OutMet: A New Metric for Prioritising Intrusion Alerts using Correlation and Outlier Analysis

Riyanat Shittu
School of Electrical &
Mathematical Sciences
City University London
United Kingdom,
riyanat.shittu.1@city.ac.uk

Alex Healing, Robert Ghanea-Hercock
Security Futures Practice
BT Adastral Park
Ipswich, United Kingdom
alex.healing@bt.com
robert.ghanea-hercock@bt.com

Robin Bloomfield
School of Informatics
City University London
London,
United Kingdom
reb@csr.city.ac.uk

Rajarajan Muttukrishnan
School of Electrical &
Mathematical Sciences
City University London
United Kingdom
R.Muttukrishnan@city.ac.uk

Abstract—**In a medium sized network, an Intrusion Detection System (IDS) could produce thousands of alerts a day many of which may be false positives. In the vast number of triggered intrusion alerts, identifying those to prioritise is highly challenging. Alert Correlation and prioritisation are both viable analytical methods which are commonly used to understand and prioritise alerts. However, to the author's knowledge, very few dynamic prioritisation metrics exist. In this paper, a new prioritisation metric - OutMet, which is based on measuring the degree to which an alert belongs to anomalous behaviour is proposed. OutMet combines alert correlation and prioritisation analysis and in given attack scenarios, is capable of reducing false positives by upto 100%. The metric is tested and evaluated using the recently developed cyber-range dataset provided by Northrop Grumman.**

I. Introduction

Intrusion Alerts are raised by an Intrusion Detection System (IDS) which is usually situated at the network's perimeter to monitor incoming and outgoing traffic. When suspicious or malicious traffic is observed by the IDS, an alert is raised. IDS's identify suspicious or malicious traffic using either a signature-based approach or an anomaly-based approach. With signature-based IDSs, traffic is matched against a set of pre-configured attack patterns called signatures. In the latter IDS type, statistical methods are used to learn the normal behaviour of network traffic and alerts are triggered when traffic deviates from normal behaviour. Both approaches have limitations – most IDSs are known to trigger a high volume of intrusion alerts. In the vast number of triggered intrusion alerts, identifying which alerts to prioritise is challenging. According to early research [1], [2], upto 99% of raised alerts can be false positives. False positives indicate alerts either triggered on normal traffic or alerts triggered on benign attacks (attacks that are non-successful or cause no network harm).

Thus, most IDS advancements in research and industry particularly focus on improving the IDS detection rate and reducing the false positive rate. In this research, the latter is focussed on by filtering false positive alerts using our newly proposed prioritisation metric, "OutMet". We focus particularly on alerts produced by a signature-based IDS. Although most signature-based IDSs provide a default priority level for each alert, it is solely based on the signature matched. This is argued to be insufficient as attacks launched on a network trigger multiple alerts therefore the context of a single attack is unlikely to be captured by analysing a single alert [3].

Alert Correlation is a common approach used in understanding this attack context[4]. In alert correlation, a set of related alerts are grouped into a higher level meta-alert which represents a single intrusion activity.

Our background research reveals the existence of other prioritisation and severity metrics however very few focus on attack context and none are known to prioritise meta-alerts.

In our work, alert correlation is used to derive higher level meta-alerts which represent attack context. Various properties which could be used to prioritise meta-alerts were investigated (Described in Section-III-B1), however, we chose to prioritise meta-alerts based on their outlier degree property. This property was chosen because outliers reflect infrequent and anomaly behaviours in attack analysis and this often correlates with true attacks.

The proposed prioritisation metric is evaluated on a recent intrusion detection dataset provided from a cyber-range experiment carried out by Northrop Grumman [5]. The effectiveness of the prioritisation metric is illustrated by comparing it to 1)A similar prioritisation metric proposed by Alsubhi et al[6], [7] and 2)An alternative prioritisation metric which ignores attack context (i.e. alert correlation). An evaluation on some of a meta-alert's properties is also carried out to measure how relevant they could be under certain attack scenarios.

The rest of this paper is organised as follows. Section-II describes the related work, Section-III details the proposed approach for alert prioritisation, Section-IV describes the results from our experiment,and Section-V describes our plans for future work.

