FuSIA: Future Situation and Impact Awareness

Jared Holsopple
Information Exploitation Group
CUBRC
Buffalo, NY, U.S.A.
Email: holsopple@cubrc.org

Shanchieh Jay Yang
Department of Computer Engineering
Rochester Institute of Technology
Rochester, NY, U.S.A
Email: jay.yang@rit.edu

***Abstract*—Recent cyber security research has focused on providing a situation awareness of computer networks by identifying incoming attacks. FuSIA: Future Situation and Impact Awareness seeks to extend this situation awareness via estimating plausible futures of ongoing attacks. Plausible futures, derived based on current progress of attacks, are projected situations that computer security analysts may use to determine appropriate actions for proactive defense. This work discusses the generalized framework of FuSIA as well as its application in cyber intrusion projection. FuSIA adopts application specific contextual information as well as provides flexibility by accommodating multiple projection algorithms. In particular, this paper presents threat projection algorithms via analyzing capability and opportunity of ongoing attacks. Plausibility scores derived from these algorithms are then combined based on Dempster-Shafer theory to provide a final fused estimate of plausible futures.**

Keywords: threat assessment, computer security, Dempster-Shafer.

I. Introduction

As computer networks increase in size and complexity, there has been an increasing need to quickly understand potential security breaches. Much research has focused on identifying single events or clusters of related events that could then be analyzed by analysts [1]. This paper extends the concepts set forth by TANDI [2] to create a general threat assessment framework, namely, FuSIA: Future Situation and Impact Awareness and apply it to cyber security. Given the current situation, FuSIA is responsible for logically generating *plausible futures* - estimates of potential future situations. These potential future situations are not to be treated as predictions, but rather as indicators to the analyst of potential future situations so they can adequately prepare or react should the situation occur.

Various tools have been developed to aid security analysts to make data more bearable to sort and understand [3]–[7]. A few of these techniques seek to correlate multiple events together into individual attacks. More recent research [2], [8] has focused on identifying the threats and impact of attacks to the computer network. These tools and techniques require knowledge of the specific computer network being monitored. When potentially malicious or suspicious activity is detected, Intrusion Detection Sensors (IDSs) send alerts to the security analysts [1]. Upon receiving alerts, analysts must quickly determine whether the activity has any current or future negative impact on the monitored network. This process consists of the determination of what has been compromised, what could be compromised, and what the impact of the current attack is on the network.

TANDI [2] was proposed to address the need to project cyber attacks into the future and introduced various performance metrics for Level 3 data fusion. TANDI combined capability with opportunity to determine the intent of the hacker. The intent of the hacker represented a projection of the hacker's next step.

The idea of projecting a current situation into the future extends beyond cyber security and applies to many other domains. Financial analysts, for example, try to project future financial situations based on a myriad of parameters. In warfare, it would be of interest to project the enemy's moves before they happen. However, it is generally impossible to accurately project every situation since there is an inherent uncertainty with projections. This work presents a framework that estimates simultaneous current situations. Information fusion plays a key role here in uncovering potential hidden plausible futures due to overwhelming and possibly conflicting estimates.

The rest of the paper is organized as follows. Section II discusses the FuSIA framework in detail illustrating how capability, opportunity, and intent are fused together to generate plausible futures. Section III illustrates an example assessment on a virtual terrain by FuSIA. Section IV concludes the paper.

II. Framework of FuSIA

Level 3 of the JDL data fusion model focuses on threat and impact assessment. Impact assessment is comprised of both the current and future impact. The future impact is determined by the impact of possible future situations. FuSIA generates future situations, or *plausible futures*, in a given environment, based on an an ontology modeling the specific relationships between objects and between activities that can occur for each object. The plausible futures can be analyzed by an impact assessment algorithm to determine the potential future impact. Figure 1 illustrates the overall FuSIA architecture.

FuSIA accepts real-time inputs known as *attack tracks*, which are assumed to be a grouping of observed events in the monitored environment. FuSIA consists of three major processing units – situation estimation, projection, and damage assessment. Situation estimation models the situation as a set of activities and affected entities or object in the monitored

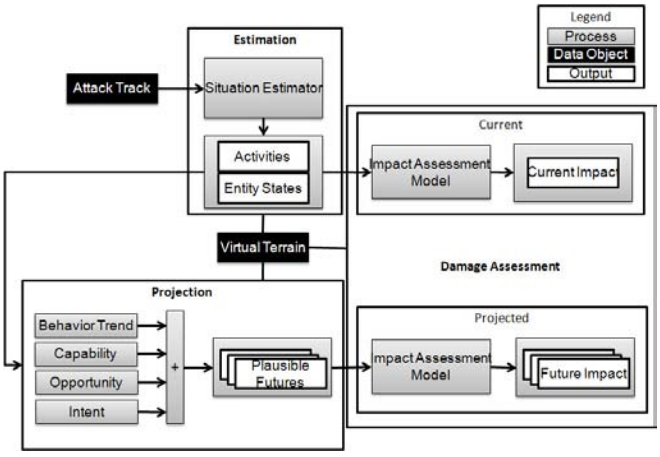

Figure 1. FuSIA determines the current situation, plausible futures, current impact and future impact.

environment. Projection identifies the plausible futures of a situation, while damage assessment uses an impact assessment model to determine the impact of the current situation to the monitored environment. This paper will focus on the situation estimation and situation projection processes of FuSIA. An example damage assessment block is described by Argauer and Yang [8].

