CLEAR-ROAD: Extraction of Temporally Co-occurring yet Rare Critical Alerts

Gordon Werner*[1] and Shanchieh Jay Yang†[1]

[1]Rochester Institute of Technology

Abstract

Intrusion detection systems generate a large number of streaming alerts. It can be overwhelming for analysts to quickly and effectively understand behavior within a network. Critical alerts occur so infrequently that it can be difficult to determine what surrounding alerts are actually related to them, providing a deep challenge to analysts. What if an analyst could provide a collection of known critical alerts and quickly receive a summary detailing their temporal behaviors within a network as well consistently co-occurring signatures that pre-empt or succeed the critical action? What if this information could be provided in near real time, with no training data, and with the capability to adapt to changing temporal patterns and relationships across signatures? The Concept Learning for Intrusion Event Aggregation in Realtime with Rare co-Occurring Alert signature Discovery (CLEAR-ROAD) answers that question, revealing consistent co-occurrences derived from alerts with similar temporal arrival patterns. Alerts are aggregated, or sequenced, based on their unique and invariant arrival patterns, not external training data. The signature patterns expressed by such temporal activity are then discovered through pattern mining techniques. A constrained databasing approach is used to reduce the number of sequences processed by an average of 90% for individual streams. Case studies are conducted to analyze the co-occurring signatures found across two real world datasets, one from a SOC operation and another from a penetration testing competition. CLEAR-ROAD is able to find consistently co-occurring signatures across streams and datasets quickly and effectively. Differences in temporal behavior are also found to lead to unique co-occurring signatures for some critical alerts. Case studies show the clear and near-immediate benefits provided to analysts by the system.

1 Introduction

Network security has never been more important than in recent times. As networks continue to grow in scale so to does the threat of malicious activity. Even with Security operation centers (SOCs) staffed with analysts dedicated to monitoring a network it is easy for them to be overwhelmed. Modern intrusion detection systems (IDSs) generate massive numbers of alerts quickly making it difficult for analysts to quickly or easily understand what is happening.

Imagine the scenario where a network suddenly sees an influx of a certain critical alert such as “ET WEB_SERVER ColdFusion administrator access." While the signature can inform analysts of general intent, they would be driven to find what additional alert signatures are being generated to construct an overall attacker “action". Even using a short time window around the critical signature, a manual query by an analyst will return a large number of unique alert signatures.

*gxw9834@rit.edu
†jay.yang@rit.edu

Proceedings of the Conference on Applied Machine Learning for Information Security, 2021

This work aims to solve a unique problem motivated through discussions with real world SOC operators. Given a set of critical signatures from a SOC analyst, can they quickly be provided with the timing information of any other alert signatures which co-occur with statistical dependence? Critical signatures are rare occurring very infrequently in a network and it may not be obvious at a glance how they relate to other alerts. While pattern mining of cyber alerts has been briefly explored in recent literature, this is the first to the author's knowledge which aims to provide such directed and clear insights into alert co-occurrence to cyber analysts.

This paper introduces and details the Concept Learning for Intrusion Event Aggregation in Realtime with Rare co-Occurring Alert signature Discovery (CLEAR-ROAD) system. Using data driven statistical processing, IDS alerts can be sequenced in real time based on their temporal arrival patterns with no external training data. Pattern mining techniques applied to constrained sequence data bases [29] allow for regular discovery of co-occurring signatures with low performance overhead.

By processing alerts in this way the additional temporal context and relationships across signatures is represented. Of the 113 critical signatures analyzed, 71 were found to have statistically co-occurring signatures. 65 of these had consistent co-occurring signatures across external sources across two IDS datasets exhibiting similar temporal behavior within the same network. In some cases, variation in temporal behavior lead to unique co-occurring signatures for each timing pattern. These results are highlighted and discussed through thorough case studies of the “GPL EXPLOIT CodeRed v2 root.exe access" and “ET WEB_SERVER ColdFusion administrator access" signatures.

The rest of this paper is organized as follows. Section 2 discusses related work in the field of alert aggregation and correlation. Section 3 details the motivation and challenges in finding co-occurring signatures. Section 4 details the CLEAR-SPADE architecture. Section 5 introduces the experimental set up, datasets used and the differences between them while Section 6 details the case studies conducted through the eyes of a SOC analyst. Section 7 concludes the paper.

2 Related Work

2.1 Alert Aggregation

IDS systems are used to raise alerts to network administrators of suspected anomalous or malicious activity [12]. Quickly and effectively processing the extreme number of alerts [4] in order to construct a clear and unified knowledge of a network's security status [13] is necessary for defense. A simple and straightforward way to reduce the overall number of alerts is through aggregation [9]. Alerts of the same type occurring close with one another are removed. The intuition here is that they are generated by ongoing activity, e.g, scanning, or alerts that are generated from the same activity by multiple scanners [8]. Traditional aggregation aims to remove ‘redundant’ alerts [21] but provides no deeper insight into the alerts presented to an analyst.

2.2 Alert Correlation

Alert correlation methods aim to process cyber alerts and provide deeper meaning to them [14]. Some early approaches provided modeling languages that allowed users to define known attack scenarios [2]. These types of approaches require a large, manually defined set of training patterns [18]. There is high overhead in creating such a dataset, especially when one considers that there are always new and emerging attack patterns. Machine learning techniques have been used to learn attack scenarios [3] or to reconstruct incoming alerts to previously clustered alerts [20], however these methods are still dependent on manual labeling of training data [14]. Rule-based approaches attempt to match prerequisite and consequent actions to cyber attack steps [1], but these too require specific attack knowledge and are unable to detect emerging attacks as they are not defined in the rules database.

