# Comparing Models of Offensive Cyber Operations

**Tim Grant[1], Ivan Burke[2] and Renier van Heerden[2]**
**[1]Faculty of Military Sciences, Netherlands Defence Academy (NLDA), Breda, The Netherlands**
**[2]Defence Peace Safety and Security department, Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa**
tj.grant@nlda.nl
iburke@csir.co.za
rvhheerden@csir.co.za

**Abstract:** Cyber operations denote the response of governments and organisations to cyber crime, terrorism, and warfare. To date, cyber operations have been primarily defensive, with the attackers seemingly having the initiative. Over the past three years, several nations (e.g. USA, UK, France, The Netherlands) and NATO have published cyber security strategies emphasising national and international collaboration. Many strategies call for the establishment of a Cyber Security Operations Centre, as well as for a better understanding of attacks. In the scientific literature, Lin (2009) and Denning and Denning (2010) have argued that offensive cyber operations deserve a more open discussion than they have received to date. Research into cyber attacks would improve the scientific understanding of how attackers work, why they choose particular targets, and what tools and technologies they employ. This improved understanding could then be used to implement better defences. Moreover, research would enable governments and other organizations to take offensive action where justified against adversaries, whether these be criminals, terrorists, or enemies. This could include responding to an (impending) attack by counter-attacking or by proactively neutralizing the source of an impending attack. A good starting point to improving understanding would be to model the offensive cyber operations process. The purpose of this paper is to find, formalise, and compare models of the offensive cyber operations process available in the open scientific literature. Seven models were sufficiently well described for formalisation using Structured Analysis and Design Technique (SADT) notation. Finally, a canonical model has been constructed by rational reconstruction. Although the model has not yet been tested, it has been reviewed by subject matter experts. The paper describes the search methodology, the SADT analysis, the shortcomings of each model, rational reconstruction, and the canonical model. Further work will include elaborating the canonical model to identify the resources needed to set up a Cyber Security Operations Centre with offensive capabilities and to cross-compare the model with the literature on attack ontologies.

**Keywords**: offensive cyber operations; process model; rational reconstruction; canonical model; formalisation; SADT

## 1. Introduction

Cyber operations denotes the response of governments and organisations to cyber crime, terrorism, and warfare, and encompasses computer network defence, exploitation, and attack. To date, cyber operations have been primarily defensive, with the attackers seemingly having the initiative (Owens, Dam & Lin, 2008). Over the past three years, several nations (e.g. USA, UK, France, The Netherlands) and NATO have published cyber security strategies emphasising national and international collaboration. Many strategies call for the establishment of a Cyber Security Operations Centre (TSO, 2009) (MinJus, 2011), as well as for a better understanding of attacks.

In the scientific literature, Lin (2009) and Denning and Denning (2010) have argued that offensive cyber operations deserve a more open discussion than they have received to date. Research into cyber attacks would improve the scientific understanding of how attackers work, why they choose particular targets, and what tools and technologies they employ. This improved understanding could then be used to implement better defences. Moreover, research would enable governments and other organizations to take offensive action where justified against adversaries, whether these be criminals, terrorists, or enemies. This could include responding to an (impending) attack by counter-attacking or by proactively neutralizing the source of the attack before it can begin. This paper makes a contribution to the literature on offensive cyber operations.

A good starting point to improving understanding would be to model the offensive cyber operations process. Although the literature is limited, nine models have been found, of which seven were sufficiently well described for analysis. Some models seem to be intuitive (Colarik & Janczowski, 2008), some are based on an analogy with similar domains (typically military Command and Control, e.g. (Veerasamy, 2010) and (Sorensen, 2010)), others are based on case studies of cyber operations

(Croom, 2010) (Dreijer, 2011) (Van Heerden & Burke, forthcoming), while yet others are based on experienced attackers' own writings (Grant, Venter & Eloff, 2007).

There is an extensive literature on cyber attack patterns (Moore, Ellison & Linger, 2001), attack trees (Schneier, 1999), attack plans (Boddy, Gohde, Haigh & Harp, 2005), attack graphs (Sheyner, 2004, and Wing, 2005), attack taxonomies (Mirkovic & Reiher, 2004), attack ontologies (Simmonds, Sandilands & Van Ekert, 2004), and attack languages (Undercoffer, Joshi & Pinkston, 2003). These representations are predominantly aimed at modelling specific attacks to an executable level of detail. At a higher level, the question arises as to whether computer intruders follow a standard attack process, which they then tailor for each specific attack. Working together with subject matter experts from the University of Pretoria's Information and Computer Security Architectures (ICSA) research group, Grant, Venter and Eloff (2007) extracted a nine-phase process model for cyber crime from computer intruders' own writings, such as (Mitnick & Simon, 2005). Subsequently, Colarik and Janczewski (2008) described a model for a cyber terrorist attack. Dreijer (2011) benchmarked both models against five cases of cyber warfare, and proposed a refined model.

In the research reported in part in this paper, we posed the following research question (RQ):

*RQ1: What resources would be needed by a Cyber Security Operations Centre in order to perform offensive cyber operations?*

Corresponding to the likely organisation in a Cyber Security Operations Centre, the scope of our research is limited to the resources needed by a professional civil and/or military group under governmental control. The resources needed by mercenaries, volunteers, or individual hacktivists are outside the scope of our research. Moreover, we assume that the legal issues associated with offensive cyber operations will have been resolved, at least for cyber counterattack and for proactive cyber defence. Legal issues are not discussed further in this paper; interested readers can consult Owens et al (2009).

We decomposed RQ1 into two sub-questions:

*RQ1.1: Can a canonical process model be developed for offensive cyber operations performed by a professional group?*

*RQ1.2: Can the set of resources needed by a Cyber Security Operations Centre be extracted from this canonical process model?*

This paper addresses RQ1.1. Our strategy was to find existing process models, formalise and compare them, extracting a canonical model. In a subsequent paper we will address RQ1.2 by elaborating the canonical model to identify resource needs.

The purpose of this paper is to find, formalise, and compare models of the offensive cyber operations process available in the open (semi-)scientific literature. Seven models were found that were sufficiently well described for formalisation using the Structured Analysis and Design Technique (SADT) notation (Marca & McGowen, 1988). Analysis shows that the seven models assume penetration of the target system, few represent target selection, attack planning, and Denial of Service attacks, and none specifically represent attack coordination within distributed groups. Finally, a canonical model has been constructed by rational reconstruction (Habermas, 1976) to address these shortcomings. Although the model has not yet been tested, it has been reviewed by subject matter experts. The paper has five sections. Section 1 is introductory. Section 2 describes the search methodology and results. Section 3 describes the analysis and formalisation of the models using SADT, and summarises the key observations arising. Section 4 describes the canonical model developed by rational reconstruction from the seven formalised models. Section 5 draws conclusions and outlines further research planned.

## 2. Searching for process models

### 2.1 Search method

Existing process models were sought by consulting subject matter experts (University of Pretoria, CSIR, and the Royal Netherlands Army), by performing web searches, by following citation chains using Google Scholar, and by filtering the proceedings of appropriate conferences (including the International Conference on Information Warfare (ICIW) and the European Conference on Information Warfare (ECIW)). A limitation in our literature survey is that we have not yet filtered relevant scientific journals, such as Computers & Security or the Journal of Information Warfare (JIW). There are also other conferences, such as the Black Hat conferences, in which cyber attack process models may have been presented.

