Situ: Identifying and Explaining Suspicious Behavior in Networks

John R. Goodall, *Member, IEEE*, Eric D. Ragan, *Member, IEEE*, Chad A. Steed, *Senior Member, IEEE*, Joel W. Reed, G. David Richardson, Kelly M.T. Huffer, Robert A. Bridges, and Jason A. Laska

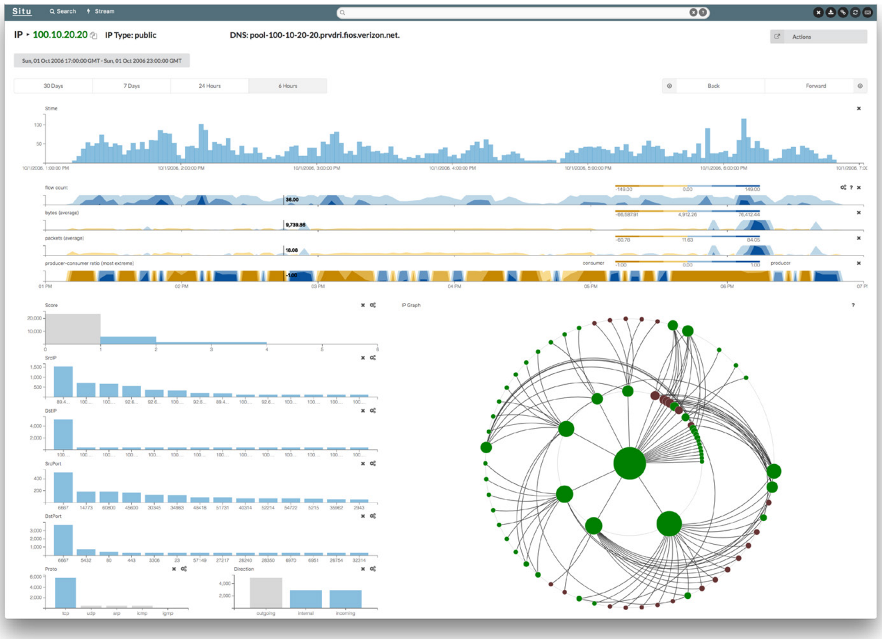


Fig. 1. The *IP Detail Page* of the *Situ* system includes a temporal histogram for selecting a time range, horizon graphs for temporal context, bar charts of field distributions for network flows of that IP, and a two-hop communication graph.

Abstract— Despite the best efforts of cyber security analysts, networked computing assets are routinely compromised, resulting in the loss of intellectual property, the disclosure of state secrets, and major financial damages. Anomaly detection methods are beneficial for detecting new types of attacks and abnormal network activity, but such algorithms can be difficult to understand and trust. Network operators and cyber analysts need fast and scalable tools to help identify suspicious behavior that bypasses automated security systems, but operators do not want another automated tool with algorithms they do not trust. Experts need tools to augment their own domain expertise and to provide a contextual understanding of suspicious behavior to help them make decisions. In this paper we present Situ, a visual analytics system for discovering suspicious behavior in streaming network data. Situ provides a scalable solution that combines anomaly detection with information visualization. The system's visualizations enable operators to identify and investigate the most anomalous events and IP addresses, and the tool provides context to help operators understand why they are anomalous. Finally, operators need tools that can be integrated into their workflow and with their existing tools. This paper describes the Situ platform and its deployment in an operational network setting. We discuss how operators are currently using the tool in a large organization's security operations center and present the results of expert reviews with professionals.

Index Terms—Network security, situational awareness, privacy and security, streaming data, machine learning, visualization

1 Introduction

Networked computing assets are routinely compromised, resulting in the exfiltration of intellectual property, the disclosure of classified information, and large financial damages. Despite the work of cyber security experts, these compromises occur regularly and the impacts are staggering. The Center for Strategic and International Studies estimated the global cost of cyber crime at $445 billion each year; in the US, these losses represent 0.6% of GDP and in Germany 1.6% [18]. While reports citing such large numbers might be considered self-serving, other effects of cyber crime are even more critical. Sophisticated attack groups at the nation-state level constantly develop new network penetration methods that current technologies cannot detect. The 2016 United States elections, with allegations of Russian hacking, are a sobering reminder of the seriousness and global impact of cyber attacks.

The most commonly deployed measures for detecting attacks on networks and systems are intrusion detection/prevention systems and antivirus software. These systems typically operate based on signatures, which use pattern matching to identify malicious activity. Although effective at detecting known attacks, these systems are unable to detect

- *John R. Goodall, Chad A. Steed, Joel W. Reed, G. David Richardson, Kelly M.T. Huffer, Robert A. Bridges, and Jason A. Laska are with Oak Ridge National Laboratory. E-mail: {jgoodall, steedca, reedjw, richardsongd, hufferkm, bridgesra, laskaja}@ornl.gov.*
- *Eric D. Ragan is with University of Florida. E-mail: eragan@ufl.edu.*

*Manuscript received 31 Mar. 2018; accepted 1 Aug. 2018.
Date of publication 16 Aug. 2018; date of current version 21 Oct. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2865029*

novel attacks or variations. More recently, tools have been developed that can detect variations of known attacks using supervised machine learning techniques (e.g., [6, 34, 35, 41, 55]). These methods train on labeled data sets containing examples of known malicious behavior. While valuable, such an approach has limitations in an operational environment. Creating labeled training data is a laborious process that requires an expert to identify malicious behavior. The models, once created, need to be continually updated. Such systems typically identify variations on the attacks in the training set, but cannot detect completely novel attacks. Another approach to identifying malicious activity is to use reputation lists of known bad actors (i.e., IP addresses known to be associated with malware). These approaches require that the bad actors be known a priori, which is likely not the case for sophisticated attacks.

All of these approaches add to an enterprise's defenses, but these automated solutions have two key limitations: 1) they will not detect all attacks (cyber security is asymmetric—the attacker doesn't need to be successful every time, but the defender does), and 2) automated solutions ignore the potential of human domain experts who understand both the domain (e.g., network protocols) and their own operational environment. Security operators need tools to help identify suspicious behavior that bypasses automated security systems.

Given the amount of data on today's networks, operators cannot be expected to discover suspicious activity without better tools. What is needed is a visual analytics approach with algorithms to distinguish the signal from the noise and visualizations to provide a meaningful context for suspicious activity so operators can determine the impact and react appropriately. Highlighting such suspicious behavior helps operators focus their limited time on the most suspicious events.

In this paper, we present *Situ*, a new visual analytics tool designed to complement existing security measures and help operators maintain situation awareness, identify suspicious behavior, and understand the context of that behavior. Situ integrates a method of unsupervised machine learning for anomaly detection with data visualization to help operators identify possible attacks, understand what makes an event suspicious, and determine the importance and impact of the event. The benefit of Situ's visual analytics design is the ability to not only highlight the detected anomalies, but to help analysts understand *why* the algorithm considers them anomalies. The interface design uses multiple linked views and pages that allow analysts to maintain an overview of network activity alongside dedicated views that allow inspection of details. The benefit of this design helps experts to better understand and trust the algorithms while still taking advantage of their experience and domain expertise when interpreting both the algorithmic output and the raw data itself.

The contributions of this paper are:

- A summarization of the most important design goals for anomaly detection and visualization systems for cyber security applications, as derived from the literature and our interactions with domain experts.

- A unique collection of anomaly scoring analytics and connected information visualization techniques that target the specific challenges of network security. The coordinated visualization techniques include several proven charting techniques as well as more innovative designs. An important technical contribution of Situ lies in the integration of these visualization and analytics techniques to form a complete system with benefits that are greater than the sum of the individual parts.

- A streaming anomaly detection platform and visualization for cyber security, with case studies demonstrating how it can be used and feedback from analysts in a production deployment.

2 Related Work

To help establish the contributions in our our presentation of Situ, we provide an overview of anomaly detection and visualization for the cyber security domain.

2.1 Anomaly Detection for Cyber Security

Applications of anomaly detection for identifying intrusions are commonplace in the literature, and the focus is primarily on accuracy—identifying the attributes and algorithms combinations to increase the overlap of detected events (anomalies) and actual positive events (attacks), e.g., [2, 12, 17, 20, 29].

Our work prioritizes a scalable, online, interpretable anomaly detection system for cyber security. Work in anomaly detection for cyber security often focuses on only some these values while de-emphasizing others. For example, some methods consider each event at only one level of granularity (e.g., [11, 54]). Multiple models at different levels of granularity facilitate the interpretation of why alerts trigger and why they are relevant. Other implementations combine analyses across multiple levels of granularity through non-comparable detector scores obscuring the relative influence of the granularity analysis [36, 48]. Some approaches (e.g., [10, 11]) require an expensive model update step such as computing the pairwise distances between all data points, which limits both the scalability and the ability for updating the model in an online fashion. Other methods focus on non-probabilistic techniques [47], which can lack the ability to communicate the confidence of the results and can be difficult to compare to other detectors.

