ThreatVectors: Contextual workflows and visualizations for rapid cyber event triage

Scott Miserendino*
BluVector, Inc.

Corey Maynard
BluVector, Inc.

Jacob Davis
Net Fusion Marketing Corporation

ABSTRACT

Cyber security operations face a daily flood of security events generated by automated security tools and analytics. These events must be rapidly and accurately triaged to remove false positives and focus investigations on those presenting the greatest risks to the enterprise and requiring immediate remediation. We introduce ThreatVectors as a contextual triage workflow and event visualization tool to aid operators in event triage. ThreatVectors use a streaming event processing framework for event correlation, aggregation and prioritization based on user definable event collections and a cyber-triage domain specific language. Triage work progress is shown using a novel progress bar matrix. Event collection visualization includes abstract event thumbnails for event overview and a dynamic filtering mechanism based on metafield hierarchies. Bulk adjudication of filtered event views and event clusters is supported. User testing on large enterprise networks indicates the approach has significant potential for aiding in identifying multievent campaigns, supporting collaborative triage and reducing total time spent triaging events.

Keywords: Triage, workflow, cyber event correlation, domain specific language, thumbnails.

1 INTRODUCTION

Cyber security analysts often work in a collaborative security environment as members of a cybersecurity operating center (CSOC) or Computer Incident Response Team (CIRT). As members of a CSOC or CIRT, analysts review and process tens to millions of different types of network events daily [1]. Due to the overwhelming number of events, analysts make educated decisions regarding whether and in what order events receive further investigation, a process known as triage. Events often go through multiple stages of review and investigation by one or more tiers of analysts before an incident is declared. CSOCs leverage a variety of automated systems including intrusion detection systems (IDS), host-based security agents, centralized security information and event management systems (SIEMs), incident tracking systems, cyber intelligence feeds and dynamic analysis engines (i.e., sandboxes) to create and curate metadata about each event. Analysts primarily rely on this automatically generated metadata when making triage decisions.

Every CSOC team uses unique criteria within its event triage workflow; yet, most workflows consist of several common steps. Analysts work through a three stage cognitive data fusion process of detection, situational awareness and threat assessment [2]. In the detection phase, events or alerts are filtered to identify those that should be further investigated. During the situational awareness phase, eligible events are prioritized and investigated including conducting additional analysis on extracted content and performing metadata correlations. In the final phase, determinations of maliciousness are made and remediation plans executed along with further association of the event to known threat actors and campaigns. Throughout these steps analysts seek to improve their understanding of network context associated with an event. Network context includes information regarding to and from IP addresses or hostnames, data transfer and application layer protocols used and application specific information (e.g., session headers for web traffic or to/from/subject for email). The network context heavily influences the final determination of maliciousness as well as informs remediation actions such as which domains to block or which hosts to reimage.

Security practitioners have found that the single most important operations metric is time to discover/contain/remediate an incident [3]. Timeliness and thoroughness of event triage are critical drivers of time to discovery. Since triage is often performed by collaborative and at times geographically distributed teams, CSOC managers are often challenged in understanding how much triage work must be done, how much is left to complete and whether outstanding work is being dominated by new or old events. Managers require methods for codifying an approach to event filtering, aggregation, correlation and prioritization that is both responsive to changes in the threat environment and can be easily shared among distributed analysts.

In this paper, we present a novel cyber event visualization and triage system called ThreatVectors. ThreatVectors consist of a streaming, user-definable event filtering and correlation module; domain-specific language (DSL) for contextual event prioritization; overview visualization of triage work status; configurable metadata pivot tool; and multistage event drill-down process including an abstract visualization called an event thumbnail to assist in rapid identification and bulk adjudication. Following a discussion of related previous works, ThreatVectors approach and components are detailed in Section 3. Section 4 contains a discussion of our web application-based implementation. In Section 5, we report system performance testing and user feedback from deployments within several large enterprise networks. Finally, we offer concluding remarks and planned future work.

2 PREVIOUS WORK

Previous work in the area of event triage has focused primarily on methods and algorithms for event and alert aggregation, correlation and prioritization. There also exists a significant body of literature addressing visualization of events and network flows to achieve cyber situational awareness but most efforts focused on visualizations for identifying network breaches rather than supporting analyst triage workflows and decisions. Several examples of domain specific languages for computer networks and cyber security exist but none dedicated to event triage.

*scott.miserendino@bluvectorcyber.com

2.1 Event aggregation, correlation and prioritization

Alert and event aggregation and correlation algorithms and systems are used to handle the large quantities of events generated by automated cybersecurity systems, heterogeneous events across multiple detection platforms, false alarm reduction, event association over multiple stages of attack, and lack of or inaccurate prioritization of events [4]. Several frameworks and technologies have been proposed to address aggregation, correlation and prioritization requirements. Mirheidari et al. [4], Spathoulas et al. [5] and Zuech et al. [6] have published survey papers covering the breadth of solutions proposed.

Mirheidari et al. categorize event aggregation and correlation algorithms into three types: Similarity-based, Knowledge-based and Statistical-based. ThreatVectors approach to event correlation and aggregation fits into the similarity-based category. Similarity-based approaches compare events based on rules over the metafields of the events occurring within a defined time window. Events that meet the rule criteria are grouped together as an event set, meta-event or cluster. The rules may range in complexity from simple direct comparisons [7] to hierarchical rulesets [8] to those derived using machine learning techniques [9]. Unlike previous work, ThreatVectors give users full control over the correlation function allowing them to define as many or as few event collections as they see fit using a SQL-like query language. All events meeting the query term are entered into the associated collection. Each collection of events constitutes a single ThreatVector view. Aggregation within a ThreatVector is performed though simple deduplication where events containing the same content (i.e., downloaded file or email attachment) are presented as a cluster. A key distinction of this work is that events are allowed to exist in multiple ThreatVector collections simultaneously; yet, adjudication only has to occur once allowing analysts to approach events from concurrent contexts.

Knowledge-based approaches differ from ThreatVectors due to their reliance on information outside of the events themselves such as network configuration and structure. If such information is available, knowledge-based approaches offer the possibility of constructing an entire attack graph [10] or history of the attack across multiple events [11] and determine possibly missed components of an attack [12]. CSOCs may not have such auxiliary information readily available. Network structure and configuration data, when it exists at all, is often stored outside of a SIEM and managed by the enterprise's information technology organization not its CSOC.