The criteria for selecting a candidate as a valid process model were as follows:

- It should describe a cyber attack or offensive cyber operations;
- It should describe the behaviour of the attacker, whether this be an individual or a group; and
- It should describe the attack process in sufficient detail for formalisation.

Following Dreijer's (2011) lead, we did not limit candidates to cyber warfare, but accepted models of cyber crime, - terrorism, and/or -warfare.

## 2.2 Models found

We found seven models, listed in Table 1 by chronological order of publication. Two additional models (Sorensen, 2010) (Veerasamy, 2010) were rejected because they did not describe a process model sufficiently clearly for formalisation.

**Table 1:** Models found and their key characteristics

| Model | Context | Basis | Attacker | DoS | Temporal aspects |
|---|---|---|---|---|---|
| Grant et al, 2007 | Crime | Hackers' writings | Lone | No | No |
| Colarik & Janczewski, 2008 | Terrorism | Analogy to cyber crime | Group | Fallback only | No |
| Damballa, 2008 | Crime | Case studies | Lone | No | No |
| Owens et al, 2009 | Warfare | Literature | Group | Yes | Yes |
| Croom, 2010 | Crime (APT) | Case studies | Group | No | No |
| Dreijer, 2011 | Warfare | Previous models and case studies | Group | Yes | No |
| Van Heerden & Burke, forthcoming | Crime & warfare | Case studies | Lone or Group | Yes | Yes |

All of the models describe a linear process representing an isolated attack by an individual attacker or a homogeneous group. None describe the coordination that would be needed by a geographically or functionally distributed group of attackers. While some of the models describe the installation of a backdoor or an advanced persistent threat (APT), none of them describe the behaviour involved in returning to a previously-penetrated target. All models identify (the equivalent of) Penetration, Action, and – apart from Damballa (2008) – Reconnaissance phases.

# 3. SADT analysis

## 3.1 SADT method

SADT notation has been chosen for analysing and formalising the process models. SADT is highly suited to specifying the behaviour of systems in terms of functional processes. The graphical notation represents the system as a network of boxes interconnected by arrows. Each box represents a process, and each arrow represents an interface between processes. Processes operate concurrently, with information passing over the interface arrows. Information is output by an information producer process and input to one or more information consumer processes. Arrows may enter a box from the left, from above, or from below. Arrows may only exit from a box from the right. Arrows entering a box from the left represent data input, and arrows exiting a box from the right represent data output. Arrows entering a box from above represent control inputs, constraining the process. Arrows entering a box from below represent the mechanisms or resources needed to perform the process. Control and resource inputs are neither consumed nor changed by the consumer process, while data inputs are transformed into data outputs. In this paper, we limit the SADT analysis to data inputs and outputs, with control and resource inputs being added to address RQ1.2 in a subsequent paper.

Processes can be decomposed into sub-processes. This is represented in SADT notation by enclosing a set of boxes within a larger box. The inputs and outputs of the larger process must match the free inputs and free outputs, respectively, of the network of sub-processes. A free input is a data input that has no producer process, and a free output is a data output that has no consumer process. A rule of thumb to aid the SADT user's understanding is that there should be three to seven sub-processes. Additional details of the rules and methodology of SADT, may be found in (Marca & McGowen, 1988).

## 3.2 Example SADT analysis

The value of SADT formalisation can be demonstrated by its application step by step to the Colarik and Janczewski (2008) model. Colarik and Janczewski describe the first phase as follows (p.xv):

> *"The first phase of an attack is reconnaissance of the intended victim. By observing the normal operations of a target, useful information can be ascertained and accumulated such as hardware and software used, regular and periodic communications, and the formatting of said correspondences."*

The first step is to give the process an appropriate name, taken from the authors' text. The obvious name is "Reconnaissance". The next step is to identify inputs from the text. The phrase "*By observing the normal operations of a target*" yields two inputs: the intended target/victim, and its normal operations. In the third step, the sole output is identified as useful information, e.g. hardware and software used, regular and periodic communications, and its formatting. The resulting SADT process is shown in **Figure 1**.

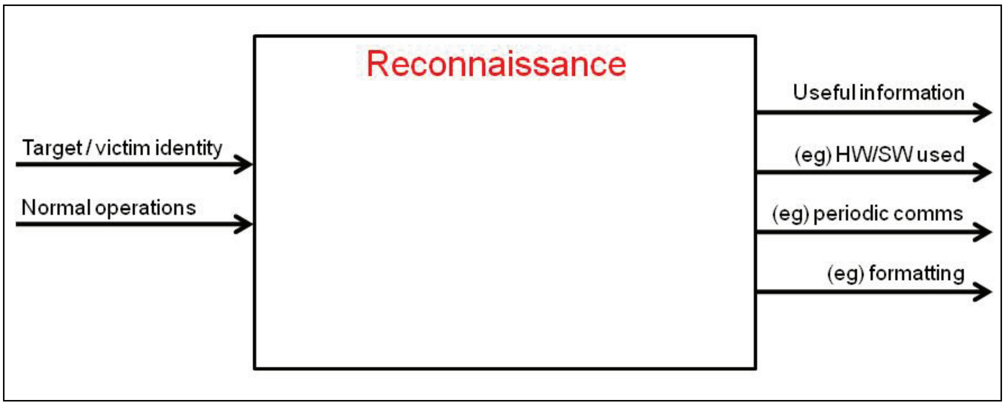

**Figure 1:** Reconnaissance phase, Colarik and Janczewski (2008)

Colarik and Janczewski (2008) describe the second phase as follows (p.xv):

> *"The second phase of an attack is penetration. Until an attacker is inside a system, there is little that can be done to the target except to disrupt the availability or access to a given service provided by the target."*

The obvious name is "Penetration". Inputs are more difficult to find in the text. Clearly, one is the identity of the system to be penetrated. Although not stated in the text, the Useful information gathered in phase 1 is likely to be essential to achieving penetration, so this has been added. The outputs are more easily identified: either penetration (violating integrity) is achieved, or – as the second sentence indicates – denial of service (disrupting availability) is the "consolation prize" if (say) no suitable malware is to hand. The description suggests a sequence of sub-phases, as shown in the resulting SADT process in **Figure 2**.

