MODELING ATTACKS

S. AL-FEDAGHI & SAMER MOEIN
Computer Engineering Department, Kuwait University, Kuwait.

ABSTRACT

The aim of this paper is to develop a general conceptual model of attack progression that can be applied to modeling of computer and communication threat risks. This paper focuses on attacks that aim at overpowering the victim/prey to gain some benefit. It examines existing models and introduces a new flow model to facilitate development of a general model of two-sided combat. The symmetry between the attacker's and defender's flow systems of signals, information, plans, decisions, and actions results in a single combat model incorporating the realms of both attacker and defender. Based on this conceptualization, it is possible to characterize the weak points and develop a map of vulnerabilities in the defender's system. Such a methodology of attack modeling provides a base for analysis in the fields of threat modeling and secure software development. Finally, this new model is applied to an SQL injection problem in web services to demonstrate implementation of a real system problem.

Keywords: Attacks, conceptual model, security, SQL injection, threat risk.

1 INTRODUCTION

Information-security attack modeling has been developed in various forms. For example, attack trees are used to analyze attacks through identification of security vulnerabilities and of compromises caused by attackers. An attack tree represents a damaging event. Branches of the tree elaborate the methods by which that event can occur. In general, an attack graph deals with the composition of an attack to produce attack paths, e.g. paths from outside that allow access to a password file.

According to Moore *et al*. [1], information system engineers cannot rely on engineering failure data – particularly attack data – to improve their designs because businesses and governments have been restrained in disclosing information about attacks because they fear loss of public confidence or exploitation of similar vulnerabilities by other hackers; however, increased interest in Internet security has resulted in increased publication of attack data. Accordingly, interest has grown in documenting attacks and investigating their patterns by security analysts and designers. Nevertheless, according to Moore *et al*. [1], 'Information system engineers need a better way to use and analyze attack data to learn from previous experience… [A] means to document information-security attacks in a structured and reusable form [is proposed] . . . based . . . on a structure called the attack tree'.

Figure 1 exemplifies a high-level attack tree in which an ACME security root node is compromised, resulting in disclosure of proprietary secrets. But such a method gives a fragmented picture of an attack, reflected in categorization of the branches, shown to the right in Fig. 1.

In the description of any phenomenon, *continuity* is an important feature that indicates uninterrupted connection and succession. In the attack tree, there are gaps in the conceptualization of an attack on ACME. First, there is the question of the enemy's awareness of ACME: Why ACME and not others? Then, the attacker collects information about ACME, and such a process is iterative, with information leading to more information. In addition, the attacker's goals, planning, and preparation should be considered.

Accordingly, the attacker starts the attack through some action, extortion, or similar; thus, instead of a piecemeal list of vulnerabilities and actions, a conceptual picture can be drawn as a script or scenario of attack progression, maintaining continuity across stages and during

ISSN: 2041-9031 (paper format), ISSN: 2041-904X (online), http://journals.witpress.com
DOI: 10.2495/SAFE-V4-N2-97-115

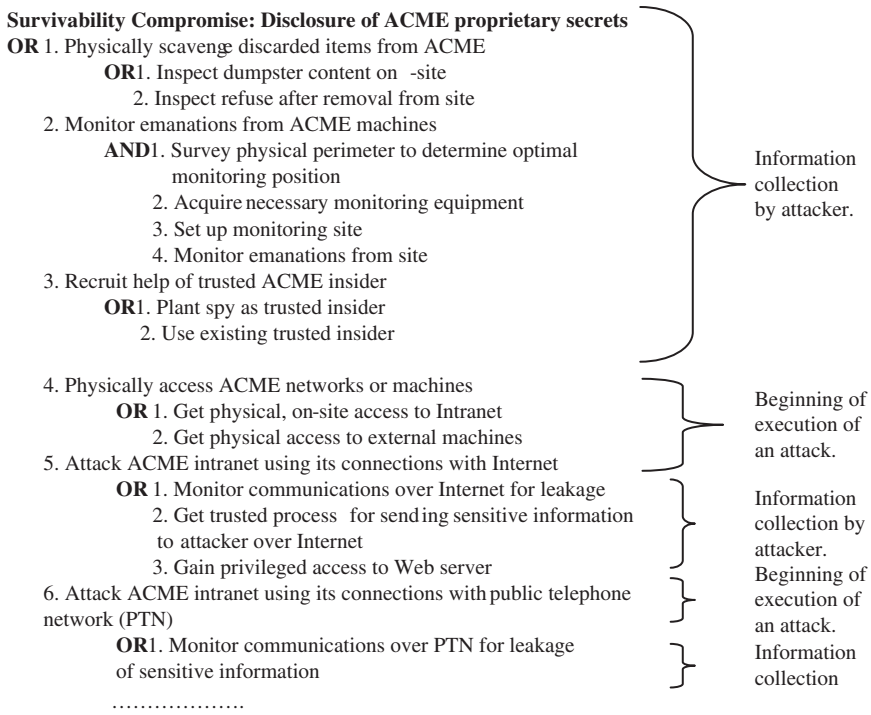

Figure 1: High-level attack tree for ACME (partially from [1]).

the attack process. While attack trees are important tools, they should be part of a more comprehensive description.

In addition to continuity of events, an attack can be viewed as a strike that triggers more subtle events of combat. Combat here refers to activities that occur after the first strike (the attack), involving struggle between attacker and defender. An interesting feature of combat is symmetrical activity, where activities of the combatants are mirrored: each becomes aware, collects information, plans, makes decisions, and acts (or counteracts). Such a conceptualization will be described in this paper.

Attack graphs are complemented by (security) use cases and misuse cases that have been used to identify security requirements, for documentation purposes and to stipulate high-level security patterns. Misuse (abuse) cases are extended use cases utilized in the specification of security threats. They include additional relationships such as *prevent*, *detect*, and *threaten*. Nevertheless, this type of description does not describe an attack; rather it identifies basic entities and relationships involved in security threats.

While such methods have proven to work well to a certain degree, the need exists for additional approaches to build a conceptual foundation for better understanding of the notion of attack.

As Alberts and Hayes [2] point out, to understand something does not mean that one can predict a behavior or an event. Prediction requires more than understanding; thus, even if one understands a phenomenon, one may not be able to predict, with anything approaching a level of usefulness, the effect(s) of that phenomenon. Prediction requires actionable knowledge, specifically the values of the variables that determine (or influence) the outcome in question. Operationally, the most that can be expected is to identify meaningfully different

alternative futures and indicators that those alternatives are becoming more or less likely over time. [2]

In this paper, a conceptual model is viewed as a representation of 'how we think (conceive) about something' [2]; in this case that something is *attack*.