II. Related Work

To the knowledge of the authors, very little work has been done on defining Alert Prioritisation metrics for IDS alerts. Porras first proposed an alert ranking framework, M-Correlator, with a prioritisation component that consisted of two security metrics - relevance and the priority scoring[8]. The relevance scoring measured the validity of an alert i.e. the likelihood of the alert being a true positive. The priority scoring measured the severity of an alert given the targeted asset's value. The priority score also combined an interest score which measured the degree to which an analyst expressed interest in the attack category the alert belonged. Using a Bayesian model they determine the overall priority of an alert based on the acquired evidence e.g. P(priority = critical — relevance = low). A limitation in their approach is that knowledge from alert correlation is not taking into account during the prioritisation despite their framework consisting of a similarity based correlation component. Since it is solely based on user and network knowledge the framework is limited to discovering known incidents while novel attack incidents remain un-prioritised.

Noel et al proposed an alert prioritising framework which used a different metric [9]. The metric calculated the proximity of an alert to a critical asset. Thus, alerts targeted at assets closer to critical assets had a higher priority over those further away. Similarly to Porras et al's framework, it only uses network knowledge and no alert or correlation context is taken into consideration.

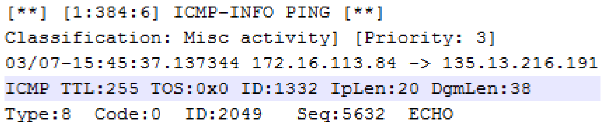
A more robust alert prioritisation system is proposed by Alsubhi et al [6], [7] who defines 7 metrics for prioritising alerts. Two of the metrics, an *alert relationship metric* and a *social activity metric*, are relevant to our work since they are based on alert correlation context. The alert relationship metric measures the degree to which the alert correlates with successive alerts whereby a high value indicates the alert is potentially a causal alert. The Social Activity metric is briefly introduced as a metric used to measure the activity of the source and destination IP addresses included in each alert.

Zomlot et al also proposed a prioritisation model for the alert correlation system they had previously presented [10] [11]. In their work on prioritisation, they use dempster-shafer to assign a degree of belief to each meta-alert (generated by the correlation system) which indicated the likelihood of true positivity given the quality of the IDS sensor which raised the alerts.

Unlike alert prioritisation, more effort has been focussed on alert correlation techniques. Using Salah et al's correlation model taxonomy, these can be classified into case-based, similarity-based and sequential-based methods. Case-based methods involve a rule language that uses expert domain knowledge to define alert types that may occur in a given attack scenario.Cuppens et al, Cheung et al, Eckmann and Kemmerer, and Cedric and Ludovic proposed LAMBDA, CAML, STATL, and ADELE respectively[12]–[15]. Although these provide high-quality correlations capturing known attacks, their limitation is that they are difficult to implement and maintain on a large-scale. In similarity methods, the correlation is based on feature similarity. Valdes and Skinner as well as Dains first introduced this approach. Although simpler to implement, such methods do not capture complex nor hidden correlations [16],[17]. More Recently,Hoffman et al proposed alert clustering technique[18]. Sequential-based correlation is more suitable for capturing causally correlated alerts with little or no apriori knowledge. Sequential-based methods include those proposed by Ning, and Debar and Wespi[19] [20]. Both used rule-like pre-requisites and consequences for correlating alerts. Qin also used abstract pre-requisites and consequences combined with statistical evaluation for correlating alerts[21]. Sequential-based alert correlation models which use little to no a-priori knowledge are based on Bayesian inference. Examples include work by [22]–[24] and [25]. Each of these alert correlation models output a set of meta-alerts represented in a graph like structure known as an *Alert Correlation Graph (ACG)*. [26] addresses how ACGs can be made useful to an analyst by simplifying the graphs using node and edge reduction techniques. However, few has focussed on how to prioritise alert correlation graphs in the event where many are produced. Based on our experiments with alert correlation models this however, is typical in real environments. In environments where a vast amount of alerts are produced, it is likely to achieve an equally vast amount of alert correlation graphs. This is one of the challenges our work aims to address.