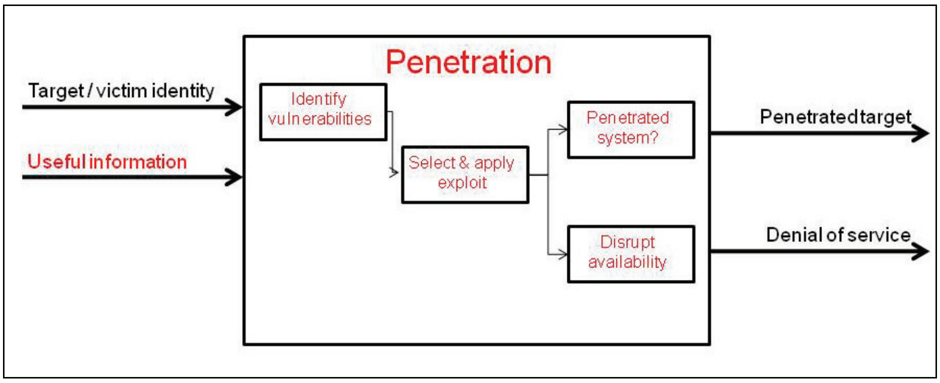

**Figure 2**: Penetration phase, Colarik and Janczewski (2008)

Colarik and Janczewski (2008) describe the third phase as follows (p.xv):

> *"The third phase is identifying and expanding the internal capabilities by viewing resources and increasing access rights to more restricted, higher-value areas of a given system."*

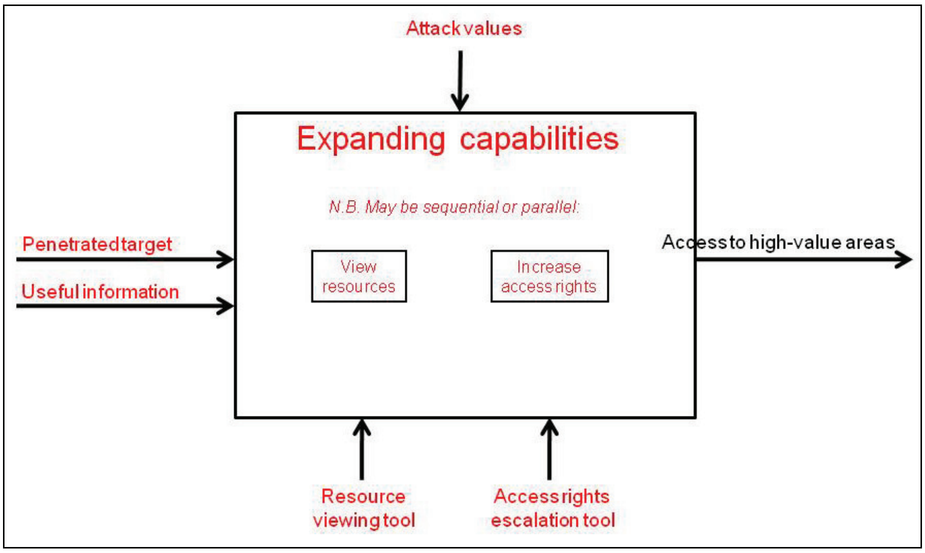

**Figure 3:** Expanding capabilities phase, Colarik and Janczewski (2008)

The obvious name would be "Identifying and Expanding the Internal Capabilities", but this is shortened to "Expanding Capabilities". No inputs are mentioned, but the output is access to the high-value areas of the target system. The target must be identified by an input, and hence "Penetrated target" and its associated "Useful information" have been added. The process also needs to know what constitutes "high value", and this is shown as a control input. The phrase "*by viewing resources and increasing access rights*" suggests two sub-phases (also implied by the "Identifying" and the "Expanding" in the long name), as well as possible resources needed. It is not clear whether the sub-phases are sequential or in parallel. The control input and resources are not needed for the purpose of this paper, but are shown for completeness. The resulting SADT process is shown in **Figure 3**.

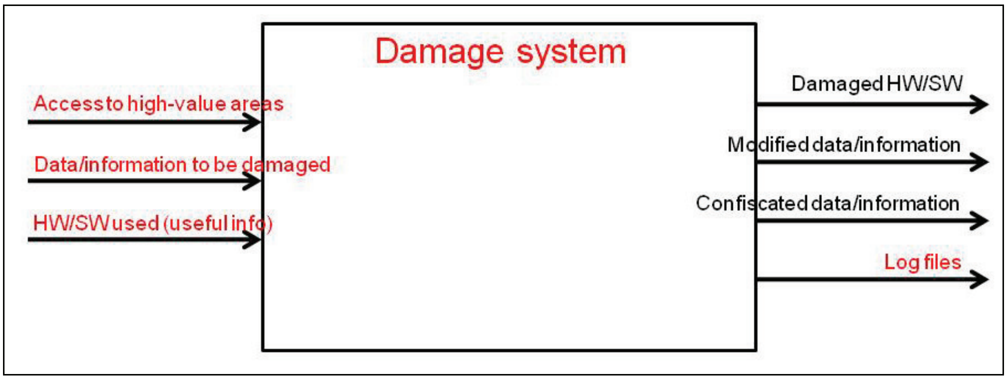
Damage system
Access to high-value areas
Data/information to be damaged
HW/SW used (useful info)
Damaged HW/SW
Modified data/information
Confiscated data/information
Log files

**Figure 4**: Damage system phase, Colarik and Janczewski (2008)

Colarik and Janczewski (2008) describe the fourth phase as follows (p.xv):

> *"The forth [sic] stage is where the intruder does the damage to a system or confiscates selected data and/or information."*

The obvious name is "Damage system". The sentence describes damaged hardware and/or software, modified information, or confiscation of data/information as the outputs. Although no inputs are mentioned, access to high-value areas is needed to damage the system. Moreover, the data/information to be damaged must be accessible, and Useful information gained in phase 1 could provide knowledge of the hardware and software configuration. Hence, these have been added as inputs. The resulting SADT process is shown in **Figure 4**. The Log files output will be discussed in the context of phase 5.

Colarik and Janczewski (2008) describe the fifth phase as follows (p.xv):

> *"The last phase can include the removal of any evidence of a penetration, theft, and so forth by covering the intruder's electronic trail by editing or deleting log files"*

The obvious name is "Removal of Evidence". The only output mentioned is the modified or deleted log files, but hiding the installed toolset has been added by example from Damballa (2008). These outputs imply as inputs Access to high-value areas, the Installed toolset, and the Log files. The latter input has to come from some preceding phase, else it becomes a free input. These files can be most easily captured immediately after the system has been damaged, in phase 4. It is for this reason that Log files has been added as an output to phase 4. The resulting SADT process for phase 5 is shown in **Figure 5**.

The final step in the analysis of the Colarik and Janczewski (2008) model is to put the individual SADT processes for phases 1 to 5 together, making the links between inputs and outputs. The resulting SADT diagram is shown in **Figure 6**, with the dashed box showing the larger process representing the whole attack. As can be seen, the Useful information output of phase 1 is one of the inputs to phases 2, 3, and 4. The Penetrated target output of phase 2 is needed as input to phase 3, and phase 3's Access to high-value areas output is an input to phases 4 and 5. The Log files, output by phase 4, are input to phase 5.