Figure 2 illustrates how FuSIA projects the current situations using one or more algorithms. Multiple algorithms may operate in parallel to determine plausibility scores $p(\cdot)$ for each situation identified in the environment. The plausibility scores derived for each situation using different algorithms are then combined using Dempster-Shafer. The combined scores based on different current situations are then further combined to generate a list of plausible futures for each object in the environment. The subsequent subsections discuss this three-step process in detail.

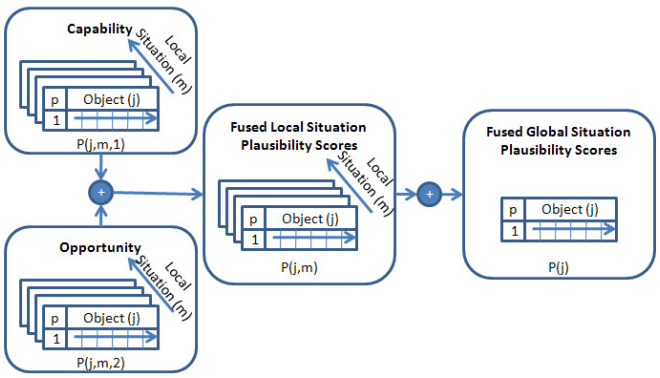

Figure 2. FuSIA's Situation Projection Architecture

A. Modeling the Current Situation

While situations in different domains can be interpreted differently, they all share the same atomic elements. We define a *local situation*, L, to be a set of related observations or inferences of the monitored environment. In other words, a local situation is a ordered collection of events, where an event is associated with an object and the activity observed for that object. One example of a local situation would be the locations and availabilities of military resources over time on a battlefield. In computer security, this would represent the progression of a multistage attack on a network. In many applications, there could be many unrelated activities taking place on the network at the same time. Therefore, we define the *global situation*, $G = \{L_1, L_2, ..., L_M\}$ as a collection of local situations, where M is the number of local situations. It should be noted that in many applications there is only a single situation that needs to be assessed, and the distinction between the local and global situation is unnecessary. However, the cyber context allows multiple, unrelated attacks to occur on the computer network and each may be modeled as a separate situation.

The actual representation of a local situation depends upon the application and is driven in FuSIA by an ontology - a contextual model that represents objects, relationships between objects, and relationships between activities that can occur for each object. Since there are different representations of an ontology for different domains, FuSIA only requires that it be represented in the language or formalism providing the expressiveness required to support the problem. For example, for computer security we use a graph-based ontology called Virtual Terrain [9] to model the monitored computer network. However, other applications may use a Bayesian network, OWL-based ontology, or some other model.

There are two requirements for an ontology to be used in FuSIA:

1) The plausibility algorithms will need to extract information from the ontology, so the format of the ontology must be supported by the plausibility algorithms.
2) The ontology must define specific *Objects of Interest*, J. This is a list of objects in the environment that can be assigned plausibility scores by the plausibility algorithms. These objects of interest represent entities within the ontology that can be acted upon and are of interest to an analyst.

B. Modeling Plausible Futures

A plausible future is defined as an event that extends from a current progression of events, i.e., a current situation. In mathematical terms, the plausible futures of a situation L may be expressed as $F = L \cup E$ where E is a set of future events.

The future events are the basis to determine the plausibility scores to various objects of interest. A *plausibility score*, $p(j)$, for object $j \in J$, corresponds to the plausibility that some event in the near future will relate to that object. A plausibility score takes on a value in the closed interval [0,1] or a value of $unknown$. A non-zero score indicates that it is logically possible for a future activity to occur on the given object. The higher the score, the more plausible it is for that event to happen. A score of zero implies that evidence suggests for

no future activity on that object. An $unknown$ threat score indicates that there is no information relating to the object that it will or will not be acted upon in the future. This is an important distinction since an $unknown$ score is processed differently than a zero score in the fusion process (see Section II-D). A plausibility score of 1, which would imply that there is complete certainty of the object to be acted upon.

There are two important issues that should be noted about the modeling of plausible futures. First, a plausibility score is not meant to be interpreted as a probability. It is an indication of how strongly the current evidence suggests that the given object could be acted upon. Also, unlike probabilities, plausibility scores are not required to add up to one among objects in the environment. Secondly, plausible futures are not meant to be treated as predictions. While a highly plausible situation might be more likely to occur, the plausible futures are simply meant to aid the analyst in identifying the future situations to quickly assess how to prepare or inhibit future events. In most applications, it would be intractable to completely and perfectly model the current situation. Likewise, it is very likely that not *all* events can be detected or inferred based on the situation and model. Whether it is quantifiable or not, there will always be an inherent error associated with the estimate of the current situation, which can propagate to errors in generating plausible futures.