Statistically driven approaches to alert correlation attempt to determine relationships across alerts without prior domain knowledge [14]. Researchers in [17] a Bayesian model was generated for each “hyper alert." Hyper alert construction leverages many basic alert aggregation principles, collecting all alerts with identical attributes such as source and destination IP and signature. Conditional probabilities are then calculated for each pair of hyper alert in an attempt to produce a causal relationship between alerts. This approach was iterated upon in [18] to provide “online" alert correlation however the approach still required extensive off-line Bayesian training using large sets of hyper alerts.

2.3 Temporal Correlation of Alerts

Many alert correlation techniques use some type of windowing when processing new alerts [5]. Alert occurrence frequencies are used when determining relationships across alert types, but the temporal relationships across alerts usually are not used in alert correlation. This seems like a missed opportunity as time series modeling has been applied effectively to network traffic, alert counts and cyber intrusion events in research. ARIMA models provided a boost in accuracy when forecasting cyber event counts [26]. Hourly counts of individual signatures have been leveraged to detect abnormalities in occurrence rates [25]. Weekly analysis of malicious activity against a commercial entity found seasonal behavior, and changes in intensity over the course of a day or week [27]. Researchers in [28] are the first to our knowledge to incorporate statistically driven aggregation of alerts based on their arrival patterns. Aggregation leveraged the notion of concept drift, the phenomena where the distributions or relationships across features change over time [15]. These changes can happen gradually or suddenly, and are very common in network traffic and other human driven systems [24]. Manual analysis of learned temporal “concepts" found signatures with consistent temporal properties and co-occurrences.

2.4 Data Mining Applied to Cyber Alerts

In [23] directed graphs were constructed based on the source and destination IPs of alerts. Pattern mining was conducted over a day’s alerts to collect association rules. It is not sufficient to simply collect association rules as they can occur without a true dependence existing [11]. In [19] sequences of hyper alerts were mined to find patterns within a network. Due to the use of hyper alerts, directional order across alert types must be estimated as a hyper alert is treated as a single event although it is made up of a number of alerts occurring at varying times within the current window. Applying similar pattern mining techniques to individual alerts should provide a clearer and more confident context to signature occurrences and relationships. The performance of sequential mining algorithms to alert and netflow data was explored in [10]. While the efficacy of the algorithms was not explored the performance results showed both that such algorithms can be run in an on-line manner with low performance overhead and that sequential database construction can severely impact performance to the point that online operation is infeasible.

Previous applications of pattern mining to cyber alert data would treat entire streams of alerts as individual sequences. Pattern mining algorithms are then applied to the group of all sequences within a network. While this allows for frequent patterns to be found across streams, it does not allow for patterns unique to streams to be explored. Creating sequences within streams is difficult as there is no clear or obvious beginning or ending point to a users “actions" found within alerts.

2.5 Data Driven Learning of Temporal Behaviors

The Concept Learning for Intrusion Event Aggregation in Realtime (CLEAR) system is able to dynamically learn the temporal arrival patterns of cyber alert streams in near real-time and with no training data [28]. CLEAR aggregates alerts as they are generated by an IDS system and is capable of detecting and ending an aggregate with a maximum delay of two alert arrivals. CLEAR employs a concept learning engine which builds overarching temporal behaviors made up of statistically similar aggregates. CLEAR's data driven approach to aggregation allows for aggregates to be more than collections of alerts with matching fields [9]. It's lack of dependence on training data and ability to learn and adapt to ongoing network traffic patterns ensures that its concepts best represent the current temporal behaviors of alerts within a network.

3 Finding Co-Occurring Signatures: Definitions and Challenges

Finding signatures that occur with a analyst specified critical signature presents a new and unique challenge. Figure 1 shows an example stream of cyber alerts over time. This mimics an analyst's view after querying for the critical alerts and those that occurred temporally near to it. Are these alerts related to one another? Is the occurrence of the critical signature statistically dependent on those around it? If so, what temporal patterns exist between them? Can this example be applied to future instances of the critical signature? These are questions raised by SOC analysts when attempting to process IDS alert data and ones that do not have easy answers. By removing a reliance on external historic data and training, analysts can be confident that the results and statistics they are being presented with are representative of their *current* network.

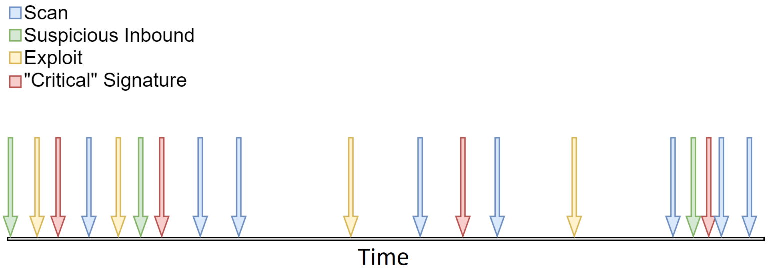

Figure 1: Illustration of IDS alert stream

3.1 Sequencing Alerts Analytically

Not every malicious entity attacks a network the same way. Further, a single attacker may deploy different strategies to reach the same goal when met with resistance. If all alerts from a single external IP are collected into a single sequence, the new co-occurrences brought on by such changes in strategy could go unnoticed. Pattern mining algorithms do not consider the number of pattern occurrences within the same sequence, only the number of sequences containing them [6]. To better understand and find co-occurring signatures a finer grained sequencing of alerts is needed.

Aggregates created by CLEAR are strong candidates for use as sequences in this context. They are created in a data driven and unsupervised way with no external training data; sequencing temporally similar alert arrivals together. CLEAR aggregates are temporally invariant and statistically unique from one another [28]. This provides great confidence that alerts are correctly sequenced based on temporal arrival patterns.