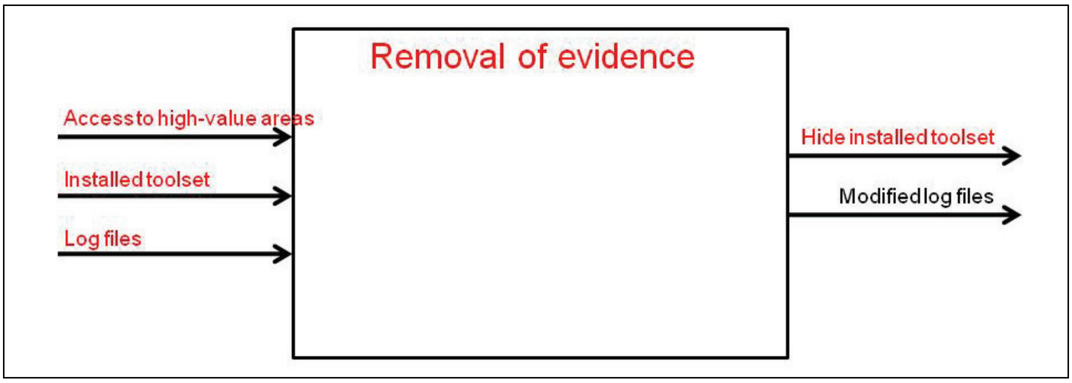

**Figure 5:** Removal of evidence phase, Colarik and Janczewski (2008)

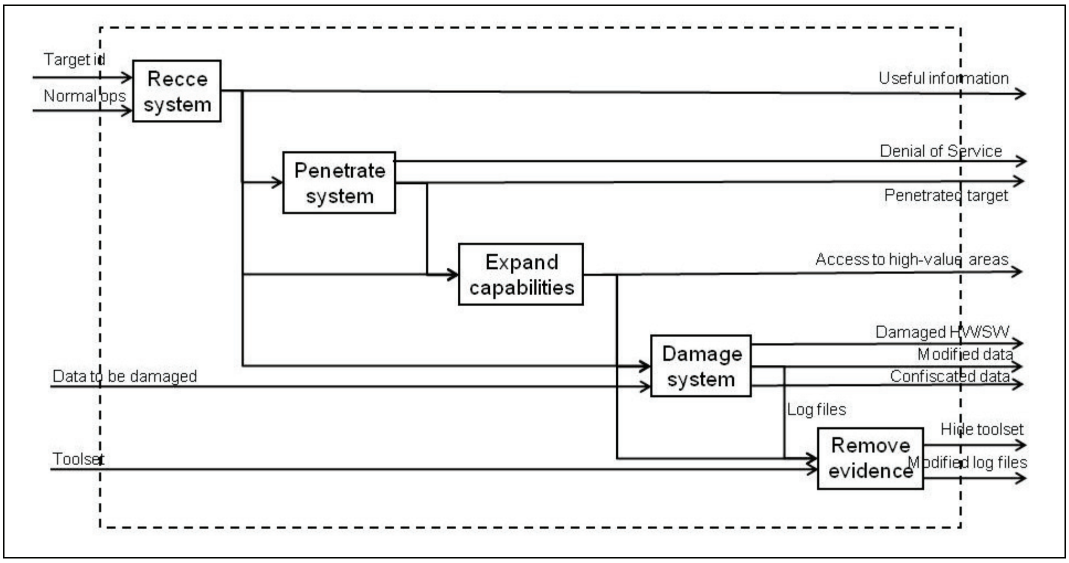

**Figure 6**: Overall process model, Colarik and Janczewski (2008)

Free inputs include which target system should be attacked (shown as Target id), which can be assumed to have been determined by governmental authorities. The Normal operations free input is obtained (from the target) by the Reconnaissance process. Phase 4's free input, Data to be damaged/modified, can be assumed to be determined as a part of the goals for the operation. Phase 5's free input Toolset is identical to the set of tool resources used in the preceding phases. For the purposes of this analysis, they can be assumed to have been provided before the operation began. Free outputs (e.g. that could go to the authorities, or used in lessons learned for future operations) include the Useful information about the target, the knowledge that the system has been penetrated or that a DoS attack has been executed, that access has been gained to the high-value areas, what damage has been achieved, and that as much evidence as possible has been removed.

## 4. Rational reconstruction

### 4.1 Method

In philosophy, rational reconstruction (RR) is defined as “a philosophical and linguistic method that systematically translates intuitive knowledge of rules into a logical form” (Habermas, 1976). RR has been applied in computing research to redesign a seminal expert system (Cendrowski & Bramer, 1984) and to formalise Boyd's (1996) Observe-Orient-Decide-Act (OODA) loop (Grant & Kooter, 2005).

In the research reported here, we used the SADT notation to represent the text describing each phase or step (i.e. sub-process) in the attack process. Using a formalisation like SADT enforces systematic analysis, and the graphical notation is Habermas' (1976) “logical form”. As we have seen, the source text often omits to specify completely the inputs and outputs of each phase. Moreover, terminology may be inconsistent between phases or steps in the same process model. For example, Colarik and Janczewski (2008) refer to an “attacker” in their second phase and an “intruder” in their fourth phase, but the knowledgeable reader realises that the two referents are identical. Such omissions and inconsistencies occur because the authors of the seven models were not writing with formal linguistic analysis in mind. In Habermas' terms, the source texts are “intuitive”. By applying the SADT rules and methodology when linking the sub-processes into the overall process, many of these omissions and inconsistencies can be rectified.

### 4.2 Comparing models

The problems of analysis are magnified when we compare the process models. Although each model is formalised using SADT, there are still differences in terminology, this time between models. For example, three of the authors “penetrate” the target system (or the equivalent in Dutch: “binnendringen”), two “deliver malware”, and one “launches” the attack. In describing copying data residing in or passing through the target system and exporting the copy to the attacker, Grant et al (2007) “extracts” it, Damballa (2008) “steals” it, Colarik and Jaczewski (2008) “confiscate” it, Dreijer

(2011) “gathers” or “collects” it (In Dutch: “vergaren”), and Owens et al (2009), Croom (2010), and Van Heerden and Burke (forthcoming) respectively “compromise”, “violate”, and “breach” its confidentiality. Our analysis interprets all of these as the same sub-process. More insidiously, “target” can refer to an organization, a computer system or network, a hardware or software component or a piece of information residing in or passing through a system or network, or a vulnerability in a component. We distinguish target organizations, systems, and vulnerabilities.

A further complication is that the process models differ in their level of detail and where they draw the boundary around the process. For example, Colarik and Janczewski (2008) go straight from Surveillance (phase 1), in which “useful information” is gathered about the target, to Penetration (phase 2). Other process models first identify vulnerabilities in the target system, e.g. Grant et al (2007), Dreijer (2011), and Van Heerden and Burke (forthcoming). Dreijer then selects malware suited to exploiting those vulnerabilities, and Croom (2010) “weaponises” the exploit by coupling it with a remote access Trojan into a deliverable payload. Damballa (2008) goes straight into malware delivery, apparently without any reconnaissance, surveillance, or intelligence gathering. Only Dreijer is sufficiently careful to model planning the attack in advance and testing the plan before penetration. Only Van Heerden and Burke recognise that there might be unintended effects from the attack and to assess the damage by post-attack reconnaissance. Only Dreijer evaluates the operation, and only Grant et al disseminate the lessons learned to their colleagues. None of the process models caters for an attack that is geographically or functionally distributed. In short, there is no “best” process model.

## 4.3 Canonical model

Since each source model differs in terminology, level of detail, and functionality, our strategy has been to form the canonical model as the union of the source models. The (sub-)phases identified in the SADT analysis of each model are the starting point for bottom-up rational reconstruction. These (sub-)phases are then synthesized into the canonical model by SADT composition, grouping them top-down into phases by the following division of responsibilities within a professional group:

1. Governmental authorities would select one or more target organisations and (counter-)attack goals based on the nature and attribution of the incoming or impending attack.