Statistical-based approaches seek to find common patterns of occurrence between events over time without apriori knowledge of attack scenarios or network infrastructure. These techniques are primarily concerned with false positive reduction. By finding event patterns, such as regularly occurring events [13] or associated/causal events [14], background noise in the sensing system can be suppressed. Statistical-based approaches are often subject to high error rates in event correlation and some techniques require large quantities of training data to be effective.

Event or meta-event prioritization is typically implemented separately from event correlation and aggregation. Event prioritization schemes range from relatively straight forward weighted sums [11] to advanced applications of machine learning [15]. Two classes of prioritization approaches exist. First, those based on prior manual analysis of similar content such as Snort rule priorities or anti-virus signature severity metrics. Second, those that attempt to predict risk or impact to the enterprise by leveraging auxiliary information such as vulnerability databases, network or host configuration databases or victim criticality assessments [11] [14] [15] [16].

ThreatVectors' approach to event prioritization or ranking allows users a mix of the analysis of similar content and auxiliary information approaches. Within each ThreatVector collection a different ranking function can be defined using our DSL. This allows users to programmatically express preferences for certain contextual information. Users may augment automated detection scores with auxiliary information about the network session involved in delivery of the suspicious content and the history of events sharing metadata similarities. Applying different rankings based on contextual grouping ensures critical metadata fields are present and allows the system to handle heterogeneous events without sacrificing prioritization accuracy.

2.2 Event Visualizations

The cyber security research community and commercial enterprises have developed many approaches and tools for assisting cyber analysts. Commercial visualizations accompanying cyber security products tend to focus on basic system-level dashboards and tabular displays of events. Often commercial systems support a search capability and provide some fixed method of event prioritization used to drive the dashboard and tabular displays. A few commercial products offer network graph-based views of aggregated events for improving situational awareness. Only basic visualizations, if any, of analytic progress exist in most commercial products.

Cyber security visualization research community efforts related to event visualization tend to fit into one of three categories: log visualization, network visualization for situational awareness and attack visualization. The goal of log visualization is the manual identification of patterns and anomalies that could indicate a network has been breached. Examples of log visualization include Backhoe [17] a generic security log browser, IDS RainStorm [18] which uses parallel axis visualization to find patterns across IDS alerts, Hao et al. offer both a web application for alert browsing [19] focusing on allowing users a high degree of customization and an alert ensemble visualization approach using a variety of interactive 2D charts [20]. ThreatVectors share Hao et al.'s ensemble approach to grouping events for visualization but uses a radically different technique for actually displaying the results.

Network visualizations attempt to place event data in context of a view of the enterprise or global network environment. Approaches include the use of node-edge graphs as exemplified by the ACCEPT tool [11], radial or petri dish displays as in the Ocelot tool [21], scalable glyph representations used by ClockView [22] and hierarchical treemaps in the IP-space mapping tool [23]. Network-level visualizations provide additional situational awareness and context relative to log visualizations while handling simultaneous display of many heterogeneous event sources. These views, however, provide analysts with little assistance in triaging individual events.

Attack visualization offers another approach that focuses on sets of correlated events belonging to a single effort by an adversary to breach a network. By combining multiple events a clearer picture of an adversary's tools, techniques and procedures may be formed along with greater understanding of the potential impact of the attack on the enterprise. Tools such as NAVIGATOR [24] and PERCIVAL [10] are good examples of attack visualizations. Often attack visualizations require significant auxiliary information outside the scope of the correlated events, such as network infrastructure, host vulnerabilities and threat capabilities, to be fully realized. The need for this auxiliary information can often limit the use of such visualizations within dynamic enterprise network environments.

ThreatVectors, a type of event log visualization, offer two unique contributions. First, it provides an analytic overview focused on meeting CSOC manager's need to understand total workflow progress and remaining work by using a novel matrix of progress bars approach which we call our ThreatTempo visualization.

Existing event visualizations tend to completely ignore the workflow monitoring aspect of cyber security operations. From the ThreatTempo visualization users can dynamically pivot into any ThreatVector collection (i.e., ensemble). The collection view provides the second unique contribution. Individual events are presented as information-dense thumbnails and thumbnail clusters with full event details available on-demand. The cyber event thumbnails are similar in concept to those proposed by Chen et al. for music files [25] but conceptually differ from thumbnail representations of network assets in a variety of commercial tools. In ThreatVectors, event thumbnails are displayed in priority order based on a ranking score and loaded using a familiar infinitely scrollable paging mechanism. A metadata-based summarization and filtering mechanism is also provided to allow users to quickly focus in on similar events. Unlike previous work, ThreatVectors are focused on reducing the time required to triage all events identified as requiring further investigation and tracking analysts' progress over time.

2.3 Cyber Domain Specific Languages

A domain specific language is a computer language specialized to a particular application domain. Previous efforts at formalizing the cyber domain focused primarily on ontology development. Mitre [26] is perhaps the most successful with its STIX [27], CybOX [28] and MAEC [29] ontologies but others [30] have also contributed. While ontologies are developed for standardizing the language and semantics associated with a domain to facilitate sharing of information, DSLs are used to express instructions within a scripting language used by computers. Cyber DSLs for cyber-physical systems [31] and intrusion detection system rules [32] have been developed. Bhatti et al. even proposed using DSLs as a basis for developing next generation network protocols that would have provably correct construction [33]. The closest implementation of a cyber DSL to ThreatVector's is Splunk's Search Processing Language (SPL) [34] [35]. ThreatVector's DSL is focused on streaming calculation of a ranking function on a per event basis where as SPL is focused on facilitating and summarizing search results within the Splunk® SIEM.

3 ThreatVectors: Approach and Components

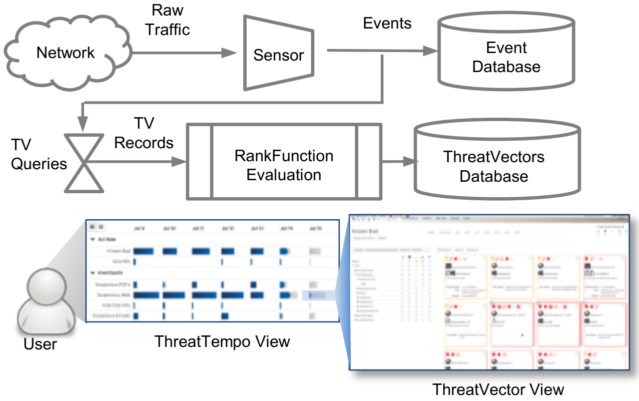


Figure 1: ThreatVectors' (TV) system architecture uses a parallel event processing path to facilitate systems integrations. Interactive visualizations show workload progress and high-level collection details with options for filtering and drill-down.