3.2 Extracting Co-Occurring Signatures

By sequencing alerts by their temporal arrival patterns it is possible to analyze and understand signature co-occurrences within specific episode types. Concepts generated by CLEAR are statistically unique from one another. Should a specific alert signature appear in multiple concepts, it is safe to assume that the signature was generated by two unique temporal alert “episodes." The temporal structure of concepts helps give context to any co-occurring signatures found within, and can potentially change which signatures are co-occurring for a given critical one. SOC analysts are concerned with the flow between alerts: which signature co-occurs prior to or after a critical one? At what timing? CLEAR aggregates are a series of successive alerts, however aggregation is based solely on the inter arrival times (IATs) of alerts. Additional analysis is necessary to discover any patterns across other alert attributes within individual concepts. Sequential pattern mining algorithms process collections of temporally ordered items in order to find patterns that occur frequently. From these sequences, rules and their statistical confidence and correlation can be derived. This well fits the goals of this research, to find co-occurring alerts and understand their temporal relationship to a critical signature. The application of sequential pattern and rule mining to CLEAR aggregates and concepts will produce co-occurring signatures unique to the temporal patterns exhibited by the episodes containing them.

4 CLEAR-ROAD Architecture

To quickly and effectively find and deliver co-occurring alert signatures and their temporal characteristics to a SOC analyst the Concept Learning for Intrusion Event Aggregation in Realtime with Rare co-Occurring Alert signature Discovery (CLEAR-ROAD) system is presented. Alerts are processed by CLEAR as they are generated by the IDS and maintain concepts and aggregates for each stream with at most a two alert delay. ROAD post-processes these concepts and aggregates with sequential pattern and rule mining. To reduce overhead and increase efficiency, sequence databases (SDBs) are constrained to only process sub-sequences containing the critical signature while still producing statistics accurately in relation to the entire database. ROAD’s processing can be manually run by an analyst at anytime or scheduled to occur at fixed intervals, processing recent historic time windows based on supplied parameters. ROAD finds all co-occurring signatures and collects and presents multiple levels of statistics to the analyst. High level summaries for critical signatures as well as in depth statistics for each co-occurring signature are collected and delivered to analysts quickly.

A flowchart describing the overall process of the presented system can be found in Figure 2. CLEAR runs in an online manner processing IDS alerts and outputting aggregates in near-real time. Each aggregate is mapped to the temporal concept it represents. These aggregate and concept mappings are then ingested by ROAD, and processed with cSPADE to extract co-occurring alert signatures. To reduce processing overhead, a constrained database is constructed using the analyst supplied critical signatures. Only aggregates containing the critical signatures are processed by cSPADE, with additional post processing done to the results to account for the aggregates not included in the SDB. Sequences and rules are then parsed, processed and tabulated before being presented to the analyst.

While this processed can be manually executed by an analyst at any time, it can also be run periodically to analyze potential changes in co-occurrence. The system can process any number of critical signatures specified by the analyst when it is invoked.

4.1 Pattern Mining of Temporal Concepts

After being processed by CLEAR, ROAD analyzes the sequenced alerts to extract co-occurring signatures. The constrained SPADE (cSPADE) algorithm is used [29] as it allows for itemsets, maintains order within sequences and can find frequent patterns that occur with gaps. IDSs produce a high volume of alerts, some of which are false positives [5] or unrelated to attack actions. cSPADE’s allowance for gaps between frequent patterns in sequences means that potential noise in the alert stream will not impact finding consistently co-occurring alerts.

cSPADE analyzes a sequence data base (SDB) which is made up of a number of individual sequences S. Each sequence S_i contains some number of events $S_i = e_{i,1}, e_{i,2}, ...e_{i,n}$ with each

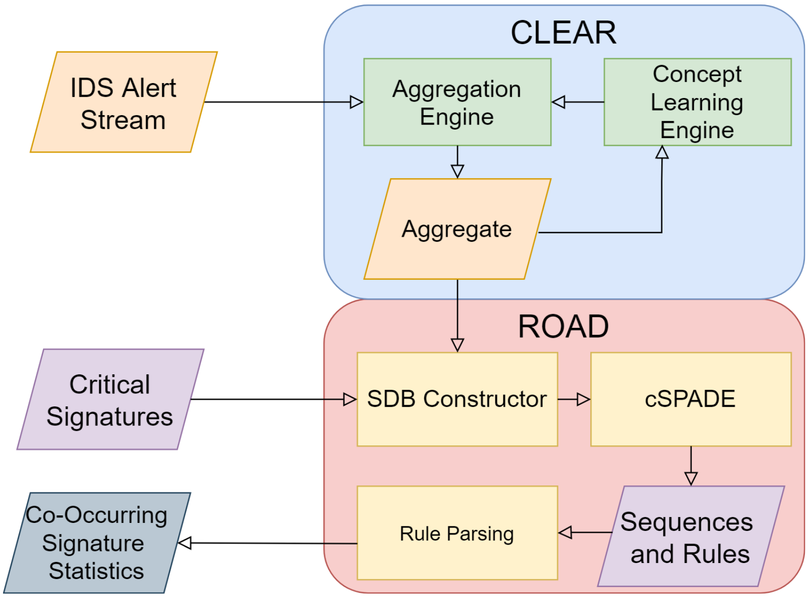

Figure 2: Flowchart of IDS Alerts Through CLEAR-ROAD

event containing a number of items $i_m \in I$ that occur at a specific time t_n. Each successive event in a sequence occurs at a time later than the previous event $e_{i,a}, e_{i,b}, t_a < t_b$. If multiple items occur at the same time they are stored in the same event $e_{i,j} = I_j | i_1, i_2, ...i_m$. cSPADE processes all sequences in the SDB looking for any sub-sequences $s \subseteq S$ which occur at a frequency higher than a user designated minimum. A sub-sequence's frequency, or support $Sup(s)$, is the count of sequences it is found in divided by the total number of sequences in the SDB N.