> Building a meaningful conceptual model is quite difficult. The most important decisions involve what to include and what not to include. When a piece of Mozart's was criticized for having "too many notes", the composer replied that the piece did not have too many or too few notes but exactly the right number of notes. So too does a model that is "fit for use". The important thing to consider is whether or not the model serves its intended purpose . . . [2]

To keep the scope of this paper manageable, it is limited to a discussion of mirror-image (two-sided) attacks with binary roles, where an attacker takes action and the defender reacts to these actions. An example of another type of attack is a ternary attack, involving an attacker, a defender (e.g. police), and a victim. The binary attack covers the most common type of attack in the area of computers. According to Cloppert [3],

> By far and away, the goals of the most sophisticated adversaries in 2009 are focused on the surreptitious acquisition of sensitive information for the purposes of competitive economic advantage, or to counter, kill, or clone the technologies of one's nation-state adversaries.

Figure 2 shows a preliminary and partial picture of what will be proposed as a conceptualization of this type of attack. The figure also illustrates what is meant by symmetry, mentioned previously.

This 'synergistic and cumulative' approach is not new, though the paper focuses entirely on its features, such as continuity (of flow of events) and cycling in the conflict between adversaries [4].

It can also be noticed that in many attack-related studies, the focus of analysis is on improving decision making. The observation–orientation–decision–action (OODA) model offers a synthesized conceptualization of information warfare for use by the Air Force [5,6]. Information is analyzed and combined with existing knowledge during the Orientation step to produce a model for decision-making.

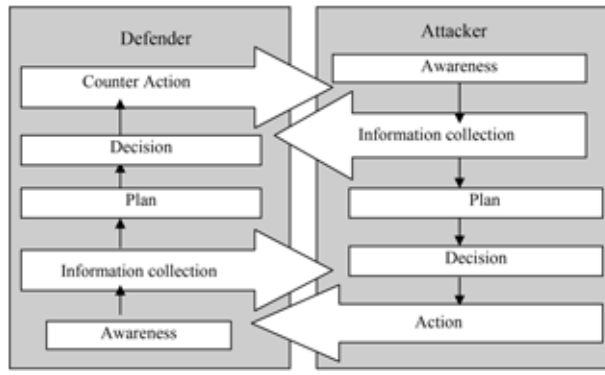

Figure 2: General picture of attacker/defender encounter.

Decision-making is an important factor in the construction of a rationality-based model, where alternatives are chosen by moving through a series of steps. In contrast, our descriptive model encapsulates structured knowledge gained from the real-world phenomenon of attacking. 'Descriptive' here refers to identifying 'attack progression' through its various phases intertwined with causal relationships (e.g. detection of the prey triggers locating it). This type of model facilitates understanding of and communication about the notion of attack.

In the next section, attacks are examined in three different contexts:

- In the animal world, where a lion pride exhibits one of the most powerful tactics for overcoming prey.
- In the U.S. Air Force, where pilots use *Attack Cycle*.
- In computer systems, where an attack has recently been described in terms of six phases.

Since this paper proposes a schematic representation of attacks, Section 3 reviews a new descriptive model, called the Flowthing Model (FM), to be used in later sections as a base for analysis of conflict modeling. Specifically, FM is utilized in Section 4 to model or redesign the three contexts of attack described in Section 2. Section 4 ends with introduction of a general attack model incorporating these various types of attacks. Section 5 discusses the notion of vulnerability in the context of FM. Since the paper emphasizes computer-related attacks, in Section 6, the FM-based conceptualization is applied to SQL injection as described in a published study case.

2 THREE KINDS OF ATTACK

For the purpose of drawing a general conceptual picture of an attack, three attacks are inspected from very diverse domains: animals, military, and computers.

2.1 Prides attack

Lions are predators that live in a group called a pride that occupies a pride area. Lionesses take the role of hunters in the majority of coordinated hunting efforts by the pride, which works as a coordinated group in effectively monitoring, selecting, pursuing, and bringing down the kill. Lionesses plan the attack by encircling the herd from different points and targeting the closest prey. Before initiating the attack, they sneak up close to their prey, taking advantage of factors such as cover and reduced visibility. The attack involves catching the victim and killing it (mostly) by strangulation, especially of large prey, by enclosing the animal's mouth and nostrils in its jaws.

This sequence of actions by lionesses involves processes that can be used as a template for modeling of attacks. Following introduction of our flow model, a conceptual model of the pride attack will be developed to represent an initial version of an attack description; it will then be enriched with details from the other two complex attack environments.

2.2 USAF attack model

The United States Air Force Intelligence Targeting Guide [7] uses the attack cycle functions shown in Fig. 3. The six mission functions of the cycle continuously interact at the decision stage in target analysis.

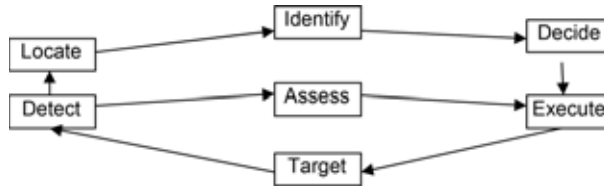

Figure 3: USAF Intelligence Targeting Guide attack cycle functions [7].

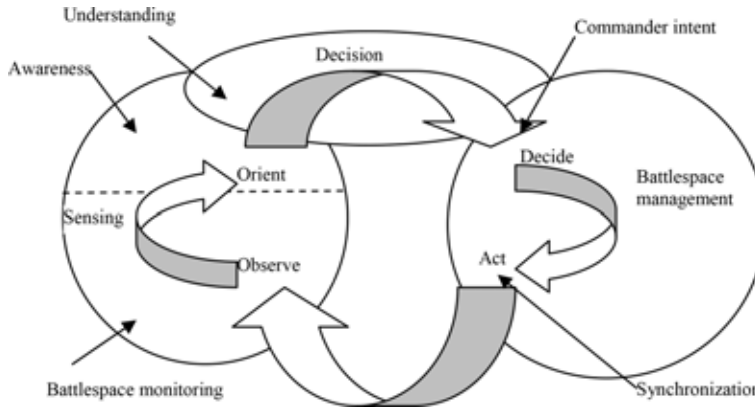

Figure 4: OODA loop processes (partially from [8]).

Comparing this model with the lioness attack methodology, the following can be observed:

- The first step for lionesses seems to be 'awareness' of something in their pride area. This can be considered detection in the USAF model.
- Accordingly, a locating process is activated.
- The next crucial step for lionesses is *identification* of the intrusion 'thing': is it potential food (e.g. wildebeest, impala, zebra, buffalo), or a challenge (adult rhinoceros, elephant)?
- The next step is not to *decide* as in the USAF model; rather, it is to collect information about the target, such as density of the pack, presence of small or weak members, etc.
- Lionesses also depart from the USAF model in the next step, which is to plan the attack before deciding. Each lioness plays a role and takes a position, encircling the herd while 'calculating' even the direction of the wind during the planned attack.