ThreatVectors are an interactive, web application for the visualization and adjudication of cyber security events generated by one or more network sensors. Our design and approach follows Shneiderman's principles of visual information: Overview first, zoom and filter, then details-on-demand, extract sub collections and allow users to view relationships among items [36]. The system conducts real-time processing of an event stream. It seeks to assist users in triaging cyber events and enables them to find malicious activity and malware more quickly and effectively. In particular, the design is heavily influenced by user feedback regarding effective event visualization for network sensors producing probabilistic or statistical detection results such as those using machine learning-based analytics for malware detection.

ThreatVectors consist of four major components shown in Figure 1 as parts of the high-level systems architecture. The four major components are a streaming ThreatVector query processing engine, a streaming rank function evaluation engine and two views (ThreatTempo and ThreatVector). The TheatVector's parallel event processing dataflow allow users to add the visualizations without altering existing event databases or data models. The only requirements on the existing data model are that it contains an event timestamp, a unique event identifier and a field for event adjudication status (i.e., has the event review been completed and what is the resulting decision regarding maliciousness).

3.1 Streaming Event Correlation and Prioritization

Each ThreatVector collection is defined by a unique query statement, a rank function expressed in terms of the ThreatVector's DSL, and a PinPoint (described in Section 3.3.1) metafield hierarchy. Every event is tested to determine what ThreatVector queries, if any, it matches. For each matched event a ThreatVector record is created in the ThreatVector database. The record contains the original event ID (a foreign key into the event database), timestamp, matched ThreatVector and prioritization score, known as a rank. Event rank is calculated using the rank function. In cases where the same event ID is associated to different pieces of content, ThreatVectors use both the event ID and a hash of the content as the unique key into the event database. An event may match on one or more queries. Each match results in a unique record. Examples of several ThreatVector definitions that have been operationally vetted are provided in Section 4.

Queries are used to define event correlations. They are expressed using the Python-Query-Language (PQL) over the metadata fields present in an event [37]. To maintain maximum flexibility and agility for the user no strict requirements are placed on the queries besides being valid PQL. Ideally, however, ThreatVectors are defined such that the collection of events has shared metadata elements not necessarily common in all event types that are useful in prioritizing the collection. For example, email events may be grouped together because they share metafields such as to, from and subject that could be useful in prioritization. Events without content may be put into their own collection because adjudication procedures are different and fewer metadata fields are available for prioritization.

When events are displayed in the ThreatVector view they are displayed in descending order by their rank. When events form a cluster, the maximum rank in the cluster is used. A ThreatVector's rank function is evaluated to determine a record's rank. Rank functions map the event metadata space to the real numbers. Since each ThreatVector has a different rank function, prioritizations are meaningful only within the context of a particular ThreatVector. Record ranks should not be compared across ThreatVectors unless the same rank function is used.

The rank functions are defined using basic arithmetic operations: addition (+), subtraction (-), division (/), multiplication (*), power (^) and parenthesis for controlling order of operations.

In addition to these operations, the DSL defines several common mathematical functions as well as several specially designed functions that operate over event metadata shown in Table 1. The user may also use real-valued constants as operands or as mathematical function arguments. Evaluation is carried out using PLY, a Python-based lexical parser and compiler.

Table 1. ThreatVector DSL function definitions.

Function Name	Description
exists(*[metafield]*)	Returns true (1) if *metafield* exists in event otherwise returns false (0)
valueOf(*[metafield]*)	Returns the real-value of *metafield* if it exists otherwise 0
numberOf(*[metafield]*)	Returns the number of entries in *metafield* if it exists otherwise 0
count(*[metafield]*)	Returns the number of occurrences of the value of *metafield* in the entire event database or 0 if field is not an attribute of the event
flagFreq(*[metafield]*)	Returns the proportion of events with that are flagged or contain at least one flagged piece of associated content across all events in the event database with the same value of *metafield*.
contains(*[metafield]*, [*substr1*, ..., *substrN*])	Returns true (1) if any substring in the list is in the string representation of *metafield* otherwise returns false (0). Individual substrings may not contain brackets. Quotes (",') will be ignored within substrings.
ceil(*arg*)	Returns the integer closest to but greater than or equal to *arg*. *Arg* may itself be a valid rank function.
floor(*arg*)	Returns the integer closest to but less than or equal to *arg*. *Arg* may itself be a valid rank function.
round(*arg*)	Returns the integer closest to *arg*. *Arg* may itself be a valid rank function.
abs(*arg*)	Returns the absolute value of *arg*. *Arg* may itself be a valid rank function.
max(*arg1*,*arg2*)	Returns the larger of *arg1* and *arg2*. *Arg1* or *arg2* may be valid rank functions.
min(*arg1*,*arg2*)	Returns the smaller of *arg1* and *arg2*. *Arg1* or *arg2* may be valid rank functions.
log(*arg*)	Returns the natural logarithm (log base e) of *arg*. *Arg* must be greater than 0. *Arg* may itself be a valid rank function.
log10(*arg*)	Returns the log base 10 of *arg*. *Arg* must be greater than 0. *Arg* may itself be a valid rank function.

3.2 ThreatTempo: A progress overview

The ThreatVectors approach to visualization consists of a three step drill-down process. Users progress from an overview of triage progress across all ThreatVectors to high-level summaries of individual ThreatVector collections to detailed event views. We found the first step missing or only superficially handled by most existing cyber event visualization schemes. Often triage progress is reported for a single timeframe as a percent of all events observed. Users, particularly CSOC managers and administrators, indicated during interviews a desire to understand analytic progress on a temporal basis with an emphasis on recent activity. The most popular time frames suggested were shift-based or daily summaries. We elected daily summaries because they were universally applicable.

Our overview of triage progress, called ThreatTempo, summarizes adjudication work by ThreatVector over several recent timeframes (see Figure 2). This allows managers, analysts and administrators to quickly identify threat sources that are producing a disproportionate or unusually small or large number of events. They can quickly understand, at a high-level, if events are deviating from their typical baseline due to activity such as a denial of service attack, new “noisy” signature or spear phishing campaign. Managers can also better understand what types of events are receiving the most attention by analysts and which are being operationally ignored or deprioritized. They can determine if their teams can handle additional eventing workloads or need to scale back on the type of events they elect to further investigate. This can lead to better system-wide sensor tuning and greater transparency throughout the enterprise.

