Novel Data Reduction Techniques in Large-Scale Network Infrastructures through Impact and Vulnerability Assessment

[1]Robert F. Erbacher, [2]Anupama Biswas, [3]Trent Cameron
[1]*U.S. Army Research Laboratory, Robert.F.Erbacher@us.army.mil*
2 Utah State University, Anupama.Biswas@aggiemail.usu.edu
3 Utah State University, Trenton.Cameron@gmail.com

Abstract

Large-scale networks generate enormous numbers of events that network analysts must parse through in order to determine which are malicious attacks and which are not. Additionally, network analysts must prioritize the events such that the most severe attacks are resolved first in order to limit the potential for damage to the network as much as possible. While there exist many data reduction and event correlation techniques for reducing the amount of data needing analysis, these techniques do not provide prioritization capabilities.

This paper discusses our novel impact and vulnerability assessment techniques geared towards the prioritization of events. This will aid network analysts and managers in identifying and resolving the most critical events first. Our techniques will work with the already existing data reduction techniques. The impact assessment technique identifies the potential impact of an attack while the vulnerability assessment identifies the likelihood that the attack will come to fruition. Combined they provide a significant advance in providing the automatic prioritization of attacks. Additionally, specific examples are provided to show how impact assessment scores would be computed in association with actual events. This impact assessment as an automated prioritization scheme will greatly improve the efficiency of the analysis process and reduce the amount of data needing to be transmitted over the network. Thus, all aspects of performance and efficiency of concern to network analysts and managers will be improved through the employment of such techniques. This is particularly true for large- scale networks with large numbers of valid connections in which the number of events needing analysis is even more substantial than typical networks.

Keywords: *impact assessment; vulnerability assessment; data prioritization; data reduction; network management*

1. Introduction

Large-scale networked resources are continuing to exacerbate the problem of network management and identifying and isolating attacks, especially sophisticated attacks. Network resources designed for large-scale connections such as national and international databases have additional issues associated with management of the network connectivity. Such databases would include the data generated from the Large Hadron Collider, the Human Genome Project, Google maps, etc. There are numerous examples of similar or related databases that allow or require large numbers of external connections. The management of such network connectivity is challenging because of large numbers of people legitimately attempting to connect to the local resources and the need to identify attacks within this morass of connectivity. In essence, attackers will be able to use the legitimate connections to obfuscate their activity. While network managers must identify and eliminate malicious connections, they must avoid negatively affecting valid connections, while simultaneously preventing the malicious activity from negatively affecting the legitimate connections. This creates a huge problem of scale, as enormous numbers of events must be analyzed in order to classify events and resolve those deemed malicious.

The goal of this research was to develop novel techniques to aid in the analysis and interpretation of this massive amount of data accurately. The chaotic nature of network traffic data makes it very difficult to differentiate normal from malicious traffic. Typically, network managers will have to resolve large numbers of events that may or may not be malicious in nature. To some extent their goal must be to prioritize the events based on the likelihood of maliciousness and potential ramifications of the event should it prove to be malicious. Thus, the question that arises is, how can automated techniques be used to prioritize events in a way that is helpful to analysts resolving the most critical events first?

More specifically, while current techniques attempt to classify events as malicious or innocuous they cannot do so with absolute certainty. This leaves huge numbers of events that the analyst must manually examine. This requires that the analyst attempt to prioritize the remaining events in order to resolve the ones with the greatest potential for negative impacts on the network and its users. Our techniques are designed to fill this gap. For instance, military networks see hundreds or thousands of attacks per day [24]. While many of these attacks will automatically be blocked by firewall rules, there will be dozens per day that are not. The network analyst must choose which attack to resolve first, essentially creating a priority order for the resolution of the attacks. The goal with such a priority list is to minimize the total impact, damage, of the attacks. Thus, the goal is to give highest priority to the potentially most damaging attacks. If a high impact attack is not given sufficiently high a priority, it will essentially be allowed to damage the network until it does get resolved. This prioritization of events is currently done in an adhoc fashion, leading to a great potential for misidentification of the potential impact of attacks. This could be especially problematic on particularly eventful days.

For this research, we developed an automated impact assessment algorithm designed to work with existing capabilities to assist analysts in prioritizing network events based on the potential severity of the identified event sequences. The impact assessment scores identify the potential (expected) impact of an attack on the network and associated resources. It identifies the extent to which resources will be degraded by the attack. Thus, our capabilities allow for the analyst to more efficiently and effectively target the events with the greatest potential impact on the network. In turn, this will greatly reduce any potential impact on the legitimate system users. Additionally, we propose the application of vulnerability assessment techniques to be used in conjunction with the impact assessment algorithm to determine the potential impact of an event or event sequence more accurately. A highly vulnerable system is more likely to be susceptible to a given attack and thus the priority for an attack against such a system must be higher. The network analyst would thus give priority to high impact attacks being targeted at highly vulnerable systems, Figure 1. A future situational awareness visualization [1][7][11][15][23] of the impact and vulnerability assessment data essentially makes the network state far more approachable and comprehensible to the network analyst.

The remainder of the paper is organized as follows. Section 2 discusses the proposed impact assessment technique. Section 3 discusses the vulnerability assessment technique that would be used to identify the likelihood that the attack would be successful and the identified impacts come to fruition. Section 4 discusses relation to previous work and examines other techniques for data reduction. This section also discusses how these techniques can be used in conjunction with our capabilities. Section 5 discusses future work and section 6 discusses conclusions.

2. Impact assessment

The primary component of this research was the development of techniques for the identification and representation of the impact of an event. Our concept with respect to the presentation of impact is to perform an analysis of a wide range of events and associate these events with associated military impacts. For instance, the most applicable association is with readiness, i.e., cyber readiness. In other words, how ready is the network to deploy missions or countermeasures? Specifically, a denial of service attack would decrease available bandwidth associated with a specific computer system. This would be associated with an impact on the readiness of the effected computer system and a reduced ability for said system to deploy operations. In highly load-balanced environments, the associated impact would be similarly reduced. Corresponding impacts must be identified, associated with the full range of analyzed attacks, and visually represented. In essence, when computing an impact assessment score, we are attempting to compute the amount of degradation of available resources. More importantly, however, is identifying the levels at which the degradation in available resources negatively affects our ability to use those resources.

It is this concept of impact that will bring together, link, and correlate groups of events in a meaningful way for the network analyst and network manager. Rather than simply identifying the existence of an event, we will be attempting to identify what this event and other events together mean to the decision maker.

In effect, the goal of these mappings is to identify the operational readiness of the cyber infrastructure as a whole, as well as individual components. The mappings are designed in a generic format such that they can easily be adjusted based on considerations of each local environment. Of particular importance will be the need to be able to change the associated value with each impact as this can vary greatly.