Contrasts in the two methodologies will become clearer when the pride attack is described in terms of our flow model.

Another interesting model, in a slightly different context, is the OODA loop-based processes shown in Fig. 4. 'This model provides a means to understand the IO [information operation] environment. It also provides a logical foundation for the IO capabilities of influence operations, network warfare operations, and electronic warfare operations' [8].

Note the similarity of this model and our sketch in Fig. 2. Figure 4 includes the actions of decide, act, observe, sensing, awareness – terms similar to those in Fig. 2. As will be shown, our conceptual model is focused on the systematic continuity of events that occur in an attack, based on the notion of flow.

2.3 Cloppert's model

In the context of computer forensics, Cloppert [9] conceptualizes the phases of an attack in six sequential stages (Fig. 5). Some phases may occur in parallel, and the order of phases can be interchanged.

The *Reconnaissance* phase involves knowing the target, e.g. browsing websites, learning the internal structure of the organization. These activities are often indistinguishable from normal activity. The *Weaponization* phase reflects 'the technique used to obfuscate shellcode, the way an executable is packed into a trojaned document, etc. Only by reverse engineering of delivered payloads is an understanding of an adversary's weaponization achieved' [9]. In the *Delivery* phase, 'the payload is delivered to its target such as an HTTP request containing SQL injection code or an email with a hyperlink to a compromised website'. The *compromise* phase includes elements of software, human, or hardware vulnerabilities. This phase results in 'the compromised host behaving according to the attacker's plan as a result of the execution of the delivered payload (e.g. running an EXE attachment to an email)'. This phase may include subphases such as 'the delivery of shellcode that pull down and execute more capable code upon execution'. The *command-and-control* phase represents the period after which adversaries leverage the exploit of a system. Communication back to 'the adversary often must be made before any potential for impact to data can be realized' [9].

Cloppert's [9] phases lump together a great deal of semantics. The basic conceptual ingredients necessary for describing an attack actually differ drastically from Cloppert's phases. First, in an attack, what is being 'transferred' between adversaries must be identified. In fencing, the duelists exchange thrusts, and in boxing, they transfer punches. Attacks in fencing and in boxing are first described in terms of thrusts and punches, respectively, that *flow* between attacker and defender. These *things that flow* between adversaries are explained in terms of the way in which they are created (generated), received, transferred, released, and processed (twisted, strong, etc.).

Things that flow from/to an attacker may be of different kinds. The attacker receives information, transfers a punch, and creates a (fault) signal. Each kind of input and output has its own sphere of flow. For example, to describe a cheating card player, his/her information, signals, and actions must be examined.

It can be observed that Cloppert's Reconnaissance phase includes signals (communication carriers), information (knowledge), and action dimensions. This phase, in Cloppert's words, includes "knowing internal structure of the organization"; hence it may include eavesdropping (signals), browsing of websites (information), and actual visits to the organization (actions).

In the next section, our new specification methodology (FM) is reviewed to provide the opportunity to scrutinize previous descriptions.

3 FLOWTHING MODEL

The FM is basically a lifecycle specification of things that flow (e.g. information). Life cycle and flow are familiar notions [10–12]. The Air Force Information Resources Management System [6,13] specifies cradle-to-grave information management in terms of the information

Figure 5: Cloppert's model of attack progression [9].

life cycle: Create®Use®Store®Destruct. This life cycle is conceptually incomplete. Suppose a piece of information is created that is then transmitted. Transmission or transfer in the channel means it enters a different stage of its life that is different from states of being created, used, stored, or destroyed. The information can be destroyed while being transferred, but is this different from being destroyed while stored? What is needed is a 'state transaction' model for life cycle that includes exclusive (i.e. being in one state excludes being in any other state) and complete states of information.

The FM was first introduced by Al-Fedaghi [12] and has been used since then in several applications such as software requirements, communication, and business processes [10,11]. This section provides a review of the basic model as described in other publications, and it includes new aspects of the model.

3.1 General view

A flow model is a uniform method for representing things that 'flow', i.e. things that are exchanged, processed, created, transferred, and communicated. 'Things that flow' include information, materials (e.g. manufacturing), and money.

To simplify this review of FM, a method of describing attack flow is introduced. 'Attack' here refers to the means of an assault such as bullets, computer virus, or direct violent action. There are five states of attack: create, release, transfer, receive, and process, as illustrated in Fig. 6, where the flowthing is an attack. *State* is here used in the sense of properties; for example, states/phases of water: liquid, solid, and gas.

The model can also be defined in terms of a transition graph comprising five stages, as will be described later in the paper, in which the stages are also create, release, transfer, receive, and process. Consider an attack such as a computer virus.

1. Creation stage: The virus has been created by a hacker. It is not yet a released virus, not yet being transferred to targets. It has not been received by anyone, or processed (e.g. analyzed). It stays in its created state until it is released.
2. Released stage: The released virus may stay in its released state for a while, because, say, the transfer channel is down. This situation is analogous to a factory that manufactures goods, then releases them for export; the goods stay in their released state until actually being transported on a certain date to a destination. Certainly, the goods have left the created state and occupy the released state. The created state is now a past state, and the released state is the current state. It is possible that later, for whatever reason, the order is canceled (in the case of a virus, the hacker can change the plan about where to send

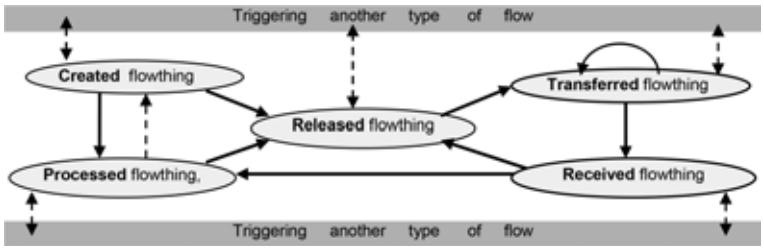

Figure 6: State transition diagram for FM with possible triggering mechanism.

the virus); hence, the goods move back to the created state for possible later release to another customer.

3. Transfer stage: In this case, the released virus is put into the communication channel. This is analogous to passengers in an airport released from passport processing and waiting to board, actually getting on the airplane.
4. Receiving stage: In this case, the virus arrives at the target site. It may be stored in its original form, or deleted immediately; however, it is in its originally received state.
5. Processing stage: In this case, the received virus is processed (changed in form), as in the case of being activated or re-engineered. Viruses can be stored, copied, destroyed, used, etc. while in any of the five specific stages; however, stored, copied, destroyed, etc. are generic states. A created virus can be stored, a released virus can be stored, a received virus can be stored, and so forth.