C. Local Situation Plausibility Algorithms (LSPAs)

Since there are typically many different aspects that need to be accounted for in generating an accurate plausible future, a single algorithm assessing the entire situation may be too complex to implement or execute. The three main aspects of threat assessment are capability (the ability to execute an attack), opportunity (the environment permits the attack to happen), and intent (the attacker must have a reason for the attack) [10]. By assessing the three aspects separately and in parallel, the overall system may be more efficient and easier for model updates. Note that it could be difficult to have a clear division of the three aspects or to analyze intent in certain applications. For example, the authors argue that it is unlikely to determine the ultimate intent of a computer hacker with the openness of the Internet.

FuSIA simplifies the assessment by using multiple parallel algorithms that generate plausibility scores for each object of interest. A local situation plausibility algorithm (LSPA) is responsible for generating plausible futures for each local situation. A LSPA is meant to mimic a single expert analyzing a situation. Different experts may often have different (and potentially conflicting) opinions of plausible futures. In many instances, a LSPA would be a heuristic algorithm for the application of interest. In the example of projecting cyber attacks, we will show an example of how two heuristic algorithms (Demonstrated Capability and Opportunity) look at the cyber security situation from different perspectives.

The plausibility scores generated by each LSPA are then input to the fusion process where an estimate of plausible futures is calculated for each local situation. If an object of interest has already been compromised or acted upon, it may not make sense to assign any plausibility score to it. Therefore, a LSPA only needs to assign plausibility scores to objects in the set $J^* \subseteq J$, where J^* represents the objects that have not yet been compromised. To further improve efficiency, any objects in J^* that are not assigned a plausibility score are automatically assigned a value of $unknown$. Formally, a LSPA, k, analyzes a local situation L_l and assigns a plausibility score $p(j,l,k)$ for object $j \in J^*$.

1) Algorithm Reliability: There may be certain situations or parameters in which a LSPA performs very well or very poorly. Therefore, each LSPA must provide a reliability score, $r(j,l,k) \in [0,1]$, for each assigned plausibility score. The reliability scores allow for more reliable assessments to be weighted higher than less reliable assessments. The reliability can be calculated based on a number of factors. E.g., the error associated with evidences supporting the calculation, or the historical performance of the algorithm's assessments.

In terms of the historical performance of the algorithm's assessments, TANDI [2] defined a number of threat prediction performance metrics that FuSIA can utilize to generate reliability scores. In particular, a compromising score has been used to evaluate the accuracy achieved by TANDI. The compromising score is defined as the relative plausibility score a single event before the object is compromised. Mathematically, let $p_t(j,l,k)$ represent the plausibility score for Object j based on algorithm k's assessment of situation L_l at time t. The compromising score, $c(j,l,k)$, is defined as:

$$c(j,l,k) = \frac{p_{t^*-1}(j,l,k)}{\max_{\forall i \in J^*} p_{t^*-1}(i,l,k)} \tag{1}$$

where t^* represents the time in which the object was compromised. The denominator of the above equation normalizes the compromising score so that it is defined relative to the maximum plausibility scores of objects in J^* at time t^*. In the case where historical performance is considered at the algorithm level, a running average of $c(k) = \text{AVG}_{j,l}(j,l,k)$ can be used as the reliability scores for all assessments.

D. Fusion Process

Each LSPA assigns a single plausibility score to each object for each local situation. The fusion process is responsible for two assessments:

1) Combining the plausibility scores, $p(j,l,k)$, for an object j based on a local situation L_l and across multiple LSPAs, into a fused plausibility score $p(j,l)$.
2) Combining the fused plausibility scores, $p(j,l)$, from the previous process to calculate a plausibility score, $p(j)$, for each object j as the global future situation.

Since each LSPA is also associated with a reliability score, the fusion process should be able to take this information to weight more reliable scores higher than less reliable scores. Likewise, if the plausibility score is $unknown$, the fusion process should not factor that assessment into the calculation. In addition, the algorithms could present potentially conflicting results, so the fusion process must be able to take this into

account. Lastly, multiple high (or low) plausibility scores should increase (or decrease) the fused plausibility score.

Dempster-Shafer Theory (DST) [11] is a method to combine multiple uncertain observations into a single fused observation. However, Zadeh showed a rather simple example in which conflicting observations created counter-intuitive results. Recently, Haenni [12] showed that introducing reliabilities to different observations help to overcome this example. Since each of the plausibility scores generated by each LSPA contains a reliability, we decided to use DST to fuse plausibility scores.