Ultimately, the impact assessment will provide a prioritization score to the network analyst. In essence, the network analyst must deal with enormous numbers of events on a daily basis, especially for larger

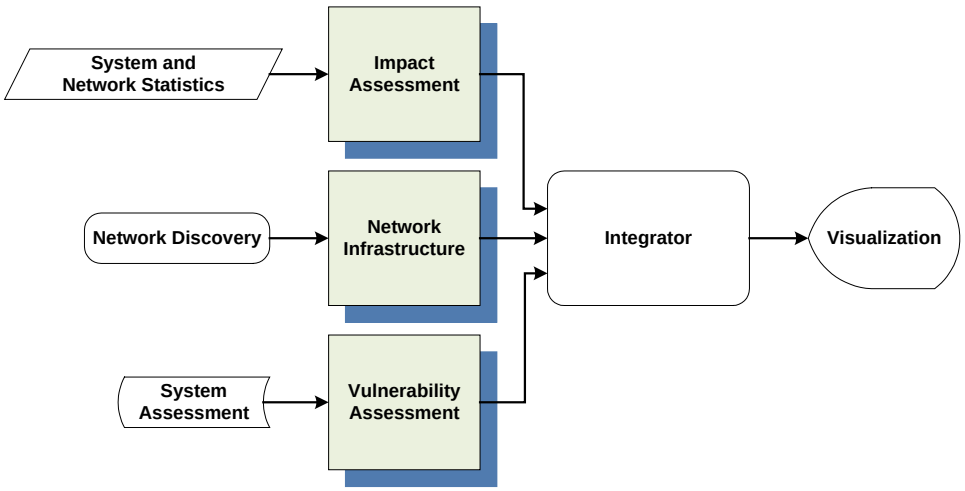

Figure 1. Diagram of impact and vulnerability assessment and integration for visualization.

organizations. These events must be prioritized in order to identify a sequence in which they will be resolved. Currently, this prioritization is adhoc and based solely on the network analyst's expectations of the event significance. The goal here is to create an impact assessment that identifies for the network analyst the current and predicted impact of each event according to the network analyst's specified metrics.

2.1. Impact assessment computation

Our impact assessment computation is derived from the work by Hariri et al. [19].

In essence, Hariri et al. provide two mechanisms by which impact can be computed. These are dependent on how the network analyst wishes to specify acceptable impact. First, the network analyst can specify the minimum acceptable resources that must always be available. Alternatively, the network analyst can specify the maximum acceptable usage of available resources. The equations specified by Hariri would then be:

- CIF(Router, FSk)=|Bfault-Bnorm|/|Bmax-Bnorm|
- CIF(Client, FSk)=|TRnorm-TRfault|/|TRnorm-TRmin|

Where CIF is the computed impact factor. Here, the assessment is per resource and the individual values for each resource would have to be summed to generate total system impact. For our purposes, we also include priority. Different systems will have different levels of importance to the network analyst, i.e., having a server heavily impacted is of greater concern to the network analyst than a common desktop. Thus, our inclusion of priority is designed to account for the variation in systems importance. Therefore:

- CIF'=CIF*$\text{Priority}_{\text{system}}$
- CIF=0..1
- 0 is normal
- 1 is abnormal
- Priority=0..1

For this research, we initially computed impact factors manually as follows:

1. Identify a set of attacks covering a wide array of attack types.
2. Simulate these known attacks to quantitatively and accurately identify their impact on resources. This formed our initial attack corpus [4].
3. Map the attacks identified in the sample attack data to these known attacks. In essence, we are attempting to map the characteristics of new simulated attacks against attacks in our known attack corpus to determine if the sequence of events should be considered an attack and to derive its future potential impact. This can result in an attack in the sample attack data being partially mapped to multiple known attacks, resulting in a list of percentages.
4. The resultant impact factors for the sample attack data are calculated.
5. These values are stored in our database for use by the network analyst, likely through situational awareness techniques.

We have performed simulations of attacks to identify the impact those attacks have on network bandwidth, CPU utilization, memory utilization, and disk utilization. The second step in the process was to manually map the identified impacts onto the attacks identified in sample attack data. These impacts could be further mapped to available services, systems, and missions. The full set of results is beyond the scope of this paper but appear fully in [4]. While we currently map the simulated (new) attack to the attacks in the attack corpus manually, this could be done in an automated manner in the future through attack graph analysis or related techniques.

Caveat: The fact that we store the potential impact rather than simply calculating it allows for the analyst to directly modify some of these values. This is critical for attacks in which the rate of resource consumption, as a primary measure of impact specification, is inappropriate. A common example for which this is the case is botnets. For botnets we will want to relate a much greater impact than the simple rate of consumption of resources implies.

2.2. Empirical formula derivation

This section provides more specifics as to how the impacts factors of the sample attack data were generated. For the purpose of this analysis, we considered the following performance parameters:

- CPU Usage
- Memory Usage
- Disk Usage
- Network Bandwidth Usage

Threshold values of the attacked system and network: These are the values of the victim network and system before the attack:

Network Bandwidth Capacity: 94.13 Mbps CPU: 3 GHz
Memory: 1 GB
Disk: 2.5 GB

For our initial example of the impact assessment computation, we considered two attacks: ICMP Flood attack and ICMP ping NMAP. All attacks were implemented to perform the simulation, amounting to ~1100 lines of code. The ICMP Flood attack was previously computed and its impact metrics stored in our attack corpus. The goal was to deploy an actual simulation of the ICMP ping NMAP attack, correlate it with the ICMP Flood attack, and determine how the ICMP ping NMAP attack's resource usage compared to that of the ICMP Flood attack's resource usage. These two attacks were compared due to their similarity. In a deployed capability, identifying this relationship would allow the network analyst to predict the future potential impact of a new attack, an attack not currently in the attack corpus.

For this evaluation, we are considering only a single system under attack. The simulations used three systems, running through a single switch, namely: the target systems, the attacker system, and the detection system running the Wireshark protocol analyzer. This was a controlled environment to measure only the impact of the attack so there were no other users on the system. The values of all parameters were measured before the attack was initiated to acquire the ground truth data. The parameters were again measured after the attack was initiated. The difference in values identified the impact of the attack.

All results presented are associated with the single target system associated with the simulation. If multiple systems were being affected then the CPU usage, memory usage, and disk usage would include the impact of the attack on all affected systems; thus scaling them appropriately. Differences in system configuration may prevent a simple linear scaling of the impact.