Other applications of anomaly detection for cyber security signal on the *probability* of an event being above a threshold [36]. In our Situ system, we use an ensemble of multinomial distributions, one per IP per statistic, with each updated in real time using a simple Bayesian update. We define anomalies as low probability events, i.e., an event whose p-value is below a fixed threshold, where the p-value is computed from the multinomial. This approach follows mathematical and empirical results of Ferragut et al. [13] and Bridges et al. [5]

Ferragut et al. [13] promote the p-value definition of anomalies, citing two main advantages. (1) *Comparability:* p-values admit quantitative comparability across detectors. This is necessary for situational awareness in the presence of many and/or evolving detectors, as scores from multiple models must be compared. (2) *Regulatability:* a theorem is provided giving a sharp bound on the likelihood of an alert in terms of the p-value threshold. This allows operators, especially in high throughput applications, to theoretically set the thresholds of the ensemble of detectors to bound the number of alerts. Ferragut et al. [13] presented experiments with cyber data sets, and the results demonstrate the efficacy of modeling simple statistics of network data for anomaly-based intrusion detection.

Bridges et al. [5] build on Ferragut et al. [13] and provide multiple theorems for mathematically understanding the relationship between the alert rate and threshold. Operationally, this provides criteria for when users can set thresholds to prescribe the alert rate in expectation (not just bound it). In addition, this method describes how deviations from the proven relationships stem from model drift, indicating a state change and the need for model retraining. These advantages are inherited by our approach for Situ.

2.2 Cyber Security Visualization

While various forms of data analysis may fall into the realm of cyber security, our research is most concerned with analysis of streaming network data that is continually arriving and updating. Systems designed for analysis of streaming data often aim to support both real-time monitoring of incoming data as well as the inspection of older data to provide context (e.g., [28, 37, 40]). Following recent updates is essential to maintain situational awareness of the state of the network in order to respond quickly and make appropriate real-world decisions [9, 21].

Streaming data introduces challenges for analysis due to the continued growth and dynamic nature of the data. Other major challenges faced while designing visual analytic tools to support cyber security analysis are scalability and flexibility. Visual analytics approaches aim to help analysts handle the overwhelming amounts of data while highlighting the most important items or patterns. For example, Gupta et al. [26] proposed a method for handling ad-hoc querying of streaming data with the CHAOS system. It provides a scalable platform for anomaly detection in data stream by first applying data reduction then implementing a computational data cube. To address the challenge of balancing of real-time and previous data, StreamSqueeze [39] uses a screen-filling technique that provides more details for events in a data stream that are closer to the current time. This technique takes into account the higher relevance of recent events while still making it possible to follow trends for the history of prior events. Another tool, VizTree [37] addresses the challenge of interpreting large time-series data by transforming the data to a symbolic representation that is visualized in trees. A later example is LiveRAC [40], a system designed for visualization of large amounts of network data using a collection of basic charts such as line charts and bar graphs. The core component of this system is a reorderable matrix of charts that employs the stretch and squish technique of accordion drawing designs. These tools are visualizations that lack the analytics required to help focus the domain experts on the most important events.

CLIQUE [4] employs a behavioral modeling approach that learns the expected activity of actors and collections of actors on a network, and then compares current activity to this learned model to detect behavior-based anomalies. To support real-time situational awareness, CLIQUE shows flow-activity levels for each actor across a range of categories (such as web, ftp, and email) as well as a summary behavioral signal that reflects actor deviation from calculated baseline behavior. Where CLIQUE uses simple statistics, the analytics in Situ take a probabilistic modeling approach, discussed in 4.3.3.

3 Design Requirements

In this section, we describe the need for an anomaly detection system in cyber security as well as the functional design requirement for both the anomaly scoring and visualization components in Situ. The description is based on our prior work, other research uncovering the work practice of security analysts, and more recent interviews and observations with numerous security analysts. The goal of our visual analytics approach is to leverage analytics to assist in managing scalability of vast quantities of network data in a streaming scenario while also allowing experts to use their judgment to review and better understand trends and details.

While automated cyber security solutions are commonplace among most organizations, and nearly all organizations will have a log collection infrastructure and dashboard, visual analytics are rare. Systems like firewalls and intrusion prevention systems can automatically block some malicious network traffic at an organization's border. Virus scanners can quarantine known-malicious malware on a host system. These are often used in conjunction in a strategy known as *defense-in-depth*. This approach is based on the intuition that any one solution cannot stop all malicious network traffic, but employing a variety of solutions increases the likelihood of stopping such traffic. These automated systems perform well at stopping known malicious behavior, but they are incapable of preventing all attacks. Therefore, domain experts need new tools that helps them identify and understand potentially malicious events in the large volumes of data collected on computer networks in today's environments.

Situ is intended to be complementary to these kinds of automated solutions by highlighting suspicious activity indicative of an attack that other tools are unable to identify. It is designed to make anomalies salient for the user while using visualization to assist in understanding the context of those anomalies. We note that events that our system highlights with high anomaly scores are not necessarily malicious, but the rarity of these events make them important for situation awareness, and our observations during both controlled tests and real-world deployments show that they often indicate malicious activity.

Below, we describe the specific requirements for both the anomaly detection and visualization components of an anomaly detection visual analytics system.

3.1 Anomaly Detection Requirements

Understanding security analysts' work practice has been the focus of numerous research projects (e.g., [7, 22, 50, 52]). Based on this past research and our own observations and interactions with security analysts, we formulated the following *anomaly detection requirements* (ADR)—functional requirements for anomaly detection tools in cyber security:

- **ADR1-Understandable scores**: Anomaly detection results must be understandable to the security analysts. We have observed that security analysts are suspicious of "black box" solutions that highlight anomalies but fail to communicate *why* something is anomalous. Anomaly detection systems should help the analysts understand an event, and the first step is to understand what makes the event abnormal.

- **ADR2-Contextualizing events**: If the first step is understanding an event, the next step is gathering additional context about it. This contextual understanding during analysis is often derived from alternate data sources and tools [23, 52], such as log files, other security tools, and web sites. Anomaly detection systems should allow an analyst to understand the context of the event that may not be directly embedded in the event itself.

- **ADR3-Comparable scores**: Typically, different data sources, distributions within the same data source, or data from different entities will result in anomaly scores that are not comparable. However, having comparable scores is integral to analysis—analysts must be able to determine what are the most anomalous events in order to prioritize analysis. Anomaly detection systems should provide scores across data types, distributions, and other variations that can be directly comparable to each other.

- **ADR4-Fast notification**: An attacker can compromise a system and exfiltrate its data quickly. If a system takes too long to discover an event and notify an analyst, the attacker may have already exited the network. Anomaly detection systems should provide timely results minimizing the time from event discovery to notification.

- **ADR5-Scalability**: Security systems that operate on network or log data need to scale to the immense volumes of those data sets [16]. Anomaly detection systems should scale to modern enterprise sizes.

3.2 Security Visualization Requirements

In addition to the above functional requirements for anomaly detection systems, we outline *security visualization requirements* (SVR) specific to the visualization component of any cyber security system:

- **SVR1-Temporal context**: Just as the anomaly detection system should provide additional data to enrich events and provide context, visualization tools should also emphasize the context of an event. This includes the temporal event context, which can be provided by displaying relevant data that happened before the event occurred [23]. Cyber security visualizations should provide the temporal context of an event.

- **SVR2-Scalability**: Like the anomaly detection system, the visualization component must scale to large enterprises. This can be achieved via data summarizations or by only showing the most anomalous events. Cyber security visualizations should scale to handle large volumes of security data.

- **SVR3-Access to raw data**: Security analysts may distrust visualization techniques, particularly those that smooth out the raw data [16]. Cyber security visualizations should provide analysts with access to the raw data for inspection.

- **SVR4-Center on enterprise assets**: Security analysts care the most about what is happening on their own network [3]. Analysts may keep up with larger security trends or even specific attacks on other enterprises, but these are only useful for adding context to attacks against the assets in their own enterprises. Cyber security visualizations should differentiate between local (within the enterprise) and remote assets and emphasize the former.

- **SVR5-Tool integration**: Many visualization tools are monolithic and do not integrate well with analysts' existing tools and data. Nevertheless, it is crucial that visualizations be capable of integrating with existing tools that analysts rely on [16, 24, 25]. Analysts trust certain tools and data sources that they understand and heavily rely on. New cyber security visualizations should integrate with existing tools and data that analysts leverage.