The flow of a flow structure among its five stages is called a flowsystem. In Fig. 6, flows are denoted by solid arrows and may trigger other types of flows, denoted by dashed arrows.

The environment in which a virus exists is called its sphere (e.g. computer, system). Consider three spheres: a hacker system and two computer systems, as shown in Fig. 7.

The hacker creates a virus that is released and transferred to computer 1. Processing the virus in computer 1 results in duplicating it and transferring to computer 2. In this example, there are three spheres: hacker, computer 1, and computer 2, each with its own virus flowsystem.

Figure 8 illustrates the triggering mechanism. Assuming SQL injection attack, the hacker creates SQL statements that are transferred to a computer SQL flowsystem and that triggers transfer of passwords to the hacker in the password flowsystem. In this case, there are two

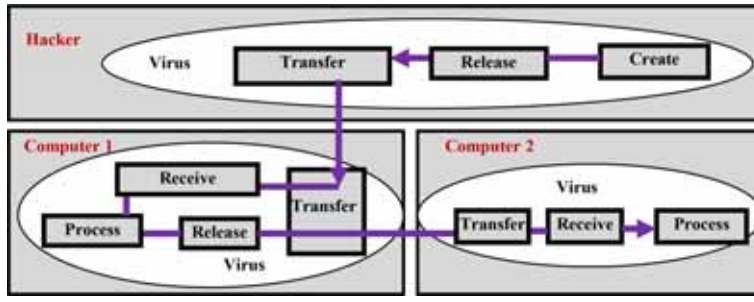

Figure 7: Virus flow from a hacker system to two computers.

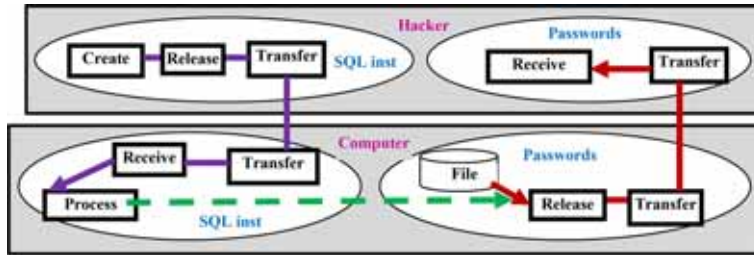

Figure 8: SQL injection triggers flow of passwords to the hacker.

spheres: the hacker system and the computer system. Each sphere has two flowsystem: an SQL statements flowsystem and a passwords flowsystem.

In Cloppert's reconnaissance phase, 'things that flow' are first identified. By 'browsing websites, pulling down PDF's, learning the internal structure of the target organization', the attacker receives, processes, creates, releases, and transfers signals, information, and actions. Thus, the attacker controls the streaming movements of flowthings (signals) among signal spheres, information spheres, and action spheres.

While the pride is sleeping under a tree, lionesses *receive* auditory and olfactory signals of something in the pride area. They process the *signals* and take (*create*, *release*, and *transfer*) action to inspect the intruder. They collect (*receive*) and *process* more data (signals) to *create* information that is *processed* to *create* more information.

Identifying the flowthings in the conceptualized system is a fundamental first step in FM. Flowthings are things that can be received, processed, created, released, or transferred. A conceptualization of a stream of flowthings may not necessarily contain all stages. For example, conceptualization of a physical airport can model the flow of passengers: arriving (received), processed (e.g. luggage and passports), released (waiting for boarding), and transferred (to planes); however, airports do not *create* passengers. In this case, the schema includes only the stages received, processed, released, and transferred.

3.2 Other characteristics

An important principle in FM is the separation of flows. Formally, FM can be specified as

$$\text{FM} = \{\textit{Receive}^*, \textit{Process}^*, \textit{Create}^*, \textit{Release}^*, \textit{Transfer}^*\},$$

where the asterisks indicate secondary stages [11]. For example, {Copy, Store, Delete, and Destroy} can represent these secondary stages. The flow between the five specific stages can be defined as the directed graph:

{(Receive, Process) (Receive, Release), (Release, Receive), (Process, Create), (Create, Process), (Process, Release), (Release, Process) (Create, Release), (Release, Create), (Release, Transfer)}

One 'inaccuracy' in this formalization is the use of arrows at (Release, Receive), (Release, Process), and (Release, Create). Each arrow denotes a 'return' flow. For example, if the communication channel is down for a long time, a decision might be made to return a message to a sender (creator, processor, or receiver) who previously released it. For simplicity's sake, our formalization does not guarantee that the released message is 'returned' to its previous state, i.e. an internal sender. If the schema represents a company, then receiving, processing, and creating information are shown as three different departments.

The formalization can be complemented with rules and constraints that permit flow from one state to another. There are properties of a flow, a phase, subphases, and a flow system. For example, a flowthing can be dated, or the number of flowthings 'inside' a phase, subphase, or system can be indicated.

Triggering in the context of FM means activation of a stage or substages that may generate a flow. Suppose the received stage is activated. In Receive, then, triggering may result in:

1. Activating a flow to Release.
2. Activating a flow to Process.
3. Mistriggering.

Mistriggering indicates that triggering has not succeeded. Triggering can specify a chain of flow; for example, a triggering in Receive can specify flow to Release or flow to Release and Transfer. In the last case, a chain of triggering is triggered.

FM is a map of possible flows the same way a city map shows possible routes. Suppose that flowsystem F1 is in a certain state. The state of F1 indicates the positions of flowthings in F1 at a particular point in time. This state can change depending on events, such as:

1. Flow from outside: another flowsystem, F2, transfers flowthings to F1.
2. Interior flow: previously arrived flowthings move automatically to the processing stage;
3. Interior triggering: the processing stage causes the creation of new flowthings.
4. Exterior triggering: another flowsystem triggers creation of a flowthing in S1.

The 'size' of flowthings seems to be an important factor in many applications. For example, according to Sarriegi *et al*. [14], 'information systems size' increases 'new risks'. Several factors influence the size of a flowsystem, including the following:

- The number of flowthings and their positions in stages.
- The number of stages and substages.
- The number of sub-flowsystems, and their hierarchical depth.

The 'size' of a flowthing can also be measured in terms of the number of flows/triggers included in the elements mentioned above.

4 FM ATTACK MODEL

The pride attack described previously can be conceptualized in an FM representation, shown in Fig. 9. Signals are received (detected) in the pride area (circle 1), triggering creation of information (2) that is processed, producing the first type of action (3) of seeking more data. Such a process reaches a threshold (4) when a plan is created, a decision made, and action taken to realize the actual attack. A simplified version is shown in Fig. 10 in terms of flows triggering flows. Mapping this pride attack to Cloppert's model, similarities and missing elements can be identified.