The severity of the attack's impact is measured by the rate at which the resources are being used. Hence, the measurements should be in such a format that represents the amount of consumption of the resources as well being comprehensible to the network analyst. Hence, the network bandwidth usage is measured in Mbps. Memory usage and disk usage are measured in GB/min, which indicates the amount of storage-based resources being used if the attack continues. For CPU resources, a typical usage based on CPU load is acquired, which has a value of *zero* for no load and a value of *one* for the full-time usage of one equivalent CPU. A difficulty arises, however, since CPU load is not directly related to other CPUs. Thus, to generalize the same here, there is a need to map the CPU load to an indicator of that CPU's capability. Here, it is mapped to the speed of the CPU, GHz, essentially indicating how much of the available processing power would be consumed by the attack. Another option would be to map to the MFLOPS rating of the processor, but this is less accessible and can be even more misleading since the attacks are not going to be optimized for efficiency of instruction usage.

ICMP Flood Attack: An ICMP Flood Attack is a denial of service attack in which one or more attacking systems send a sufficient number of ICMP echo requests to overload the target system's resources. This prevents the target system from being able to respond to any future requests, even valid ones. The values were recorded during simulation of the attack for a period of 20 minutes.

Available Network Bandwidth during Attack: 39.0 Mbps
NW Bandwidth Usage: (94.13 - 39.0) Mbps*60/1024/8= .4038 GB/min *300 =
121.14 GB/min CPU Usage: 1.26 GHz/20 = 0.063GHz/min
Memory Usage: 0.53GB /20 = 0.0265 GB/min Disk Usage: 2.5 GB/20 = 0.125 GB/min
Total Resource Usage: 121.14 GB/min + 0.063GHz/min + 0.0265 GB/min + 0.125 GB /min =
121.3545

Thus, Total Resource Usage is a representation of the total resources used by the attack. We convert all values to similar scales of units, namely giga* per minute. A side effect of this is that the network bandwidth becomes an extremely low value in comparison with the other values; this is seen below where we compute maximum availability values. To compensate we multiple the network bandwidth by a scale factor, namely 300. This could also be construed as a priority value. For instance, should network bandwidth be considered the most important resource then it could be multiplied by a greater amount.

Network Bandwidth Capacity: 94.13 Mbps *60/1024/8=.6894 GB/min *300 =
206.82 CPU: 3 GHz*60 = 180 GHz/min
Memory: 1 GB*60 = 60 GB/min
Disk: 2.5 GB*60 = 150 GB/min
Total Resource Availability: 206.82 +180+60+150 =596.82
Percentage of Total Resources Used by Attack = 121.3545/596.82 *100%= 20.3335%

One thing that becomes clear very quickly is that most of the values are extremely small, except for network bandwidth usage. The percentage of available network bandwidth being used is very high but the other values are extremely small. This is representative of ICMP Flood Attack being an older denial of service attack using many network packets that current network services handle efficiently. An alternative computational strategy would be to compute the individual usage percentages for each resource and then average them together.

This is an example of an attack in our attack corpus. The interpretation of the specific impact of this attack would be up to the individual network analyst. What is more important is when a new attack is identified and we can associate that attack with an attack already in our attack corpus. Then this would allow the network analyst to predict the impact of the new attack. For instance, we can simulate an ICMP Ping NMAP attack, which is similar to and would be mapped to the ICMP Flood Attack.

ICMP Ping NMAP: This is a network scan attack. In essence, it is an indication that nmap was used to generate a sequence of pings of the target network. This is often seen as a precursor to more direct and malevolent attacks. Since there is no direct way to measure the usage of the mentioned parameters, we have assumed certain values given below. It is also assumed that these values were calculated over a period of 20 minutes.

Available NW Bandwidth during Attack: 50.5 Mbps
NW Bandwidth Usage: (94.13-50.05) *60/1024/8 = .3229 GB/min *300 = 96.8555
CPU Usage: 0.6 GHz/20 = 0.03 GHz/min
Memory Usage: 0.5 GB /20 = 0.025 GB/min
Disk Usage: 2.5 GB/20 = 0.125 GB /min
Total Resource Usage: 96.8555 + 0.03 + 0.025 + 0.125 = 97.0355
Ratio of resources used compared to ICMP Flood Attack = 97.0355/121.3545*100% =
79.9603%

Thus, the impact of an ICMP Ping NMAP attack is 79.96% of the impact of ICMP Flood Attack. In other words, the ICMP Ping NMAP uses approximately 79% of the resources used by ICMP Flood Attack. This will essentially allow a network analyst to predict the future impact of an attack.

2.3. Generalization of the formula

The threshold values of the attacked system and network are the values before the attack and are represented as follows:

- Let the availability of the resources of the attacked system be defined as:

  AV_{CPU}: Amount of CPU available for usage in GHz

AV_{MEM}: Amount of memory available for usage in GB AV_{DSK}: Amount of disk available for usage in GB

AV_{NW}: Amount of network available for usage in Mbps.

- Let the values of the above parameters for the simulated attack for time period T1 minutes be defined as:

  S_{CPU}: Percentage CPU usage by simulated attack

  S_{MEM}: Percentage memory usage by simulated attack S_{DSK}: Percentage disk usage by simulated attack

  S_{NW}: Network bandwidth available by simulated attack (in Mbps)

- Let the Priority Factor be defined as :

  PF: Priority factor which is assigned a value 1.

- Let the Priority value for each of the parameters be defined as follows:

  $CPU_{PRIORITY\_VALUE}$ = 1 min/GHz

  $MEMORY_{PRIORITY\_VALUE}$ = 1 min/GB

  $DISK_{PRIORITY\_VALUE}$ = 1 min/GB

  $NWBANDWIDTH_{PRIORITY\_VALUE}$ = 1 min/GB

- Let the usages of the above parameters for the simulated attack in T1 minutes be defined as:

  $US_{CPU} = (AV_{CPU} * S_{CPU} * PF * CPU_{PRIORITY\_VALUE}) / T1$

  $US_{MEM} = (AV_{MEM} * S_{MEM} * PF * MEMORY_{PRIORITY\_VALUE}) / T1$

  $US_{DSK} = (AV_{DSK} * S_{DSK} * PF * DISK_{PRIORITY\_VALUE}) / T1$

  $US_{NW} = ((AV_{NW} - S_{NW}) * PF * NWBANDWIDTH_{PRIORITY\_VALUE}) / T1$

- Total resources used by the simulated attack (AS_{TOTAL}) in T1 minutes is as follows:

$$\mathbf{US_{TOTAL} = US_{CPU} + US_{MEM} + US_{DSK} + US_{NW}} \quad (1)$$

It is important to note that the specific terms for the different components will be factored out in equation 3. The goal with adding the different terms is to relate the total impact of the attack while treating these terms as equally important. This differs from Hariri et al. [19] in which each component is individually checked to determine if it is anomalous and the percent of anomalous components is determined. We felt this was insufficient since it allows attacks to go unrecognized should they limit the number of impacted components, even if those components are significantly impacted. By adding the individual components, we ensure that all impacts of an attack are represented and presented to the network analyst. The individual priority terms for each component can be used should one of the component values be too small relative to the other components.