III. OutMet: The proposed prioritisation metric

Figure 1 represents a sample *Snort* IDS alert output in text format. Only certain alert attributes are used in our analysis process. Each alert is represented using a 6-tuple a = $(\alpha_1, \alpha_2, ..., \alpha_6)$ where each element of the tuple is the attribute value of the following attributes: source IP, source port, destination IP, destination Port, priority (default) and intrusion type respectively.



```
[**] [1:384:6] ICMP-INFO PING [**]
Classification: Misc activity] [Priority: 3]
03/07-15:45:37.137344 172.16.113.84 -> 135.13.216.191
ICMP TTL:255 TOS:0x0 ID:1332 IpLen:20 DgmLen:38
Type:8  Code:0  ID:2049   Seq:5632  ECHO
```

Fig. 1. Snort IDS Alert Example

For each alert that is received, the OutMet is calculated over a series of steps:

A) *Alert Correlation:* A heuristic similarity measure is used to correlate alerts into meta-alerts. Each new received alert is either added to a new meta-alert or joins a previously existing meta-alert. Meta-alerts are represented as alert correlation graphs. An alert correlation graph is a directed acyclic graph $G = (E, V, W)$ where each vertex, $v \in V$ represents a single intrusion alert, each edge $e_{v_i,v_j} \in E$ indicates a correlation between two alert $v_i, v_j \in V$ and W contains the weight of each edge indicating the correlation strength between two alerts.

B) *Alert Correlation Graph Comparison:* The difference between each two or more meta-alerts is computed. A set of additional features which are unique to meta-alerts are extracted and used in the distance measurement. Graph Edit Distance (GED)[27] is applied to compare the graph structure of alert correlation graphs.

C) *Alert Correlation Graph Prioritisation:* Given a set of meta-alerts, A prioritisation value is assigned to each meta-alert based on the degree to which it differs to other meta-alerts. We use Local Outlier Factor, LOF [28] to derive this value. Furthermore, we refer to a meta-alert with a prioritisation value greater than P_θ as an outlier meta-alert. Such a meta-alert is an alert correlation graph that differs to the rest of the graph set to a degree which causes suspicion to arise. Each low-level IDS alert assumes the prioritisation value of its containing meta-alert. Non-prioritised meta-alerts (those with prioritisation values less than P_θ) are filtered and labelled as false-positives.

The next section details each step accordingly.

A. Alert Correlation

Given a stream of intrusion alerts, the most recent-alert, a_j is added to a pre-existing alert correlation graph, $g_{\max}$ as follows:

$$\mathrm{g}_{\max} \leftarrow \arg\max\{C(a_j, g_i)\}_{i=1}^{n} \text{ if } \max\{C(a_j, g_i) > \theta\} \tag{1}$$

Let it be assumed that G is the set of all pre-existing alert correlation graphs and |G| = n. If n = 0 then a new alert correlation graph, g_0 is initialised and a_j becomes the first alert in g_0. Otherwise, the Correlation Strength between a_j and any alert correlation graph is defined as:

$$C(\mathrm{a}_j, \mathrm{g}) \leftarrow \max\{C(\mathrm{a}_i, \mathrm{a}_j) : \mathrm{a}_i \in \mathrm{g}\}_{i=1}^{m} \tag{2}$$

where m is the number of alerts in a given alert correlation graph that occurred less than T_θ seconds apart from a_j. A new alert correlation graph may also be initialised if $\max\{C(\mathrm{a}_j, \mathrm{g}_i)\}_{i=0}^{n}$ is less than a defined threshold θ.

The Correlation Strength, $C(a_i, a_j)$ between two alerts is given as:

$$C(a_i, a_j) = \frac{\sum\limits_{f \in F} \omega_f \times f}{\sum\limits_{f \in F} \omega_f} \tag{3}$$

1) Features: The correlation strength between two alerts is dependent on the degree to which the alerts share similar features. In the equation 3, F is a set of correlation features. Based on the importance of each feature, each feature is assigned a weight ω_k. Each is described subsequently.