- **SVR6-Collaboration**: Security analysts do not work alone and often collaborate—either together within an organization, with other IT specialists, or across organizational boundaries [21, 38, 52]. Some environments are inherently collaborative, such as operations centers [44]. Analysts also need to communicate about incidents, either across shifts or between different levels (or tiers) of analysts. Cyber security visualizations should facilitate collaboration and communication about events.

Each of these requirements point to a higher level requirement for security systems to augment the domain expertise of analysts by highlighting salient data and helping them develop a comprehensive understanding and trust of the results. Security analysts are domain experts with a tacit understanding of their environments, which is central to security work practice [21, 22, 52]. Thus, tool designers should create tools that automate the tedious work analysts currently do and provide methods that allow them to leverage this tacit knowledge.

While we designed Situ based on these requirements for anomaly detection and visualization systems for cyber security, we also provide this summary as a reference for other designers.

4 System Design

Situ is a visual analytics system that consists of two main components. First, a **streaming anomaly detection system**, which ingests and parses event streams from multiple sources, enriches events with additional context, provides an anomaly score for each event according to several different models of past behavior, and sends the enriched, scored events to a data store or message queue. Second, a **web-based visualization system**, which interacts with an API that provides access to the scored events to provide analysts with a way to explore events and understand their context.

4.1 Data

Situ is primarily being used with three different input data types: network flows, firewall logs, and web proxy logs. In this context, a network flow represents an aggregation of a set of packets exchanged by a pair of systems. See Table 1 for an example flow record. Most of the firewall log messages contain information about connections that are successfully opened, torn down, or are denied due to some policy violation. The web proxy messages describe events where a web request violates an organization's content retrieval policy or is attempting to retrieve something potentially malicious, as malware often does.

Flow Record Example	
Time	09:58:32.912
Protocol	tcp
SrcIP	192.168.1.100
SrcPort	59860
SrcPackets	201
SrcBytes	508526
DstIP	172.16.100.10
DstPort	80
DstPackets	595
DstBytes	1186562

Table 1. Flows record meta-data of IP communications.

4.2 Architecture

The anomaly detection system's architecture was designed to meet the requirements of **ADR4-fast notifications** and **ADR5-scalability**. To enable **fast notifications**, Situ has low latency (measured in milliseconds) event processing. It is possible to automate actions by outputting events above a threshold score to a message queue, where another job could read from the queue and send an email, automatically create a ticket, or perform some other response. To address the **scalability** requirement, Situ is a fully distributed system, as shown in Fig. 2. Each event is routed to a scoring node based on a hash of the IP address so that events with the same IPs are consistently routed to the same node, which allows the scoring behavior models for an individual IP to be all on the same node. The router nodes handle data ingesting and parsing, while the scorer nodes perform enrichment, scoring, and data output. The distributed nature of the system means that Situ is not limited by the volume of the data since more nodes can be added to handle a greater load.

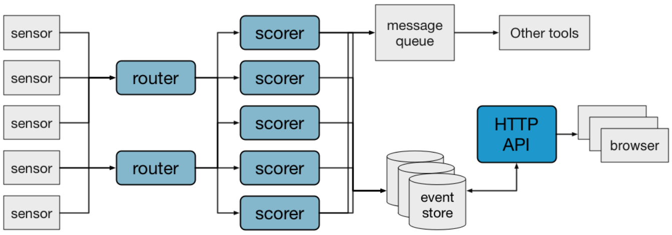


Fig. 2. The architecture of Situ's streaming anomaly detection system.

The web-based visualization system consists of an HTTP API and a visual interface that runs in the analyst's browser.

4.3 Streaming Anomaly Detection System

Here we describe the event processing for Situ's anomaly detection.

4.3.1 Ingestion and Parsing

Situ has a versatile set of data ingestion options, including: reading directly from a computer's network interface or a *pcap* file [51], reading from a networking device via *NetFlow v9* [32] or *IPFIX* [33], reading from a network flow collection tool [46], or reading from one of several message queues, including *Nanomsg* [42], a brokerless queue, and several brokered queues, including *Apache Kafka* [19], *Nats* [43], and *RabbitMQ* [45]. This flexibility makes it possible to integrate Situ with different workflows and different log collection infrastructures. For message queues, parsing input messages is specified in a comma separated value (CSV) or regular-expression based configuration file. Of these options, the most scalable and most widely used with Situ are Argus for network traffic related data and Kafka for all other data types.

4.3.2 Enrichment

After parsing the data, Situ enriches events to provide analysts with additional context, addressing requirement **ADR2-Contextualizing events**. The system enriches events with the country of the IPs, tagging IPs on blacklists, adding asset metadata for internal IPs, and assigning a role based on port activity for each internal IPs.

Because analysts often look up the country of origin for the IPs involved in the event, Situ will automatically attempt to determine the country in which each IP address is located. Situ automates this step in the investigation by providing context with the event. Using a configurable set of blacklists (lists of known malicious IP addresses), Situ will automatically tag events that include these malicious IP addresses. This enrichment information provides insight into the maliciousness of an event. It can also point to the type of attack being executed. Using an enterprise's asset configuration database, Situ can be configured to automatically label the owner of a device, the protection zone the device is in, and additional contextual information about assets that is available. This information allows analysts to prioritize the events that are investigated based on the importance of the assets and protection zone that may be compromised.

In most large environments, it is nearly impossible for cyber security analysts to know what roles a system plays, making it difficult to achieve situational awareness and adequately diagnose or prioritize an attack. For example, an intrusion alert describing an attack against a workstation would probably be lower priority than one against an organization's primary domain controller, but without knowing the role of a machine it would be impossible to make this determination. Additionally, it may be useful to know which systems are performing more than one role in the enterprise, which could be important for resource planning and security. For example, it would be important to know that a system is operating both as a web server and as a DNS server because this would not be the best security posture. This kind of labeling of asset roles is used to enrich the data within Situ.

We have developed a library that builds temporal behavioral models from flow data [31]. Internal IP addresses are clustered based on historic network port usage for a specific range of time. The HDBScan clustering algorithm is used to create IP groups based on their port behaviour. Also included in these clusters are role labels for known, commonly used roles, including web server, mail server, domain name server, and SSH server. We build these behavior models offline and periodically update them. Situ uses these models to label any internal IPs in a streaming event with a set of roles. Investigations into highly anomalous events can be prioritized based on the importance of the machines involved.

4.3.3 Anomaly Scoring

After events are ingested, parsed, and enriched, the system will score each event according to the internal IP addresses (IPs within the enterprise) within that event. Each event is scored according to all applicable behavior models, described below, using the same algorithm.

Algorithm For each observation (e.g., flow, IP per time-window) we extract a set of statistics of interest and perform anomaly detection for each statistic. This gives a multi-faceted view of each event but uses single-feature detectors that are understandable to operators.

For anomaly scoring, we estimate a multinomial distribution from previously observed data, compute the p-value of newly observed data, and update the multinomial to accommodate these new observations. Initially, multinomials are given a uniform distribution of one count per bin, and a standard Bayesian update is performed upon receipt of new data. Notationally, using bins $i = 1, ..., k$, we set $f_0(i) = 1/k$. Upon receipt of a new, say n^{th} observation, x_n, we compute the p-value; $pv_{f_n}(x_n) = \sum f_n(i)$, with sum over $\{i \in 1, ..., k : f_n(i) \leq f_n(x_n)\}$. Next, the model f_n is amended to accommodate the new observation. We obtain f_{n+1} by incrementing both the total observation count (denominator) and the count of the x_n's bin (numerator). In short, we use a multinomial distribution and a uniform prior, then iteratively compute the maximum a posteriori (MAP) estimate.

The anomaly score given to each event, x (observed statistic) is defined as $-\log_{10}(pv(x))$, so high p-value events are given scores near 0 and low p-value events receive large scores (e.g., a score of 6.0 indicates a one-in-a-million event). Unlike the direct use of event probability, the p-value captures relationships among event probabilities. This enables computation of a threshold amenable to online updating.

P-value anomaly scoring is chosen because it satisfies several of our requirements:

1. **Understandable scores**: In support of **ADR1**, because multinomials are essentially histograms, they are straightforward to explain to non-mathematicians. Analysts can easily understand that a high anomaly score indicates the bin for that feature is small compared to other bins in past behavior.

2. **Comparable scores**: Supporting **ADR3**, P-values are comparable across distributions. For example, a one-in-a-million event is a meaningful score regardless of the distribution. As our application involves distributions of heterogeneous data and distributions that change over time, quantifiable comparability is needed for operators to prioritize anomalous observations.

3. **Fast notification**: Multinomials are simple data structures, but this simplicity leads to very fast p-value computations and updates, supporting **ADR4**.