- Let the identified parameter values for a new attack for a time period T2 minutes be defined as:

  N_{CPU}: Percentage CPU usage by new attack

  N_{MEM}: Percentage Memory usage by new attack

  N_{DSK}: Percentage Disk usage by new attack

  N_{NW}: Network bandwidth usage by new attack (in Mbps)

- Let the usages of the above parameters for the new attack in T2 minutes be defined as:

  $UN_{CPU} = (AV_{CPU} * N_{CPU} * PF * CPU_{PRIORITY\_VALUE}) / T2$

  $UN_{MEM} = (AV_{MEM} * N_{MEM} * PF * MEMORY_{PRIORITY\_VALUE}) / T2$

  $UN_{DSK} = (AV_{DSK} * N_{DSK} * PF * DISK_{PRIORITY\_VALUE}) / T2$

  $UN_{NW} = ((AV_{NW} - N_{NW}) * PF * NWBANDWIDTH_{PRIORITY\_VALUE}) / T2$

- Total resource usage by the new attack in T2 minutes is as follows:

$$\mathbf{UN_{TOTAL} = UN_{CPU} + UN_{MEM} + UN_{DSK} + UN_{NW}} \quad (2)$$

Thus, the impact severity percentage calculation is as follows:

$$\mathbf{(UN_{TOTAL} / US_{TOTAL}) * 100} \quad (3)$$

Equation 3 is particularly critical as it essentially eliminates the remaining terms leaving a value without GHz, Mb, GB, etc. Additionally, this converts the total impact into a percentage. This helps to neutralize the deviation is scale of terms from equations 1 and 2.

2.4. Assigning severity score

- The severity score is assigned a value on a scale of 0-5. There is the possibility that decimal values will be computed.

- For each of the simulated attacks, calculate the total impact severity percentage for each of the parameters.
- The impact severity percentage is the total resource used while the simulated attack was active divided by the total resources available under normal conditions. [This approach has been taken because we can witness the impact when the simulated attack is active].
- A higher impact severity percentage signifies a high severity, hence a higher severity score. A lower impact severity percentage signifies a lower severity score.
    - Under 5%: 5 (System needs a reinstall of software, needs to go offline)
    - 5% - 35% : 0 (System Behaves Normally)
    - 35% - 50% : 1
    - 50% - 75% : 2
    - 75% - 85% : 3
    - Above 85%: 4 (System behaves abnormally)

The assumption being made with these scores is that a system *reporting* minimal or no impact at all from an attack is likely compromised or already in a failed state. At the very least, the network analyst must examine the system/scenario to ensure that the results are appropriate. Thus, such systems are given a higher severity score. As with all such performance metrics, these values would be adjusted by network managers/analysts to be suitable for their own environment.

2.5. Impact analysis algorithm

The impact analysis algorithm states the procedure for detecting when a system/network is under attack along with the calculation of the impact severity percentage. It is as follows:

1. The resource availability values signify that the systems are functioning to their full potential. Make available such values for each of the identified performance metrics of the target machine.
2. Create the attack database by simulating a set of known attacks and populating the database with the amount of resources used while the system was under attack as well as with the resource availability values.
3. Attack graphs are used to detect the attacks as well as to map the attacks to various simulated attacks in the database. Once the attack is mapped to a simulated attack, calculate the resource usage using Equation 1.
4. The detection goes in a stepwise manner:
    - Check the network bandwidth for an increase in the number of packets of a particular type. An increase in a particular type of protocol can signify an attack.
    - Check the CPU usage for an increase in usage against the optimal value. A very high rate of increase signifies an attack.
    - Check the memory usage for an increase in usage. A very high rate of signifies an attack.
    - Check the disk usage for an increase in usage. A very high rate of increase signifies an attack.
5. While the attack is in progress, use Equation 2 to calculate the total resource usage.
6. Use Equation 3 to calculate the impact severity percentage.

2.6. Validation example

This section demonstrates the application of the impact assessment technique to a simulated attack. The same three-system configuration used previously was similarly used in this example with an actual implementation of a simulated attack. In this simulated attack, the attacker sent a large number of overlapped IP fragments. This particular attack maps to the Nestea attack, one of the attacks we simulated to create an initial attack corpus [4]. The Nestea attack is a Linux specific denial of service attack based on the sending of IP fragments to a target system. It relies on an “off by one IP header bug” in the Linux defragmentation code. It can cause vulnerable systems to crash. Any open applications while the attack is in progress will likely lose data. Table 1 presents the results collected over the 150-minute period during which the attack was run. All values show the increased usage of the given resources from the pre-attack state.

Table 1. Performance values during the new attack.

Time (in mins)	0	15	30	45	60	75	90	105	120	135	150
CPU Usage (%)	46.86	51.59	52.43	52.50	53.55	53.90	54.09	54.58	55.06	56.18	57.50
Memory Usage (%)	46.30	50.50	50.73	51.65	51.91	52.25	52.74	53.60	53.77	53.86	54.03
Disk Usage (%)	83.00	83.00	83.00	83.00	83.00	83.00	83.00	83.00	83.00	83.00	83.00
NW BW Usage (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The impact severity is computed below. The threshold values of the victim's system/network before the attack are:

- Network bandwidth capacity: 94.13 Mbps
- CPU : 3 GHz
- Memory : 1 GB
- Disk: 2.5 GB

Nestea attack: These values were recorded during the 150-minute period of the simulated attack.

- Available NW bandwidth during attack: 94.13 Mbps
- NW bandwidth usage: (94.13–94.13)* (1 min/Mbps) /150 = 0.0
- CPU usage : (1.725 GHz) *(1 min/GHz) /150 = 0.0115
- Memory usage : (0.5403GB) * (1 min/GB) /150 = 0.003602
- Disk usage : (2.5 GB) *(1 min/GB) /150 = 0.0167

Total resource usage: 0.0 + 0.0115 + 0.003602 + 0.0167 = 0.031802

Simulated attack: These values were calculated while the system/network was under a simulated attack for a 150-minute period.

- Available NW bandwidth during attack: 94.10 Mbps
- NW bandwidth usage: (94.13–94.13)*(1 min/Mbps) /150 = 0.0
- CPU usage : (1.705 GHz) *(1 min/GHz) /150 = 0.01134
- Memory usage : (0.5303 GB) *(1 min/GB) /150 = 0.003535
- Disk usage : (2.5 GB) *(1 min/GB) /150 = 0.0167

Total resource usage: 0.0 + 0.01134 + 0.003535 + 0.0167 = 0.031575

Impact Severity Percentage: (0.031575/0.031802) * 100 = 99.28%

Thus, the impact of the new attack is 99.28% of the impact of a Nestea attack. This means the simulated attack uses approximately 99.28% of the total resources used by a Nestea attack.