There are four possible outcomes for each plausibility score generated: the outcome is plausible and the assessment is reliable (PR), the outcome is plausible and the assessment is unreliable (PU), the outcome is not plausible and the assessment is reliable (NR), or the outcome is not plausible and the assessment is not reliable (NU). We can therefore define the frame of discernment for DST to be:

$$\theta = \{PR, PU, NR, NU\} \tag{2}$$

We can then use the following mass function in DST combination:

$$m_{j,l,k}(A) =$$

$$\begin{cases} p(j,l,k)r(j,l,k) & \text{for } A = \{PR\} \\ p(j,l,k)\left[1 - r(j,l,k)\right] & \text{for } A = \{PU\} \\ \left[1 - p(j,l,k)\right] r(j,l,k) & \text{for } A = \{NR\} \\ \left[1 - p(j,l,k)\right] \left[(1 - r(j,l,k)\right] & \text{for } A = \{NU\} \end{cases} \tag{3}$$

The plausibility score can be extracted by calculating the belief of the PR and NU outcomes:

$$p(j,l) = Bel(\{PR, NU\}) \tag{4}$$

Likewise, the reliability can be extracted by calculating the belief of the PR and NR outcomes:

$$r(j,l) = Bel(\{PR, NR\}) \tag{5}$$

Now that the plausibility scores have been mapped to a mass function, fused plausibility scores, $p(j,l)$, can then be generated for each object and local situation. Extending this further, the $p(j,l)$'s can then be fused together to create a global plausibility score, $p(j)$.

As described previously, an unknown plausibility score implies that there is no evidence supporting or discounting the possibility of a future event occurring on that object. Therefore, the fusion algorithm must be able to ignore $unknown$ plausibility scores. Also, if all $p(j,l,k) = unknown$ for a given object, it must follow that $p(j,l) = unknown$.

III. Applying FuSIA to Computer Security

This section presents an example of FuSIA applied to computer security. The ontology will be described to illustrate how to model location situations. Two computer security LSPA's will be discussed, followed by the fusion of the plausibility scores and an interpretation of the results.

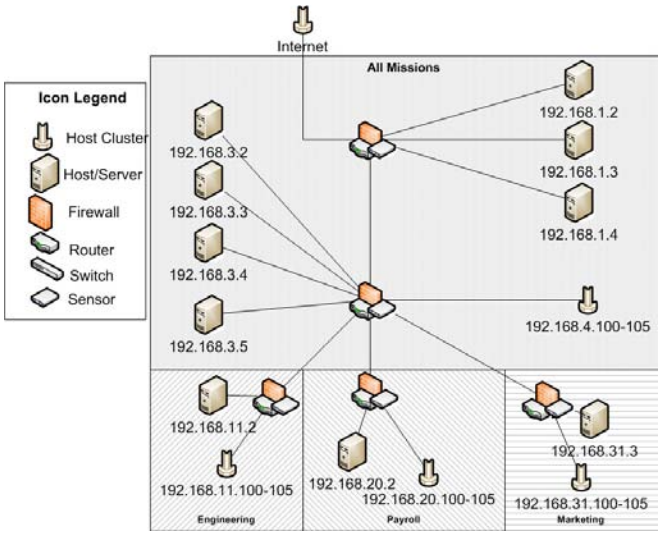

Figure 3. Example Virtual Terrain

A. Virtual Terrain

The virtual terrain [9] is a security-based representation of a computer network. The virtual terrain models not only the physical topology of the network (hosts, routers, firewalls, etc.), but also the configurations (routing rules, firewall rules, etc.) and vulnerabilities of the network or objects in the network. Like a true computer network, each host contains a list of services that can be remotely exploited. Each of these services contain a list of vulnerabilities that correspond to the equivalent IDS alerts and log files. The virtual terrain also contains firewall rules, which can be used to infer if traffic was filtered before it ever reached the destination. The information in the virtual terrain allows inferences on the current situation to be made based on observed events. One example inference that can be made is to determine if an attack on the network was successful. Figure 3 illustrates the example network used in [8] and [9], which will also be used as the ontology in the example presented in this section.

When monitoring a computer network, an analyst may wish to focus his attention on critical hosts, services, missions, or users on the network. When any of these entities are compromised (or close to being compromised), the analyst must be able to react in a timely manner to resolve that situation. While current technologies [5], [6] are able to identify related security events, they do not directly map these events to an estimate of the current state of the various entities. The entities for which the state is estimated are referred to as the *objects of interest*, J. FuSIA exploits information from the virtual terrain estimate the state of the objects of interest for a local situation. For simplicity, our example will only focus on four hosts and the associated services in the computer network shown in Figure 3 as the objects of interest.

B. Modeling the Current Situation

Most computer networks are monitored by IDSs and security logs [1]. These components provide a large volume of data

indicating specific security-related events in the computer network. In general, each event contains the following attributes:

- Type of Event
- Source IP Address
- Source Port
- Destination IP Address
- Destination Port
- Protocol (TCP, UDP, ICMP, etc.)
- Timestamp
- User name

Tools such as INFERD [5] and ArcSight [6] are able to correlate these security-related events into individual *attack tracks*, which in their basic form are groupings of related events. An example attack track is shown below which is an excerpt of a scenario in [8]. The scenario begins by scanning and intruding the external web server. The web server is then used as a stepping stone to attack the e-mail and FTP servers.