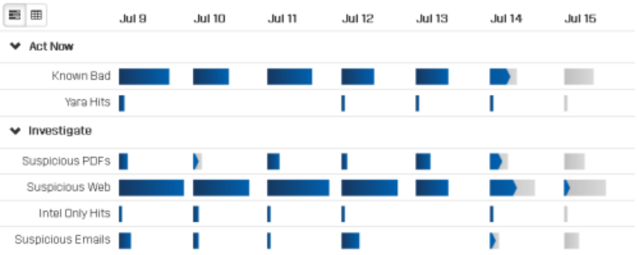


Figure 2: ThreatTempo triage progress matrix shows work breakdown and completion level over a seven day period.

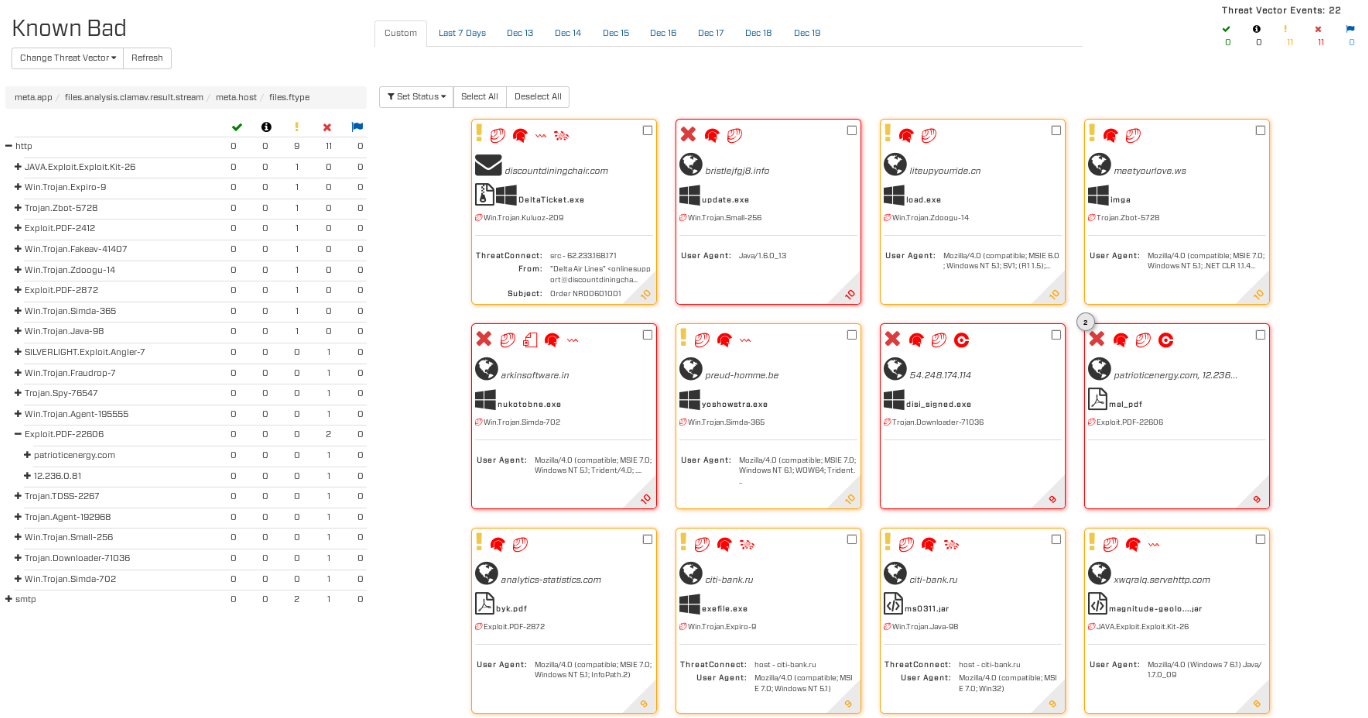
The ThreatTempo visualization is a matrix of progress bars. Our goal was to represent simultaneously the relative amount of work across each ThreatVector and the progress in completing that work. Each cell in the matrix consists of a gray bar whose width is scaled to reflect the relative number of records in the cell. We found two scalings to be beneficial. First, each bar is scaled by the maximum number of events in the matrix (see Equation 1 and Figure 2). Second, each bar is scaled based on the maximum number of records per ThreatVector row (see Equation 2).

$$w_{i,j} = \frac{0.8{*}r_{i,j}}{\max\limits_{t\in TVs}\left(\max\limits_{d\in Days}(r_{t,d})\right)}\ \text{, or} \qquad (1)$$

$$w_{i,j} = \frac{0.8{*}r_{i,j}}{\max\limits_{d\in Days}(r_{i,d})}\ , \qquad (2)$$

where w is the width factor, r is the number of records, i is the cell row, and j is the cell column. The factor of 0.8 is used to ensure some whitespace between adjacent bars. Since the display can be dynamically sized by the browser, the actual bar width is the cell width determined by the browser times the width factor.

As records are adjudicated within each cell a blue progress arrow fills the gray bar. When a cell reaches 100% completion the entire bar is turned blue. The two bar sizing methods differ primarily in the ability to visualize the progress arrow. The second method produces larger bars and is therefore easier to view progress. This method, however, distorts the relative quantity of records when comparing across ThreatVectors. For systems where ThreatVectors are relatively balanced in size, typically within an order of magnitude of each other, we’ve found users prefer the first method. To support deployments or time frames when one cell vastly dominates the matrix, we’ve incorporated an option to dynamically switch to a textual display of the numbers of adjudicated and total records per cell.