3. Vulnerability assessment

The validation example in the last section demonstrates the importance of vulnerability assessment. The simulated attack can have disastrous consequences. However, this is only the case if a targeted system is vulnerable to the attack. The goal with vulnerability assessment in the long term is to determine if a specific system is vulnerable to the current attack. This is currently not feasible, especially when we consider the case of zero day attacks. Thus, the goal is to generate an indicator identifying the approximate overall vulnerability of each system. Thus, while Linux systems should have long ago been patched against the Nestea attack specifically, we can assume that a similar bug will be identified in the future and the simulation shows the results of the impact assessment under such similar circumstances. Systems with a higher overall vulnerability indicator will more likely be susceptible to a given attack.

In order to provide an effective representation of vulnerability we began with the Common Vulnerability Scoring System (CVSS, http://www.first.org/cvss/) [32][36] which specifies specific algorithms and equations for assigning a numerical vulnerability score to a system. We then examined how such a scoring system could be integrated into a primarily automated system and how the components of the scoring system could be accurately evaluated and updated based on events.

The goal with vulnerability assessment is to identify the relative susceptibility of a system to attack. A system with a low vulnerability score is less likely to have a successful attack against it. Thus, a network analyst will be more concerned with attacks against systems deemed more vulnerable. This will be further impacted by the priority of the system itself; i.e., network analysts will be more concerned with attacks against a server than a common desktop.

3.1. Generating CVSS scores

We took a multi-pronged approach to the generation of CVSS scores. First, we realized that as CVSS is becoming a defacto standard, many of the common scanning applications are including it, such as Nessus. Given that tools such as Nessus will be constantly updated and will have extensive and robust understanding of vulnerabilities, beyond anything we could implement, it was determined that such tools would be the primary providers of CVSS scores. As a second prong, we created a tool to generate our own CVSS score from the results of nmap and netstat, using a weighted average as follows:

- openPortsFactor = 1.0;
- incomingPacketsDiscardedFactor = 0.01;
- inputICMPMessageFailedFactor = 0.01;
- icmpMessagesFailedFactor = 0.01;
- failedConnectionAttemptsFactor = 0.001;
- badSegmentsReceivedFactor = 0.01;
- packetsToUnknownPortsReceivedFactor = 0.01;
- packetsRejectedInEstablishedConnectionFactor = 0.02;
- sackRetransmitsFailedFactor = 0.005;
- connectionsAbortedFactor = 0.005;
- connectionsFactor = 1.0;

These values can easily be adjusted by network managers and analysts to best suit their specific needs depending on their specific concern with attack types and associated appearance in the network data.

3.2. CVSS score collation

Given the multiple methods of generating CVSS scores, we created a java applet to collate the scores from individual machines. This java applet has a database of applications to examine for potential CVSS scores. It then uses a weighted average of the available scores. Note that the weighted average can give all weight to one application and no weight to others. This would be advisable if Nessus and our own custom capability were available. Given the accuracy and robustness of Nessus vs. our custom capability, we would wish to only use Nessus and ignore our own results should Nessus be available. Figure 2 shows the process for collating the CVSS scores from all available applications onto a database for use by the visualization application.

More specifically, the database of applications would include the following data:

- **Location** - Location of the file on the drive.
- **Importance** - Importance of this piece of data.
- **Description** - Description of data.
- **Penalty** - Penalty given to final score if data is not present.
- **Rating scale** – Normalization value (divisor).
- **Line #** - What line the data is on.
- **Char #** - What character on that line.

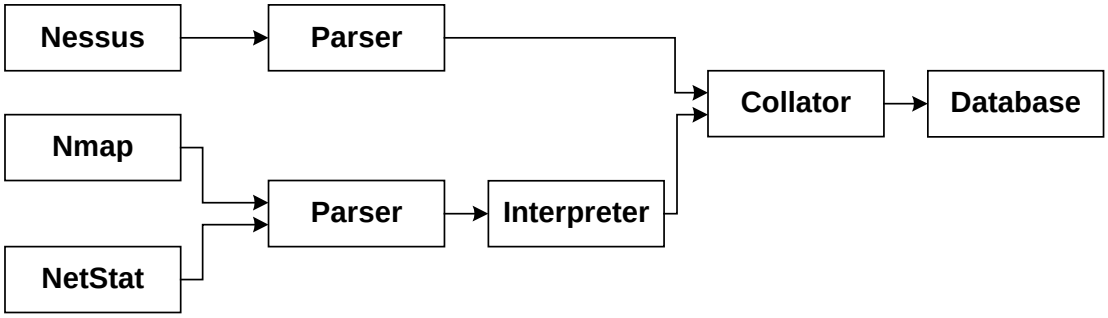

Figure 2. Our process for collating CVSS scores into a database for use by the visualization environment.

Given a set of database entries, the following process is used to generate the final CVSS score:

- Each score is acquired and normalized.
- Each score is weighted by importance.
- If an application CVSS result file is not present, zeros are added based on the specified penalty.
- A weighted average is generated.
    - Possibly include Dempster-Shafer Weighting.
- The score is displayed to the user (*Optional*).
- The score is sent to the database for use by the network analyst.

4. Relation to previous work

Ultimately, the goal with our research is to provide a more effective and usable data reduction technique for network managers. This is a necessity due to the scale of the current problem and the insufficiency of current techniques such as data reduction, automated classification, and intrusion detection systems. The data reduction domain focuses on three primary techniques, namely: event correlation, clustering, filtering, and dimensional reduction [13].

Our technique provides data reduction while simultaneously providing prioritization information for the network analyst. Existing capabilities may provide data reduction but do not provide the prioritization. The goal in this section is to provide examples of other relevant techniques for data reduction. We also identify how they relate to our technique. These techniques can be used in conjunction with our proposed technique.

This research applies to the greater domain of developing novel metrics for cyber security analysis and comparison. Such metrics are critical for the continued advancement of cyber security at large. Related work by Jiang et al. [20] evaluates the security of wireless networks. Such a metric could easily be integrated into our framework to provide impact assessment of not just hosts but networks as well.

4.1. Event correlation

Basic event correlation tools attempt to associate sequences of events with higher-level attack concepts. One of the most well known such tools is snort [22][31]. Snort actually possesses a hierarchy of attack information and when it generates an alert based on the correlation of multiple events can provide the network analyst with details on all levels of the attack hierarchy. This correlation of events into specific attacks greatly reduces the volume of information needing analysis but does not aid in the prioritization of events that is the primary goal of our research. In fact, such event correlation could be used by our techniques as an attack mapping technique. Haines et al. [18] evaluate typical event correlation techniques.

There is an additional category of interactive event correlation techniques. These techniques are typically designed around database queries [2][10] and are designed more for post mortem analysis. The interactive nature of these techniques makes them ineffective for our purposes. As with other event correlation techniques, they do not aid in the prioritization process.