1) SSH Buffer Overflow
   140.203.195.48:80 → 192.168.1.2:80
2) WEB-MISC http directory traversal
   140.203.195.48:80 → 192.168.1.2:80
3) WEB-IIS .asa HTTP header buffer overflow attempt
   140.203.195.48:80 → 192.168.1.2:80
4) POP3 USER overflow attempt
   192.168.1.2:110 → 192.168.1.3:110
5) FTP adm scan
   192.168.1.2:21 → 192.168.1.4:21
6) FTP ADMw0rm ftp login attempt
   192.168.1.2:21 → 192.168.1.4:21

FuSIA treats each attack indepedently and as a local situation. In fact, IDS alerts in each attack track are observations of a local situation, and, thus, need to be referenced to the virtual terrain so as to estimate the state of objects in the network.

For each step in an attack track, the situation can be determined by answering the following two questions:

1) Was the observation indicative of a successful attack step?
2) Did the event cause the compromise of any objects of interest?

To answer the first question, the algorithm for detecting illogical attacks proposed in [8] can be executed. This algorithm first looks at the targeted host using the virtual terrain to see if it is vulnerable to the given attack. If it is vulnerable then it analyzes the path the traffic took to infer if it was filtered out before it reached the target. If the target is both vulnerable and the traffic was not filtered before it reached the target, then the attack can be assumed to be successful. If an attack was not successful, it is still of interest because the hacker demonstrated the ability to execute that attack. If an attack was successful, the next step is to determine if it compromised any objects of interest. Using these algorithms, objects in the virtual terrain can be assigned one of the following object states:

1) Normal - the current situation has no effect on the object.
2) Attacked - the object was unsuccessfully attacked.
3) Discovered - the object was discovered, but was not compromised.
4) Partially Compromised - the object was compromised but is not in full control by the attacker.
5) Compromised - the object is in complete control by the attacker.

Figure 4 shows the state of four hosts and associated services from the virtual terrain after the *first five* steps of the above attack were executed. We will look at the sixth step once the plausible futures have been generated to compare FuSIA's output with the last event. The external web server is not vulnerable to the *SSH Buffer Overflow* attack, so it was not successful and did not compromise anything on the network. The *WEB-IIS .asa HTTP header buffer overflow attempt* and *POP3 USER overflow attempt* both indicate the complete compromise of the external web server and mail server, respectively. The fifth step is simply a scan of the FTP server, but does not actually compromise the service or host. Since there are a number of hosts and services that have already been compromised, FuSIA does not need to assign plausibility scores to these objects. The objects of interest, J^*, that have not yet been fully compromised in this example are the host and service objects corresponding to the External FTP Server and Internal Web Server.

C. LSPA Algorithms

The TANDI framework [2] fused capability with opportunity to derive the intent of the hacker. FuSIA extends this methodology by defining two different LSPA's - capability and opportunity. The capability generates plausibility scores based on the attacks he has executed. The opportunity algorithm analyzes traffic flow using the virtual terrain in order to determine which hosts and services are actually accessible by the compromised hosts. Each of these algorithms will be described in how they assign plausibility scores to hosts and services.

1) Demonstrated Capability: One LSPA generates plausible futures based on the capability of the hacker. Ideally, capability would be modeled using an accurate behavioral model. However, not much is currently known about the modeling the behavior of a hacker [13] and such a model could arguably change as technology evolves. Therefore, this LSPA will focus on the demonstrated capability (DC) of the hacker. If a hacker has demonstrated knowledge of a certain attack or service, it is plausible that the same (or similar) attack can be executed in the future. The following four parameters are used to assign plausibility scores for a service:

1) p_{ss} is the plausibility score where a host that contains a service has been discovered but not compromised.
2) p_{su} is the plausibility score where a host that has not been discovered contains the same service as one discovered on another host.
3) p_{sk} is the plausibility score where a host that has been discovered contains the same service as one discovered on another host.