Known Bad

Threat Vector Events: 22

✓	i	!	✗	⚑
0	0	11	11	0

meta.app / files.analysis.clamav.result.stream / meta.host / files.ftype

	✓	i	!	✗	⚑
− http	0	0	9	11	0
+ JAVA.Exploit.Exploit.Kit-26	0	0	1	0	0
+ Win.Trojan.Expiro-9	0	0	1	0	0
+ Trojan.Zbot-5728	0	0	1	0	0
+ Exploit.PDF-2412	0	0	1	0	0
+ Win.Trojan.Fakeav-41407	0	0	1	0	0
+ Win.Trojan.Zdoogu-14	0	0	1	0	0
+ Exploit.PDF-2872	0	0	1	0	0
+ Win.Trojan.Simda-365	0	0	1	0	0
+ Win.Trojan.Java-98	0	0	1	0	0
+ SILVERLIGHT.Exploit.Angler-7	0	0	0	1	0
+ Win.Trojan.Fraudrop-7	0	0	0	1	0
+ Trojan.Spy-76547	0	0	0	1	0
+ Win.Trojan.Agent-195555	0	0	0	1	0
− Exploit.PDF-22606	0	0	0	2	0
+ patrioticenergy.com	0	0	0	1	0
+ 12.236.0.81	0	0	0	1	0
+ Trojan.TDSS-2267	0	0	0	1	0
+ Trojan.Agent-192968	0	0	0	1	0
+ Win.Trojan.Small-256	0	0	0	1	0
+ Trojan.Downloader-71036	0	0	0	1	0
+ Win.Trojan.Simda-702	0	0	0	1	0
+ smtp	0	0	2	1	0

discountdiningchair.com
DeltaTicket.exe
Win.Trojan.Kuluoz-209
ThreatConnect: src - 62.233.168.171
From: "Delta Air Lines" <onlinesupport@discountdiningcha...
Subject: Order NR00601001
10

bristlejfgj8.info
update.exe
Win.Trojan.Small-256
User Agent: Java/1.6.0_13
10

liteupyourride.cn
load.exe
Win.Trojan.Zdoogu-14
User Agent: Mozilla/4.0 (compatible; MSIE 6.0 ; Windows NT 5.1; SV1; (R1 1.5);...
10

meetyourlove.ws
imga
Trojan.Zbot-5728
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4...
10

arkinsoftware.in
nukotobne.exe
Win.Trojan.Simda-702
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; ...
10

preud-homme.be
yoshowstra.exe
Win.Trojan.Simda-365
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident. ..
10

54.248.174.114
disi_signed.exe
Trojan.Downloader-71036
9

2
patrioticenergy.com, 12.236...
mal_pdf
Exploit.PDF-22606
9

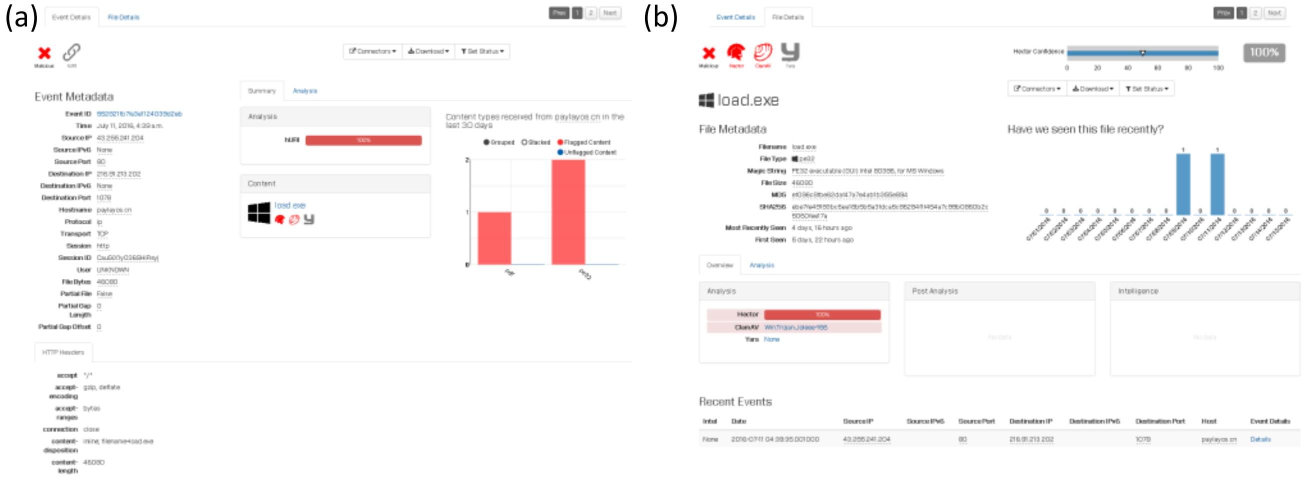
analytics-statistics.com
byk.pdf
Exploit.PDF-2872
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; InfoPath.2)
9

citi-bank.ru
exefile.exe
Win.Trojan.Expiro-9
ThreatConnect: host - citi-bank.ru
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
9

citi-bank.ru
ms0311.jar
Win.Trojan.Java-98
ThreatConnect: host - citi-bank.ru
User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Win32)
9

xwqralq.servehttp.com
magnitude-geolo....jar
JAVA.Exploit.Exploit.Kit-26
User Agent: Mozilla/4.0 (Windows 7 6.1) Java/1.7.0_09
9

Figure 3: ThreatVector view showing PinPoint summaries and filter selection on the left and a thumbnail grid on the right. Date range filter is available above the thumbnail grid. Cluster size annotations shown when the same content is observed in the event database multiple times within the date range. Thumbnails are sorted by their rank score placing those with highest priority first (top-left of grid). Yellow and blue thumbnails indicate those in need of further investigation while red and green thumbnails indicate fully adjudicated events.



Event Metadata

Event ID [illegible]
Time July 11, 2016, 4:39 a.m.
Source IP 43.255.241.204
Source IPv6 None
Source Port 80
Destination IP 216.81.213.202
Destination IPv6 None
Destination Port 1078
Hostname paylayos.cn
Protocol ip
Transport TCP
Session http
Session ID [illegible]
User UNKNOWN
File Bytes 46080
Partial File False
Partial Gap Length 0
Partial Gap Offset 0

HTTP Headers

accept */*
accept-encoding gzip, deflate
accept-ranges bytes
connection close
content-disposition inline; filename=load.exe
content-length 46080

load.exe

File Metadata

Filename load.exe
File Type pe32
Magic String PE32 executable (GUI) Intel 80386, for MS Windows
File Size 46080
MD5 [illegible]
SHA256 [illegible]
Most Recently Seen 4 days, 16 hours ago
First Seen 6 days, 22 hours ago

Hector 100%
ClamAV [illegible]
Yara None

Recent Events

Intel	Date	Source IP	Source IPv6	Source Port	Destination IP	Destination IPv6	Destination Port	Host	Event Details
None	2016-07-11 04:38:35.001000	43.255.241.204		80	216.81.213.202		1078	paylayos.cn	Details

Figure 4: ThreatVector thumbnail drill-down views into a thumbnail cluster. (a) Network event details including layer 3 to 7 metadata about the event and network metadata analytic results. (b) Content details including file metadata and file-based analytic results.