4. **Scalability**: Because p-values are computationally fast, more operations can occur in a time period, facilitating scalability, **ADR5**.

There are additional advantages of this approach, including:

- P-values are regulatable, meaning, the expected number of events with an anomaly score over a set threshold α is computable a prioi. This allows operators to set a single threshold for the many evolving detectors to prevent flooding a downstream system.

- Model drift is detectable by comparing the expected number of high p-value events to the observed number.

See our previous works for mathematical theorems and empirical verification of the p-value anomaly scoring advantages [5, 13–15].

While admittedly simple, multinomials are robust when fit to a large number of observations, provide very fast p-value computations and updates, and are easily visualized and understood; hence, they are an ideal choice for our setting.

Behavior Models The Situ system creates several behavior models for each internal system in the enterprise. These contexts are updated as each flow event is processed. Each of these contexts has a temporal analogue that maintains a separate model for workday hours, evening hours, and weekend hours. Having multiple contexts not only increases the coverage of attack vectors that the system can identify, but also helps analysts answer the question of *why* something is anomalous, helping to satisfy our requirement of *ADR1-Understandable scores*. If an analyst knows which behavior model was most anomalous, it can help them discover the reason for the anomalousness.

The *bytes per packet* context models a system's typical byte quantity per packet. A desktop system used primarily for web browsing will mostly create small packets for the HTTP(S) request and receive large packets containing the response. If a system like this starts generating many large outbound packets, which may indicate exfiltration of data, this model will recognize it as anomalous.

The *DNS request rate* context models the recent quantity of DNS requests a system has made. This context is primarily designed to recognize a BotNet infection. Typically, when a system is infected with a BotNet, it begins to generate many DNS requests as the BotNet tries to locate a command and control system.

The *non-ephemeral traffic* context models the amount of traffic inbound to and outbound from non-emphemeral, non-privileged ports. This context tracks ports 1024 to 32767 since most Linux distributions use ports greater than or equal to 32768 as ephemeral, despite the Internet Assigned Numbers Authority (IANA) standard of 49152. It is designed to recognize when a system begins to interact with some non-privileged service on another system or begin running a non-privileged server that other systems are interacting with.

Similar to the non-ephemeral context, the *privileged port traffic* context models the amount of traffic inbound to privileged ports (less than 1024) and the amount of traffic outbound to privileged ports. This is based on our previous work [30], which showed that the role of a host can be characterized by the use of private ports in flow data. This context represents all of the privileged network services that a system provides, as well as all of the privileged network services it interacts with. The intent of this model is to recognize any change in behavior in how a system is communicating with its peer's privileged services, and to recognize when any new privileged services are started.

The *producer-consumer ratio* context models the amount of network traffic a system produces relative to how much it consumes defined as $(source\ bytes - destination\ bytes)/(source\ bytes + destination\ bytes)$. Typically, a system in a server role will be mostly a producer, and a desktop system will mostly be a consumer. This context is intended to recognize that a system is shifting roles from its previous behavior. A system switching from mostly a consumer toward being a producer could possibly indicate the exfiltration of data.

Similar to the privileged port context, the *privileged port bytes per packet* context models a systems typical bytes per packet for each privileged port. This context is intended to recognize a situation like if a DNS server (which normally receives small packets on port 53) starts to receive large packets, or if a web server (which mostly receives small request packets) starts receiving large requests. Either of these situations may indicate attempts to transfer data in a concealed manner.

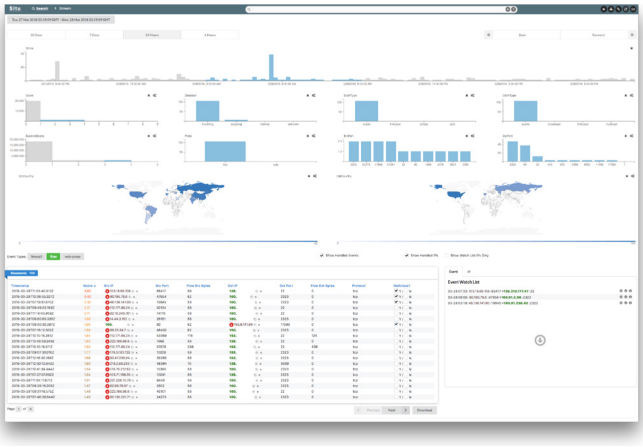


Fig. 3. *Event Search Page* showing the details of an event, relevant temporal context, and the raw data.

4.3.4 Data Output

Similar to the flexibility of data ingestion, Situ has several options for outputting scored events. These include the same message queues that can be used for ingesting data, Nanomsg, Kafka, Nats, and RabbitMQ, as well as several data stores, including Elasticsearch, an open-source distributed key-value store [8], and Splunk, a commercial log aggregation system [49].

4.4 Visualization System

After data is streamed through Situ and output into a data store, an API exposes the data to users via an HTTP/JSON API. While the anomaly score allows an analyst to determine outliers from normal behavior, a score without context is not necessarily enough to determine if an anomalous event is also malicious. The visualization provides the ability to filter events based on characteristics of the underlying data and to see the context of an event. The goal of the visualization is to enable analysts to quickly characterize anomalous events.

IP addresses are special in network security data; they represent the *who* in the cyber domain. There are two fundamentally different kinds of IP addresses: *internal*, those within the enterprise network that are being protected, and *external*, which is the rest of the IP space and represents potential attackers. Throughout the visualization, IP addresses are differentiated by color to meet the design requirement of **SVR4-Centering on enterprise assets**.

Situ's visual interface includes multiple pages to support different functionality and types of analysis. The goal was to support both overview and detail to enable a sense of high-level context as well as details on demand. The *Event Search Page* provides an overview of network data and supports filtering or selection of specific data items, and the *Event Detail Page* and *IP Detail Page* allow inspection of details of items of interest.

4.4.1 Common features

In addition to the specialized page views, some common features are always available regardless of the page the user is on. A search bar is available for filtering data that will auto-complete field names and values, and it also allows complex boolean searches beyond what clicking selections in the charts allow. There are also actions available that allow downloading data or copy the application's URL (as all application state changes are reflected in the URL). The former facilitates the design requirements of **SVR5-Tool integration** and **SVR3-Access to raw data**—data can be downloaded as JSON and analysts can use their command line tools that they currently use. The latter facilitates the design requirement of *collaboration*, as it allows the visualization's state to be saved, shared, and stored in an existing system and retrieved by other analysts at a later date.

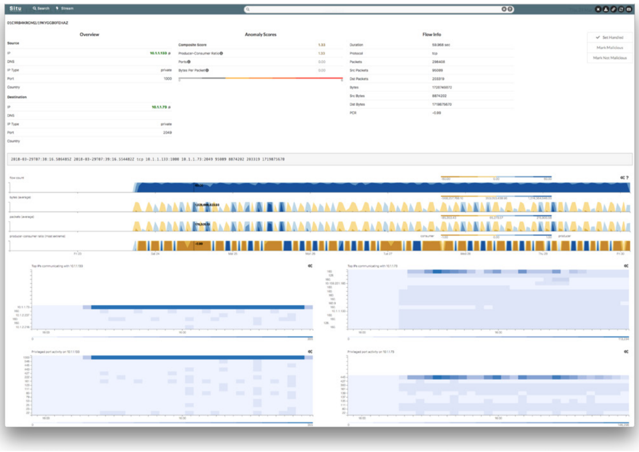


Fig. 4. *Event Detail Page* showing the details of an event, relevant temporal context, and the raw data

4.4.2 Event Search Page

The *Event Search Page*, shown in Fig. 3 presents the user with a visual overview of the data via a collection of scented widgets [53] that show the data distribution of the most important fields while providing a way to quickly filter data in multiple coordinated views. A temporal histogram at the top of the page allows initial selection of a time range of interest. Additionally, a series of bar charts/histograms are provided for additional user-defined fields, and map visualizations show countries.

These data summaries are intended to meet the design goal of **SVR2-Scalability** by summarizing the data while also allowing users to quickly drill into events of interest. For example, incoming events with high anomaly scores from the past 30 minutes and bytes greater than zero would show incoming traffic that was not blocked at the firewall. At the bottom of the page are the individual events shown in a sortable table. The selections in the bar charts/histograms determine which individual events are shown in the table. Data shown in the table can also be downloaded, which addresses the **SVR5-Tool integration** and **SVR3-Access to raw data** requirements.

Finally, there are *watchlists* at the bottom right that allow the user to save and share events and IP addresses of interest. This can be useful for watching for suspicious or known-malicious IP addresses. It can also be used to share lists of malicious IP addresses that have been discovered as part of the analytic process with other users. These are designed to facilitate the visualization design requirement of **SVR6-Collaboration**, which is specifically cited by one user in 5.4.