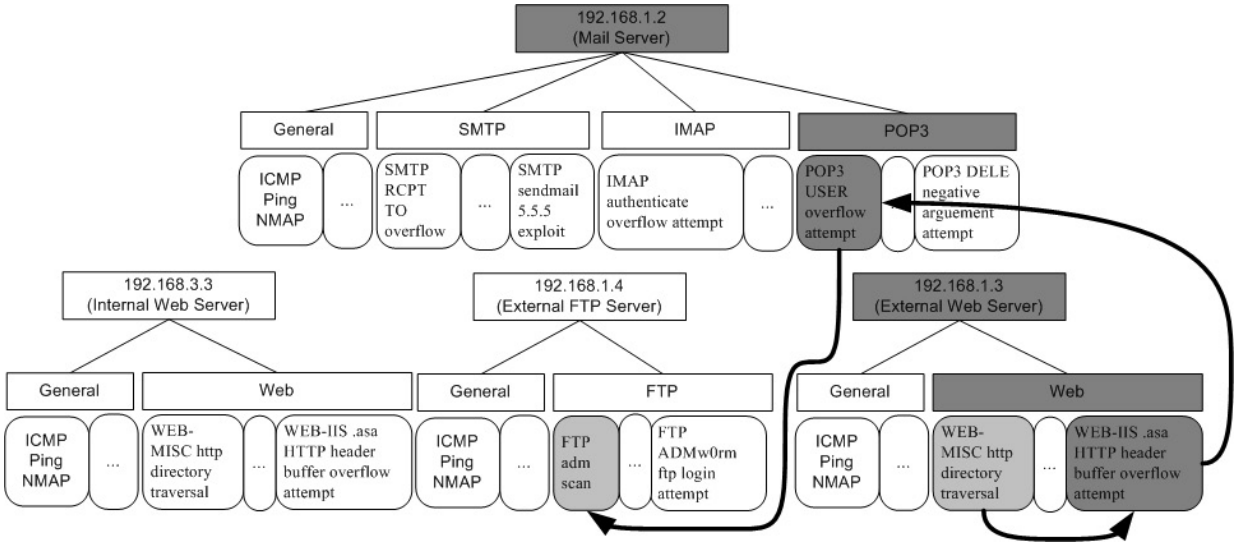

Figure 4. The estimated local situation of the network based on the attack track. Darkened hosts and services indicate compromised entities. The solid arrows indicate the sequence of the exploits.

4) p_{cu} is the plausibility score where a host that has not been discovered but contains the same service as one that has already been compromised.

Figure 5 illustrates the execution of the DC algorithm. The algorithm first looks at each compromised and discovered services and determines if there are any other hosts with those services. Using the parameters above, the plausibility scores are assigned to each service. In this example, the Web service in the external web server was compromised. This same service is running on the internal web server, so the algorithm sets the plausibility score of that web service to p_{cu} (which is 0.8 in this example). The FTP service has also been discovered on the FTP server, so that service is assigned the value of p_{ss} (which is 0.9 in this example). If the algorithm needs to assign multiple values to a service, the maximum of the values is used as the plausibility score.

The plausibility scores for each host are calculated using the plausibility scores assigned to the host's service. The plausibility score for the host is the maximum weighted plausibility score of the services. Some services do not run at a system level on a host, so even if that service is compromised, it does not completely compromise the host. Therefore the weights of each service are defined based on the privilege level the service is running at. The external FTP server's FTP service is running at the system level, so the plausibility score for the host is set to 0.9. Likewise, the internal web server's web service is also running at the system level, so it is assigned a value of 0.8.

2) Opportunity: The opportunity exposed to a hacker depends on the progress he made on the network. In a tightly configured network, some servers or hosts are hidden behind one or more firewalls. Hackers may use compromised machines as stepping stones to penetrate the network. Referencing to the virtual terrain, the opportunity algorithm analyzes the firewall rules between the compromised machines and the rest of the network and determines the level of threats imposed on each machine.

In the example, the two compromised hosts (192.168.1.1 and 192.168.1.3) are in the same subnet as the External FTP Server (192.168.1.4), and the Internal Web Server is separated by two firewalls and is in a different subnet - recall Figure 3.

The algorithm determines the closed ports between all pairs of compromised and non-compromised hosts using the known firewall rules.[1] The closed ports between a compromised host, i, and non-compromised host, j, are denoted by C_{ij}. Let S_{jk} denote the set of ports the kth service of host j. Define d_{jk} as the discounting factor for services that are still in the normal state, i.e., they have not been discovered, attacked, or compromised. Let $d_{jk} \in (0, 1)$ if the service is still in the *normal* state and $d_{jk} = 1$ otherwise. The following parameters define the plausibility score, p_{jk} assigned to that service:

$$p_{jk} = \begin{cases} p_{open} * d_{jk} & \text{for } C_{ij} \cap S_{jk} = \{\} \\ p_{closed} * d_{jk} & \text{for } C_{ij} \cap S_{jk} = \varnothing \\ p_{partopen} * d_{jk} & \text{otherwise} \end{cases} \quad (6)$$

where p_{open}, p_{closed}, and $p_{partopen}$ all are in the closed interval of $[0, 1]$.

The final plausibility score assigned to a service is the maximum of the values assigned based on the above equation. A host's plausibility score is the maximum plausibility score of its child services.

Figure 6 shows the plausibility scores assigned to the hosts. In this example, $p_{partopen} = 0.6$, $p_{open} = 0.8$ and the discounting factor is 0.5. The FTP service on the External FTP Server is assigned a value of p_{open} since there are no

[1] An efficient algorithm, such as breadth-first search, is under investigation to provide scalable implementation of the opportunity algorithm.

