Information assurance: a cyber security storm map

Denise Ferebee*, Dipankar Dasgupta and Qishi Wu

Department of Computer Science,
The University of Memphis,
375 Dunn Hall, Memphis, TN, 38152-3240 USA
E-mail: dferebee@memphis.edu
E-mail: ddasgupt@memphis.edu
E-mail: qishiwu@memphis.edu
*Corresponding author

Abstract: Cyber-security issues affect organisations at all levels. In this article, we will discuss how to apply a visualisation and event correlation tool to facilitate the analysis of data, understanding of data, and dissemination of information to all affected parties. The visualisation shows an overall view of security events or storms that are occurring on a network while providing information in reference to severity and a propagation pattern. The tool can potentially provide an early warning so that events or storms can be proactively mitigated. Thus, organisations can make business decisions by determining or understanding the relationship between the computing devices and the business/information technology services they make-up.

Keywords: cyber-security; information visualisation; levels of abstraction; event correlation; security event.


Biographical notes: Denise Ferebee received her PhD in Computer Science from the University of Memphis in 2012. She has published research articles in highly reputed conference proceedings. Her main research interests include cyber security and information and scientific visualisation.

Dipankar Dasgupta is a Professor of Computer Science and the Founding Director of the Center for Information Assurance at the University of Memphis. His research interests are tracking real-world problems through interdisciplinary cooperation with special interests in artificial immune systems, genetic algorithms, neural networks, multi-agent systems and their applications. One of his current areas of research is in applying biologically-inspired computing to cyber security. He has published more than 200 research papers in book chapters, journals and international conferences. He received three best paper awards in international conferences, and in editorial board of five journals.
Qishi Wu received his BS in Remote Sensing from Zhejiang University, China in 1995, his MS in Geomatics from Purdue University in 2000, and his PhD in Computer Science from Louisiana State University in 2003. He was a Research Fellow in the Computer Science and Mathematics Division at Oak Ridge National Laboratory during 2003–2006. He is currently an Associate Professor with the Department of Computer Science at the University of Memphis. His research interests include parallel and distributed computing, computer networks, sensor networks and cyber security.

1 Introduction

On a daily basis, we are notified of current cyber-security events. These types of events consist of the following: mobile threats, malware, crimeware, hacktivism, etc. These events make national and international news. CNN on March 12, 2013 featured an online article about international cyber-attacks being on the rise (Abdullah, 2013).

Organisations across the global landscape are constantly trying to gain more tools in their cyber-security plans to help thwart cyber threats. This requires individual contributors to analyse large volumes of security data in order to make informed decisions. These individual contributors, as shown in Figure 1, collaborate and share information to make decisions. However, it is a daunting task without a common understanding. Therefore, the security data that a subject matter expert (SME) reviews may be too detailed (i.e., it may need to be put into a broader scope). Information technology (IT) environments are changing. They now include cloud computing, extensive data management needs (i.e., the ‘big data’ issue), etc. Thus, there is a need for a more innovative solution. Therefore, this paper will focus on providing a cyber-security tool that analyses time-series, geospatially related data and produces visualisation.

The tool that we will use will take a page from meteorology. Meteorology and cyber-security share major aspects such as: gathering large volume of data and making it easy for users to discern. Therefore, the purpose of this research is to show how security visualisations (i.e., a cyber-security storm map) can be used to bridge the gap of information understanding/consumption and perform impact analysis from an IT and a business aspect.