4.4.3 Event Detail Page

When the user clicks on an event in the event table on the *Event Search Page*, the user is taken to the *Event Detail Page*, as shown in Fig. 4, which provides multiple visualizations and tools intended to meet the design requirements. Horizon graphs of several flow fields and heatmaps of IP addresses support **SVR1-Temporal context** to the event. These visualizations prioritize showing trends and patterns since this is most important for context. Additional meta-data such as DNS names and countries of IPs supports **ADR2-Contextualizing events**. The raw data is explicitly highlighted in the middle of the page, in support of **SVR3-Access to raw data**. Additionally, buttons to support **SVR5-Tool integration** allow users to query their other tools and APIs (e.g., SANS, WatchGuard) to find more information about remote IP address; additional tools can be added through a simple addition to configuration. To facilitate **SVR6-Collaboration**, users can mark an event as *handled* to signal to each other who is working or completed working on what. Also, anomaly scores are broken down by their behavior models to facilitate the requirement of **ADR1-Understandable scores**.

4.4.4 IP Detail Page

When the user clicks on an IP address in the event table on the *Event Search Page* or on an IP in the *Event Detail Page*, the user is taken to the *IP Detail Page*, as shown in Fig. 1. Meta-data at the top of the page show the DNS name and other information to support **ADR2-Contextualizing events**, the temporal histogram and horizon graphs provide support for **SVR1-Temporal context**, and other features on the *Event Search Page* are also shown on this page to meet those same requirements. There is also an IP graph shown here that shows the IP of interest at the center, and the IP addresses that the IP has communicated with (meeting the selected filters) arranged on a ring nearest the center, and then the IP addresses those IPs communicated with on the outer ring. This egocentric graph layout provides a familiar presentation for experts while prioritizing communications with the selected IP. This supports **ADR2-Contextualizing events** to facilitate understanding of communication patterns and highlight suspicious activity, like a remote IP (red nodes) communicating with a lot of internal IPs (green nodes).

5 Evaluation

In this section, we present several evaluations: an evaluation of the algorithm and behavior models to determine if the anomalies in available test data are malicious, two case studies presenting a synthetic scenario with known attacks and a real-world scenario, and feedback from analysts using Situ in production at a security operations center (SOC).

5.1 Anomaly Detection Algorithm Evaluation

We evaluated the Situ system using the 5s12 naive attack and 5s20 multiple stepping stones attack scenarios from the Skaion 2006 IARPA Dataset [1], a synthetic cyber attack data set. The Skaion data used for this case study was "generated by capturing information from a synthetic environment, where benign user activity and malicious attacks are emulated by computer programs."

For both of these scenarios, the attack data was integrated into the provided background data with approximately two hours of offset so that Situ would be able to observe benign traffic before the attack begins. These integrated pcap files were then processed into network flows using the Argus flow tool. The resulting flows were ingested by Situ, and the top 1,000 highest anomaly scored flows were recorded. The portion of these flows involving the known attacking systems were computed. See Table 2 for the test results. For the two scenarios tested, 92.5% and 97.8% of the most anomalous traffic was part of the attack scenario.

While the "not part of attack" count in the table is somewhat analogous to a false positive rate typically reported by intrusion detection systems, it is important to note that these concepts are not identical. The flows here that are not part of the attack may not be malicious in the attack scenario, but they are still anomalous—they may be a misconfiguration in the test network or an artifact of the test itself. In reality, an analyst or operator would likely want to know about such misconfigurations that may not represent an attack but could be a future security threat. This is where the visual analysis would come in.

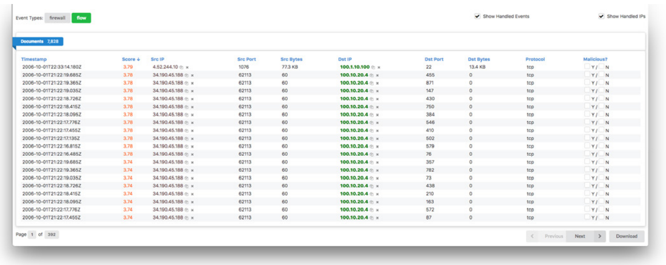
Attack Scenario	Part of known attack	Not part of attack
5s12	925	75
5s20	978	22

Table 2. Test Results.

5.2 Case Study 1: Skaion Data Set

To evaluate the performance of the Situ visualization tools in analyzing network flow data, the background data and attack data consisting of approximately 800,000 flows for the Skaion 5s20 attack scenario were visually examined. This attack involves multiple attackers who attempt, and in some cases succeed, in compromising a host inside the target network. Once they have succeeded, they try to move laterally through the network.

The benefit of Situ is the integration of the automated analytics and the visualization to enable analysts to quickly filter out less interesting data and understand the most anomalous data. An analyst may start their search for attacks by taking advantage of the analytics in the



Event Types: firewall flow

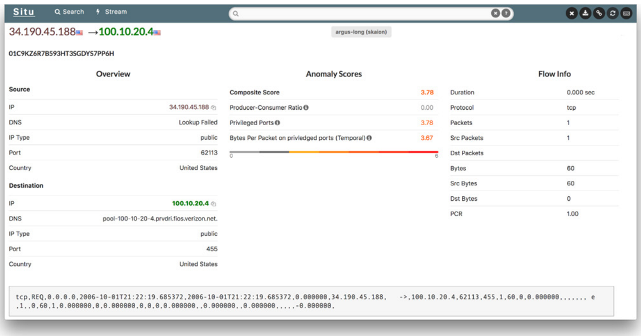
☑ Show Handled Events ☑ Show Handled IPs

Documents 7,828

Timestamp	Score ↓	Src IP	Src Port	Src Bytes	Dst IP	Dst Port	Dst Bytes	Protocol	Malicious?
2006-10-01T22:33:14.180Z	3.79	4.52.244.10	1076	77.3 KB	100.1.10.100	22	13.4 KB	tcp	□ Y / □ N
2006-10-01T21:22:19.685Z	3.78	34.190.45.188	62113	60	100.10.20.4	455	0	tcp	□ Y / □ N
2006-10-01T21:22:19.365Z	3.78	34.190.45.188	62113	60	100.10.20.4	871	0	tcp	□ Y / □ N
2006-10-01T21:22:19.035Z	3.78	34.190.45.188	62113	60	100.10.20.4	147	0	tcp	□ Y / □ N
2006-10-01T21:22:18.726Z	3.78	34.190.45.188	62113	60	100.10.20.4	430	0	tcp	□ Y / □ N
2006-10-01T21:22:18.415Z	3.78	34.190.45.188	62113	60	100.10.20.4	750	0	tcp	□ Y / □ N
2006-10-01T21:22:18.095Z	3.78	34.190.45.188	62113	60	100.10.20.4	384	0	tcp	□ Y / □ N
2006-10-01T21:22:17.776Z	3.78	34.190.45.188	62113	60	100.10.20.4	546	0	tcp	□ Y / □ N
2006-10-01T21:22:17.455Z	3.78	34.190.45.188	62113	60	100.10.20.4	410	0	tcp	□ Y / □ N
2006-10-01T21:22:17.135Z	3.78	34.190.45.188	62113	60	100.10.20.4	502	0	tcp	□ Y / □ N
2006-10-01T21:22:16.815Z	3.78	34.190.45.188	62113	60	100.10.20.4	579	0	tcp	□ Y / □ N
2006-10-01T21:22:16.485Z	3.78	34.190.45.188	62113	60	100.10.20.4	76	0	tcp	□ Y / □ N
2006-10-01T21:22:19.685Z	3.74	34.190.45.188	62113	60	100.10.20.4	357	0	tcp	□ Y / □ N
2006-10-01T21:22:19.365Z	3.74	34.190.45.188	62113	60	100.10.20.4	782	0	tcp	□ Y / □ N
2006-10-01T21:22:19.035Z	3.74	34.190.45.188	62113	60	100.10.20.4	73	0	tcp	□ Y / □ N
2006-10-01T21:22:18.726Z	3.74	34.190.45.188	62113	60	100.10.20.4	438	0	tcp	□ Y / □ N
2006-10-01T21:22:18.415Z	3.74	34.190.45.188	62113	60	100.10.20.4	210	0	tcp	□ Y / □ N
2006-10-01T21:22:18.095Z	3.74	34.190.45.188	62113	60	100.10.20.4	163	0	tcp	□ Y / □ N
2006-10-01T21:22:17.776Z	3.74	34.190.45.188	62113	60	100.10.20.4	572	0	tcp	□ Y / □ N
2006-10-01T21:22:17.455Z	3.74	34.190.45.188	62113	60	100.10.20.4	87	0	tcp	□ Y / □ N

Page 1 of 392

Fig. 5. Events filtered in the *Event Search Page* to show only incoming traffic with the highest anomaly scores.