Association rules can be mined from frequent sequences to derive potential relationships between item occurrences [7]. An association rule is defined over the directional relationship of two frequent sub-sequences $A \rightarrow B$. The support of the rule $Sup(A \rightarrow B)$ is the proportion of sequences the rule is found in within the SDB. Rules have a confidence value shown in (2) which define the rate of occurrence of B given the appearance of A.

$$Support = Sup(A) = \frac{A}{N} \tag{1}$$

$$Conf(A \rightarrow B) = \frac{Sup(A \rightarrow B)}{Sup(A)} \tag{2}$$

A number of metrics have been used to analyze the “importance" of individual rules [7] based on the specific needs of the miner. Lift is one such metric and it measures how likely it is for the consequent of the rule to occur in relation to the antecedent based on the frequency of each occurring individually. A lift higher than 1 indicates that the occurrence of the two parts of the rule are directly correlated with one another, their occurrence together in a sequence is more likely than random chance. The calculation of lift can be found in (3). Lift is a beneficial metric in finding co-occurring alerts given that critical alerts are inherently rare. It is likely that any potential co-occurring alert will have a much higher frequency than the critical alert, therefore it is imperative to ensure that their co-occurrence is not mere chance, but that it is statistically probable that they occurred in relation to one another.

$$Lift = \frac{Sup(A \rightarrow B)}{Sup(A) \cdot Sup(B)} \tag{3}$$

Lift is a powerful metric as it can potentially find co-occurring signatures in as little as one co-occurrence. With one co-occurrence, the lift calculation simplifies to $\frac{N}{A \cdot B}$ where A and B are the counts of the individual signature occurrences.

4.2 Constrained SDB Generation

As noted in [10], how the SDB is constructed can severely impact the performance of pattern mining algorithms. CLEAR's aggregates are leveraged as sequences as they represent a stationary alert arrival pattern [28]. Each stream is independently processed with found sequences and co-occurring signatures across streams found through analysis of ROAD's statistical results.

SDBs are further constrained by discarding any sequences not containing the critical signature. After being processed by cSPADE, the resulting support statistics are scaled to account for the unconstrained database. Figure 3 shows the boxplot of the overall size reduction of the SDB obtained when analyzing only aggregates containing the critical signature for all critical signatures found in the CPTC dataset. SDBs saw on average a 90.1% reduction in sequences when including only aggregates containing the signature.

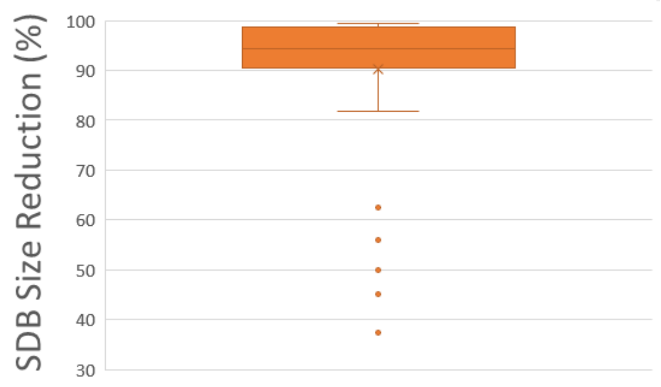

Figure 3: Plot of SDB size reduction for all concepts

5 Initial Experimentation and Co-Occurring Signature Findings

Experiments were conducted over two unique sets of Suricata [22] cyber intrusion alerts. The first set of data was captured in a real world SOC operation (RSOC) environment. IP addresses were obfuscated to maintain privacy with an additional alert field indicating which, if any, IPs were external to the network. Alerts were captured and aggregated by CLEAR in real time during the summer 2020. Alert correlation was later developed and integrated and was therefore conducted offline.

The second dataset was collected at the 2018 National Collegiate Penetration Testing Competition (CPTC) [16]. CPTC is a yearly college competition where a number of teams execute parallel penetration testing operations against a common "client" network. There were nine teams competing with each provided identical personal and target networks. Such rigid structure and duplication in network infrastructure allows for greater confidence in potential results. Knowing that eight groups with the same goals are acting on a network leads to an expectation of similar behaviors across teams. This should therefore manifest in consistent alert correlation across the dataset.

Data streams were created based on "adversary" IPs. For RSOC this was done using the external IPs captured by the IDS system. In CPTC this was done using the known IPs given to each of the team members. By parsing the data in this way each stream can be most closely interpreted as a single attacker's behavior, and is a common approach to parsing intrusion alerts in research.

The CPTC dataset resulted in 50 streams made up of nearly two hundred thousand alerts over the course of the day of the competition. The RSOC environment by comparison saw six hundred and twenty thousand alerts in its first day alone generated by over one thousand external IP addresses. Although the RSOC dataset experienced a larger overall number of alerts, they are spread over a large number of relatively short streams. On average, RSOC streams contain 378.9 alerts while CPTC streams contained 2097.6 alerts. It is very common in a real world scenario for an external IP to connect to a network and generate only a small number of alerts. In many cases these are false alarms raised by non-malicious use of the network. Even when malicious however, it is still not uncommon for an attacker to connect to a network with a unique IP address in order to conduct a short burst of activity. The CPTC competition by contrast provided individuals with unique IP addresses to be used for the ten hour competition. This results in a small number of very long streams, contrasting the RSOC dataset.