This paper is separated into seven parts. Section 2 covers related topics that discuss some of the current cyber-security threats and information visualisation techniques used as a foundation for a cyber-security storm map. Section 3 covers the applied methodology for approaching cyber-security visualisation. Section 4 covers the implementation details of the methodology covered in Section 3. Section 5 covers the experiments and results performed. Section 6 discusses the conclusions reached from the experiments and results covered in Section 5. Section 7 covers future work and other potential applications of the methodology and techniques used.
2 Related topics

In order to provide a cyber-security storm map that provides information for the average layperson, we have to establish an understanding of the current cyber-security threats that organisations are facing. Next, gain an understanding of the security mechanisms used in the IT environment to establish a sense of assurance. Finally, gain an understanding of the process that the SMEs go through to correlate security mechanism data in order provide it to needed individuals and organisations such as the organisations in Figure 1 (Cockton, 2008; McCarthy and Wright, 2007; Wania et al., 2006). Therefore, this section will focus on these areas in order to present a basis for a cyber-security storm map.
2.1 Cyber threats

Cyber threats stem from e-mails to online searches where people can purchase information about you, your family and friends. McAfee predicts ransomeware kits for mobile devices to increase. Rootkits are predicted to change in order to use master boot record (MBR) to thwart current cyber-security protection techniques. Use of Trojans such as Citadel is on the rise (Abdullah, 2013; Dasgupta and Ferebee, 2013; McAfee, 2013). Many security mechanisms protect the targeted environments of the fore mentioned threats (e.g., firewalls, application/business methods, policies, etc.) (Bishop, 2004).

These mechanisms produce a large volume of monitoring data used for logging and alerting; thus, requiring some form of maintenance by an administrator, developer or security engineer. Even though, there are tools that are able to analyse mechanism specific data (i.e., switch, server, firewall, mobile device, etc.); the correlation of this data is difficult and time consuming.

2.2 Information visualisation

Visual information processing consists of three stages listed as follows:
1. extracting low-level properties of the visual scene by parallel processing
2. pattern perception
3. sequential goal-directed processing (Ware, 2004).

In order to take advantage of known representations, we are going to base our visualisation on a weather map from meteorology and call it a cyber-security storm map. It will make the information easy to understand for all individuals and organisations involved (i.e., the average layperson).

A security storm map is constructed by answering the following questions and making use of the user’s visual perception (Ferebee et al., 2012):

- What tasks the user is going through to conclude the severity of a security breach?
- What are the gaps in the current methods?
- What are the user roles?
- What purpose/value does the information serve in order to mitigate risk?

Therefore, we will show how security visualisations (i.e., a cyber-security storm map) can be used to bridge the gap of information understanding/consumption and perform impact analysis from an IT and a business aspect.

3 Methodology

This section covers the methodology used to create the security storm map as shown in Figure 2. In order to create an automated means of stepping through the process of finding a pattern in events that are occurring, we will utilise the fundamentals of visualisation and event correlation. The patterns formed will help in analysis and decision making in order to mitigate the risk associated with the security event.
3.1 Holistic view of security

The section will focus on creating a security analysis application by using sense-making, visualisations, event correlation, and domain knowledge in order to provide cyber-security events in a user-friendly manner.

The first step on this path is to understand the event visualisation-analysis process. It consists of the following four stages (Ware, 2004):

- collection and storage of data
- preprocessing to transform the data into something understandable to the end-user
- display hardware and the graphics algorithms that produce an image on the screen
- human perceptual and cognitive system (i.e., perceiver).

Next, there are three phases to create the cyber-security tool needed which are as follows:

- Phase 1 consisted of security data collected by mechanisms and tools such as firewalls, application rules, operating system monitoring, etc. This is emulated by creating a security event generator that selects security vulnerabilities with the premise that they have been exploited (Ferebee et al., 2012).
- Phase 2 consisted of transforming the data from phase 1 via the correlation engine. The data is summarised; metadata links are created, correlates, and classified in order to make it understandable (Ferebee et al., 2012).
- Phase 3 consisted of taking the data from phase 2 and presenting it to the end-user for processing in order to perceive the significance of the event presented (Ferebee et al., 2012). It is necessary to understand what the end-user goes through to find meaning in the data (i.e., to synthesise a sense-making process) (Dervin, 1998). This is essential to accomplishing phase 3.