(i) Time Proximity (f_1) . This feature represents the time proximity between two alerts. It is derived as a sigmoid function such that the time proximity between two alerts decreases as the time between them increases.

$$f_1 = \frac{1}{1+e^t} \text{ where: } t = \frac{|t_{a_i} - t_{a_j}|}{T_\theta} \tag{4}$$

(ii) IP Similarity (f_2). This compares the source and the destination IP of a_i to the source and destination IP of a_j respectively. It is a common similarity measure that indicates that a_i and a_j are targets to a similar destination node or/and are from a similar attacker. The higher the value, the more likely this statement holds true.
A common IP similarity measure is applied for all IP address features. The IP Similarity Measure $S(ip_1, ip_2)$ is the common subnet mask between any two IPs as shown in Table I.

TABLE I
IP SIMILARITY

172.16.113.20	10101100 . 00010000 . 01110001 . 11001111
172.16.115.20	10101100 . 00010000 . 01110011 . 00010100
Common Mask	11111111 . 11111111 . 11111100 . 00000000
	22/36 = 0.61

(iii) Cross IP Similarity (f_3). This compares the source and the destination IP of a_i to the destination and source IP of a_j respectively. This feature indicates that a_j is a responsive intrusion to a_i. For example, if DestIP_{a_i} == SourceIP_{a_j} it could indicate that a_i was a successful attempt to exploit DestIP_{a_i}. After this success,a_j could indicate that this host is now performing intrusive activities. On the other hand, a_j could be an echo alert to a_i. In this case not only is the above condition satisfied but also SourceIP_{a_i} == DestIP_{a_j}. Since f_2 and f_3 conflict each other, i.e. the relationship between two intrusions is likely to be one or the other but not both, we select only one of the features based on the feature with the highest similarity.

(iv) Port Similarity (f_4). This feature indicates 1 if both alerts share the same destination port and 0 if they don't.

2) Process: To add a_j to $\mathrm{g}_{\max}$, a_j is added as a vertex in $\mathrm{g}_{\max}$ and the value of $C(a_j, g_{\max})$ is assigned to the edge between a_j and $\mathrm{a}_{\max} \in g_{\max}$ for $a_{\max}$ is the alert which produced the correlation strength. Algorithm 1 describes the entire correlation process.

3) Complexity: For each incoming alert a_j, the time complexity to find the most optimal alert correlation graph to add a_j to is O(N) for N is the total number of alerts in all the pre-existing alert correlation graphs with timestamps that satisfy $T_{a_J} - T_{a_i}$.

N increases as the number of incoming alerts increases. It is also likely to increase if T_θ increases. Thus, though the complexity for correlating a single alert is reasonable, in an environment where hundreds of incoming alerts are received per second, the task of alert correlation becomes highly complex.

An intuitive approach to ensuring the complexity is feasible is to ensure N is always of a reasonable size. To achieve this, sampling is used to select a set of M alerts from N given M is significantly less than N. A random approach described in work by Bateni et al was adopted[29].

B. Alert Correlation Graph Comparison

Each alert correlation graph represents a meta-alert where all the low level IDS alerts contained within it are part of

Algorithm 1 Correlation Process

```
function CORRELATE($a_j$)
    $a_j$                                          ▷ an incoming alert
    $G$                     ▷ A set of pre-existing alert correlation graphs
    if $G == \emptyset$ then
        INITIALISEGRAPH($a_j$)
    else
        $m = 0$
        for all $g \in G$ do
            for all $a_i \in g$ do
                if $C(a_i, a_j) > m$ then
                    $m = C(a_i, a_j)$
                    $g_{\max} = g$
                    $a_{\max} = a_i$
                end if
            end for
        end for
        if $m > \theta$ then
            $g_{\max}(V) \leftarrow g_{\max}(V) \cup v(a_j)$
            $g_{\max}(E) \leftarrow e_{a_{\max}, a_j}$
        else
            INITIALISEGRAPH($a_j$)
        end if
    end if
end function

function INITIALISEGRAPH($a_j$)
    $g \leftarrow$ new graph
    $g(V) \leftarrow a_j$
end function
```

the same or similar intrusion. Therefore, it may be meaningful to distinguish similar meta-alerts from highly dissimilar meta-alerts. This may help in identifying common intrusion activities and non-frequent intrusive activities.