2. Intelligence analysts would select and gather information about the target systems owned or used by the target organisations.

3. Planners would plan the counterattack in detail, obtain and prepare the resources needed, and test the plan and resources in a simulated environment.

4. Cyber operatives would rehearse and execute the tested plan, aiming to achieve the goals set by the authorities.

5. The whole group would evaluate the operation, archive related information, and identify and disseminate lessons learned.

The resulting canonical model has three levels of decomposition, as shown in **Figure 7**.

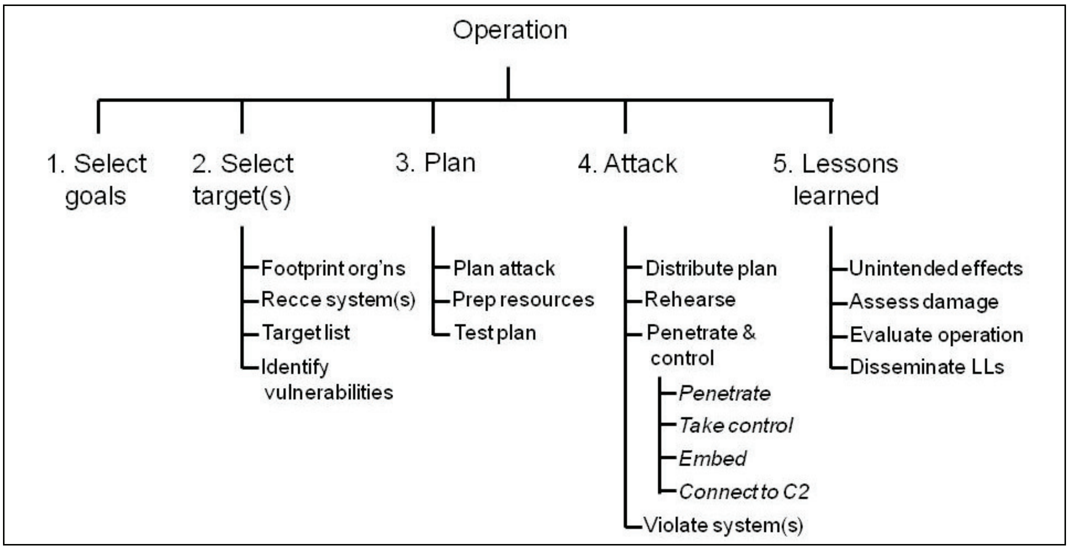

**Figure 7**: Canonical model: breakdown into phases and sub-phases

At the lowest level, the Penetrate and Control process – itself part of the Attack phase – composes the Penetrate, Take control, Embed, and Connect to C2 sub-processes, as shown in **Figure 8.**

At the middle level, there are five phases: Select goals (**Figure 9**), Select targets (**Figure 10**), Plan (**Figure 11**), Attack (**Figure 12**), and Lessons learned (**Figure 13**).