Figure 2 Security visualisation methodology (see online version for colours)

The basis for the correlation engine is steeped in the sense-making process (i.e., data foraging) as shown in Figure 3 (Dervin, 1998). For the purposes this research, the correlation engine will forage through the data and determine a pattern. Next, a meaning
is determined and the significance of the data is established. The information is presented visually for analysis of the problem or situation. After the analysis has been completed, an action or plan is made.

**Figure 3** Sense-making of a security event (see online version for colours)

![Diagram](image)

Such visualisation needs to take advantage of the human vision system and visual analysis process to represent a large amount of data in a meaningful way. This cuts down on the analysis and mitigation time.

Using the fore mentioned concepts, how do we establish a holistic visualisation? This is accomplished by putting security events in the context of an organisation’s services and locations; a relationship between devices and application services is defined. Information overload is reduced by abstracting network components by service and then by location. A drill-down from geographical location shows each business or IT service that is available and any potential virus or security event propagation. The time laps feature allows one view to view the recovery rate as the patches or antivirus mitigations has started. Because of the defined metadata, emerging relationships that typically go unnoticed because the data cannot be observed as a whole or because of sheer volume is available.

There will be three views of the data provided:

1. the storm map view
2. the services view
3. the device view.

First is the storm map view. The storm map view shows the geographically dispersed data centre locations and the security vulnerabilities that are occurring; thus, providing a similarity between the cyber-security weather/storm map and a meteorological weather
map. To highlight the propagation of security threats/storms, icons will be used. As in a meteorological weather map like Doppler radar, icons are used to highlight high and low wave fronts (with different colours), hurricanes and snowstorms making them visible and intuitive. From each geographic location, the user can drill down and see the business and IT services that are available at that location (i.e., the service view).

From the device view, the user is able to see devices that make-up the services and the events that are occurring. Thus, providing the SME with the ability to see which portions of the network are plagued by events and determine impact based on business need.

3.2 Event correlation

The event correlation module consists of the following components: a time-series data collector, a correlation engine to correlate the detected security events, a RMT-based correlation network constructor, a known event database, and event identifier (Ferebee et al., 2012). After the events are correlated, the correlation networks are de-noised and compared to the known events database. If there was no similarity, the correlation network was added to the database and then provided to the user as visualisation (Wu et al., 2009a, 2009b, 2009c).

We worked under the premise of abstracting devices up to a business and IT service offering (i.e., many devices make-up a service). As well, many companies are geographically dispersed and therefore, their services reside globally and these issues should be taken into account that service locations and service offerings can affect other locations and offerings.

First, the current storm \( S_i \) is compared to a known storm \( S_k \) by comparing correlation networks \( C_1 \) from the matrix set representation of each storm \( C_{ak} \). We calculate the similarity \( m \) based on the following measurement:

**Correlation network similarity**

\[
m = w_S \cdot \sum_{e \in E_S} \left( 1 - |\rho(e^s) - \rho(e^k) | \right) \\
+ w_{BR} \left( \frac{|E_{BR}|}{|E^s|} \sum_{e \in E_{BR}^s} \rho(e) + \frac{|E_{BR}|}{|E^k|} \sum_{e \in E_{BR}^k} \rho(e) \right) \\
- w_{NI} \left( \frac{|E_{NI}|}{|E^s|} \sum_{e \in E_{NI}^s} \rho(e) + \frac{|E_{NI}|}{|E^k|} \sum_{e \in E_{NI}^k} \rho(e) \right),
\]

where \( w_S, w_{BR}, \) and \( w_{NI} \) are weighted coefficients for three subsets of edges and \( |E| \) represents the number of edges in \( E \) (Wu et al., 2009a, 2009b, 2009c).