To achieve this, the alert correlation graphs are compared using a distance metric. The distance between two alert correlation graphs, d(g$_1$, g$_2$) is the normalised value of four weighted features combined. The features are described.

1) Features:

1) Interval Rate (I). The interval rate indicates the average time interval in milliseconds between any two alerts with an alert correlation graph. An alert correlation graph with a low interval rate indicates that the alerts occurred with rapid succession. This is often found to be the case in denial of service attacks.

$$I = \sum_{k=1}^{n} \frac{t_k}{n} \tag{5}$$

$$\text{and} \quad d(I_{g_1}, I_{g_2}) = |I_{g_1} - I_{g_2}|$$

$t = \{t_1, t_2, ...t_n\}$ is a list of all the time intervals where t_1 is the interval between the first two alerts in a graph, t_2 and next two and so on.

2) Time Duration (TD). The time duration indicates how long the intrusion activity lasted and is measured as the time interval between the first alert v_1 and last v_n alert in the alert correlation graph.

$$TD = |t_{v_1} - t_{v_n}| \tag{6}$$

$$\text{and} \quad d(TD_{g_1}, TD_{g_2}) = |TD_{g_1} - TD_{g_2}|$$

The time duration feature provides more information when observed alongside the interval rate. E.g. If a high time duration and a low interval rate is observed then the alert correlation graph is likely to contain a high number of alerts.

3) Incoming Rate (IR). This is the ratio of incoming intrusions in the alert correlation graph. Domain knowledge is required to distinguish internal IP addresses from external.

$$IR = \frac{\text{\# of incoming Alerts}}{\text{\# of Alerts}} \tag{7}$$

$$\text{and} \quad d(IR_{g_1}, IR_{g_2}) = |IR_{g_1} - IR_{g_2}|$$

This feature may prove useful when trying to understand the context of the intrusive behaviour. For instance, an alert correlation graph with a higher value (i.e. higher incoming rate) could potentially indicate a DoS attack and a graph with a lower value (i.e. higher outgoing rate) would indicate that the internal host may be being used to perform malicious activities. A graph with an average value could indicate a constant activity between an internal host and an intruder. A case where an attack pattern graph consists of many ICMP pings and echo replies would yield an average value.

4) Graph Structure (gs). The nodes and edges of an alert correlation graph reflect the behaviour of the intrusion activity and the relationship between alerts of the correlation graph at varying times and stages of the intrusion activity. The graph structure of any two alert correlation graph is compared using “Graph edit distance” (GED) [27]. GED represents the distance between any two graph structures sg$_1$, sg$_2$ (of g$_1$, g$_2$ respectively) by counting the minimal number of actions required to transform sg$_1$ into sg$_2$ by manipulating sg$_1$ using a number of operations such as node deletion/insertion, edge deletion/insertion and node/edge substitution. Algorithm2 details the edit distance process.

2) Process: In Algorithm 2, Lines 2 & 3 are used to simplify the alert correlation graph. The complexity of calculating the GED between any two graphs is bound by |V| and |E|. In order to minimize this complexity, the nodes and edges in the alert correlation graph are aggregated and simplified. Therefore, an aggregated graph g’ of g is a graph where all the nodes (i.e. alerts) in g with the same intrusion type are aggregated in g’ and all edges in g with labels e$_{v_i,v_j}$are relabelled E$_{t(v_i),t(v_j)}$ in g’ where t(v) is the value of the vertex’s (i.e. alert’s) intrusion type. Furthermore, edges in g’ with identical labels are aggregated.

3) Comparison: The distance between any two alert correlation graph is:

$$D(g_1, g_2) = \frac{\sum\limits_{i=0}^{n} \omega_{pi} \times p_i}{\sum\limits_{i=0}^{n} \omega_{pi}} \tag{8}$$