34.190.45.188 →100.10.20.4

argus-long (skaion)

01C9KZ6R7B593HT3SGDYS7PP6H

Overview

Source	
IP	34.190.45.188
DNS	Lookup Failed
IP Type	public
Port	62113
Country	United States

Destination	
IP	100.10.20.4
DNS	pool-100-10-20-4.prvdri.fios.verizon.net.
IP Type	public
Port	455
Country	United States

Anomaly Scores

Composite Score	3.78
Producer-Consumer Ratio	0.00
Privileged Ports	3.78
Bytes Per Packet on privledged ports (Temporal)	3.67

Flow Info

Duration	0.000 sec
Protocol	tcp
Packets	1
Src Packets	1
Dst Packets	
Bytes	60
Src Bytes	60
Dst Bytes	0
PCR	1.00

```
tcp,REQ,0.0.0.0,2006-10-01T21:22:19.685372,2006-10-01T21:22:19.685372,0.000000,34.190.45.188,   ->,100.10.20.4,62113,455,1,60,0,0.000000,,,,,,, e
,1,,0,60,1,0.000000,0,0.000000,0,0,0,0.000000,,0.000000,,0.000000,,,,,-0.000000,
```

Fig. 6. Details of an event's context help analysts discover why the event was anomalous.

visualization by filtering out the events the system has determined as normal (i.e. low anomaly scores), as shown in Fig. 5 in the *Event Search Page*. Using the temporal histogram the analyst can see where in time the attack approximately began. By selecting the the spike in anomalous traffic, the analyst filters down the network flow data to a more manageable list of events Fig. 5. Once the analyst has filtered the search view to their liking, they may select an event to show the *Event Details Page*, as shown in Fig. 6.

The *Event Details Page* provides the analyst with context about why an event scored as highly anomalous, supporting the **ADR2-Contextualizing events** requirement. This page displays the results of the enrichment and scoring process. In this case, the *privileged ports* model scored the highest, as the external IP was communicating to port 445. To support **SVR1-Temporal context** about the communication patterns between hosts, the page shows several heatmaps to show what other IPs the source and destination IPs have recently communicated with. In this case, the heatmaps indicate that the external IP has been communicating to a handful of internal hosts, and the IP has been talking to many privileged ports on those hosts. Clicking on the external IP in the event brings up the *IP Details Page*, as shown in Fig. 7.

Here, the analyst is presented with an IP graph, as described in 4.4.4, to show the context of communications. By using edges and the color of the nodes, the analyst can see which nodes an external node communicated with. Clicking on any of the nodes shifts the view to that of the selected node, allowing the analyst to search for lateral movement. By correlating the IP graph with the raw pcap logs, it is possible to determine that the host in Fig. 7 34.190.45.188 either compromised or attempted to compromise all the internal nodes on the innermost ring of the IP graph.

5.3 Case Study 2: Real-world Use Case

Here, we walk through a real-world example demonstrating how Situ is being used in production at a large (5000 users) organization's Security Operations Center (SOC), which utilizes Situ as one of the tools they use on a daily basis. As in the previous case study, network flow data is available for analysts to search. In addition to flow data, Situ also processes data from Cisco ASA firewall logs. These firewall logs have many fields that are not available in the network flow data, but IPs and ports are still a prominent feature. In incident response, Tier 1

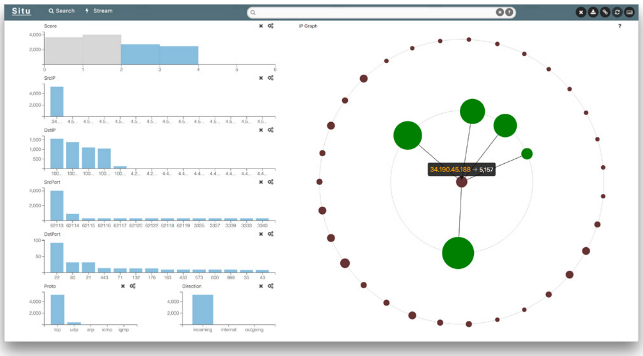


Fig. 7. Part of the *IP Details Page*, showing the IP graph of hosts communicating with anomalous IP.

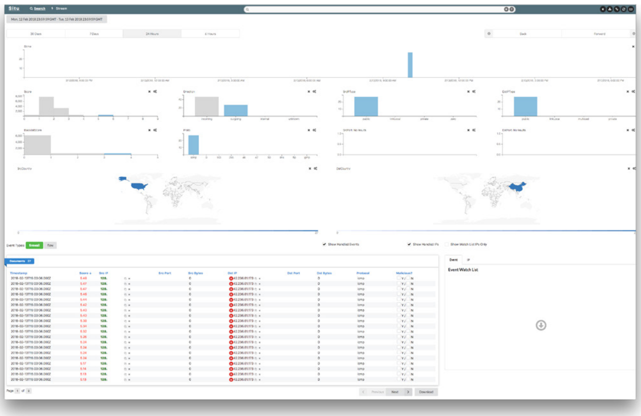
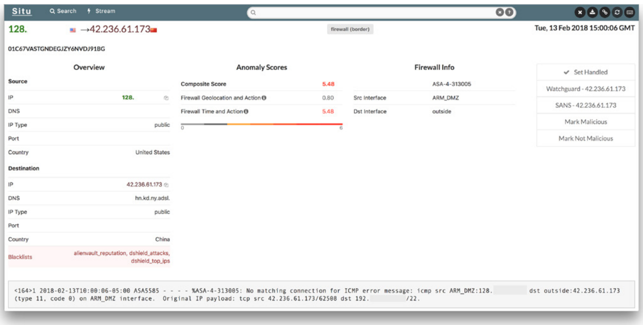


Fig. 8. *Event Search Page* showing only blacklisted IPs with high anomaly scores.



128. →42.236.61.173 firewall (border) Tue, 13 Feb 2018 15:00:06 GMT

01C67VASTGNDEGJZY6NVDJ91BG

Overview

Source	
IP	128.
DNS	
IP Type	public
Port	
Country	United States
Destination	
IP	42.236.61.173
DNS	hn.kd.ny.adsl.
IP Type	public
Port	
Country	China
Blacklists	alienvault_reputation, dshield_attacks, dshield_top_ips

Anomaly Scores

Composite Score	5.48
Firewall Geolocation and Action	0.80
Firewall Time and Action	5.48

0 6

Firewall Info

	ASA-4-313005
Src Interface	ARM_DMZ
Dst Interface	outside

✓ Set Handled
Watchguard - 42.236.61.173
SANS - 42.236.61.173
Mark Malicious
Mark Not Malicious

```
<164>1 2018-02-13T10:00:06-05:00 ASA5585 - - - - %ASA-4-313005: No matching connection for ICMP error message: icmp src ARM_DMZ:128.[redacted] dst outside:42.236.61.173
(type 11, code 0) on ARM_DMZ interface.  Original IP payload: tcp src 42.236.61.173/62508 dst 192.[redacted]/22.
```

Fig. 9. *Event Details Page* showing the anomalous firewall log data.

analysts typically are the first-level responders that focus on quickly triaging events, whereas Tier 2 analysts perform more in-depth analysis, and Tier 3 analysts focus on more sophisticated investigations. The following is based on a real event described to us by Tier 1 and Tier 3 analysts.

The Tier 1 analyst uses Situ daily to inspect the anomalous events identified by the analytics that may not be caught by other commercial tools, which primarily consist of a commercial intrusion prevention systems that utilizes rules derived from the vendor's experts and crowd-sourced feedback collected from their customers. The analyst looks at firewall logs for events with high anomaly scores that have also been flagged as being on multiple blacklists from the enrichment process. The analyst visually sorts the data based on the following criteria: logs with high anomaly scores (5-6), IPs on more than two blacklists, and communications going out of the enterprise (see Fig. 8; note that internal IP addresses have been redacted). The IPs in this list all have anomaly scores indicating that this type of event only occurs, on average, once in 100,000 events (scores around 5 in a log-10 scale). Additionally, the analyst notes that the destination IP has been flagged as being on three blacklists. Clicking on a row with a high score brings up the *Event Details Page*, shown in Fig. 9.

The event details for this firewall log indicate that the *firewall time and action* model is the reason this event has been flagged with such a high anomaly score. At this point, the analyst uses an internal host look-up system to ascertain that the IP communicating outbound is a router. Since the router should not be communicating to a blacklisted IP, the analyst opens a ticket so a Tier 3 analyst can investigate further.