4.2. Principal component analysis

Principal component analysis [26][30] is geared towards reducing the dimensionality of data. Principal component analysis identifies the variables (dimensions) which account for the variability in the data. The reduction of data parameters can help focus network analyst attention on the most important data attributes. This technique does not reduce the actual number of data elements itself. Thus, additional capabilities are needed for data reduction, though principal component analysis could be used with any other data reduction techniques. To date, such data reduction techniques have not proved effective against network data due to the variability of such data and the fact that attackers can simply vary irrelevant network packet parameters to throw off such dimensionality reduction techniques. Our work focuses on the reduction of data elements rather than the dimensionality reduction provided by principal component analysis.

Our approach is to perform data reduction by algorithmically transforming the raw data into a more abstract concept that will have greater value and meaning to the network analyst. This results in a dimensional reduction while at the same time increasing effective value of the data elements; i.e., along the lines of the quote by Aristotle “The whole is greater than the sum of its parts”. Thus, this algorithm designed specifically for cyber security will be more effective in this scenario than generalized techniques such as PCA.

4.3. Clustering

With the large volume of data available, many of the data elements and parameters will be closely associated. This must take context and past activity into account. Additionally, workstations will clearly be treated differently than servers. Thus, clustering [21] can prove enormously effective at data reduction and is greatly needed [6]. Additionally, such clustering can greatly improve the analysis process by assisting in correlation activities. Clustering can be applied according to two main metaphors. First, manual clustering allows a network analyst to directly group data elements or specify rules for the creation of clusters. Alternatively, algorithmic approaches can be applied for the automatic clustering of data elements [33]. Such clustering techniques can be applied both in conjunction with and independently of our proposed impact assessment metric. The clustering will reduce the number of individual activities a network analyst must cope with. In essence, the impact assessment could be considered a clustering algorithm itself as it groups event sequences based on their level of severity. Adhoc clustering is not effective since by grouping elements together it is essentially hiding data and can actually aid in obfuscating attacks.

4.4. Filtering

An extension to the idea of clustering is allowing the network analyst to identify elements to remove from analysis interactively, i.e., the concept of a new view of the data. Filtering can be applied through interaction, algorithmically, or other means. Through filtering, we wish to remove the elements deemed non-contributive to the problem under inspection. Fully automated filtering can require that decisions be made as to what data is relevant and what data is not. Such concepts are often used by attackers to obfuscate their activities. Collaborative filtering [16] is particularly relevant to the large-scale nature of the problems at hand. In essence, one of our goals is to filter out events deemed to have little or no malicious impact on the network automatically, allowing the network analyst to focus on truly significant events more effectively. Filtering can similarly be applied to remove events resolved through other techniques such as intrusion detection systems or deemed to be of lesser priority due to the systems or services being attacked. In other words, additional filtering techniques can be used to assist the impact assessment algorithms.

4.5. Automated classification of network events

Techniques exist for the automatic classification of network events. Our goal is not to replace these techniques but to provide additional capability on top of such techniques to aid analysts in prioritizing events that may need investigation. Where possible, the goal is to use these techniques to classify events as malicious or innocuous and to associate an event sequence with known attacks in the attack corpus such that the impact of a new attack can be predicted and thus prioritized. Thus, our capability can be applied to any classification scheme, such as:

- Attack graph analysis [3][27] – Attack graphs are most commonly used to classify attacks as early as possible based on known event steps such that analysts can identify attacks they know to be a concern to their network. A more complete use of attack graphs incorporates vulnerability information to correlate attacks with known vulnerabilities. The latter is often used a priori to identify and resolve network vulnerabilities. As attack graph analysis can associate event sequences with exact attacks, the identification of potential impact associated with the event sequence is likely to be more accurate.
- Attack Correlation [25] – Attack correlation techniques are related to attack graph techniques in that they attempt to identify attack scenarios through the analysis of event sequences. Ning et al. [25] identify attack scenarios through the identification of prerequisites and consequences.
- Neural network classification [35] – Neural networks are circuits of interconnected processing elements that learn and identify patterns in data. Complex neural networks can be applied to attempt classification of specific attacks in contrast to simply malicious versus innocuous activity.
- Statistical classification [5] – Statistical classification, a form of data mining, groups events based on their statistical similarity. Any statistical technique could essentially be applied to determine the similarity of events. Thus, the goal here is to identify the similarity of an event sequence with other malicious events or with other specific attacks.

The goal here is not to consider the accuracy of current classification techniques but simply to provide examples of the techniques that our impact assessment technique can work with. As improvements are made with the underlying classification algorithms for network security, the resulting impact assessment will receive similar improvements in accuracy. While many of these techniques will independently reduce the number of alerts the network analyst needs to deal with they do not aid in the prioritization task that is the primary goal of our research.

4.6. Intrusion detection systems

Intrusion detection systems, such as snort [22], attempt to determine if events and event sequences are malicious through comparison with a set of rules. Such systems can be effective at eliminating common attacks, especially if the rules and algorithms are carefully crafted to match the policies of the network. However, such systems are generally unable to detect unknown sophisticated attacks. Thus, the network manager must still analyze a large number of events in order to determine if any are malicious or will have a negative impact on the network. Thus, our impact assessment algorithm can work with such intrusion detection techniques to provide the next level of capability, aiding network managers in determining which remaining events should be analyzed next and which will fall below the threshold of significance of impact.

In essence, network intrusion detection systems may detect an attack or related malicious event on the network. However, such systems will identify events arbitrarily and will not aid network managers in the identification of the severity of the events; thus not aiding in the prioritization of events for remediation.

5. Future work

Our immediate plan for the future is to continue work on the automated impact assessment research as it has the greatest potential to influence the field. Other research we have identified that needs to be done includes:

- Creation of command console capabilities to allow for the long-term management of events.
- Creation of a more complete set of simulated attacks and mappings of events in the attack corpus.
- Automatically insert new attacks seen in the wild and verified as hostile into the attack corpus, assuming sufficient data associated with the new attack was observed.
- Automatically derive priority factors based on the types of services, the criticality of those services, and the frequency of use of those services.
- Automated generation of vulnerability assessment data.
- Integration of trust and reliability metrics; i.e., a reputation scoring system [34]. This will aid analysts in identifying how much credibility to associate with impact assessment scores from different sources.

The mapping of events to impact, while focusing on the cyber domain, will be designed into the environment in such a way that mappings for additional domains, once identified can easily be plugged into the environment. We have begun developing situational awareness visualization techniques that are directly focused on the representation of these impact and vulnerability assessment values. Discussion of these visualization techniques is beyond the scope of this paper.

5.1. Automated impact assessment computation

The current research focused on the manually mapping of the impact assessments to sample attack data, services, and missions. In the long term, the goal will be to fully automate the impact analysis process. We have identified the process by which this can be done, Figure 3, as follows:

1. Simulate a wide range of known and existing attacks to identify their impacts on CPU utilization, network bandwidth utilization, memory utilization, and disk utilization. Simultaneously, identify what other resources should be measured.
2. When a new attack stream is identified, the associated events will be passed through a classifier, such as a neural network or attack graph. The results of the classifier will be percentage matches against known attacks. These percentages would then be mapped to the impact/utilization matrices to identify the potential impact of this new attack. Identification of the most effective classifier and its configuration will be a research question during the development of the automated impact assessment process.