IDS signatures contained in the CPTC dataset were manually mapped by SOC experts to corresponding attack stages. For the following results, any alert mapped to the attack stages "arbitrary code execution, brute force credentials, command and control, data exfiltration, and privilege escalation" were treated as critical.

5.1 Quantitative Summary of CLEAR-ROAD

Table 1 shows the breakdown of critical signatures found in the CPTC dataset by attack stage. Of the 113 signatures present CLEAR-ROAD found at least one co-occurring signature for 71 (62.8%) of them. The final column of the table shows regularly co-occurring signatures, those that co-occur with the critical signature at a rate of at least fifty percent across the entire dataset.

Table 1: Summary Results for CPTC Critical Signatures

Atk. Stage	Tot. Crit. Sig.	w/co-sig	w/reg. co-sig	In RSOC	Same Co-Sig
ARB. CODE EXE	55	35	34	8	1
BRUTE FORCE CREDS	4	3	2	3	1
COM. & CON.	19	9	9	10	8
DATA EXFIL	26	19	16	6	1
PRIV ESC	9	5	4	3	2

The final two columns of the table list the counts of critical signatures also found in the RSOC database. Of the 30 signatures found in both datasets, a total of 14 had the same co-occurring signatures in both datasets. The command and control and privilege escalation critical signatures made up a majority of those that had similar co-occurring signatures across datasets.

5.2 High Level Summary Results for Critical Signatures

Table 2 shows high level summary results for individual critical signatures from both datasets. Even at this high level, key insights can be made regarding the occurrence patterns of critical signatures to help determine where to focus. As expected, critical signatures are extremely rare in the set of all generated IDS alerts for both datasets, in most cases a critical alert occurs fewer than one in one hundred alerts. However even with this rarity these signatures appear in aggregates with a high number of unique signatures. Were an analyst to manually query the critical alerts and the signatures surrounding them it would still be quite difficult for them to determine which are truly co-occurring.

By comparison, even just high level results can provide immediate feedback to an analyst. The bottom four rows of the table highlight a group of critical signatures that co-occur with one another. These four signatures are all variations of alerts that notify specific configuration options in a PHP URI. It is clear that the occurrence of these signatures are related given their identical statistics. Most likely they are borne from an attacker sweeping the network for a number of potential PHP vulnerabilities. The signatures also co-occur with extremely high lift, indicating that they occur together in aggregates and can very rarely be found separate from one another.

Table 2: Summary Results of Critical Signature Correlation

Cri.Sig.Abr.	Dataset	Rarity(%)	Agg.Sigs.	μ Lift	μ IAT
CFADMN	CPTC	1.39	43	4.67	6.1 ms
CFADMN	RSOC	1.39	19	1.45	1.2 s
CFAPIA	CPTC	0.16	27	4.52	1.5 ms
CFAPIA	RSOC	0.07	14	1.75	170 ms
CFUTIL	CPTC	0.16	27	4.52	1.5 ms
CFUTIL	RSOC	0.04	11	1.75	322 ms
DRUPAL	CPTC	0.13	20	5.78	10.8 ms
DRUPAL	RSOC	0.02	15	1.33	1.6 s
CDERED	CPTC	0.39	15	7.8	15.8 s
SMPURI	CPTC	0.04	9	196	2.6 ms
SSPURI	CPTC	0.04	9	196	2.6 ms
DIFURI	CPTC	0.04	9	196	2.6 ms
OBDURI	CPTC	0.04	9	196	2.6 ms

To better explore the CLEAR-ROAD system and how it can impact an analyst's ability to understand network activity, results are presented as case studies. The only assumption made in collecting results is that the analyst has a known collection of individual alert signatures that are considered "critical" and that this was provided to the system at run-time. In the presented experiments critical signatures were those categorized by SOC experts as "command and control," "privilege escalation," "arbitrary code execution" and "data exfiltration." Examples chosen best highlight certain findings regarding co-occurrence consistency but are not the only examples contained within the datasets.

6 Case Studies

6.1 Case Study 1: CodeRed

The critical signature "CDERED" (GPL EXPLOIT CodeRed v2 root.exe access) was only observed in the CPTC dataset, but was an attack vector leveraged by a number of the teams giving good insight into pattern consistency across users. The CodeRed worm attempts to connect to random hosts in the hope of finding a Microsoft IIS web server. Figure 4 shows a selection of boxplots representing the IATs of alerts in the concepts from 7 teams containing the signature. Each of the two temporal "modes" could potentially see unique or additional correlated signatures possibly accounting for the differences in presentation of the critical signature across streams. Having this initial high-level context helps frame the analyst's expectations as they delve deeper into the statistics presented regarding individual co-occurring signatures.

Table 3 shows statistics relating individual co-occurring signatures with the critical signature under analysis. The third column represents the count of aggregates containing the critical and co-occurring signature while direction columns indicates the order of occurrence between the two signatures. The lift column shows the average lift for all occurrences of the signature pairs while the IAT column shows the average arrival time between the critical and co-occurring signatures. The IATs and appearances of the co-occurring signatures match with the frequencies and timings of the concepts seen in Figure 4. There is a strong consistency in these co-occurring signatures with the critical signature independent of the source, but dependent on the timing. Thanks to the structure of the CPTC competition each individual attacker was targeting a copy of the same network. With the same configuration, the same action generated the same set of co-occurring signatures, even among any other alerts generated by other actions being taken within the network.