In the next subsection, the military-based USAF model and computer-based Cloppert's model are examined to reveal additional operations that can be incorporated into the pride attack model.

4.1 Revisiting the USAF attack model

Cross-examining the FM and USAF attack model, the following can be observed with respect to USAF model steps.

Detect, *locate*, and *identify*: These steps fall in the Signals–Information–Actions triangle of Fig. 10. *Detect* denotes the process of receiving signals about an object in the attacker sphere. *Locate* and *Identify* indicate processing and creation of information, triggered by received data. These steps seem to be elements in what the USAF calls *intelligence*.

Decide: These steps are suddenly followed by 'decide'. This is reasonable since the model represents an air attack, where an attack is a reflexive action. In addition, according to USAF [7], 'the attack cycle works on four assumptions . . . [including] direction and guidance provided for each of the six steps. . . . Execution planning prepares input for and supports the

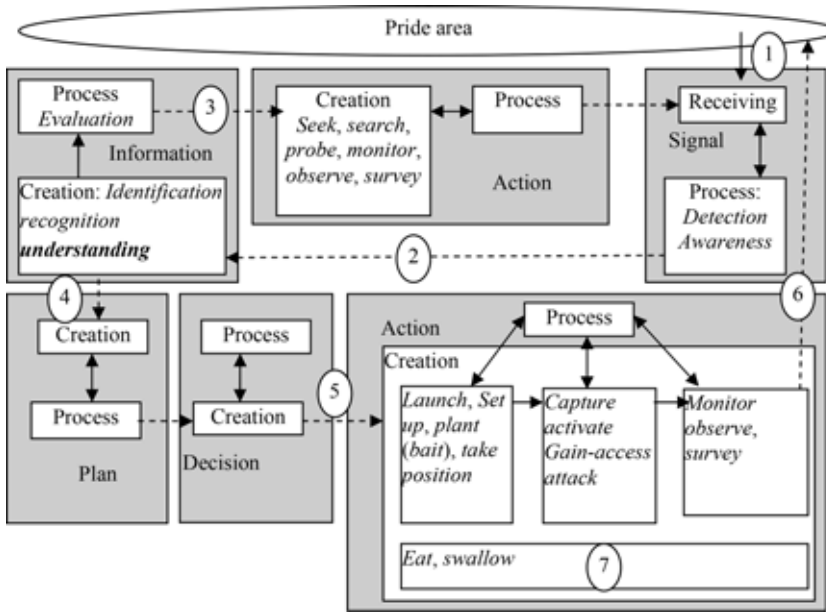

Figure 9: Conceptual model of pride attack.

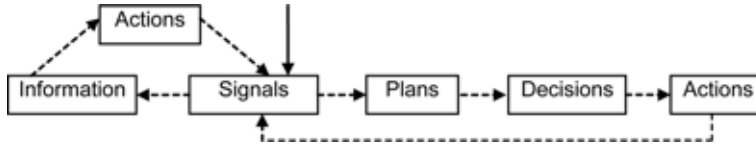

Figure 10: Pride attack modeled in terms of flows triggering flows.

actual tasking, construction, and subsequent execution by weapon systems'. The real reflexive action comes from the defender. As will be explained later, upon receiving a sudden attack action, the defender reacts in two ways: with a reflexive reaction and with a settled reaction involving the signal–information–action triangle and planning. *Execute*: This phase represents actions in FM. Thus, in general, 'decide–execute' and 'decisions–actions' in FM are aligned consecutively in the two models.

Execute: This phase represents actions in FM. Thus, in general, 'decide–execute' and 'decisions–actions' in FM are aligned consecutively in the two models.

Target: This term includes the meanings 'capture or destruction' and 'disruption, degradation, neutralization, and exploitation, commensurate' [7]. These are classified as actions in FM and performed after the attack is initiated. Attacks come in different kinds. In addition, the term 'exploitation' is used in Cloppert's attack model. These issues will be discussed in the next section.

The USAF report [7] discusses at length the term 'targeting' as an intersection of intelligence and operation. It talks about types of targets and target development and relates targeting to enemy, goal, objectives, and so forth. In FM, this means that some account of the 'target or victim, or defender' must enter the conceptual description of the attack. This seems

reasonable since offensive activities are interwoven with defensive activities. Some attack aspects may be well described from a defensive point of view; vice versa, some forms of resistance can be well viewed from the perspective of the attacker.

Assess: This phase in USAF can be mapped in the FM model by the arrow linking Actions to a triangle with Signals and Information. According to the USAF [7], 'After mission execution, the quality of the whole process is assessed. Improvements in force employment, munitions design and situation assessments emerge from this appraisal of post-strike data.

. . . The product of this phase is tailored to the decision makers'. 'Assess' may also indicate a high-level judgment such as that the attack is successful. Handling of this additional aspect in FM will be discussed later.

4.2 Revisiting Cloppert's attack model

Reconnaissance phase: From the attacker's perspective, reconnoitering is an offensive operation designed to obtain information before a battle; however, collection of information usually continues during battle. The three flows to the left – signals, information, and actions – reflect a more general view than the 'reconnaissance phase'.

Weaponization phase: This phase is missing from the pride attack. It can be pictured as a detour that leads to actions included in plans (e.g. acquiring software in preparation for the attack).

Delivery phases and compromise/exploit: Delivery phase is the action (attack) phase. It can be noted that actions are of different types. Actions in the reconnaissance phase are directed toward the outside environment, including the potential target (e.g. lionesses move closer to a specific kind and size of prey). The action in the delivery phase is directed at the prey itself. In 'delivery', 'compromise', and 'exploit', the semantics become clearer as who is doing what to whom is specified. The direction of the action is also important. In 'delivery', actions are created by the attacker and received by the defender. 'Compromise' seems to indicate that the defender does something. 'Exploit' seems to indicate that the attacker does something. Cloppert's compromise/exploit phase can be viewed as an action by the attacker towards the defender.

In attacks such as the pride's attack, however, actions are reciprocal between the attacker and the defender. From the defender's point of view, such action is called resistance. An attack is met with resistance. By 'compromise', it seems that Cloppert shifts from the attacker's sphere to the defender's sphere. 'The compromise phase will possibly have elements of software vulnerability, a human vulnerability aka "social engineering", or a hardware vulnerability' [9]. For us, this implies that to gain a complete conceptual picture of an attack, it is necessary to extend the description to the defender sphere, and this is done when Fig. 10 is redrawn to include aspects of Cloppert's model (Fig. 11).