We have chosen to use the National Vulnerability Database (NVD, 2012; Common Vulnerabilities and Exposures, 2010) and the Common Vulnerability Scoring System (CVSS, 2010) because security device, software, and network device manufactures have standardised their vulnerability references in this manner. The data has been expanded from a matrix \( M \) that only consisted of event and time interval data to a set of matrices \( L_n \) where \( n \) is the number of business/organisation locations. We reference each location \( L_n \) geospatially (i.e., by latitude and longitude). Each row in \( L_n \) is a CVSS vulnerability \( V_i \) where \( I \) is the number of different types of vulnerabilities that occurred at a location \( L_n \).
Each column in $L_n$ is a time stamp $T_j$ where $J$ is the number of time intervals (Ferebee et al., 2012).

We used a set of matrices $D_p$ where $P$ is the number of matrices in the set which will be two. $D_1$ is the matrix where each row is CVSS vulnerability $V_i$ and each column is a time interval $T_j$. $D_2$ is the matrix where each row is a CVSS vulnerability $V_i$ and each column is the number of occurrences at a location $L_n$. The purpose of $D_2$ is to be the basis of a correlation between vulnerabilities and locations (i.e., vulnerabilities occurring at one location can affect other locations and we are using this to capture that information) (Ferebee et al., 2012).

Each set of matrices $D_p$ will be used to create two correlation networks $C_n$. The set of networks $C_n$ represents a security event/storm. The correlation matrix $C_1$ characterises the storm at a location $L_n$. The correlation matrix $C_2$ will only be used to determine what other locations might be involved or affecting the current security storm. Each correlation network will be compared to the database of known storms in order to find a match.

4 Implementation details

In this section, we will review an application of the prototype. The five major components of the cyber-storm map system are a security event generator, security event database, data retrieval services, an event correlation and classification engine, and an end-user visualisation application. These components are outlined in this section.

4.1 Event data generator

Once, the NVD’s Common Platform Enumeration (CPE) Dictionary was chosen as a basis (Common Vulnerabilities and Exposures, 2010), definitions of various hardware, software, and operating system profiles were created without having to implement each component as shown in Figure 5. The event data generator was developed to create the security events based on the profiles provided. Because of the compactness of the application, a laptop was chosen for the hosting platform. The laptop hardware consisted of a Toshiba Satellite X205 with 4 GB of RAM, two internal 250 GB hard drives, an Intel® Core™2 Duo CPU T5450 (Ferebee et al., 2012).

The Java application created multiple events simultaneously during a time interval and recorded the event data in a database for logging purposes and to be used by other system components. This was accomplished by designing it as multi-threaded application that could write to a database where the threads were created at specified time window, thus requiring a scheduling component (i.e., Java’s Executors thread pool and scheduling components). This provided a means of having an interval timer and multiple event selections (Ferebee et al., 2012).

Next, a MySQL (2012) database was integrated into the system and a database connection pooling was provided for performance. Therefore, the example provided by Oracle for JDBC was followed (MySQL, 2012; Oracle, 2012). It provided a way to cache database connections to be used by another component and cut down on connection creation time; thus, completing the components needed for the class definitions for the data generator (Ferebee et al., 2012).
4.2 Cyber-security storm map

The cyber-security storm map visualisation was designed in Adobe Flex. It provides a richer and rapid programming environment in order to create a good user interface. Hence, the application is hosted on Toshiba Satellite X205 with 4 GB of RAM, two internal 250 GB hard drives, an Intel® Core™2 Duo CPU T5450 (Ferebee et al., 2012).

The UI is made of three basic components consisting of a navigation component, application simulation/security warnings component, and a visualisation component as shown in Figure 4. Also, it has a preferences component to show the environment profiles as shown in Figure 5.

Figure 4  Cyber-storm map – geographical view (see online version for colours)

Figure 5  Cyber-storm map – environment profile preferences (see online version for colours)
The collapsible preferences component on the left contains the weather map, a network layout, and historical date preferences tabs as shown Figure 4. These tabs allow the user to customise their visualisation component. Next, the user can pan/zoom either a weather map or a services layout map when drilling down into a locations network design in order to see detail data about security events from the visualisation component in the middle. Finally, the show simulation/warnings component allows the user to see the types of warnings that are occurring (Ferebee et al., 2012).