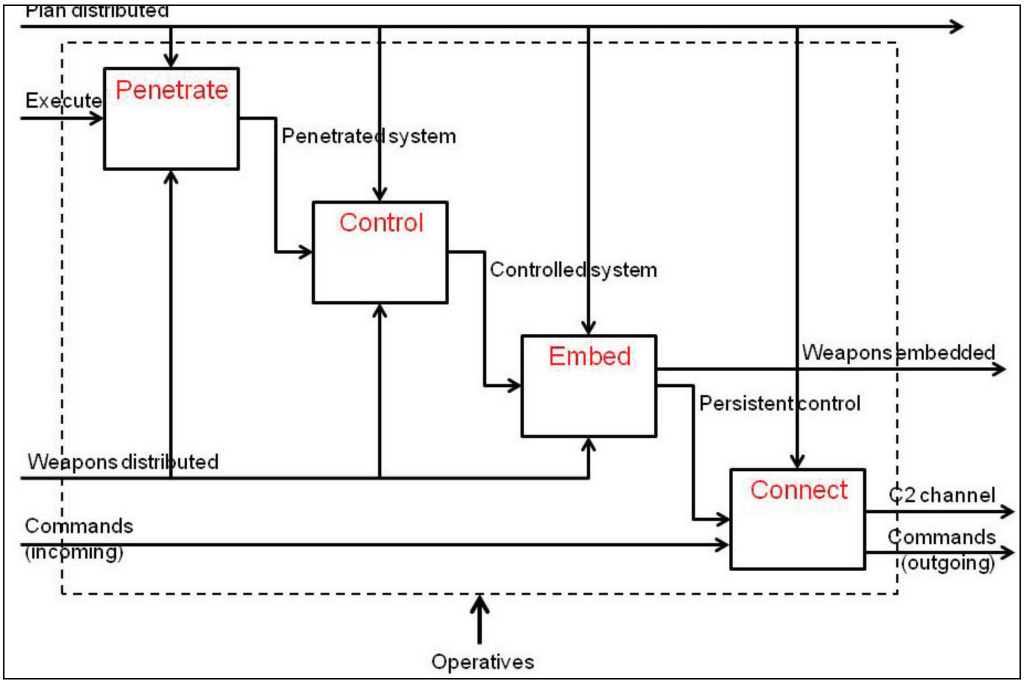

**Figure 8**: Canonical model: Penetrate and control sub-phase

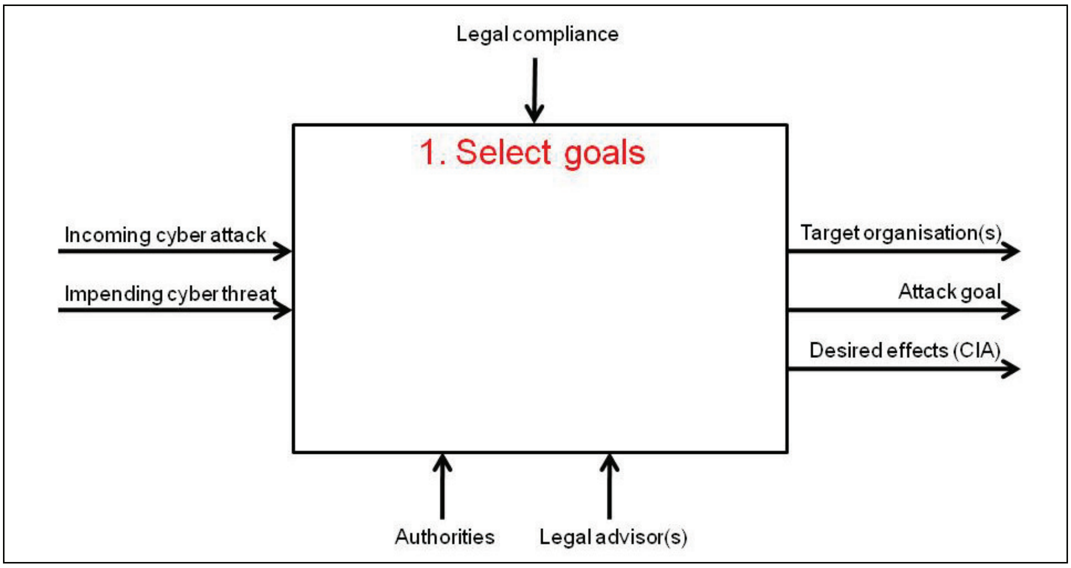

**Figure 9**: Canonical model, step 1: Select goals

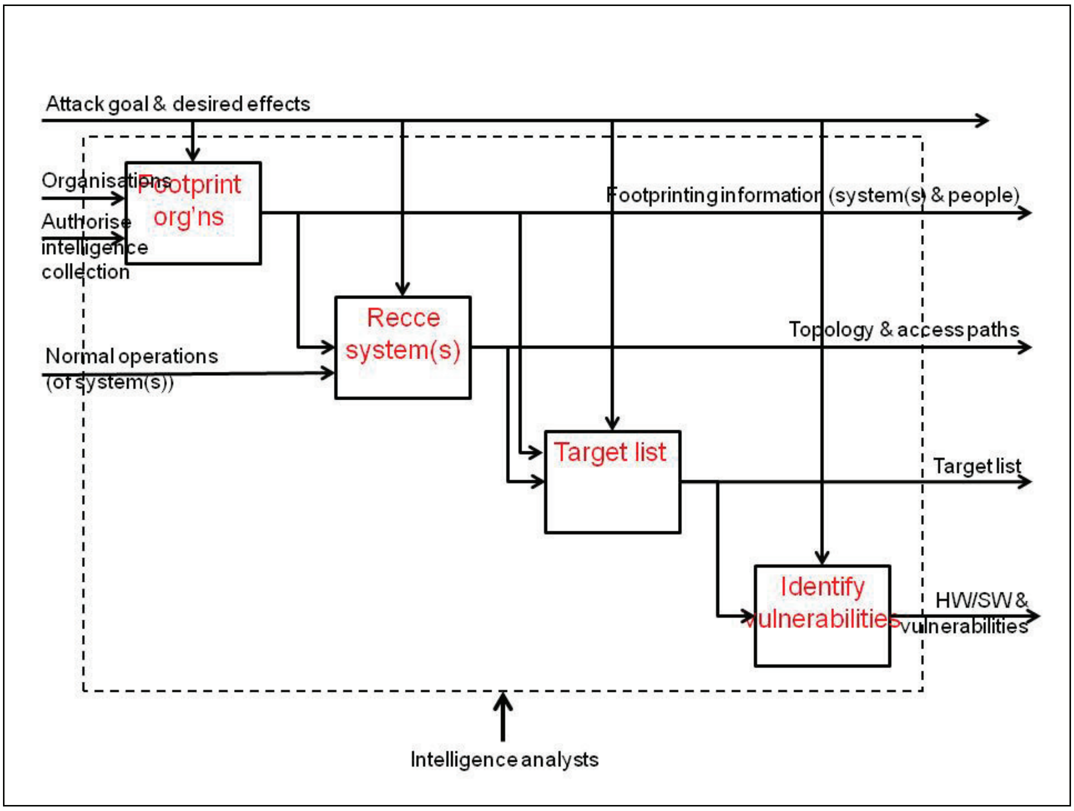

**Figure 10**: Canonical model, step 2: Select targets

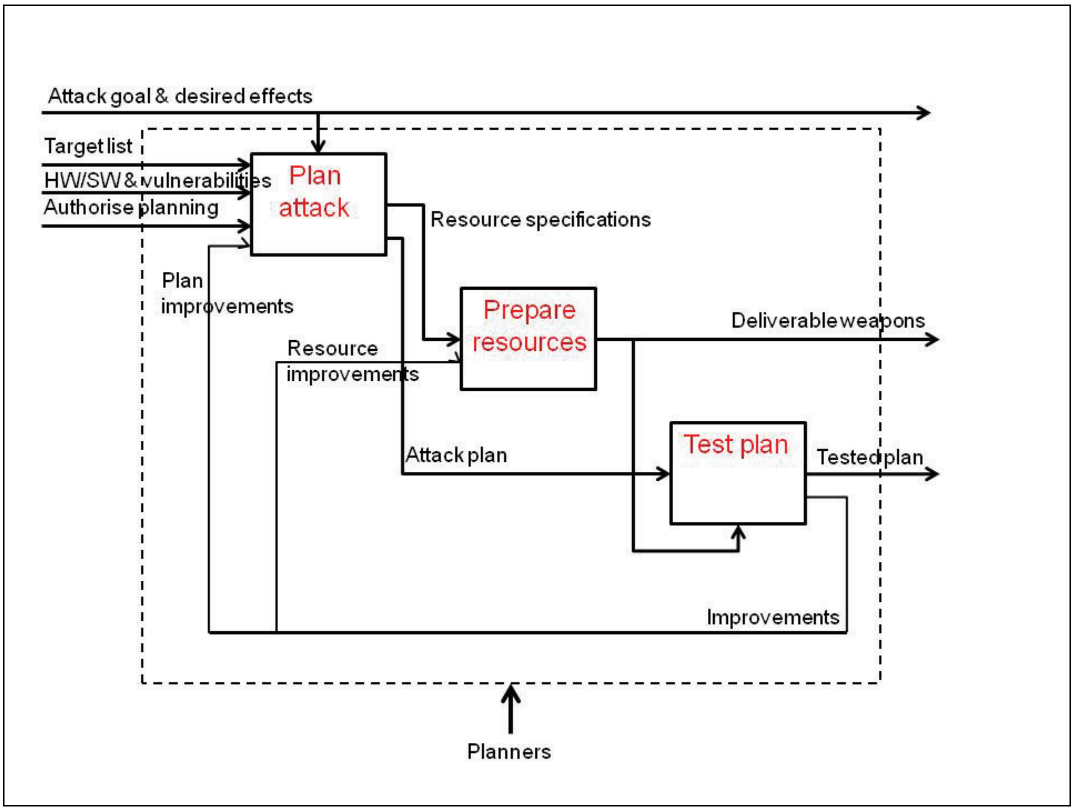

**Figure 11**: Canonical model, step 3: Plan

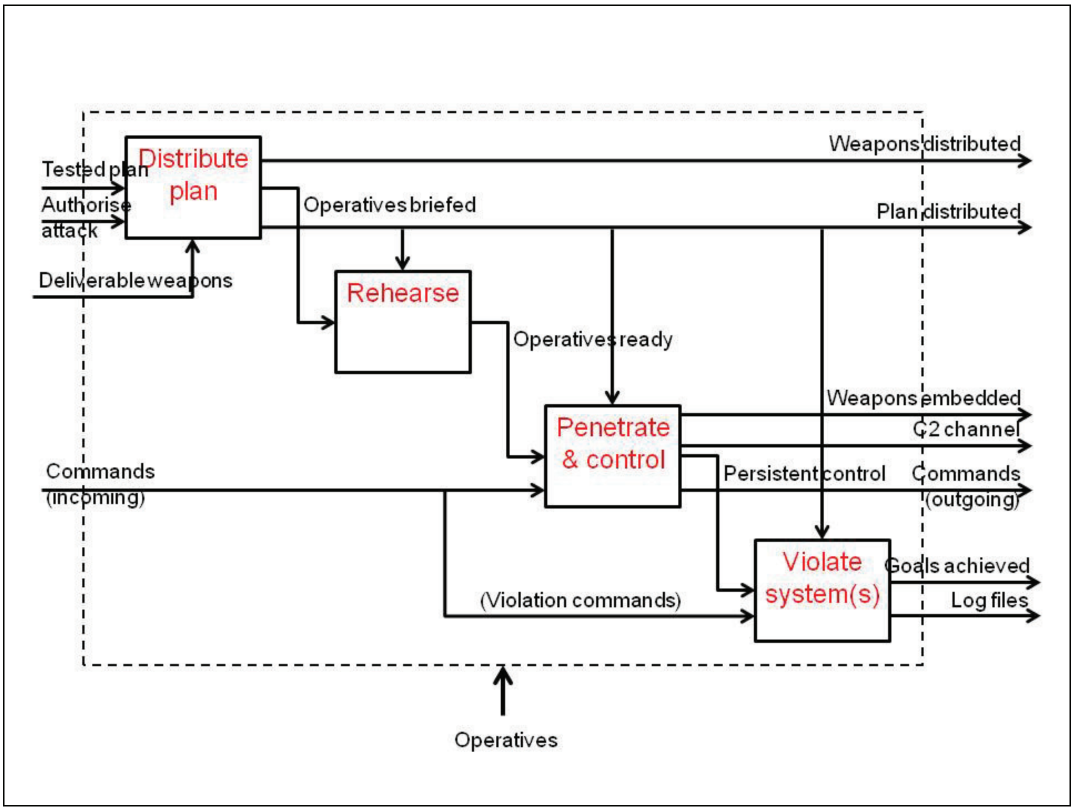

**Figure 12**: Canonical model, step 4: Attack

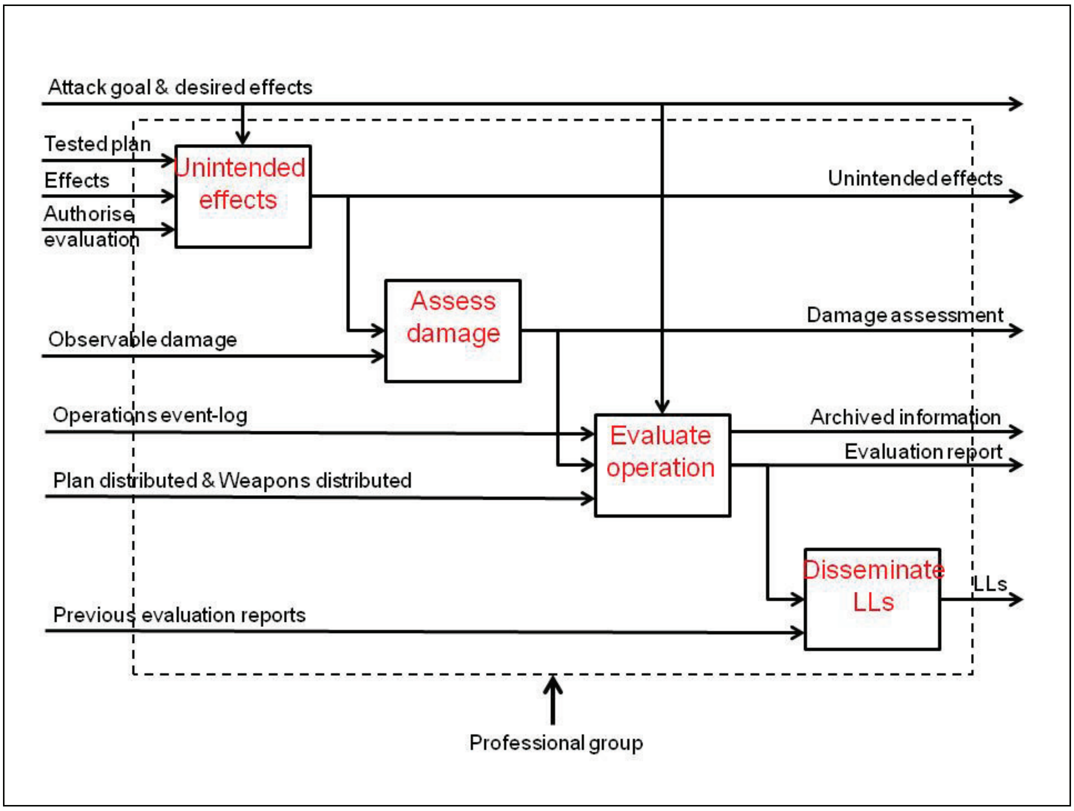

**Figure 13**: Canonical model, step 5: Lessons learned

At the top level, the Operation process composes these five phases, as shown in **Figure 14**.