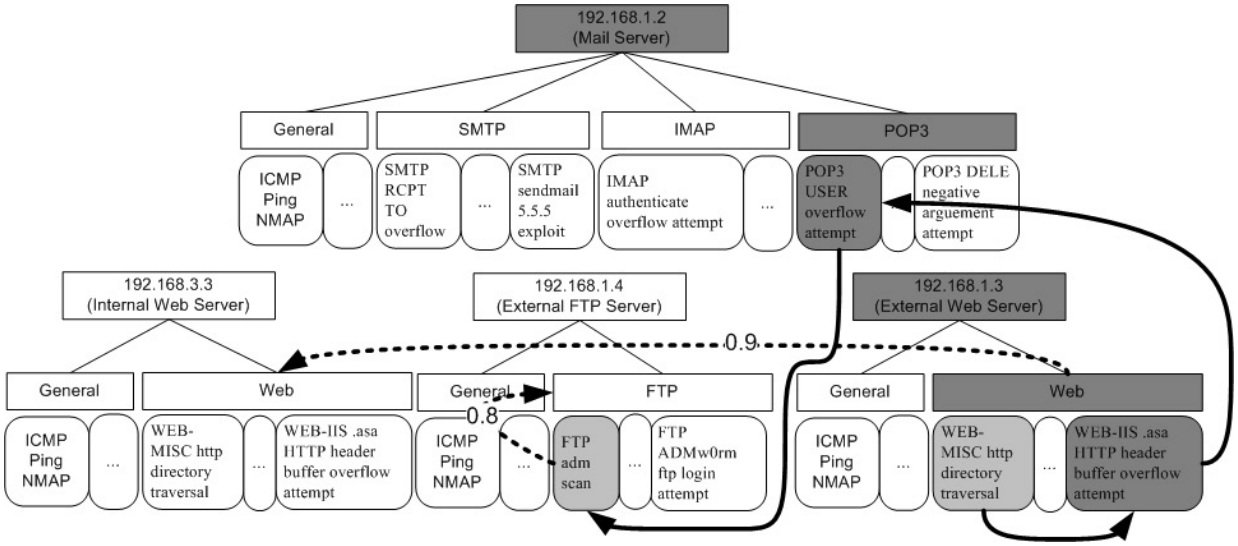

Figure 5. Example Capability Algorithm Execution

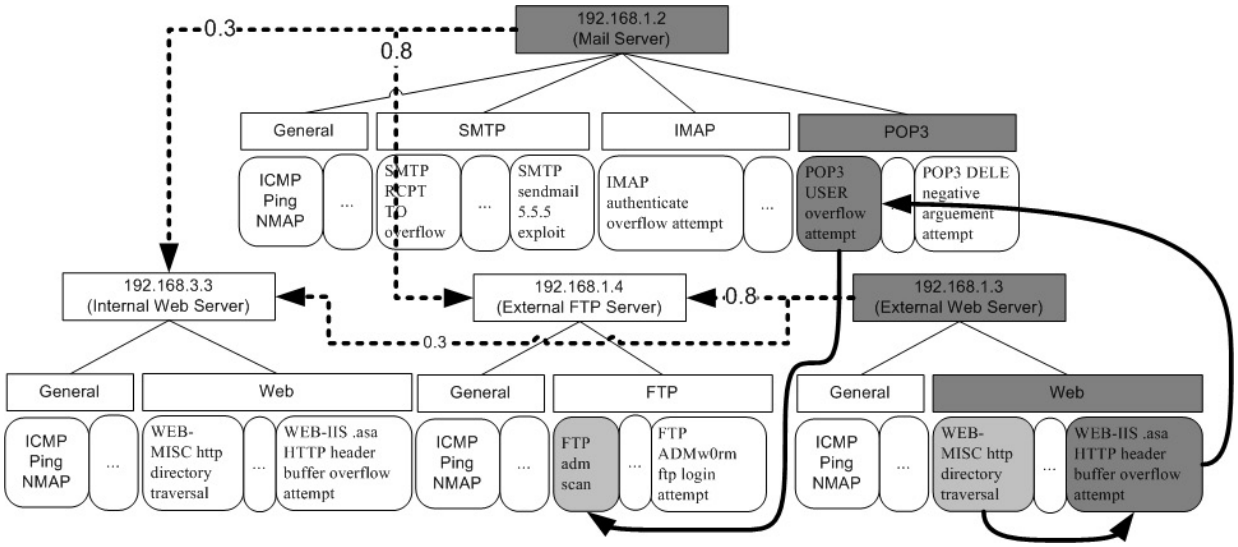

Figure 6. Example Opportunity Algorithm Execution

restrictions on the ports between it and the two compromised hosts. That score is then propagated up to the host. The Internal Web Server is assigned the value of $p_{partopen} * d = 0.3$ since there are some firewall restrictions between it and the two compromised hosts.

3) Example Assessment using Dempster-Shafer Fusion: Figure 7 shows the plausible futures due to the first five steps of the example attack. The top portion of the table shows the plausibility scores for the External FTP Server and the bottom portion shows the plausibility scores for the Internal Web Server. The sixth step in the example attack is the compromise of the External FTP Server. Examining the plausibility scores of both hosts, one can identify the compromise of the External FTP server as being more likely than the Internal Web Server. This deduction makes sense because the hacker exhibits knowledge about the FTP server by scanning it, which can be a direct indication that the hacker wishes to exploit that service. However, it may not have been as obvious that the Internal Web Server could also be compromised. The capability algorithm reveals that the hacker has demonstrated his skill in exploiting web services and the opportunity algorithm reveals that the network, in fact, permit malicious traffic to reach the Internal Web Server from the compromised external servers.