**Figure 6** Security events – location correlation of relationship diagram (see online version for colours)

The security events will be correlated on a location by location basis and super imposed at the geographic level onto a map creating a security storm map as shown in Figure 6. Questions that can be answered based on the data at this level are as follows:
Are the current security events related by geographical location?

What is the max severity of the events linking a location?

What are the percentages of the severities of the events linking the locations?

(Ferebee et al., 2012)

The services view drill down shows security events that are correlated on an event-by-event basis. By drawing lines between services that have an event-by-event relationship (i.e., there is a relationship between event A and event B), we are able to impose a location level, then a services level, and finally a node level event relationship where we are able to answer the following questions:

1. Are the current security events related by service?
2. What is the max severity of the linked events of the services?
3. What are the percentages of the severities of the linked events of the services?

4.3 Even correlation and classification module

The correlation and classifier module consists of the following components: a data loader, a correlation engine, an event level classifier, and a location level classifier. The raw event data type ceDataLoader is used for summarising and creating the metadata. The database contains data that is summarised over an interval window by location and as a global data (i.e., all events by type per location and is used to do the location-by-location correlation).

The corrEngine data type is used to correlate the summarised matrix of events-by-time-interval. The loadMatrix function performs a MySQL (2012) query where the events are aggregated by type over an interval range. Each interval range is a sliding window meaning that if there are 24 intervals created in a simulation and the interval window is size 15; all matrices are created using a sliding window until we reach the max interval. Each matrix is correlated before de-noising, and the eigenvalues are calculated using the CULA LAPACK libraries and the data is stored in the database (CULA Tools, 2012; NVIDIA, 2012).

The ceClassifier data type handles the de-noising and the classification of matrices and following the methodology sections, the nearest neighbour distribution is calculated. Next, it is used to determine a cut-off value in order to de-noise the matrix. The de-noise matrix function uses the Boost C++ distribution functions to perform the Chi squares test verification (Ferebee et al., 2012; Boost C++ Libraries, 2012).

The ceClassifier data type using the similarity equations defined in its similarity function finds the similar matrices/storms that match the current correlated storm (Ferebee et al., 2012; Wu et al., 2009a). The default value of the weight coefficient for the shared interval edges is one. The weight coefficient for the bridging edges is 0.5. The weight coefficient for the non-shared internal edges is set to the value of 0.5. If there are no similar matrices/storms the current storm matrix that is being evaluated is added as a new storm to the classified storms table (Ferebee et al., 2012).
4.4 Database and data model

The database is hosted on a VMware virtual machine running Ubuntu 10.x. The VM has 1 GB of RAM and one CPU. The database software is MySQL 5.x. It is setup for remote connections on port 3306. Because of the connection creation issues in Java and C++ the connection, timeout was set to 15 seconds instead of the default of 5 seconds Ferebee et al., 2012.

By parsing the vulnerabilities available from the NVD and the hardware profiles available from CPE (National Vulnerability Database, 2012), a data model was created for the cyber storm map system. Perl was used to parse the XML formatted profiles and insert the data into the database. The database consisted of five table categories:

1. NVD profiles and metadata (i.e., all of the tables dealing with the vulnerability values, hardware profiles, and their metadata)
2. geospatial environment data (i.e., the geographic location, services, and device specific information)
3. location specific unclassified storm data (i.e., the entire event correlation prior storm classification)
4. location specific classified storm data (i.e., the known storms and their metadata)
5. global geospatial storm data (all of the location-to-location correlation of events) (Ferebee et al., 2012).