3.3 ThreatVector View

From the ThreatTempo matrix users can pivot into an analyst-centric view of a particular ThreatVector collection. The ThreatVector view is shown in Figure 3. It consists of several parts including a date picker at the top that allows users to switch between several predefined time spans or to a custom one, on the far left is the PinPoint dynamic filtering tool and on the right is a thumbnail grid. The visualization adopts a common user experience found in thumbnail viewers. An infinite scroll is used to load additional content. The infinite scroll allows the view to be performant even when large numbers of thumbnails are in the collection. Thumbnails are sorted by priority (i.e., rank) from top left in the grid to bottom right. Thumbnails can be clicked-on to receive a detailed view presented as a modal overlay on the ThreatVector page. Examples of the detailed event and content views for a thumbnail cluster of two events are shown in Figure 4.

ThreatVectors support five statuses of event adjudication during the triage process. The info status (white "i" in black circle) indicates events that are recorded for audit or informational purposes and have not been flagged by sensors for further review. The suspicious status (yellow exclamation point) indicates events flagged by the sensor but not yet reviewed by the analyst. Review status (blue flag) indicates events marked by an analyst for additional further study. Trusted status (green checkmark) is set by analysts to mark an event as a false positive after manual review. Malicious status (red X mark) is set by analysts to mark an event as a true positive after manual review. Events marked as malicious can be used by higher tier analysts in CSOCs as the basis of incident creation, remediation planning or tuning defensive systems such as firewalls and web proxies. Users may change event status directly on the thumbnail either one-at-a-time or in bulk or from within the detailed drill-down modal.

3.3.1 PinPoint

ThreatVector collections often contain events with similar properties or metadata. PinPoint is our dynamic filtering tool that allows analysts to view multi-level statistical summaries of events in the collection timeframe under view. A PinPoint metafield hierarchy is configured for each ThreatVector independently to take advantage of the ThreatVector context. For example, ThreatVectors based on events flagged by a signature-based analytic could use the signature name as one level in the metafield hierarchy (as shown in Figure 3). By allowing hierarchal summaries to be tailored to a particular context there is no need to develop a one-sizes fits all solution that works for signature and non-signature based sensors, email and web based events, events with file content and those without.

PinPoint displays each unique value for each metafield tier with its parent expanded. Parent values may be expanded to show the next lower tier by clicking the plus symbol in their row. In Figure 3, for example, the two unique “meta.app” values are “http” and “smtp”. Within the “smtp” events there are nineteen unique ClamAV signature strings. The event related to the expanded signature string had “meta.host” values of “patrioticenergy.com” and “12.236.0.81”.

Each unique value row has a breakdown of events containing that value by adjudication status. Users may click on any cell in PinPoint including the top-row event statuses and metadata value row names to dynamically filter the thumbnail grid. When a PinPoint cell is selected only the thumbnails meeting all conditions along the PinPoint metafield hierarchy and with the selected adjudication status will be displayed (still sorted by rank).

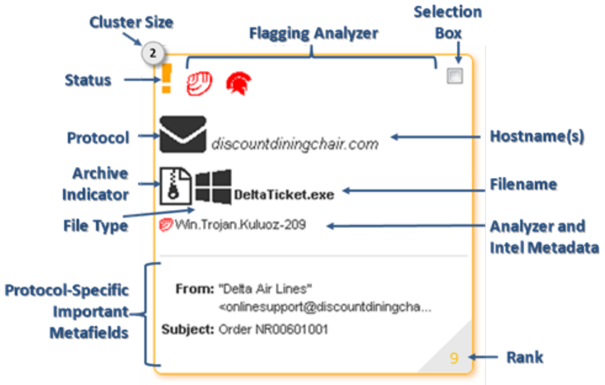
3.3.2 Thumbnails and Thumbnail Clusters

ThreatVector thumbnails and thumbnail clusters are generated on-demand when the view pages new records onto the screen. The thumbnails and thumbnail cluster visualizations are constructed using HTML templates within the Django web framework. Each thumbnail or thumbnail cluster is treated as an embedded foreign object within a single HTML5 scalable vector graphic (SVG) element representing the grid. Each thumbnail is a fixed size of 300x300 pixels with a 15 pixel-wide margin and retrieved with an AJAX GET request.

The thumbnail HTML template and associated Django view define a mapping of the event metadata schema or schemas into the generalized components of the thumbnail visualization shown in Figure 5. Glyphs are used where possible to represent common metadata fields such as protocol, file type, analyzers or intel sources and reduce the use of text. Within the Django view, different event metadata schemas or databases can be combined to produce a thumbnail.

Thumbnails were selected instead of the traditional row-based display to maximize the display of critical text fields that cannot be represented as glyphs such as the hostname, AV signature, user-agents, filename, email subjects and intel information. In row-based displays, particularly those that limit rows to a single line of text contain many metafields, and wish to avoid horizontal scrolling, column widths greatly constrain the number of characters that can be displayed per metafield. Therefore, applications using these types of tabular displays tend to leave this critical information for their detailed view. By elevating this critical data to this summary screen event adjudication can occur more quickly by avoiding pulling up the detailed view every time.

Thumbnail clusters are dynamically formed. Records sharing the same content are visually combined into a single thumbnail cluster. Since the adjudication status of a piece of content should be the same for all observed instances. We use the SHA256 hash as the unique content identifier.



discountdiningchair.com

DeltaTicket.exe

Win.Trojan.Kuluoz-209

From: "Delta Air Lines" <onlinesupport@discountdiningcha...

Subject: Order NR00601001

Figure 5: Annotated event thumbnail cluster showing two emails received with the same suspicious email attachment (a Zip archive containing a Windows executable)

4 THREATVECTORS: AN IMPLEMENTATION

ThreatVectors were implemented in a BluVector™ cyber threat hunting and detection network security appliance using the Django web framework. BluVector sensors provide multiple analytic engines for malware detection which append metadata to events generated from a Bro network monitoring cluster. BluVector’s analytic engines include both signature-based approaches (ClamAV and Yara) along with a machine learning-based detection engine called Hector. The Hector detection engine produces confidence scores for all content between 0 and 100 with 0 indicating high degree of certainty the content is benign and 100 indicating a high degree of certainty the content is malicious. BluVector also offers a machine-learning based analyzer for web URLs called hUri which produces scores in a similar manner to Hector based on a lexical analysis of the hostname from which the content was downloaded. The sensor also produces events based on cyber intel indicators of compromise (IoCs) that may not be associated with any file content.