The FTP and web services running on their respective hosts have an identical score as their hosts. This is intuitively correct since a compromise of either service would also lead to the compromise of the host. Recall that the opportunity algorithm propagates the plausibility score from the host if the relevant ports or protocols used are blocked. In this situation,

	External FTP Server					
			General		FTP	
	Plaus	Rel	Plaus	Rel	Plaus	Rel
Capability	0.80	0.40			0.80	0.40
Opportunity	0.80	0.80	0.80	0.80	0.80	0.80
Fused	**0.70**	**0.73**	**0.80**	**0.80**	**0.70**	**0.73**
	Internal Web Server					
			General		**Web**	
	Plaus	Rel	Plaus	Rel	Plaus	Rel
Capability	0.90	0.40			0.90	0.40
Opportunity	0.30	0.80	0.30	0.80	0.30	0.80
Fused	**0.63**	**0.73**	**0.30**	**0.80**	**0.63**	**0.73**

Figure 7. Example Dempster-Shafer Combination

the plausibility score of the service would differ from the plausibility score of the host since the opportunity algorithm assigned them different scores.

Recall Figure 1, which shows how FuSIA generates plausible futures not only for identifying future situations, but also for determining future impact. In most computer networks, the data stored on the internal network is often more critical than the data stored on the external servers. The internal web server would generally contain information that should only be accessible to employees and would contain sensitive information. The fact that it is plausible for the hacker to attack the internal web server could be cause for concern, especially if it might not have been obvious to the analyst. The plausible futures could be used by an impact assessment system to identify the impact each plausible futures might impose.

IV. Conclusions/Future Direction

This paper presented a framework named FuSIA to generate potential future situations, called plausible futures, to aid in the calculation of future impact and illustrated an example of how it is applied to computer security. FuSIA generates plausible futures by executing multiple algorithms in parallel that each generates estimates of plausible futures with associated reliability. These (potentially conflicting) estimates are then fed into a fusion process, driven by DST, that weights the assessments based on the reliability to calculate a final fused estimate of the situation.

An ongoing work is to test FuSIA over a large dataset to analyze its performance and the effect of parameters defined in each LSPA. In addition, FuSIA will incorporate more LSPAs, such as the behavioral model presented in [13], as well as an impact assessment algorithm to determine current and future impact. Finally, the authors plan to apply FuSIA to other application domains.

Acknowledgements

This work is funded through the National Center for Multisource Information Fusion (NCMIF) grant under the technical supervision of AFRL/IFEA. The authors would like to thank Adam Stotz, Moises Sudit, John Salerno, Michael Hinman, and George Tadda for their comments toward the development of FuSIA.

References

[1] T. Bass, "Intrusion detection systems and multisensor data fusion," *Communications of the ACM*, vol. 43, no. 4, Apr. 2000.

[2] J. Holsopple, J. Yang, and M. Sudit, "TANDI: Threat assessment of network data and information," in *Proceedings of SPIE, Defense and Security Symposium*, vol. 6242, April 2006, pp. 114–129.

[3] S. Noel, E. Robertson, and S. Jajodia, "Correlating intrusion events and building attack scenarios through attack graph distances," in *Proceedings of ACSAC*, December 2004.

[4] O. Dain and R. K. Cunningham, "Fusing a heterogeneous alert stream into scenarios," in *Proceedings of ACM Worksop on Data Mining and Security*, December 2001.

[5] M. Sudit, A. Stotz, and M. Holender, "Situational awareness of a coordinated cyber attack," in *Proceedings of International Data Fusion Conference*, Quebec City, Quebec, CA, July 2007.

[6] ArcSight, "ArcSightESM," `http://www.arcsight.com` (accessed March 2008).

[7] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," in *Proceedings of the 1998 workshop on New security paradigms*. New York, NY, USA: ACM Press, 1998, pp. 71–79.

[8] B. Argauer and S. J. Yang, "VTAC: Virtual terrain assisted impact assessment for cyber attacks," in *Proceedings of SPIE, Defense and Security Symposium*, March 2008.

[9] J. Holsopple, B. Argauer, and S. J. Yang, "Virtual terrain: A security-based representation of a computer network," in *Proceedings of SPIE, Defense and Security Symposium*, March 2008.

[10] E. Little, G. Rogova, and A. Bourry-Brisset, "Theoretical foundations of threat ontology for data fusion applications," DRDC-Valcartier, Tech. Rep. TR-2005 -269, November 2005.

[11] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[12] R. Haenni, "Shedding new light on zadeh's criticism of dempster's rule of combination," in *Proceedings of 8th International Conference on Information Fusion*, July 2005.

[13] D. Fava, J. Holsopple, S. J. Yang, and B. Argauer, "Terrain and behavior modeling for projecting multistage cyber attacks," *2007 10th International Conference on Information Fusion*, 9-12 July 2007.