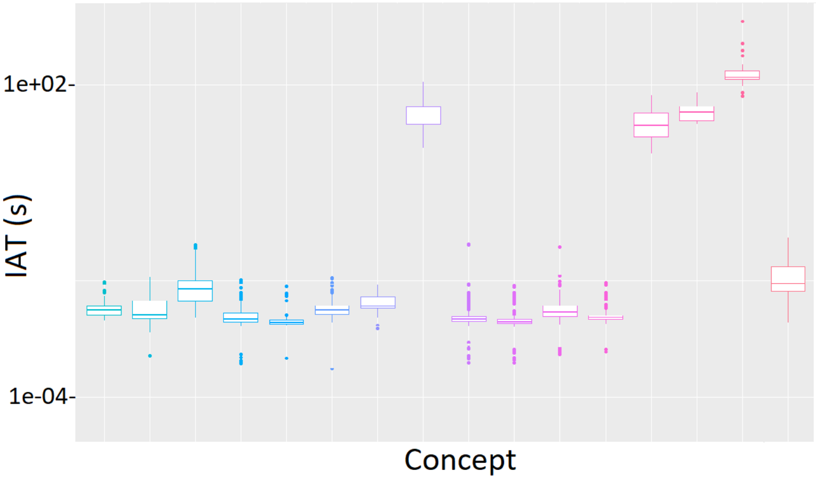

Figure 4: Boxplots of concepts containing “CDRED" signature in CPTC dataset

Table 3: Detailed Co-Occurring Signature Statistics

Cri.Sig.Abr.	Co.Sig.Abr.	Apps.	Dir	μ Lift	μ IAT
CDERED	ISAPIA	28	cri→co	18.9	150 ms
CDERED	ROOTA	22	co→cri	24.5	17 ms
CDERED	MSAAC	4	co→cri	21	5 ms
CDERED	JEXBO	6	co→cri	3	34.5 s
CDERED	DTLEAK	4	cri→co	6	28 s

To fully understand the relationships between critical and co-occurring signatures and any potential attack vectors executed, analysis of the individual signatures and their causes are needed. The first group of co-occurring signatures “ROOTA" (GPL WEB_SERVER / root access), “MSAAC" (GPL EXPLOIT /msadc/samples/ access) and “ISAPIA" (GPL EXPLOIT ISAPI .idq access) co-occur with CDERED quickly, with IATs measuring in milliseconds. Signatures ROOTA and MSAAC are triggered by an attempt to access specific directories of an IIS server, the usual target of the CDERED worm. Successful access is what most likely lead to the code red exploit being deployed against the server, generating the critical signature shortly after the initial access. Signatures vary commonly followed the code red alert an indicates a successful buffer overflow on a IIS server. This could indicate the worm was successful in finding and exploiting an IIS server.

The second temporal behavior highlights a completely different attacker action. The signature “JEXBO" (ETPRO WEB_SERVER JexBoss Common URI struct Observed 2 (INBOUND)) relates to a java platform testing tool that has been used in ransomware attacks such as “SamSam." Jexboss is used in conjunction with web servers so it is not unreasonable to assume that it was used as a potential injection vector for the code red worm. About 30 seconds after the Jexboss signature, the code red signature was alerted followed by the “DTLEAK" (ETPRO WEB_SERVER Possible Information Leak Vuln CVE-2015-1648) which alerts of a potential data leak through the opening of a command terminal.

Figure 5 Illustrates the differences in timing between the found correlated signatures and a potential timing flow of the overall attack. Such a plot makes clear the average timings between critical and co-occurring signatures while also highlighting the temporal discrepancy between the two “modes" that CDERED is seen in in the network. Having the knowledge of these co-occurring signatures can help a SOC analyst to adjust their defenses to appropriately react when a ROOTA or JEXBO alert is generated with in the network to prevent future potential exploits

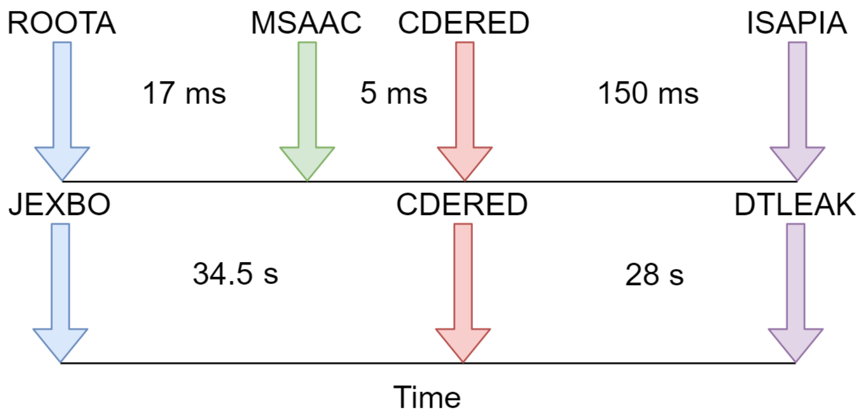

Figure 5: Timing illustration of unique signature co-occurrences related to Code Red

6.2 Case Study 2: ColdFusion

The critical alert signature “CFADMN" (ET WEB_SERVER ColdFusion administrator access) occurred in both datasets with each experiencing very different appearance rates for co-occurring signatures. While both datasets saw “CFAPIA" (ET WEB_SERVER ColdFusion adminapi access), “CFUTIL" (ET WEB_SERVER ColdFusion componentutils access) and “DRUPAL" (ET EXPLOIT Possible CVE-2014-3704 Drupal SQLi attempt URLENCODE 1), they did not co-occur with CFADMN in the CPTC dataset with the same frequency as in RSOC. This may seem strange given that most of these co-occurring signatures are directly related to ColdFusion servers, but in fact highlights another strength of CLEAR-ROAD’s data driven processing. Just as different network topologies can cause unique timing characteristics within streams, so too can the maintenance and configuration of the infrastructure. The CFADMN alert is raised when the administrator of an Adobe ColdFusion web server is remotely accessed. This is simply one of a number of alerts that can be raised when a malicious entity is attempting to access a ColdFusion server. CFAPIA, CFUTIL and CFPWDA (ET WEB_SERVER ColdFusion password.properties access) all alert on different methods of gaining access to a ColdFusion server. If not configured properly, it is possible to retrieve the component utils page, the administrator page or the adminapi page of a ColdFusion server through a standard web call. It is possible for sensitive infomation, such as login credentials to be stored in plaintext in the component utils pages.