3. Remote agents [8], along the lines of CyberCraft [28][29], would be used to automatically determine what services are running on each system, what users are using these systems, and how much of each resource is typically required by that system. This information combined with the results of phase 2 will be used to identify the impact of the new attack on users, systems, and services.
4. The user could then specify priorities for the systems/services, and associate systems and services with higher level concepts such as tasks or missions. The user could also manually assign services to systems that cannot run the remote agent or for large groups of desktops that are configured identically.

The computed impact potential would be passed to appropriate databases, logs, situational awareness capabilities, and the networks analysis and managers.

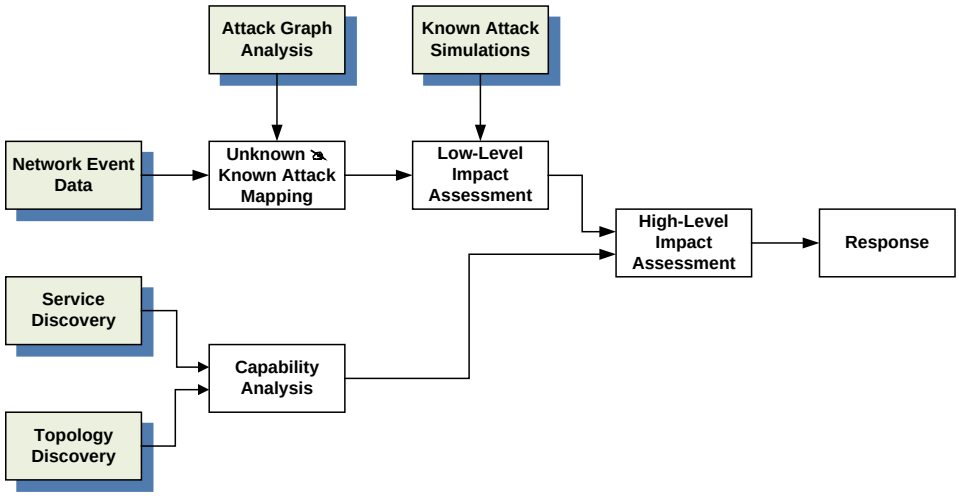

Figure 3. Proposed process for the automatic generation of impact assessment.

5.2. Determining maximum and available network bandwidth

When determining the impact of an attack on the network it is helpful to determine the actual available network bandwidth for each critical system. This would essentially examine the available bandwidth over the slowest or the most heavily used network link. The impact assessment algorithm identifies the impact of the events on network availability. However, we must determine the actual network bandwidth available over the slowest links relevant to an affected system to determine the full impact of the event. In other words, the impact assessment algorithm provides an indication of how much network bandwidth an attack will use. We propose that the impact assessment algorithm can be further enhanced with the results of algorithms for determining actual network bandwidth availability to create a more complete evaluation of the impact of an attack, such as with the Ford-Fulkerson algorithm [9][14]. Determining this bandwidth availability could be done using typical applications of the Ford-Fulkerson algorithm. The Ford-Fulkerson algorithm essentially determines the maximum flow in a network.

Using techniques such as the Ford-Fulkerson algorithm could additionally be used to identify redundancy and lack thereof in the network. More specifically, the techniques could be used to identify a set of linked systems on a network whose bandwidth can be jointly used to relay information across the networks of an organization. In addition, this techniques could help identify critical systems that if attacked will disconnect communications between the networks of an organization. These systems can be identified during the runs of this algorithm under a simulated attack and can be protected with extra measures of limited accessibility and few running processes. The available bandwidth on each system is used to determine the flow and capacity of each communication path between the associated systems on the network.

6. Conclusions

This research presents significant advances in the automatic generation of impact assessment data. Additionally, we advance the development of vulnerability assessment and show how vulnerability

assessment can be used in conjunction with the impact assessment data. This research focused on how these values can be generated. We identified manual means through which these values can currently be computed and future work that will ultimately allow these values to be computed automatically. The goal is for these techniques to be integrated into next generation situational awareness environments. This will have numerous direct benefits to network managers and network analysts, including:

- Reduce the volume of information needing analysis by network analysts and managers, improving their performance and effectiveness.
- These techniques will make situational awareness environments more accessible and usable when these values are represented as opposed to the low-level events as are typically represented in such environments.
- Provide direct representations of consequences of events and thus aiding network analysts in prioritizing events. This will directly improve the effectiveness of network analysts.
- Reduce the amount of information transmitted over the network. The impact assessment can easily be computed on distributed nodes, especially routers. Thus, only the impact assessment scores would need to be transmitted to the network analyst for a high-level network assessment. This reduction in network traffic will improve network performance overall.

Combined, these capabilities will go a long way towards improving the cyber management challenges being seen in today's network environments and will be seen in future large-scale networks. This will consequently aid in maintaining a more secure network in the long run, allowing sophisticated and substantial attacks to more readily be identified and resolved.

In addition to improving the efficiency of the network analysts, the proposed capability has additional benefits towards large-scale collaborative environments. First, the focus on impact and vulnerability assessment essentially amounts to a distributed data reduction technique. This reduces demands on the network by reducing the computation required by the visualization environment. Additionally, it puts much lower strain on any database system attempting to store the network data. This is exemplified by Fink et al. [12] in which they identified that DOE networks currently generate 500 million events daily and network analysts can only analyze 25-30% of the events they'd like to. Much of the time in the analysis process is associated with database accesses [12][17]. The proposed data reduction technique distributes the computation and reduces the volume of data, reducing the network and database impact and increasing the amount of data feasibly analyzable by network analysts. Finally, the ability to prioritize attacks more effectively and target the attacks of greater potential threat allows network analysts to provide a more secure environment while continuing to support the connections needed for large-scale connections.

7. Acknowledgements

This research was funded in part by AFRL under project #FA8750-07-C-0163 at Utah State University. This research became the primary focus of Anupama Biswas' Master's Thesis [4]. Many students played significant roles in the performance of the project, including: Anupama Biswas, Anusha Davuluri, Chris Harris, Srinidhi Kakani, Stephen Miller, Steena Montiero, Sarah Moody, Rian Shelley, and RB Whitaker.

References

[1] Eugene C. Adam, "Fighter cockpits of the future," *Proceedings of 12th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pp. 318-323, 1993.

[2] Javed Aslam, Sergey Bratus, David Kotz, Ron Peterson, Brett Tofel, Daniela Rus, "The Kerf toolkit for intrusion analysis," *IEEE Security & Privacy*, Vol. 2, No. 6, pp. 42-52, 2004.