ThreatVectors were configured as detailed in Table 2. Six ThreatVector collections were defined and grouped into either an Act Now category or Investigate category based on whether the events were flagged by a signature-based tool or Hector without an accompanying anti-virus signature hit. Rank functions for Hector-based events assign higher rank to those events with higher Hector score, corroboration from intel, come from hosts that have a high flagging frequency and come from relatively rare hosts (i.e., the first time a host generates an event is given a higher rank than later events given all other things are equal). Rank functions for signature-based events produce higher ranks for events with greater degrees of corroboration from other analytic engine and intel. Note that the query field is used to limit some types of flagged events such as intel-only events with very small numbers of bytes transferred out of the enterprise. The suspicious PDF collection is limited by a minimum Hector score to help focus analysts due to the particularly high volume of this filetype.

Table 2: ThreatVector definitions used in BluVector implementation

ThreatVector Name	Group In	Query	RankFunction	PinPoint Hierarchy
Known Bad	Act Now	files.flags == 'clamav' and clamav not in ['Heuristics.Encrypted.PDF','Heuristics.Encrypted.Zip','Heuristics.Encrypted.7Zip']	numberOf("files.flags") + exists("intel.providers")	meta.app -> files.analysis.clamav.result.stream -> meta.host -> files.ftype
Yara Hits	Act Now	files.flags == 'yara'	numberOf("files.flags") + exists("intel.providers")	meta.app -> files.analysis.yara.result.rule-> meta.host -> files.ftype
Suspicious PDFs	Investigate	files.ftype == 'pdf' and files.flags == 'hector' and files.analysis.hector.result.confidence >= 0.8 and files.flags != 'clamav'	0.5*valueOf("files.analysis.hector.result.confidence") + 0.25*flagFreq("meta.host") + exists("intel.providers") + 0.25*max(1/count("meta.host"),0.1)	meta.app -> meta.host -> files.fname
Suspicious Web	Investigate	meta.app == 'http' and files.ftype != 'pdf' and files.flags == 'hector' and files.flags != 'clamav'	0.3*valueOf("files.analysis.hector.result.confidence") + 0.2*flagFreq("meta.host") + 0.3*valueOf("analysis.huri.result.confidence") + exists("intel.providers") + 0.2*max(1/count("meta.host"),0.1)	meta.host -> files.ftype -> meta.headers.user-agent
Suspicious Emails	Investigate	meta.app == 'smtp' and files.ftype != 'pdf' and files.flags == 'hector' and files.flags != 'clamav'	valueOf("files.analysis.hector.result.confidence") + 0.25*flagFreq("meta.host") + exists("intel.providers") + 0.25*max(1/count("meta.host"),0.1)	meta.host -> files.ftype -> meta.headers.subject
Intel Only Hits	Investigate	intel == exists(True) and files.flags == exists(False) and ((bv_intel.conn.orig.size >= 256) or files == exists(True))	numberOf("intel.providers") + 0.5*flagFreq("meta.host") + 0.5*max(1/count("meta.host"),0.1)	meta.app -> meta.host -> files.ftype -> intel.flagged_fields

5 RESULTS AND USER FEEDBACK

ThreatVector system performance tests were conducted using a tap of a saturated 1 Gbps network link at an enterprise gateway supporting tens of thousands of end users. Over seven 24-hour periods, the maximum observed average record creation rate in one period was 8.2 records/minute. This serves as a lower-bound on system ingest performance. Web application load times for the ThreatTempo visualization remained below two seconds for ThreatVector databases of up to the maximum tested size of 45,000 records in the seven day window. ThreatVector views had paging load times between approximately between 1 and 5 seconds. Individual thumbnails are returned asynchronously after initial page load in between 1 and 5 seconds.

BluVector security appliances with the ThreatVectors system were deployed onto several enterprise networks and users from two of the deployments were interviewed after at least one month of working with the system. Sensors were deployed as part of the enterprises' security gateway or provided a passive tap of gateway traffic used for experimental evaluations. Users including analysts, system administrators and CSOC managers were interviewed after the evaluation period and asked to compare the ThreatVectors workflow and visualizations with the preexisting tabular view of events.

User feedback on the tabular display within the BluVector system showed a significant degree of frustration in identifying which order to adjudicate events. This feeling was exacerbated by the "black-box" nature of Hector analytic engine which provides no detail on how it arrived at a particular score. Prioritization and adjudication of Hector-only flagged events was particularly difficult and lead to an overall feeling of the system being too complicated to use.

Users who evaluated the ThreatVector system found it much easier than that tabular display to navigate to high priority events. The preconfigured nature also made the system feel less complicated to use. They also noted the ability to better "see" multievent campaigns in near real-time, particularly phishing campaigns that produced large thumbnail clusters and shared key metafields such as from addresses or subjects even if the content hash was different. CSOC managers appreciated the ability to see the collaborative work progress by their team, even though their team was apprehensive about having their work progress clearly monitored. Analysts generally liked the idea of being able to categorize and influence event prioritization schemes but none stated that they made any changes to the default configuration provided. Analysts were able in several circumstances to adjudicate events directly from the thumbnail view and did not have to resort to viewing the full details of the event saving a significant amount of time.

6 CONCLUSIONS AND FUTURE WORK

The ThreatVectors workflow and event visualization system provide analysts, CSOC managers and system administrators with a highly customizable approach to reducing the time spent triaging events. Furthermore, it enables them to more quickly recognize metaevents such as spear phishing campaigns. Several design improvements were identified in testing and user trials including providing per event score breakdowns and a simpler configuration method. Additionally, the ThreatTempo visualization could be made to better handle larger disparities in collection size by providing an option for log scaling of the progress bar widths.

While designed and partially implemented as a sensor and event database/schema agnostic solution, further generalization of the implementation is left as future work and must be completed for the system to successfully operate outside of a BluVector sensor. Other future work includes enhancements to the thumbnail views to add more graphical elements and perhaps adding support for dynamically sizing thumbnails.

REFERENCES

[1] C. Zimmerman, Ten Strategies of a World-Class Cybersecurity Operations Center, Bedford, MA: MITRE Corporation, 2014.

[2] A. D'Amico, K. Whitley, D. Tesone, B. O'Brien and E. Roth, "Achieving Cyber Defense Situational Awareness: A Cognitive Task Analysis of Information Assurance Analysts," in *HUMAN FACTORS AND ERGONOMICS SOCIETY*, Orlando, FL, 2005.