Algorithm 2 EditDistance between graph structure

```
function EDITDISTANCE(sg'$_1$, sg'$_2$)
    sg'$_1$ ← Aggregate(sg$_1$)
    sg'$_2$ ← Aggregate(sg$_2$)

    L ← maximum cost allowed
    Q ← $\emptyset$                          ▷ A queue sorted by minimum path cost
    $v_i$ ← random vertex from sg'$_1$

    for v$_j$ in sg'$_2$ do
        s = new substitutePath(v$_i$,v$_j$)
        Q ← Q ∪ s
    end for
    d ← new deletePath(v$_i$)
    Q ← Q ∪ d
    while true do
        e = Q.firstPath()
        if e.isComplete() then
            return e
        end if
        if e.cost() > L then                 ▷ The maximum cost has been exceeded
            return L
        end if
        EXTEND(e, sg'$_1$, sg'$_2$, Q)
    end while
end function

function EXTEND(e, sg'$_1$, sg'$_2$, Q)
    if g'$_1$(V) ⊆ e == true then
        v$_i$ = next vertex in g'$_1$ : $v_i \notin$ e
        for v$_j$ ∈ sg'$_2$ do
            s ← e ∪ new substitutePath(v$_i$,v$_j$)
            Q ← Q ∪ s;
        end for
        d ← e ∪ new deletePath(v$_i$)
        Q ← Q ∪ d
    else
        for v$_i$ ∈ sg'$_2$ do
            if v$_i$ ∉ e then
                i ← e ∪ new insertPath(v$_i$)
            end if
        end for
    end if
end function
```

C. Alert Correlation Graph Prioritisation

A prioritisation value is assigned to each alert correlation graph based on its dissimilarity to a set of other alert correlation graphs, G. The prioritisation value is a real number between 1 and 4 (inclusive) where 1 indicates the least priority and 4 the highest priority.

$$p(g) = \begin{cases} 1 & 0 \leq \text{nlof(g)} \leq 0.25 \\ 2 & 0.25 < \text{nlof(g)} \leq 0.50 \\ 3 & 0.50 < \text{nlof(g)} \leq 0.75 \\ 4 & 0.75 < \text{nlof(g)} \leq 1 \end{cases} \tag{9}$$

$$nlof(g) = \frac{lof(g)}{\max\{lof(g_i)\}_{i=0}^{|G|}}$$

nlof(g) is a normalised value which represents the degree to which g is dissimilar with respect to a set of neighboring graphs in otherwords, it is the degree to which g is an “outlier”. This is more coherently illustrated in 2

In figure 2, each point represents a single alert correlation graph. Graphs closer together indicate the graphs are similar. Both g$_1$ and g$_2$ are highly outliers and would have high prioritisation values. g$_1$ is a local outlier to C_1 since it is somewhat similar to the members of cluster 1 but also varying. g$_2$ on the other hand is a global outlier as it is highly dissimilar to all other alert correlation graphs.

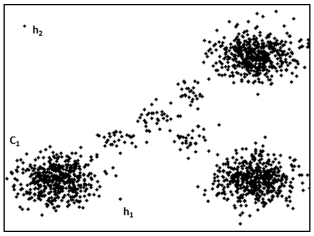


Fig. 2. Outlier Example

1) Process: lof(g) is computed over three steps:

i Derive g’s kth reachanbility distance and neighborhood:

$$rd_k(g, g_j) = \max\{kDist(g), D(g, g_j)\}$$

ii Derive g’s local density: This is the inverse of the average reachability distance which is defined in equation 10.

$$lrd(g) := 1/\frac{\sum_{g_j \in N_k(g)} rd_k(g, g_j)}{|N_k(g)|} \tag{10}$$

iii Derive lof(g):

$$lof(g) = \frac{\sum_{o \in N_g} \frac{lrd_g}{lrd_{g_j}}}{|N_g}| \tag{11}$$

Finally, we calculate nlof of each meta-alert, map it to its respective prioritisation value and drop all alerts with a prioritisation value less than P$_\theta$.