The command-and-control phase: This phase seems to encompass all elements previously discussed. It includes the actions of collecting information during the attack, processing it, planning the next move in the field, making decisions, and taking the next action.

Exfiltration: This phase involves, in Cloppert's words, 'taking the data' [9]. This indicates 'pulling' flow from the defender to the attacker.

4.3 Revisiting the FM attack model

Taking these additional details into consideration, Fig. 11 shows the resulting conceptual model of the struggle between the attacker and the defender.

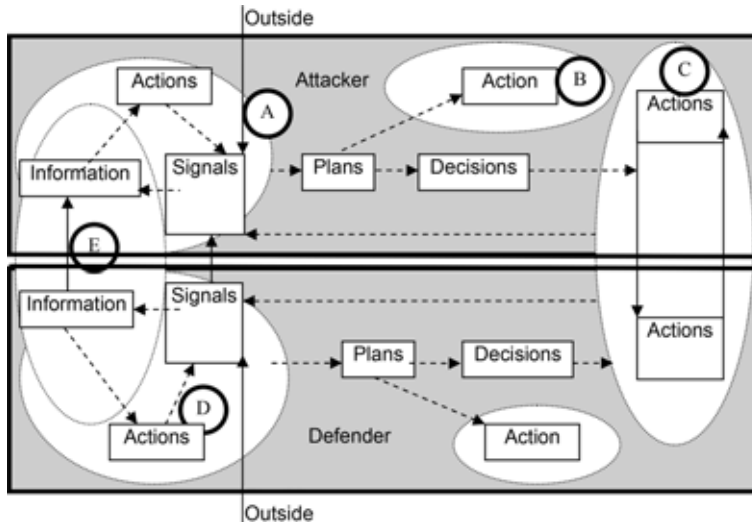

Figure 11: Streams of flows of attacker and defender.

The figure includes the spheres of the attacker and those of the defender. Including matters related to the defender is necessary for a complete model of attacks. This principle is implicitly indicated in the USAF model (Fig. 3) in the discussion about types of targets, enemy, goal, objectives, and so forth. These terms refer to the object under attack. The same idea is embedded in Cloppert's compromise phase, where the focus shifts to the defender.

Many processes can be found in the conceptualization of attacks and resistance to attacks. The first of such processes is the Signals–Information–Actions triangle in FM that corresponds partially to intelligence in the USAF model and to reconnaissance in Cloppert's model.

In Fig. 11, dotted oval A contains streams of signals, information, and actions. These three streams of flow correspond to circles 1, 2, and 3 in Fig. 9, a model of a pride's attack. Initially, the attacker receives signals from its environment (outside), but during the attack it also receives signals from the defender itself. Lionesses receiving buffalo noise 'pull up' updated signals about the buffalo's condition. In military battles, communication signals are constantly analyzed. Cloppert's command-and-control phase involves collection of signal–information during the attack. Hence, in Fig. 11, signals flow between the attacker and the defender.

Defender's oval E is the counterpart of oval A. The defender also receives signals from the environment and processes them. They trigger information that in turn triggers more signal–action (e.g. who is attacking me? size? direction? how large?). Oval E is the signal–information–action of the defender.

Returning to the attacker's sphere, it can be seen that Cloppert's weaponization phase can be attached to the 'plan' flow (oval B) as a type of action in preparation for the attack. These actions involve creation of a weapon.

After the decision to attack is made, it is time to take action, when the action (attack) flows from the attacker to the defender (oval C), indicated by a solid arrow from oval C to oval A and corresponding to 'delivery phases' and partially to Cloppert's compromise/exploit phase. Figure 11 indicates, however, that the defender may in its turn deliver reflexive counter actions, shown in oval C by a solid arrow from the defender to the attacker.

Alternatively, the defender may respond in a settled way: upon receiving the action, it triggers 'pulling up' signals in oval E (strength of the attack, time frequency, and so forth). Here a complete cycle is performed, with the defender receiving signals and processing them for information that triggers defensive action. The defender processes data from its environment and from the attack itself.

Finally, dotted oval D represents Cloppert's exfiltration, where the attacker 'pulls up' data from the defender's information sphere.

4.4 Combat model

The striking feature of Fig. 11 is the symmetry between the attacker's and defender's systems of flowsystems. Both collect information, plan, decide, and take action. Except for the fact that the attacker strikes first, the rest of the scenario is identical: repeated systems of flowsystems that represent a two-sided *combat*. Combat here means a hand-to-hand struggle or clash between two adversaries in which each side tries to wrest or maintain possession of something.

The OODA model was mentioned as a conceptualization of information warfare used by the Air Force [5,6]. It can be said that OODA models the attacker's sequence of actions. Our combat conceptualization takes a more comprehensive view, with the conflict represented from a third party's perspective.

Conventional stochastic combat models conceptualize combat as a sequence of independent events or interaction equations and center on the notion of attrition. For example, the salvo exchange model 'described combat as a pulse of offensive combat power designed to instantaneously penetrate an adversary's active defenses and cause damage to an adversary's platforms' [15].

A 'combat network' is conceptualized by Cares [15] as nodes defined as elements in a process, including sensors, deciders, influencers, and targets:

> Sensors receive observable phenomena from other nodes and send them to deciders. Deciders receive information from sensors and make decisions about the present and future arrangement of other nodes. Influencers receive direction from deciders and interact with other nodes to affect the state of those nodes. A target is a node that has some military value but is not a sensor, decider or influencer.
>
> A single node can contain the attributes of a sensor, influencer, decider or a target [15].

Besides its different context, it is clear that such an approach represents a type of conceptualization dissimilar to the FM-based representation. It can be seen that 'sensing' (the stage of receiving in FM), deciding (a flowsystem in FM), influencing (an outside triggering flowsystem in FM), and a target (a flow in FM, e.g. information resource flows to the attacker) are a heterogeneous mix of 'things' compared with the FM representation.

Capitalizing on the symmetry between the attacker's and defender's systems of flowsystems, reflected in Fig. 11, redundant features can be eliminated to arrive at the flow-based combat model shown in Fig. 12.

In FM there is no attacker or defender, only combatants. Flowsystems trigger each other, or flowthings flow from one flowsystem to another. The same system of flowsystems represents the combatant just as one flowsystem represents different spheres in the previously described FM (Fig. 6). The reflexive edges denote possible flows between flowsystems of the same flowthings.

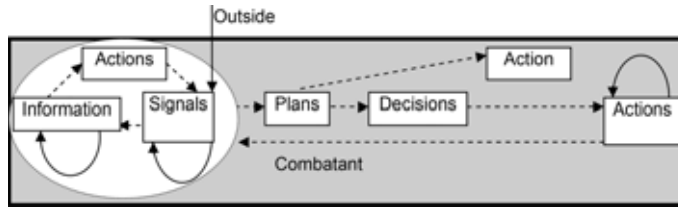

Figure 12: Flow-based combat model.