[3] Paul Ammann, Duminda Wijesekera, Saket Kaushik, "Scalable, graph-based network vulnerability analysis," *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, pp. 217-224, 2002.

[4] Anupama Biswas, "Impact and Analysis of System and Network Attacks," Master's Thesis, Department of Computer Science, Utah State University, Defended 11/2008.

[5] Alvaro A. Cardenas, J. D. Tygar, "Statistical Classification and Computer Security," *NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security*, Poster Paper, 2007.

[6] Eugene Casey, "Forensic Network Analysis Tools," *Digital Forensics Research Workshop*, 2003,

http://www.dfrws.org/dfrws2003/presentations/Brief-Casey.

[7] Mica R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, 37(1), pp. 32-64, 1995.

[8] David Chess, Benjamin Grosof, Colin Harrison, David Levine, Colin Parris, Gene Tsudik, "Itinerant Agents for Mobile Computing", *Journal IEEE Personal Communications*, Vol. 2, No. 5, pp. 34-49, 1995.

[9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Section 26.2: The Ford-Fulkerson method," *Introduction to Algorithms* (Second ed.), MIT Press and McGraw-Hill, pp. 651–664, 2001.

[10] Steven T. Eckmann, Giovanni Vigna, Richard A. Kemmerer, "STATL: An Attack Language for State-Based Intrusion Detection," *Journal of Computer Security*, Vol. 10, No. 1, pp. 71–104, 2002. www.cs.ucsb.edu/~vigna/publications.html

[11] Robert F. Erbacher, Kenneth L. Walker, Deborah A. Frincke, "Intrusion and Misuse Detection in Large-Scale Systems," *Computer Graphics and Applications*, Vol. 22, No. 1, pp. 38-48, 2002.

[12] Glenn A. Fink, Chris L. North, Alex Endert, Stuart Rose, "Visualizing Cyber Security: Usable Workspaces," *Proceedings of the 2009 Workshop on Visualization for Cyber Security*, pp. 45-56, 2009.

[13] Imola K. Fodor, "A survey of dimension reduction techniques," *LLNL technical report*, June 2002, UCRL-ID-148494.

[14] Lester R. Ford, Delbert R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics* 8, pp. 399–404, 1956.

[15] Stefano Foresti, James Agutter, Yarden Livnat, Robert Erbacher, Shaun Moon, "Visual Correlation of Network Alerts," *Computer Graphics and Applications*, Vol. 26, No. 2, pp. 48-59, 2006.

[16] David Goldberg, David Nichols, Brian M. Oki, Douglas Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, Vol. 35, No. 12, pp. 61-70, 1992.

[17] John R. Goodall, "Visualization is Better! A Comparative Evaluation," *Proceedings of the 2009 Workshop on Visualization for Cyber Security*. October, pp. 57-68, 2009.

[18] Joshua Haines, Dorene Kewley Ryder, Laura Tinnel, Stephen Taylor, "Validation of sensor alert correlators," IEEE Security & Privacy, Vol. 1, No. 1, pp. 46-56, 2003.

[19] Salim Hariri, Guangzhi Qu, Tushneem Dharmagaddam, Modukuri Ramkishore, Cauligi Raghavendra, "Impact Analysis of Faults and Attacks in Large-Scale Networks," *IEEE Security and Privacy*, pp. 49-54, 2003.

[20] Yuan Jiang, Dongming Jiang, "The Security Assessment Method of Wireless Sensor Network with Interval Grey Linguistic Variables," *IJACT: International Journal of Advancements in Computing Technology*, Vol. 3, No. 10, pp. 85-91, 2011.

[21] Sotiris Kotsiantis, Panayiotis Pintelas, "Recent Advances in Clustering: A Brief Survey," *WSEAS Transactions on Information Science and Applications*, Vol. 1, No. 1, pp. 73-81, 2004.

[22] Jack Koziol, *Intrusion Detection with Snort*, Sams Publishing, 2003.

[23] Lundy Lewis, Gabriel Jakobson, John Buford, "Enabling Cyber Situation Awareness, Impact Assessment, and Situation Projection," *Proceedings of IEEE SIMA/MILCOM*, 2008.

[24] Robert McMillan, "Cyber attacks on US military jump sharply in 2009," PC World, 11/20/2009, http://www.pcworld.idg.com.au/article/327075.

[25] Peng Ning, Yun Cui, Douglas S. Reeves, "Constructing attack scenarios through correlation of intrusion alerts," *Proceedings of the 9th ACM conference on Computer and Communications Security*, ACM Press, pp. 245-254, 2002.

[26] Karl Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, Vol. 2, No. 6, pp. 559–572, 1901.

[27] Cynthia Philips, Laura Painton Swiler, "A Graph-Based System for Network-Vulnerability Analysis," *Proceedings of the 1998 Workshop on New Security Paradigms*, ACM Press, pp. 71-79, 1998.

[28] Paul W. Phister, Dan Fayette, Emily Krzysiak, "CyberCraft: Concept Linking NCW Principles with the Cyber Domain in an Urban Operational Environment," AF Research Lab, Presented at the DODCCRP Conference 2005.

[29] Paul W. Phister, Dan Fayette, Emily Krzysiak, "The CyberCraft Concept Linking NCW Principles with the Cyber Domain in an Urban Operational Environment," *MILITARY TECHNOLOGY*, Vol. 31, No. 9, pp. 123-131, 2007.

[30] Calyampudi Radhakrishna Rao, “The Use and Interpretation of Principal Component Analysis in Applied Research,” *Sankhya* A 26, pp. 329 -358, 1964.

[31] Martin Roesch, “Snort: Lightweight Intrusion Detection for Networks,” *Proceedings of Usenix 13th Systems Administration Conference (LISA 99)*, Usenix Association, pp. 229-238, 1999. www.usenix.org/publications/library/proceedings/lisa99/roesch.html.

[32] Peter Mell, Karen Scarfone, Sasha Romanosky, “The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems,” NIST Interagency Report 7435, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, August 2007.

[33] Pan Wang, Shun-yi Zhang, Xue-jiao Chen, “A Novel Reputation Reporting Mechanism Based on Cloud Model and Gray System Theory,” *IJACT: International Journal of Advancements in Computing Technology*, Vol. 3, No. 10, pp. 75 ~ 84, 2011.

[34] Qian Wei-ning, Zhou Ao-ying, “Analyzing Popular Clustering Algorithms from Different Viewpoints,” *Journal of Software*, Vol. 13, No. 8, pp. 1382-1394, 2002.

[35] Zheng Zhang, Constantine Manikopoulos, “Investigation of neural network classification of computer network attacks,” *Proceedings of the International Conference on Information Technology: Research and Education*, pp. 590-594, 2003.

[36] http://www.first.org/cvs