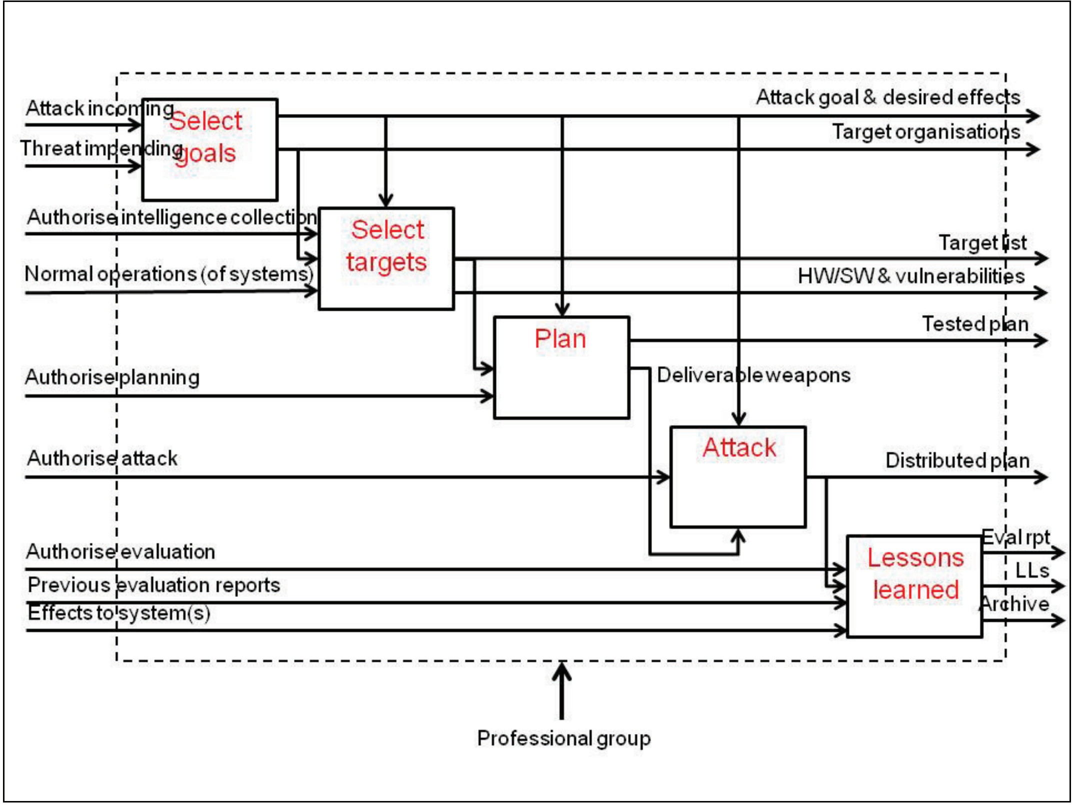

**Figure 14:** Canonical model, operation

## 5. Conclusions and further research

In the research reported in this paper, we found seven sufficiently well-described models of the cyber attack process in the open literature. Each model has been formalised from its textual description using the SADT graphical notation, permitting comparison of functionality and terminology. A canonical model has been constructed by forming the union of the formalised source models, taking into account the likely division of responsibilities within the professional group associated with a Cyber Security Operations Centre.