4.5 Data retrieval services

Adobe Flex requires a HTTP connection to retrieve data. Therefore, PHP was used to create a database connection to the MySQL database, perform a query, and output the query results in XML. The XML is then parsed by the Flex application (Ferebee et al., 2012). These parsed XML files provide the data to the end-user visualisation in order to show the correlated relationships.

5 Experiments and results

This section describes the test setup, and output from test runs used to examine the implementation details mentioned in Section 4 in reference to the visualisation and event correlation.

5.1 Experiments

The validation experiments used fall into three categories:

1. hardware related
2. software related
3. mixed hardware and software related.

A picked vulnerability is assumed or classifies it as being exploited.
In the hardware directed experiments, the purpose is to select vulnerabilities that only reference hardware. Therefore, hardware events are randomly selected. The hardware experiments will have parameter settings as shown in Table 1.

In the software directed experiments, the purpose is to select vulnerabilities that only reference software. Therefore, software events are randomly selected. The software experiments will have parameters settings as shown in Table 2.

**Table 1** Hardware event directed experiment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of locations</td>
<td>11</td>
</tr>
<tr>
<td>Number of services</td>
<td>2–3 services per geographic allocation</td>
</tr>
<tr>
<td>Devices</td>
<td></td>
</tr>
<tr>
<td>Installed software</td>
<td>1–3 software titles per device</td>
</tr>
<tr>
<td>Time interval size</td>
<td>300 seconds or 5 minutes</td>
</tr>
<tr>
<td>Number of time intervals</td>
<td>31</td>
</tr>
<tr>
<td>Security events</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
</tr>
</tbody>
</table>

*Source: Ferebee et al. (2012)*

**Table 2** Software event directed experiment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of locations</td>
<td>11</td>
</tr>
<tr>
<td>Number of services</td>
<td>2–3 services per geographic allocation</td>
</tr>
<tr>
<td>Devices</td>
<td></td>
</tr>
<tr>
<td>Installed software</td>
<td>1–3 software titles per device</td>
</tr>
<tr>
<td>Time interval size</td>
<td>300 seconds or 5 minutes</td>
</tr>
<tr>
<td>Number of time intervals</td>
<td>32</td>
</tr>
<tr>
<td>Security events</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
</tr>
</tbody>
</table>

*Source: Ferebee et al. (2012)*

**Table 3** Software and hardware combined events directed experiment

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of locations</td>
<td>11</td>
</tr>
<tr>
<td>Number of services</td>
<td>2–3 services per geographic allocation</td>
</tr>
<tr>
<td>Devices</td>
<td></td>
</tr>
<tr>
<td>Installed software</td>
<td>1–3 software titles per device</td>
</tr>
<tr>
<td>Time interval size</td>
<td>300 seconds or 5 minutes</td>
</tr>
<tr>
<td>Number of time intervals</td>
<td>20</td>
</tr>
<tr>
<td>Security events</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
</tr>
</tbody>
</table>
**Table 4** Base metric percentages for Memphis, TN – hardware directed example

<table>
<thead>
<tr>
<th>Count between (0, 3.9)</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count between (4, 6.9)</td>
<td>6</td>
</tr>
<tr>
<td>Count between (7, 10)</td>
<td>3</td>
</tr>
<tr>
<td>Total Vul</td>
<td>12</td>
</tr>
<tr>
<td>percent_low</td>
<td>0.166667</td>
</tr>
<tr>
<td>percent_medium</td>
<td>0.5</td>
</tr>
<tr>
<td>percent_high</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Source: Ferebee et al. (2012)

**Table 5** De-noised matrix for Memphis, TN – hardware directed example

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2001-0861</td>
<td>1</td>
<td>0</td>
<td>0.066226617</td>
<td>0.106198847</td>
<td>0</td>
</tr>
<tr>
<td>CVE-2005-2186</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.066226617</td>
<td>0</td>
</tr>
<tr>
<td>CVE-2005-2552</td>
<td>0</td>
<td>0.066226617</td>
<td>1</td>
<td>0.106198847</td>
<td>0</td>
</tr>
<tr>
<td>CVE-2005-2988</td>
<td>0.222456098</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CVE-2006-0259</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CVE-2007-0963</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Source: Ferebee et al. (2012)

