04] Peng Ning Dingbang Xu, *Building Attack Scenarios through Integration of Complementary Alert Correlation Methods,* Proceedings of 11th Annual Network and Distributed System Security Symposium (NDSS'04), pages 97–111, February 2004

[Rog84] Michael L. Maudlin, Guy Jacobson, Andrew Appel, Leonard Hamey; *ROG-O_MATIC: A Belligerent Expert System*, Department of Computer Science, Carnegie Mellon University, May 1984

[Rowe] N. C. Rowe, *Counterplanning Deceptions to Foil Cyber-Attack Plans*, Proc. 2003 IEEE-SMC Workshop on Information Assurance, West Point, NY, June 2003

[Yua05] Yufei Yuan, Brian Detlor, *Intelligent Mobile Crisis Response Systems*, Communications of the ACM, Vol. 48, No. 2, February 2005

ANNEX A: APPLICABLE STANDARDS

ADVL - Application Vulnerability Description Language

Project sponsor: OASIS (approved standard as of June 8th 2004)

The AVDL specification defines a standard XML format that allows entities (such as applications, organizations, or institutes) to communicate information regarding web application vulnerabilities. The OASIS AVDL Technical Committee was formed to create an XML definition for exchanging information about the security vulnerabilities of applications exposed to networks. For example, the owners of an application use an assessment tool to determine if their application is vulnerable to various types of malicious attacks. The ADVL toolset includes:

1. The assessment tool - records and catalogues detected vulnerabilities in AVDL format (XML).
2. The application security gateway - uses the AVDL information to recommend the optimal attack prevention policy for the protected application.
3. A remediation product uses the same AVDL file to suggest the best course of action for correcting the security issues.
4. A reporting tool uses the AVDL file to correlate event logs with areas of known vulnerability.

OVAL - Open Vulnerability and Assessment Language

Project sponsor: MITRE

The OVAL Definition Schema is a standard, common schema developed by MITRE and members of the OVAL Community Forum to serve as the language framework for writing OVAL vulnerability, compliance, and patch definitions in XML. OVAL definitions are gold standard tests that check local computers for software vulnerabilities, configuration issues, and patches. OVAL's standardized schemas allow a wide range of computer security professionals to discuss the technical details of determining whether a vulnerability, configuration issue, or patch is present on a system. In addition, tool vendors or developers of security software may download the schema as input for OVAL-compatible information security products and services. There is an official schema for each of the operating systems (OS): Windows, UNIX, and Linux.

The OVAL System Characteristics Schema defines a standard XML format for storing system configuration information. This configuration information includes operating system parameters, installed software application settings, and other security relevant configuration values. The purpose of this schema is to provide a "database" of system characteristics against which the OVAL definitions can be compared to analyze a system for vulnerabilities, configuration issues, and patches. The schema also defines a standard system characteristics exchange format that can be incorporated into a variety of tools and services. For instance, the OVAL XML Definition Interpreter is one example of an application that generates data in the OVAL System Characteristics Schema format and makes it available to other applications.

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.)

Cinnabar Networks Inc.
265 Carling Ave., Suite 200
Ottawa, ON K1S 2E1

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.)

Dynamic Asset Protection & Risk Management Abstraction Study (U)

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

Henderson, G.; Bacic, E.;Froh, M.

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

November 2005

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

47

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

14

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 Ave., Ottawa, ON 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)

W7714-5-3138

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.)

DRD-5-038

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

DRDC Ottawa CR 2005-205

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.)

Unlimited

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).

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.)

computer network defence; network security; defensive posture; risk management