The Tier 3 analyst uses the integrated tools embedded in the Situ visualization to check the reputation of the external IP by clicking the relevant buttons, which execute queries to third party reputation sites that the analysts already use in their daily workflow. Adding new integrations is only a matter of adding a line in the configuration file. Providing tight integration with third-party tools supports **SVR5-Tool integration** to allow anomalous events to be more quickly characterized as malicious or benign and situates our tool within analysts' current workflows.

Seeing that the IP has a malicious reputation, the Tier 3 analyst then uses a commercial packet capture collection tool to pull the raw pcap data for the associated IPs. (In future work, we can integrate such pcap data directly into the Situ visualization.) By analyzing the pcap data, which is predominantly ICMP traffic, the analyst notices that the communications have a decreasing *Time To Live* (TTL) value. This condition indicates that the external IP is trying to map the organization's IP space with a tool such as a traceroute. This is rarely legitimate and is often part of the reconnaissance phase of an attack. The analyst also notes that the firewall is blocking the return traffic. Since the return traffic is being blocked, the Tier 3 analyst closes the ticket.

Although this event was already being blocked by firewall rules, the event was not flagged by the intrusion prevention system. Comments from the the Tier 1 analyst indicate that the only way this event would have been found is to manually search through firewall logs. The Tier 1 stated that finding the event provides the SOC with an awareness of potential bad actors and their tactics.

This case study demonstrates the benefit of visual analytics in combining the power of analytics to focus analysts' attention to the most atypical events and visualization to provide the visual context to understand those events within a real-world analyst's workflow.

5.4 Domain Analyst Feedback

As noted in Sect. 5.3, Situ is installed at a large organization in production as part of the SOC's daily tasking. The system ingests network flows (approximately 400 million flows per day) and firewall logs (approximately 1 billion events per day). This provides some indication of the scalability of the system; running on a small cluster of 6 nodes Situ processes an average of 16,000 events per second. In addition to drilling into the specific use case described by the same analysts in the previous section, we also observed five analysts—three of which are Tier 1 analysts and two are Tier 2—from the SOC using the tool. The analysts are experts with experience ranging from 2 to 10 years in network security. Observations were conducted over a period of six months in multiple sessions (approximately one hour each). We also solicited analyst feedback over email over a 12 month period.

In general, the analysts thought Situ filled a gap in the existing commercial security tools; specifically complementing their rule-based intrusion prevention system and their block lists on the border firewall. A Tier 1 analyst (primarily concerned with Triage) said:

> "Situ has been used to detect abnormal exfiltration of data, including by authorized and unauthorized users."

Another analyst (Tier 3), who primarily looks for malicious traffic that automated intrusion prevention systems have not found, reported:

> "Instances of erroneous IP traffic can be detected by Situ".

In both of these cases, we observed of analysts that Situ identified certain traffic to be anomalous that their intrusion prevention system and blacklists failed to flag. In these cases, the analysts were able to confirm that the anomalous traffic was also malicious by looking up the IP address of the source of the attack on a trusted blacklist of known malicious IP addresses.

The biggest downside the users revealed is inherent in any anomaly detection system—not all anomalies are malicious. This takes getting used to as they are much more used to looking at 'alerts' from an intrusion detection system that suggests something is malicious. We expect this to be a potential barrier to adoption of anomaly detection systems and was overcome in our deployments only through repeated interactions in which we explained the potential value and we integrated their feedback into the tool.

Analysts also thought the customizability of the visualization tool was key to its utility. A Tier 3 analyst commented:

> "The ability to customize Situ for each search is vital in narrowing down the parameters to detect specific anomalous network traffic."

This same analyst also called attention to specific features:

> "The Event/IP watchlist feature is invaluable as it allows us to keep track of interesting—although not necessarily malicious—IPs over time to identify anomalous traffic patterns."

This feature came as a result of the researchers' experience working with cyber security analysts and observing that analysts wanted to track certain IPs that they either believed were suspicious or knew to be malicious. Similar features have also been reported previously [24, 25].

The importance of collaboration in IT work in general [27] and in cyber security in particular [21] has long been reported, but many existing tools ignore this fact. Situ emphasizes sharing state to make it easy to integrate into existing ticketing systems. Referring to Situ's collaboration support, a Tier 3 analyst said:

> "The unique ability to embed saved searches and dashboards into URLs as easy bookmarks is a great optimization that makes sharing routine tasks dramatically easier."

We observed analysts copying the URL and pasting it into their ticketing system in order to retrieve the state that led them to create the ticket and to share the ticket with higher tier analysts. This facilitates collaboration and communication, per the **SVR6-Collaboration** requirement.

Research has also demonstrated the need to integrate with existing tools and data sources, per the visualization requirement **SVR5-Tool integration**. A Tier 1 analyst commented:

> "Integration with other data sources allows for correlation of traffic and minimizes the amount of time required for an analyst to search across multiple platforms for significant events."

While the analysts appreciated the ability quickly filter data by interacting with the summary visualizations and found the IP graph useful in understanding communication patterns, they did not understand the utility of the horizon graphs and heatmaps, at least initially. When we asked why they were not using these, we realized they did not understand what they were showing. After explaining these and integrating pictorial help screens, several expressed that they could see their utility, but we did not observe them interacting with these views. We expect this is, at least in part, because as Tier 1 and 2 analysts they are more focused on researching an event, rather than trying to discover trends or new behavior as a Tier 3 analyst would.

6 Conclusion

This paper presents Situ, a streaming anomaly detection system and visualization for discovering and explaining suspicious behavior in computer network traffic and logs. The contributions of the paper include a description of functional requirements for such a system, the anomaly scoring algorithm and models, the interactive visualization with integrated tooling, case studies that walk through two scenarios demonstrating the utility of the system, and feedback from analysts in a real-world production deployment of the tool. The system design uses multiple linked views and pages to support overview first and details on demand to better understand both the data and the context behind system's anomaly detection. The *Event Search Page* first summarizes the anomaly detection results and provides an overview of network activity, and the scented widgets allow analysts to select specific items of interest for further inspection. To accommodate scalability, the widgets, horizon graphs, and heatmaps prioritize an overview of trends to emphasize context at the expense of numerical accuracy (which is less important for assessing trends and patterns).

Situ is currently deployed in two real-world deployments. In the first, as described in section 5.4, the system is installed at a large organization's SOC and ingests about 16,000 events per second. We have an ongoing relationship with the analysts in the SOC and continue to solicit their feedback and ideas as they use the data. This feedback is used to evolve feature designs into an innovative system that assists analysts in an otherwise daunting task. There is also a second installation of Situ in the same organization, but within a different group that manages a supercomputer. This group uses Situ to score flows from a network device and store the flows in a Splunk instance.

Feedback from analysts demonstrates the importance of end users not only having access to the results of analytics, but also that they have the means to understand those results via intuitive visualizations. Visual analytics relies on the integration of both quality human judgment and machine automation; our experience with domain experts emphasizes the importance of understanding in order for a system to be successful and relied upon in the real world. Our evaluations revealed that Situ's multiple views do help analysts to better understand the system. Our evaluations also indicate the importance of the experts understanding the visualizations, which suggests that a combination of simple visualization designs will often be preferred over more advanced interfaces. Therefore, for practical installation of visual analytics systems in operational settings, understandability of familiar visual representations is just as important as understandability of the algorithmic support.

Situ is a visual analytics system that complements existing security tools and helps analysts gain situation awareness, identify suspicious behavior, and understand the behavior's context. It is an exemplar of the type of system that is needed to meet the escalating cyber security challenges against today's network environments.

Acknowledgments

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the US Department of Energy. The US Government retains and the publisher, by accepting the article for publication, acknowledges that the US Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

Research sponsored by the Laboratory Directed Research and Development Program of ORNL, managed by UT-Battelle, LLC, for the U.S. DOE. This work was supported by the Dept. of Homeland Security Science & Technology Directorate, HSARPA, Cyber Security Division under the Transition to Practice program. This research is also supported in part by the DARPA XAI program under Grant N66001-17-2-4031. The data referenced in this paper was created by Skaion Corporation with funding from IARPA.

REFERENCES

[1] Skaion 2006 IARPA dataset. 2006. doi: 10.23721/112/1354736

[2] T. Ahmed et al. Multivariate online anomaly detection using kernel recursive least squares. In *26th INFOCOM*, pp. 625–633. IEEE, 2007.

[3] R. Ball, G. A. Fink, and C. North. Home-centric visualization of network traffic for security administration. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, VizSEC/DMSEC '04, pp. 55–64. ACM, New York, NY, USA, 2004. doi: 10.1145/1029208.1029217

[4] D. M. Best, S. Bohn, D. Love, A. Wynne, and W. A. Pike. Real-time visualization of network behaviors for situational awareness. In *Proceedings of the seventh international symposium on visualization for cyber security*, pp. 79–90. ACM, 2010.