**Table 6** Correlated event base metric scores for hardware directed example

<table>
<thead>
<tr>
<th>Pearson’s correlation</th>
<th>Source event</th>
<th>Source event base metric score</th>
<th>Destination event</th>
<th>Destination event base metric score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CVE-2001-0861</td>
<td>5</td>
<td>CVE-2001-0861</td>
<td>5</td>
</tr>
<tr>
<td>0.222456098</td>
<td>CVE-2001-0861</td>
<td>5</td>
<td>CVE-2005-2988</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>CVE-2005-2186</td>
<td>1.9</td>
<td>CVE-2005-2186</td>
<td>1.9</td>
</tr>
<tr>
<td>0.066226617</td>
<td>CVE-2005-2186</td>
<td>1.9</td>
<td>CVE-2005-2552</td>
<td>7.5</td>
</tr>
<tr>
<td>0.066226617</td>
<td>CVE-2005-2552</td>
<td>7.5</td>
<td>CVE-2005-2186</td>
<td>1.9</td>
</tr>
<tr>
<td>1</td>
<td>CVE-2005-2552</td>
<td>7.5</td>
<td>CVE-2005-2552</td>
<td>7.5</td>
</tr>
<tr>
<td>0.106198847</td>
<td>CVE-2005-2552</td>
<td>7.5</td>
<td>CVE-2005-2988</td>
<td>5</td>
</tr>
<tr>
<td>0.222456098</td>
<td>CVE-2005-2988</td>
<td>5</td>
<td>CVE-2001-0861</td>
<td>5</td>
</tr>
<tr>
<td>0.106198847</td>
<td>CVE-2005-2988</td>
<td>5</td>
<td>CVE-2005-2552</td>
<td>7.5</td>
</tr>
<tr>
<td>1</td>
<td>CVE-2005-2988</td>
<td>5</td>
<td>CVE-2005-2988</td>
<td>5</td>
</tr>
</tbody>
</table>

Source: Ferebee et al. (2012)
In the software and hardware combined directed experiments, the purpose is to select any vulnerabilities. Therefore, any software or hardware events are randomly selected. The software and hardware combined experiments will have parameter settings as shown in Table 3.

This example starts from a specific geospatial location and covers the event correlation, de-noising/filtering, and classification processes. The data values for the de-noised matrix for the Memphis, TN are shown in Table 5. Table 4 shows the values from the percentage of low, medium, and high base metrics for this location from the source event value. The base metric scores for the source events for the Memphis, TN example are shown in Table 6. Each of these tables is used in the visualisation to provide correlated events and severity.

5.2 Output

There are three types of tests performed. The software and hardware directed test provides events that have not hardware or software specific. The hardware specific test was to select events based on the hardware available in the enterprise which consisted of but not limited to the following: servers, firewalls, switches, etc. The software specific test was to provide events that are significant only to the software installed in the environment which consisted of but not limited to the following: web server software, various operating systems, database software, etc. Between the three types of tests, there were 214 local storms created 26 global storms (i.e., these are a correlation of events location-by-location).

During the de-noising process, a cut-off value was incremented from 0.01 to 0.9 at every 0.01 interval and produced better cut-off values. Through the event correlation and classification process, the security events were summarised by event type per time interval per location. Next, we proceeded to the correlation, classification, and similarity steps. The effort progressed to the similarity step. However, it showed signs of inconsistency (i.e., no similarities were found) thus, leaving an opportunity for additional tuning in future approaches to this technique. We believe that selecting the similarity weight values based on an organisation’s view of importance will provide a better way of finding similar networks.