5 ATTACKS AND VULNERABILITY

As illustrated previously, an attack is a flowthing, A virus can be created, released, received, processed (e.g. activated), and transferred. Note that a flowthing is not necessarily a distinct 'object'. For example, if a virus copies itself, the new copy is a different flowthing; thus, there are two copies of the same virus, and one of them can be stored while the other can be transferred to another computer. Flowthings can have properties of a flow, a phase, subphases and a flow system. For example, a flowthing can be dated; the number of flowthings 'inside' a phase, subphase, or system can be indicated, etc.

FM is a framework for flowsystems that only orders them and helps make them understandable and comprehensible. It defines roles, patterns, responses, and so forth, and it defines particular ways to analyze policies, practices, routines, etiquette, and so forth. It also highlights consequences or outcomes of actions and intuits the effects of what is being done.

6 APPLICATION: SQL INJECTION

In this section, an FM-based conceptualization is applied to a sample attack. Specifically, the SQL injection described in Friedl [16] is discussed, where the following scenario describes this type of attack:

> "SQL Injection" is a subset of the unverified/unsanitized user input vulnerability ("buffer overflows" are a different subset), and the idea is to convince the application to run SQL code that was not intended. If the application is creating SQL strings naively on the fly and then running them, it's straightforward to create some real surprises.
>
> The login page [of the target] had a traditional username-and-password form, but also an email-me-my-password link; the latter proved to be the downfall of the whole system.[16]

Before the modeling of such a scenario begins, the preliminary phases of the attack are drawn, as shown in Fig. 13. First, the attacker focuses on a certain site. Friedl's attack model does not include the cause of the attacker's awareness of the target. He was asked to examine a certain intranet site used by a company's employees and customers. This was his first step in collecting information about the target.

> [The target intranet] appeared to be an entirely custom application, and we had no prior knowledge of the application nor access to the source code: this was a "blind" attack. A bit of poking showed that this server ran Microsoft's IIS 6 along with ASP.NET, and this suggested that the database was Microsoft's SQL server: we believe that these techniques can apply to nearly any web application backed by any SQL server. [16]

However, in general, 'targeting is a "special form" of intentions, meaning current intelligence suggesting plans for imminent attack against specific targets' [17]. Assuming that the

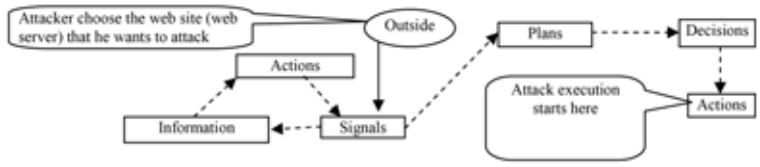

Figure 13: SQL injection web-based model.

attacker has targeted the site, our modeling of the initial steps of the attacker assumes that the attacker starts from scratch and can be divided into the following phases.

6.1 Pre-execution of attack

> To conduct offensive operations, a professional thoroughly investigates their target, carefully researching areas where they may be vulnerable. Discoveries that are potentially useful in an attack are noted, and then reviewed to develop an attack plan that is most likely to succeed with the least chance of being detected or contained. [18]

In Fig. 13, the attacker collects information about the victim's database server, such as names of tables, software used, version, etc. There are two types of attacker: inside the system, retrieving or modifying data with privileges needed to gain access, and outside the system, an attacker that tries to connect to, say, a web-based database. To do so, the attacker collects all information needed and begins making a plan to reach the point of decisive attack action.

Note the rhythm in Fig. 13 that reflects a continuing narrative. FM-based modeling provides such a continuity of logical events that occur in the attack process. For a comprehensive security analysis, several issues can be studied: what attracts hackers to web pages? How to minimize access to security-related information from signals? How to monitor coordinated actions that prompt signals to squeeze further information? Who are potential adversaries (e.g. users of the site)? What are their motivations and goals? How much inside information do they have? [19]

6.2 Attack execution

So, how did Friedl [16] begin executing the attack?

> The login page had a traditional username-and-password form, but also an email-me-my-password link; When entering an email address, the system presumably looked in the user database for that email address, and mailed something to that address. Since my email address is not found, it wasn't going to send me anything.
>
> So the first test in any SQL-ish form is to enter a single quote as part of the data: the intention is to see if they construct an SQL string literally without sanitizing. When submitting the form with a quote in the email address, we get a 500 error (server failure), and this suggests that the "broken" input is actually being parsed literally [16].

The attack is ended by accessing the file of the e-mails, including passwords.

By graphically representing the flow of information involved in communication between a (supposed) hacker and a system, conditions that lead to the appearance of vulnerability can be recognized. Figure 14 shows the combat conceptualization involved in Friedl's scenario described above. Note the missing flowsytems in the defender's sphere. Apparently, the

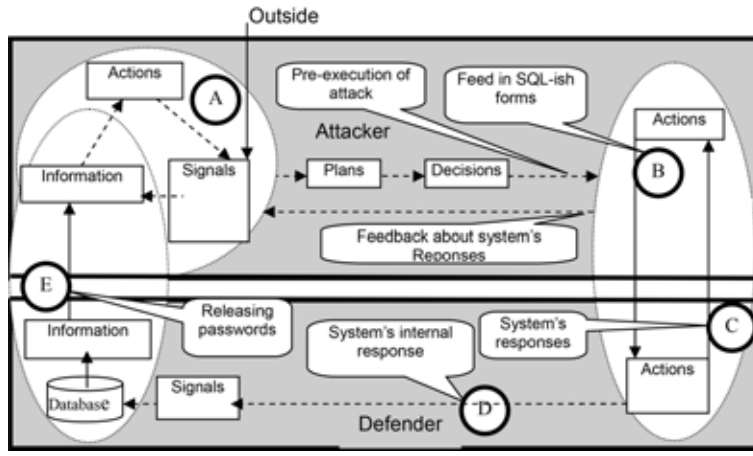

Figure 14: Applying streams of flow of attacker and defender to the SQL injection.