[5] R. A. Bridges, J. D. Jamieson, and J. W. Reed. Setting the threshold for high throughput detectors: A mathematical approach for ensembles of dynamic, heterogeneous, probabilistic anomaly detectors. In *2017 IEEE International Conference on Big Data (Big Data)*, pp. 1071–1078. IEEE, Boston, MA, USA, Dec 2017. Extended version `https://arxiv.org/abs/1710.09422`. doi: 10.1109/BigData.2017.8258031

[6] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016.

[7] 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. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 49(3):229–233, 2005. doi: 10.1177/154193120504900304

[8] Elasticsearch. Elasticsearch: Restful, distributed search & analytics — elastic, 2018. [Online; accessed: 2018-03-29].

[9] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human factors*, 37:32–64, 1995.

[10] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, and P. Dokas. Minds-minnesota intrusion detection system. *Next generation data mining*, pp. 199–218, 2004.

[11] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pp. 77–101. Springer, 2002.

[12] E. Ferragut et al. Automatic construction of anomaly detectors from graphical models. In *CICS*, pp. 9–16. IEEE, 2011.

[13] E. Ferragut et al. A new, principled approach to anomaly detection. In *ICMLA*, vol. 2, pp. 210–215. IEEE, 2012.

[14] E. Ferragut et al. Detection of anomalous events, June 7 2016. US Patent 9,361,463.

[15] E. Ferragut et al. Real-time detection and classification of anomalous events in streaming data, Apr. 19 2016. US Patent 9,319,421.

[16] G. A. Fink, C. L. North, A. Endert, and S. Rose. Visualizing cyber security: Usable workspaces. In *2009 6th International Workshop on Visualization for Cyber Security*, pp. 45–56, Oct 2009. doi: 10.1109/VIZSEC.2009.5375542

[17] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In *Proceedings of the 6th International COnference*, p. 8. ACM, 2010.

[18] C. for Strategic and I. Studies. Net losses: Estimating the global cost of cybercrime: Economic impact of cybercrime ii. Technical report, McAfee, June 2014.

[19] A. S. Foundation. Apache kafka, 2018. [Online; accessed: 2018-03-29].

[20] S. Garcia et al. An empirical comparison of botnet detection methods. *Comp. & Sec.*, 45, 2014.

[21] J. R. Goodall, W. G. Lutters, and A. Komlodi. I know my network: Collaboration and expertise in intrusion detection. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, CSCW '04, pp. 342–345. ACM, New York, NY, USA, 2004. doi: 10.1145/1031607.1031663

[22] J. R. Goodall, W. G. Lutters, and A. Komlodi. Developing expertise for network intrusion detection. *Information Technology & People*, 22(2):92–108, 2009. doi: 10.1108/09593840910962186

[23] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi. Focusing on context in network traffic analysis. *IEEE Computer Graphics and Applications*, 26(2):72–80, March 2006. doi: 10.1109/MCG.2006.31

[24] J. R. Goodall and M. Sowul. Viassist: Visual analytics for cyber defense. In *2009 IEEE Conference on Technologies for Homeland Security*, pp. 143–150, May 2009. doi: 10.1109/THS.2009.5168026

[25] J. R. Goodall and D. R. Tesone. Visual analytics for network flow analysis. In *2009 Cybersecurity Applications Technology Conference for Homeland Security*, pp. 199–204, March 2009. doi: 10.1109/CATCH.2009.47

[26] C. Gupta, S. Wang, I. Ari, M. Hao, U. Dayal, A. Mehta, M. Marwah, and R. Sharma. Chaos: A data stream analysis architecture for enterprise applications. In *Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on*, pp. 33–40. IEEE, 2009.

[27] E. Haber, E. Kandogan, and P. Maglio. Collaboration in system administration. *ACM Queue*, 8:10, 12 2010.

[28] M. Hao, D. A. Keim, U. Dayal, D. Oelke, and C. Tremblay. Density displays for data stream monitoring. In *Computer Graphics Forum*, vol. 27, pp. 895–902. Wiley Online Library, 2008.

[29] C. Harshaw et al. Graphprints: Towards a graph analytic method for network anomaly detection. In *11th CISRC*, pp. 15–19. ACM, 2016.

[30] K. Huffer and J. Reed. Situational awareness of network system roles (SANSR). In *12th CISRC*. ACM, 2017.

[31] K. M. T. Huffer and J. W. Reed. Situational awareness of network system roles (sansr). In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, CISRC '17, pp. 8:1–8:4. ACM, New York, NY, USA, 2017. doi: 10.1145/3064814.3064828

[32] I. E. T. F. (IETF). Cisco systems netflow services export version 9, 2004. [Online; accessed: 2018-03-29].

[33] I. E. T. F. (IETF). Specification of the ip flow information export (ipfix) protocol for the exchange of flow information, 2013. [Online; accessed: 2018-03-29].

[34] S. T. Ikram and A. K. Cherukuri. Improving accuracy of intrusion detection model using pca and optimized svm. *Journal of computing and information technology*, 24(2):133–148, 2016.

[35] M. S. Khan, S. Siddiqui, and K. Ferens. Cognitive modeling of polymorphic malware using fractal based semantic characterization. In *Technologies for Homeland Security (HST), 2017 IEEE International Symposium on*, pp. 1–7. IEEE, Waltham, MA, USA, 2017.

[36] C. Krügel et al. Service specific anomaly detection for network intrusion detection. In *Proc. Sym. Ap. Comp.*, SAC'02, pp. 201–208. ACM, New York, NY, USA, 2002. doi: 10.1145/508791.508835

[37] J. Lin, E. Keogh, S. Lonardi, J. P. Lankford, and D. M. Nystrom. Viztree: a tool for visually mining and monitoring massive time series databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 1269–1272. VLDB Endowment, 2004.

[38] A. Malviya, G. A. Fink, L. Sego, and B. Endicott-Popovsky. Situational awareness as a measure of performance in cyber security collaborative work. In *2011 Eighth International Conference on Information Technology: New Generations*, pp. 937–942, April 2011. doi: 10.1109/ITNG.2011.161

[39] F. Mansmann, M. Krstajic, F. Fischer, and E. Bertini. Streamsqueeze: a dynamic stream visualization for monitoring of event data. In *Visualization and Data Analysis 2012*, vol. 8294, p. 829404. International Society for Optics and Photonics, 2012.

[40] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. Liverac: interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1483–1492. ACM, 2008.

[41] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula. Efficient approaches for intrusion detection in cloud environment. In *Computing, Communication and Automation (ICCCA), 2016 International Conference on*, pp. 1211–1216. IEEE, Noida, India, 2016.

[42] Nanomsg. About nanomsg, 2017. [Online; accessed: 2018-03-29].

[43] Nats. Nats - open source messaging system, 2017. [Online; accessed: 2018-03-29].

[44] C. L. Paul. Human-centered study of a network operations center: Experience report and lessons learned. In *Proceedings of the 2014 ACM Workshop on Security Information Workers*, SIW '14, pp. 39–42. ACM, New York, NY, USA, 2014. doi: 10.1145/2663887.2663899

[45] Pivotal. Rabbitmq: Messaging that just works, 2018. [Online; accessed: 2018-03-29].

[46] QoSient. Argus: Auditing network activity, 2017. [Online; accessed: 2018-03-29].

[47] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou. Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 265–274. ACM, 2002.

[48] S. Song, L. Ling, and C. Manikopoulo. Flow-based statistical aggregation schemes for network anomaly detection. In *Networking, Sensing and*

Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on, pp. 786–791. IEEE, 2006.

[49] Splunk. Siem, aiops, application management, log management, machine learning, and compliance — splunk, 2017. [Online; accessed: 2018-03-29].

[50] M. Stolze, R. Pawlitzek, and A. Wespi. Visual problem-solving support for new event triage in centralized network security monitoring: Challenges, tools and benefits. In *IT-Incident Management & IT-Forensics - Erste Tagung der Fachgruppe SIDAR der Gesellschaft für Informatik, 24. - 25. November 2003 in Stuttgart, Deutschland*, p. 0, 2003.

[51] Tcpdump/Libpcap. Tcpdump/libpcap public repository, 2017. [Online; accessed: 2018-03-29].

[52] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov. Preparation, detection, and analysis: the diagnostic work of it security incident response. *Information Management & Computer Security*, 18(1):26–42, 2010. doi: 10.1108/09685221011035241

[53] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[54] J. Zhang and M. Zulkernine. Anomaly based network intrusion detection with unsupervised outlier detection. In *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 5, pp. 2388–2393. IEEE, 2006.

[55] Y. Zhu, J. Liang, J. Chen, and Z. Ming. An improved nsga-iii algorithm for feature selection used in intrusion detection. *Knowledge-Based Systems*, 116:74–85, 2017.