IV. Experiments and Results

The experiments were based on an attack scenarios contained in the cyber-range dataset provided by Northrop Grumman[5]. Each attack was carried out over a day’s period. Our objective is to illustrate the usefulness of OutMet in filtering out false positive alerts. We choose to use three metrics: evaluation techniques.

i False positive Rate Comparison (FPR): This compares the false positive rate in the alert dataset before and after applying the OutMet prioritisation metric. A good performance will indicate a lower FPR. The false positive rate is measured below:

$$\text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{\# of incorrectly prioritised alerts}}{\text{\# of prioritised alerts}} \tag{12}$$

ii True Positive Rate (TPR): The true positive before OutMet is applied is unknown hence no before and after comparison can be made. However, since the number of true positives before the OutMet application is known, we can atleast measure the TPR of the OutMet analysis. A good performance will indicate a higher TPR.

$$\text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{\# of correctly prioritised alerts}}{\text{\# of true positive alerts}} \tag{13}$$

iii Reduction Rate (RR): This measures the degree to which OutMet filters alerts. It takes a lesser preference to TPR and FPR but we have used it to be useful in scenarios where the TPR and FPR are uncertain. For example, if a low FPR and a high TPR is observed, the evaluation is that the results are good even if the reduction rate is very low.

$$\mathrm{RR} = \frac{\text{\# of alerts after prioritisation}}{\text{\# of alerts before prioritisation}} \tag{14}$$

A. *Attack Scenario – DMZ Attack*

In this attack an attacker launches an attack on a web server situated on the DMZ zone of a medium size network (200 clients). The intrusion detection system raises alerts during the web server attack however it also raises various alerts on non-malicious traffic. In total, ≈20% of the alerts triggered by the IDS were false positives triggered on normal traffic related to email content, host pinging *(are you alive?)*, and unreachable servers. The web server attack included casual port scanning of the DMZ servers followed by intrusive scans to discover and exploit vulnerable services using a brute force approach. During the entire day course, 3226 DMZ intrusion alerts were raised.

TABLE II
PARAMETERS USED

θ	T_θ	k	P_θ	$\omega(f_1)$	$\omega(f_2)$
0.7	30 mins	10	3	0.25	0.35
$\omega(f_3)$	$\omega(f_4)$	ω(I)	ω(TD)	ω(IR)	ω(gs)
0.25	0.15	1	1	1	1

Based on the truth score of the Attack Scenario provided, only 649 of the alerts were false positives. We set the expected TPR, FPR and RR rate are 100%, 0%, and 20% respectively. In other words, this means that the metric should prioritise 2577 alerts, and eliminate 649 alerts. Using the parameters in TABLE II, OutMet prioritised 1,853 and filtered the remaining. A TPR of 71.91% and FPR of 0.32% were achieved.

Figure 3 shows a set of Alert Correlation Graps with their assigned priority values. Due to the frequent communication between servers on the network, many ICMP Ping and Reply alerts were triggered by the intrusion detection System. This resulted in many frequent alert correlation graphs such as 3(b) which capture the Ping-Reply behaviour between servers. Since these graphs were consistently observed within the intrusion alert traffic, a low priority value of 1 or 2 (depending on the number of pings and replies in a single graph) was often assigned to such a graph.

TABLE III
RESULTS FROM DMZ ATTACK ANALYSIS

	TPR	FPR	RR
Expected	1	0	0.20
OutMet	0.719	0.003	0.426
Alushbhi et al	0.95	0.184	0.202
OutMet *(No AC)	0	1	0

Fig3(a) illustrates an alert correlation graph which captures the behaviour of an outsider sending suspicious email to a client residing on the network. After studying the network topology and configurations it was discovered that packets were routed from the outsider to the DMZ mail server and from the DMZ mail server to the internal mail server were the mail content becomes available to the local client. Many graphs (although with variations of size and noise) captured this network behaviour. These graph patterns were less frequent than that observed in 3(b) and were often assigned a priority value of 2.

Finally, Fig3(c) shows the alert correlation graph of the real attack launched by the attacker on a DMZ web server. Most of the attacks in this graph were targeted to exploit web vulnerabilities. This graph was therefore assigned a higher priority value since it varied highly to the past observed behaviour (i.e. graphs) such as those in in Fig3(a) and Fig3(b).

The Alert Relation Metric proposed by Alsubhi et al prioritises alerts based on how similar an alert is to other alerts. [7].

A comparison between the results of Alsubhi et al and our work show that we have successfully reduced the false positive rate despite a lesser true positive rate. We discuss methods for improving the true positive rate in Section V.