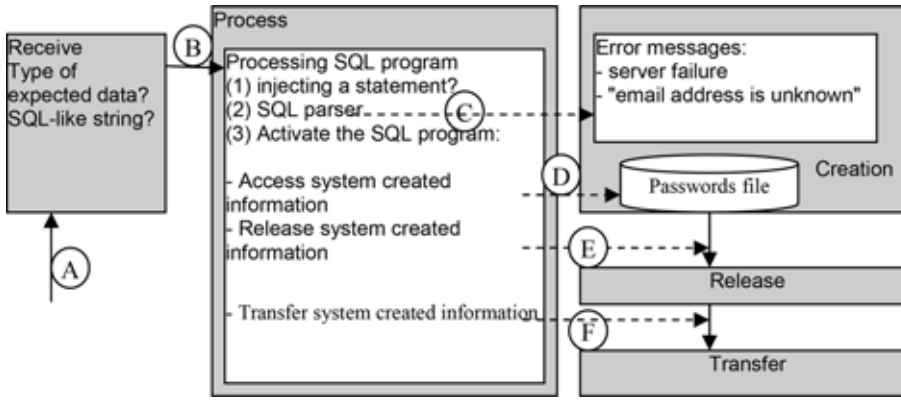

Figure 15: Modeling SQL injection attack.

system is not designed for combat. Input is received at point B, triggering an automatic reaction to all types of received input.

Figure 15 shows a partial FM-based view of the vulnerabilities in that system. First, the receive stage of the (victimized) system should set an alarm when input is the type of data that resembles an SQL-like string (circle A). At the processing stage (circle B), the sequence of processes should be designed with special security consideration given to the following:

1. Processes that inject statements into a program.
2. An SQL parser that creates errors, with careful consideration to reporting an error.
3. Activating of SQL programs that do the following:

Access a system file: A system file is created information that should have special security protection, compared with received or processed information.

Release information: The released stage should be in a special alert state when released information is created information.

Transfer information: The transfer stage is the last line of defense for catching unauthorized output.

These points in the stream of flow represent locations of necessary security checkpoints. Thus, multilevel checkpoints can be established in the developed system. These points can be used to develop checks against other types of attacks (basic operations). The method reveals critical locations along the stream of flow and identifies fine-grained operations that may cause a system alert.

7 CONCLUSION

This paper proposes a general conceptual model for uniform specification of attack progression in various phases. The model provides a more exact description in which attack-related 'things' (e.g. information, plans, decisions, and actions) are separated into different streams of flow with six specific internal operations: create, release, transfer, arrive, accept, and process. Three kinds of attacks are examined from very diverse domains: the animal world, the U.S. Air Force, and computer systems. The conceptual descriptions include the spheres of the attacker and the defender. Such attack modeling provides the possibility of more elaborate analysis in the field of security, and in computer and communication threat risks.

Further work would involve experimenting with modeling of real environments such as describing actual computer attacks. Another aim is to utilize the FM representation of attacks in other applications, such as design of protection strategies and development of security policies.

REFERENCES

[1] Moore, A.P., Ellison, R.J. & Linger, R.C., *Attack Modelling for Information Security and Survivability*. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep, 2001. CMU/SEI-2001-TN-001.

[2] Alberts, D.S. & Hayes, R.E., Understanding Command and Control. DoD Command and Control Research Program, 2006, available at http://www.dodccrp.org/files/Alberts_UC2.pdf

[3] Cloppert, M., *Security Intelligence: Introduction*. SANS Institute Computer Forensic Blog, July 22, 2009, available at https://blogs.sans.org/computer-forensics/2009/07/22/security-intelligence-introduction-pt-1/

[4] Johnson, D., *Effects-based Operations: A New Operational Model?* U.S. Army War College, 2002, available at http://www.iwar.org.uk/military/resources/effect-based-ops/ebo.pdf

[5] Brumley, L., Kopp, C. & Korb, K., *The Orientation Step of the OODA Loop and Information Warfare*, 2006, available at http://www.csse.monash.edu.au/courseware/cse468/2006/Lectures/OODA-Loop-BKK-IWC7-2006.pdf

[6] Schechtman, GM., *Manipulating the OODA Loop: The Overlooked Role of Information Resource Management in Information Warfare*. 1996, available at http://www.au.af.mil/au/awc/awcgate/afit/schec_gm.pdf

[7] USAF Intelligence Targeting Guide, *Chapter 1: Targeting and the Target. Air Force Pamphlet 14-210 Intelligence*, 1998, available at http://www.fas.org/irp/doddir/usaf/afpam14-210/part09.htm

[8] Smith, D.J., *Information Operations Primer*. U.S. Army War College, 2006, available at http://www.iwar.org.uk/iwar/resources/primer/info-ops-primer.pdf

[9] Cloppert, M., *Security Intelligence: Attacking the Kill Chain*. SANS Institute Computer Forensic Blog, October 14, 2009, available at https://blogs.sans.org/computer-forensics/2009/10/14/security-intelligence-attacking-the-kill-chain/

[10] Al-Fedaghi, S., Conceptual software testing: a new approach. *International Review on Computers and Software*, **8**(**8**), pp. 1832–1842, 2013.

[11] Al-Fedaghi, S., How the pride attacks. *9th European Conference on Information Warfare and Security*, Thessaloniki, Greece, July 1–2, 2010. Republished in: *Leading Issues in Information Warfare and Security Research*, Vol. 1, pp. 1–19, ed. Julie J. C. H. Ryan, Academic Publishing: UK, 2012.

[12] Al-Fedaghi, S., Some aspects of personal information theory. *7th Annual IEEE Information Assurance Workshop (IEEE-IAW 2006)*, United States Military Academy, West Point, NY, 2006, available at http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01652066

[13] Department of the Air Force, *Vistas: Air Force Information Resources Management Strategic Plan*, 1995. HQ USAF: Washington, DC.

[14] Sarriegi, J.M., Santos, J., Torres, J.M., Imizcoz, D. & Plandolit, A., Modeling security management of information systems: analysis of a ongoing practical case. *The 24th International Conference of the System Dynamics Society*, July 23–27, Nijmegen, The Netherlands, 2006.

[15] Cares, J.R., *An Information Age Combat Model*. Alidade, 2004, available at http://www.alidade.net/recent_research/IACM.pdf

[16] Friedl, S.J., *SQL injection attacks by example*, Steve Friedl's Unixwiz.net Tech Tips, October 10, 2007. http://www.unixwiz.net/techtips/sql-injection.html

[17] Bejtlich, R., Threat model vs. attack model, TaoSecurity: Richard Bejtlich's blog on digital security and the practices of network security monitoring, incident response, and forensics, June 12, 2007, available at http://taosecurity.blogspot.com/2007/06/threat-model-vs-attack-model.html

[18] Johansson, K., *The Offensive Operations Model*, v. 2.1. KSAJ, Inc., 2004 (accessed), available at http://www.penetrationtest.com/penetration_test_information_security_whitepapers/Offensive_Operations_Model.pdf

[19] Brown, K., *The .NET Developer's Guide to Windows Security: What is Attack Modeling*, 2007, available at http://alt.pluralsight.com/wiki/default.aspx/Keith.GuideBook/WhatIsThreatModeling.html