5.3 Visualisations

A walkthrough of the experiments will be covered in this section. The data used is based on the information presented in Tables 4 to 6. Figure 7 shows a geo-spatial map that allows for the users to see which enterprise locations are active (i.e., pictured by white map markers) and which enterprise locations are inactive (i.e., pictured by grey map markers).

Figure 7 shows hardware directed security storms and shows no globally correlated security events at the enterprise network level (i.e., lines showing correlations between locations). However, storm clouds with orange lighting are visible over geographical areas where security events are occurring where the lightning signifies that the highest percentage of correlations that are occurring have a CVSS base score of between 4 and 6.9 for the source as shown in Figure 7. Essentially, 16% of the correlations had a low base metric score, 58% had a medium base metric score, and 25% of the correlations had
a high base metric source event score as shown in Table 4 which is based on the correlated source and destination events found in Table 6.

**Figure 7** Security storm map with storms (see online version for colours)

![Security storm map with storms](image)

*Source:* Ferebee et al. (2012)

**Figure 8** Security event at service level view of the hardware directed experiments (see online version for colours)

![Security event at service level view](image)

*Source:* Ferebee et al. (2012)

By double clicking on the location marker of a geographical location, the end-user can drill-down to detailed data; thus, taking them to a service level diagram that shows all of the connected services for a location as shown in Figure 8. Shown are the enterprise services that are available at the Memphis, TN data centre (i.e., e-mail and web application services). A red arrow signifying that there is a large percentage of correlated events that have occurred where the source events have a high base metric range (i.e., between 7.0 to 10) between device pairs that make-up this service as shown in Figure 8.
Figure 9, the device detail view of the current security storm, shows the source event and destination event correlation pairs that occur between devices. The lines are coloured as follows: a yellow line signifies a low source event base metric score, an orange line signifies medium source event base metric score, and a red line signifies a high source base metric score. If there is a source event and destination event between a set of devices, a line is drawn between them and it is coloured based on the source event base metric score of the highest percentage of correlated event types (i.e., that have a low, a medium, or a high base metric source score). If the highest percentage of source events has a low base metric score, then the line will be coloured yellow. If medium, it will be coloured orange. If high, it will be coloured red.

Figure 9  Security events at the device view of the hardware directed experiments (see online version for colours)

Source: Ferebee et al. (2012)

Figure 10  Service level view of the software directed experiments (see online version for colours)

Source: Ferebee et al. (2012)
During the software directed experiment, security events appear on the visualisation as shown in Figure 10. Pictured are security events that are affecting multiple services (i.e., e-mail and web application). The source security event base metric score is greater than 4.0 and less than 6.9.

**Figure 11** Device view of security events in the software directed experiment (see online version for colours)

![Figure 11](image)

Source: Ferebee et al. (2012)

After double-clicking on the e-mail services, the device-to-device security event relationships are visible as shown in Figure 11. There are directed lines between each of the nodes where security event pairs are common. The orange line signifies the base metric range of the source event of correlated security event pair.

6 Conclusions

The event correlation component is essential in implementing the holistic visualisation consisting of a programmatic implementation of the sense-making process. It provides the metadata relationships (i.e., discovered patterns) that are needed to produce the visualisation. Thus, leading to the implementation details of these components and how they are used together to produce a security visualisation to reach multiple audiences.

The security storm map fulfilled holistic view of security need by leveraging prior end-user knowledge of weather maps. Security events are classified as storms and show the relationships between geographical areas. More meaningful context of the security data was achieved by showing which components (i.e., locations, services, and devices) have related security events.

7 Future work

The proposed methods for event correlation and security visualisation open many opportunities for additional research. It can be expanded to provide real-time event
Next, the methodology and event correlation techniques can be married with massively multiplayer online role-playing game (MMORPG) to simulate cyber-security battle fronts to how simultaneous security events affect each other. Similarly, a national centre for information dissemination and citizen awareness would find the techniques used in this research value because of the ability to provide consumable information for multiple audiences.

References