[3] Ponemon Institute LLC, "Security Metrics to Manage Change: Which Matter, Which Can Be Measured?," Ponemon Institute, Traverse City, MI, 2014.

[4] S. Mirheidari, S. Arshad and R. Jalili, "Alert Correlation Algorithms: A Survey and Taxonomy," in *Proceedings of 5th International Symposium of Cyberspace Safety and Security*, Zhangjiajie, China, 2013.

[5] G. Spathoulas and S. Katsikas, "Methods for post-processing of alerts in intrusion detection: A survey," *International Journal of Information Security Science,* vol. 2, no. 2, p. 64, 2013.

[6] R. Zuech, T. Khoshgoftaar and R. Wald, "Intrusion detection and Big Heterogeneous Data: a Survey," *Journal of Big Data,* vol. 2, no. 3, 2015.

[7] H. Debar and A. Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts," in *International Symposium on Recent Advances in Intrusion Detection*, Davis, CA, 2001.

[8] F. Valeur, G. Vigna, C. Kruegel and R. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, no. 3, pp. 146 - 169 , 2004.

[9] C. Mironeanu, M. Craus and C. Butincu, "Intrusion detection using alert prioritization and multiple minimum supports," in *2015 14th RoEduNet International Conference - Networking in Education and Research*, Craiova, 2015.

[10] M. Angelini, N. Prigent and G. Santucci, "PERCIVAL: Proactive and rEactive attack and Response assessment for Cyber Incidents using Visual AnaLytics," in *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Chicago, IL, 2015.

[11] A. Kim, M. Kang, J. Luo and A. Velazquez, "A Framework for Event Prioritization in Cyber Network Defense," Naval Research Laboratory, Washington, DC, 2014.

[12] P. Ning, D. Xu, C. Healey and R. St. Amant, "Building Attack Scenarios through Integration of Complementary Alert Correlation Methods," in *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2004.

[13] J. Viinikka, H. Debar, L. Me and R. Seguier, "Time series modeling for IDS alert management," in *ACM Symposium on Information, computer and communications security (ASIACCS)*, Taipei, Taiwan, 2006.

[14] X. Qin and W. Lee, "Discovering Novel Attack Strategies from INFOSEC Alerts," in *Symposium on Research in Computer Security*, Sophia Antipolis, France, 2004.

[15] M. Bierma, J. Doak and C. Hudson, "Learning to Rank for Alert Triage," in *12th International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, 2013.

[16] K. Alsubhi, E. Al-Shaer and R. Boutaba, "Alert prioritization in Intrusion Detection Systems," in *IEEE Network Operations and Management Symposium*, Salvador, Bahia, 2008.

[17] S. Bratus, A. Hansen, F. Pellacini and A. Shubina, "Backhoe, a packet trace and log browser," in *IEEE Workshop on Visualization for Cyber Security (VizSec)*, Cambridge, MA , 2008.

[18] K. Abdullah, C. Lee, G. Conti, J. Copeland and J. Stasko, "IDS RainStorm: Visualizing IDS Alarms," in *IEEE Workshops on Visualization for Computer Security (VizSec)*, Minneapolis, MN, 2005.

[19] L. Hao, C. Healey and S. Hutchinson, "Flexible Web Visualization for Alert-Based Network Security Analytics," in *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Atlanta, GA, 2013.

[20] L. Hao, C. Healey and S. Hutchinson, "Ensemble visualization for cyber situation awareness of network security data," in *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Chicago, IL, 2015.

[21] D. Arendt, R. Burtner, D. Best, N. Bos, J. Gersh, C. Piatko and C. Paul, "Ocelot: User-Centered Design of a Decision Support Visualization for Network Quarantine," in *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Chicago, IL , 2015.

[22] C. Kintzel, J. Fuchs and F. Mansmann, "Monitoring large IP spaces with ClockView," in *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Pittsburgh, PA, 2011.

[23] S. Miserendino, C. Maynard and W. Freeman, "Configurable IP-space maps for large-scale, multi-source network data visual analysis and correlation," in *Visualization and Data Analysis* , San Francisco, CA, 2014.

[24] M. Chu, K. Ingols, R. Lippmann, S. Webster and S. Boyer, "Visualizing Attack Graphs, Reachability, and Trust Relationships with NAVIGATOR," in *IEEE Symposium on Visualization for Cyber Security (VizSec)*, Ottawa, Ontario, Canada, 2010.

[25] Y.-X. Chen and R. Klüber, "ThumbnailDJ: Visualizing music collections based on visual thumbnails," in *International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010.

[26] L. Obrst, P. Chase and R. Markeloff, "Developing an Ontology of the Cyber Security Domain," in *STIDS*, Fairfax, VA, 2012.

[27] Mitre, "Structured Threat Information eXpression (STIX)," 2016. [Online]. Available: http://stixproject.github.io/.

[28] Mitre, "Cyber Observable eXpression (CybOX)," 2016. [Online]. Available: https://cyboxproject.github.io/.

[29] Mitre, "Malware Attribute Enumeration and Characterization (MAEC)," 2016. [Online]. Available: https://maecproject.github.io/.

[30] Z. Syed, A. Padia, T. Finin, L. Mathews and A. Joshi, "UCO: A Unified Cybersecurity Ontology," in *AAAI Workshop on Artificial Intelligence for Cyber Security*, Phoenix, AZ, 2016.

[31] S. Pradhan, A. Dubey, A. Gokhale and M. Lehofer, "CHARIOT: A Domain Specific Language for Extensible Cyber-Physical Systems," in *Proceedings of the Workshop on Domain-Specific Modeling*, Pittsburgh, PA, 2015.

[32] K. Chotvorrarak and Y. Limpiyakom, "Domain Specific Language for Detecting Intrusion Signatures with Genetic Search," *International Journal of Security and Its Applications,* vol. 8, no. 2, pp. 125-138, 2014.

[33] S. Bhatti, E. Brady, K. Hammond and J. McKinna, "Domain Specific Languages (DSLs) for Network Protocols (Position Paper)," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, Montreal, QC , 2009.

[34] Splunk, "Splunk Enterprise Search Manual 6.4.2," Splunk, Inc., San Francisco, CA , 2016.

[35] Splunk, "Splunk Enterprise Search Reference 6.4.2," Splunk, Inc., San Francisco, CA , 2016.

[36] B. Shneiderman , "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," in *Symposium on Visual Languages*, Boulder, CO, 1996.

[37] A. Horev, "A python expression to MongoDB query translator," GitHub, 2014. [Online]. Available: https://github.com/alonho/pql.