There are several limitations to this research. The SADT analysis and rational reconstruction have been done by a single analyst. Ideally, this should be done using a Delphi process involving multiple analysts working independently, followed by comparing the SADT diagrams and canonical models. Although the canonical model presented in this paper has been reviewed by subject matter experts, it has not been tested by simulation or use.

This paper contributes to the scientific body of knowledge by formalising, comparing, and rationally reconstructing existing models of the attack process in cyber crime, terrorism, and warfare. It also offers to make a practical contribution in that the authorities and Cyber Security Operations Centre personnel could use the canonical model to develop doctrine and Standard Operating Procedures, for training, and in simulations and operations.

Several areas for further research have been identified. Firstly, the canonical model needs to be tested before it could be used “in anger”. It could be compared with case studies, with other approaches (including attack languages, taxonomies, and ontologies), with hacker tool functionality, and by using the model in a simulated environment. Secondly, the canonical model could be elaborated with control and resource inputs, enabling RQ1.2 to be answered. The tools and technologies needed by a Cyber Security Operations Centre could be identified from the resource inputs, while analysis of the control inputs would provide guidance on doctrine and governance. Thirdly, our research has not addressed the command and control aspects of offensive cyber operations. These aspects deserve deeper study once the resources, doctrine, and governance arrangements for offensive cyber operations have been clarified.

## Acknowledgements

We are grateful to Professor Hein Venter, Computer Science Department, University of Pretoria, South Africa, for bringing the authors into contact with one another.

## References

Boddy, M., Gohde, J., Haigh, T., and Harp, S. (2005). “Course of Action Generation for Cyber Security using Classical Planning”. In Proceedings of International Conference on Automated Planning and Scheduling (ICAPS’05).

Boyd, J.R. (1996). The Essence of Winning and Losing. Unpublished lecture notes, Maxwell Air Force Base, AL.

Cendrowski, J. and Bramer, M.A. (1984). “A rational reconstruction of the MYCIN consultation system”, International Journal of Man-Machine Studies, Vol 20, pp 229-317.

Colarik, A. and Janczowski, L, (2008). “Introduction to Cyber Warfare and Cyber Terrorism”. In Janczewski and Colarik. Cyber Warfare and Cyber Terrorism. Hershey: Information Science Reference, pp xiii-xxx.

Croom, C. (2010). “The Cyber Kill Chain: A foundation for a new cyber security strategy”. High Frontier, Vol 6, no 4, pp 52-56.

Damballa, (2008). Anatomy of a Targeted Attack. White paper, Damballa, Inc.

Denning, P.J. and Denning, D.E. (2010). “Discussing Cyber Attack”. Communications of the ACM, Vol 53, no 9, pp 29-31.

Dreijer, D. (2011). Offensieve Cyberoperaties: Een onderzoek naar de fasering en uitvoering van offensieve cyberoperaties die plaatsvinden in de context van een internationaal conflict. Unpublished bachelor dissertation, Netherlands Defence Academy, Breda, The Netherlands [In Dutch: Offensive Cyber Operations: A study into the phases and execution of offensive cyber operations occurring in the context of an international conflict].

Grant, T.J. and Kooter, B.M. (2005). “Comparing OODA and Other Models as Operational View Architecture”. In Proceedings of the 10th International Command & Control Research & Technology Symposium (ICCRTS’05) (McLean, VA, USA, June 13-16, 2005). US DoD CCRP, Washington DC, USA, paper 196.

Grant, T.J., Venter, H.S., and Eloff, J.H.P, (2007). “Simulating Adversarial Interactions between Intruders and System Administrators using OODA-RR”. In Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT’07), ACM International Conference Proceedings.

Habermas, J. (1976). Communication and the evolution of society. Beacon Press, Toronto.

Lin, H. (2009). “Lifting the Veil on Cyber Offense”. IEEE Security & Privacy, Vol 7, No 4, pp 15-21.

Marca, D., and McGowen, C.L. (1988). SADT: Structured Analysis and Design Technique. McGraw-Hill, NY.

MinJus. (2011). De Nationale Cyber Security Strategie. Ministerie van Justitie en Veiligheid [Ministry of Justice and Safety], The Hague, Netherlands [In Dutch: The National Cyber Security Strategy].

Mirkovic, J. and Reiher, P. (2004). “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms”. ACM SIGCOMM Computer Communication Review, Vol 34, no 2.

Mitnick, K.D., and Simon, W.L. (2005). The Art of Intrusion: The real stories behind the exploits of hackers, intruders & deceivers, Wiley Publishing, Inc.

Moore, A.P., Ellison, R.J., and Linger, R.C. (2001). Attack Modeling for Information Security and Survivability. Technical note CMU/SEI-2001-TN-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Owens, W., Dam, K., and Lin, H. (2009). Technology, Policy, Law, and Ethics Regarding U.S. Acquisition and Use of Cyberattack Capabilities. Washington D.C.: The National Academies Press.

Schneier, B. (1999). “Attack trees: Modeling security threats”. Dr Dobb’s Journal.

Sheyner, O. (2004). Scenario Graphs and Attack Graphs. PhD dissertation, technical report CMU-CS-04-122, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

Simmonds, A., Sandilands, P., and Van Ekert, L. (2004). “An Ontology for Network Security Attacks”. Applied Computing, LNCS 3285, pp 317-323.

Sorensen, C.B.L. (2010). Cyber OODA: Towards a conceptual cyberspace framework. Masters thesis, School of Advanced Air and Space Studies, Air University, Maxwell AFB, AL.

TSO. (2009). Cyber Security Strategy of the United Kingdom: Safety, security and resilience in cyber space. The Stationery Office, Her Majesty’s Government, London.

Undercoffer, J., Joshi, A., and Pinkston, J. (2003). “Modelling Computer Attacks: An ontology for intrusion detection”. LNCS 2820, pp 113-135.

Van Heerden, R. and Burke, I, (forthcoming). “Network Attack Model”. Submitted to ICIW 2012.

Veerasamy, N. (2010). A High-level Mapping of Cyberterrorism to the OODA Loop. In Proceedings of the 5th International Conference on Information Warfare and Security, Ohio, USA, pp 352-360.

Wing, J.M. (2005). “Scenario Graphs Applied to Security”. In Proceedings of Workshop on Verification of Infinite State Systems with Applications to Security. (Timisoara, Romania, March 2005).