Table 4: Detailed Co-Occurring Signature Statistics for CFADMIN Critical Signature

Cri.Sig.	Co.Sig.	Dataset	Apps.	Dir	μ L	μ IAT
CFADMN	CFAPIA	RSOC	1630	cri→co	1.81	0.5 s
CFADMN	CFAPIA	CPTC	2	cri→co	27.6	184 us
CFADMN	CFUTIL	RSOC	903	cri→co	1.82	1.52 s
CFAPIA	CFUTIL	CPTC	1	co↔cri	13	2 8ms
CFADMN	DRUPAL	RSOC	158	co→cri	1.61	1.97 s
CFADMN	DRUPAL	CPTC	1	co→cri	3	70.6 ms
CFADMN	CFPWDA	CPTC	4	co→cri	23	71 ms
CFADMN	NMAPSC	CPTC	82	co↔cri	1.93	5.5 ms
CFADMN	PHPINA	CPTC	31	cri↔co	11.5	20.1 ms
DRUPAL	STREX	RSOC	370	co→cri	1.2	1.84 s

The discrepancy in co-occurrence numbers across datasets is most likely caused by effective security measures taken within the RSOC environment. ColdFusion servers in RSOC are most likely well updated and defended to avoid vulnerabilities that allow for data leaks through external accesses. Therefor most attackers will attempt to access the administrator, api, and component utils unsuccessfully. The CPTC competition however intentionally builds networks with vulnerabilities for teams to attack. It is likely that a ColdFusion server used in the competition could be accessed in this way, meaning teams did not need to attempt both CFUTIL and CFADMN accesses as frequently throughout the competition. Another indicator that the CPTC ColdFusion servers were poorly maintained is the “CFPWDA" (ET WEB_SERVER ColdFusion password.properties access) which exploits a vulnerability that provides the attacker with hashed administrator passwords. This exploit has been patched as of 2013, but it is not unlikely that such an out of date server was intentionally included in a penetration testing competition. Figure 6 illustrates the co-occurrences and timing for signatures related to CFADMN in both datasets.

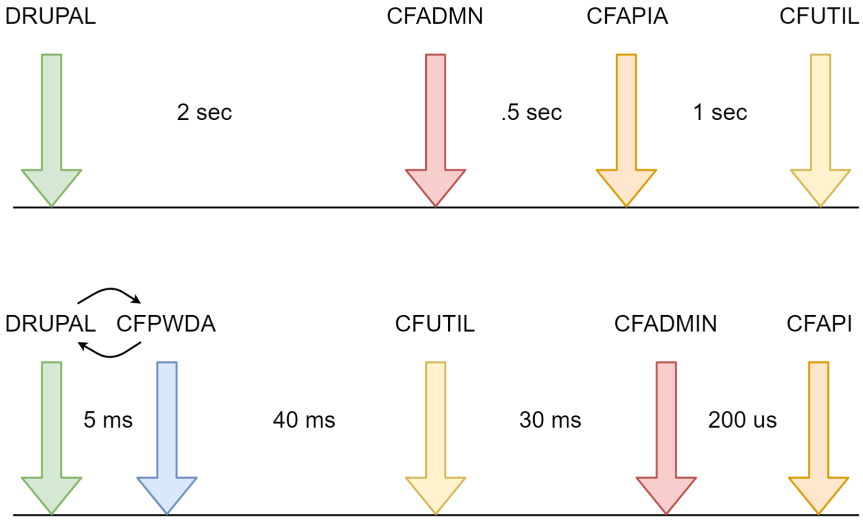

Figure 6: Timing illustration of Cold Fusion and co-signatures in RSOC (top) and CPTC (bot)

Also interesting is the order of the three signatures across datasets. Since all three are independent vectors of attacking a ColdFusion server order does not truly matter, however there is a clear difference between the core approach used by CPTC teams and real world entities. Interestingly both datasets see alerts with the “DRUPAL" (ET EXPLOIT Possible CVE-2014-3704 Drupal SQLi attempt URLENCODE 1) co-occurring before CFADMN. The DRUPAL signature can alert to potential SQL injection attacks against Drupal 7 web servers. Drupal servers are not based on the same coding language as ColdFusion servers. Most likely attackers are targeting web servers using a script, this theory is given more evidence by the co-occurrence of the “NMAPSC" (ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)) in CPTC and the “STREX" (ET EXPLOIT Apache Struts 2 REST Plugin XStream RCE (ProcessBuilder)) exploit commonly seen preceding DRUPAL in RSOC. Likely each network is being scanned for web servers with potential vulnerabilities which when found are targeted. The variation in ordering would indicate different scripts are being used in each network.

7 Concluding Remarks

It is infeasible to expect an analyst to be able to manually query and process all instances of a specific critical alert in a timely manner. While some automated approaches to alert correlation exist they require extensive training sets requiring manual labeling and are not focused on providing fast, intuitive feedback to an analyst in real time. SOC analysts are interested in knowing which signatures are actively co-occurring with certain “critical" signatures within their networks. Given the high number of alert signatures and the rate at which alerts are generated, it is nearly impossible to expect an analyst to discover these co-occurring signatures manually.

The CLEAR-ROAD system is able to quickly and effectively provide this key information to analysts defending a network. A real world SOC operation's alert data was processed and co-occurring signatures were found. Deeper analysis of the signatures produced gave strong supporting evidence towards the co-occurrences truly stemming from an attacker's action. By first processing alert arrivals to learn the temporal behaviors within a stream, co-occurring signatures and patterns are given an additional dimension of context. As seen in the first case study some critical signatures are used in different ways resulting in unique temporal patterns and co-occurring signatures. CLEAR-ROAD was able to find consistent alert co-occurrences across streams and across datasets with unique alert timings and vastly different stream characteristics.

CLEAR-ROAD's approach to SDB construction saw on average a 90.1 % reduction in the number of sequences processed with no impact to pattern or rule mining results. This allows for much lower processing times, providing an analyst with insights even quicker. Systems such as this that aim to provide new and unique perspective to SOC analysts are necessary. While automation of IDS systems and models is inevitable, the human element can never be fully removed from cyber defense. Developing and providing tools such as CLEAR-ROAD allow for smarter and more proactive defense.

References

[1] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *Proceedings 2002 IEEE symposium on security and privacy*, pages 202–215. IEEE, 2002.

[2] F. Cuppens and R. Ortalo. Lambda: A language to model a database for detection of attacks. In *International Workshop on Recent Advances in Intrusion Detection*, pages 197–216. Springer, 2000.

[3] O. Dain and R. Cunningham. Fusing a heterogeneous alert stream into scenarios. In *Applications of Data Mining in Computer Security*, pages 103–122. Springer, 2002.

[4] H. Debar and A. Wespi. Aggregation and correlation of intrusion-detection alerts. In *International Workshop on Recent Advances in Intrusion Detection*, pages 85–103. Springer, 2001.

[5] H. T. Elshoush and I. M. O. Alert correlation in collaborative intelligent intrusion detection systems—a survey. *Applied Soft Computing*, 11(7):4349–4365, 2011.

[6] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.

[7] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58–64, 2000.

[8] M. Husák and M. Čermák. A graph-based representation of relations in network security alert sharing platforms. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 891–892. IEEE, 2017.

[9] M. Husák, M. Čermák, M. Laštovička, and J. Vykopal. Exchanging security events: Which and how many alerts can we aggregate? In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 604–607. IEEE, 2017.

[10] M. Husák, J. Kašpar, E. Bou-Harb, and P. Čeleda. On the sequential pattern and rule mining in the analysis of cyber security alerts. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pages 1–10, 2017.

[11] P. Lenca, B. Vaillant, P. Meyer, and S. Lallich. Association rule interestingness measures: Experimental and theoretical studies. In *Quality Measures in Data Mining*, pages 51–76. Springer, 2007.

[12] H. Liao, Y. Lin, C.and Lin, and K. Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

[13] F. Maggi, M. Matteucci, and S. Zanero. Reducing false positives in anomaly detectors through fuzzy alert aggregation. *Information Fusion*, 10(4):300–311, 2009.

[14] S. Mirheidari, S. Arshad, and R. Jalili. Alert correlation algorithms: A survey and taxonomy. In *Cyberspace Safety and Security*, pages 183–197. Springer, 2013.

[15] J. Moreno-Torres, T. Raeder, R. Alaiz-RodríGuez, N. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[16] J. Pelletier. Collegiate penetration testing competition, 2018. `https://nationalcptc.org/`.

[17] X. Qin and W. Lee. Discovering novel attack strategies from infosec alerts. In *European Symposium on Research in Computer Security*, pages 439–456. Springer, 2004.

[18] H. Ren, N. Stakhanova, and A. Ghorbani. An online adaptive approach to alert correlation. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 153–172. Springer, 2010.

[19] R. Sadoddin and A. A. Ghorbani. An incremental frequent structure mining framework for real-time alert correlation. *computers & security*, 28(3-4):153–173, 2009.

[20] R. Smith, N. Japkowicz, and M. Dondo. Clustering using an autoassociator: A case study in network event correlation. In *IASTED PDCS*, pages 613–618, 2005.

[21] J. Sun, L. Gu, et al. An efficient alert aggregation method based on conditional rough entropy and knowledge granularity. *Entropy*, 22(3):324, 2020.

[22] Suricata. Suricata open source ids, 2020. `https://suricata-ids.org/`.

[23] J. Treinen and R. Thurimella. A framework for the application of association rule mining in large intrusion detection infrastructures. In *International Workshop on Recent Advances in Intrusion Detection*, pages 1–18. Springer, 2006.

[24] P. Vaz de Melo, C. Faloutsos, R. Assunção, and A. Loureiro. The self-feeding process: a unifying model for communication dynamics in the web. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1319–1330. ACM, 2013.

[25] J. Viinikka, H. Debar, L. Me, A. Lehikoinen, and M. Tarvainen. Processing intrusion detection alert aggregates with time series modeling. *Information Fusion*, 10(4):312–324, 2009.

[26] G. Werner, S. Yang, and K. McConky. Time series forecasting of cyber attack intensity. In *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, CISRC '17, pages 18:1–18:3, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4855-3. doi: 10.1145/3064814.3064831. URL `http://doi.acm.org/10.1145/3064814.3064831`.

[27] G. Werner, S. Yang, and K. McConky. Leveraging intra-day temporal variations to predict daily cyberattack activity. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 58–63. IEEE, 2018.

[28] G. Werner, S. Yang, and K. McConky. Near real-time intrusion alert aggregation using concept-based learning. In *Proceedings of the 18th ACM International Conference on Computing Frontiers*. ACM, 2021.

[29] M. Zaki. Sequence mining in categorical domains: incorporating constraints. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 422–429, 2000.