In our research our argument is that alert correlation aids in capturing attack context. Therefore, analysing the output of alert correlation is more effective than analysing raw alerts without correlation. To validate this argument, we performed OutMet on low-level alerts without first correlating the alerts i.e. Each alert is assigned an OutMet value based on the degree to which it is an outlier. (Note that the initial results assigned OutMet values to alerts based on the degree to which its parent Meta-alert was an outlier). In this case, Euclidean distance is used as the distance measure between any two alerts. As the results show, no alerts were successfully prioritised.

V. FUTURE WORK

The future work of this research focusses around three core areas. Firstly, in some attack cases, outlier alerts many not correspond to real attacks, therefore OutMet may not be suitable for all attack scenarios. To address this, it is aimed to use other prioritisation metrics used alongside OutMet.

Secondly, our approach to prioritisation is based on static analysis, i.e. at every set interval, a set of recent meta-alerts are extracted from the database and their prioritisation values are calculated. Once done, prioritised meta-alerts are then flagged for the attention of a security analyst. Although low intervals could be near real-time it requires high computational power to run the prioritisation component frequently. On the other hand larger time intervals may result in detecting an attack too late. To address this, real-time prioritisation is being investigated which are based on Incremental local outlier detection[30].

Lastly, OutMet is highly reliant on the quality of the correlated alerts (i.e. meta-alerts). If meta-alerts contain heterogeneous alerts from different intrusive behaviours then it is highly likely that OutMet will produce a high false positive

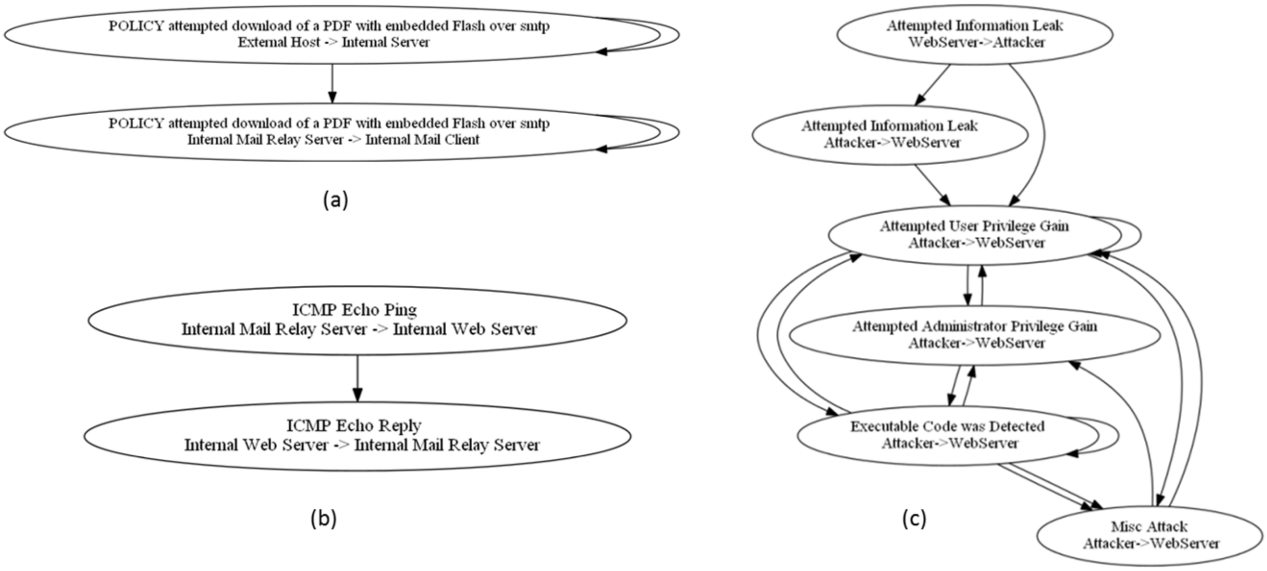

POLICY attempted download of a PDF with embedded Flash over smtp
External Host -> Internal Server
POLICY attempted download of a PDF with embedded Flash over smtp
Internal Mail Relay Server -> Internal Mail Client
(a)
ICMP Echo Ping
Internal Mail Relay Server -> Internal Web Server
ICMP Echo Reply
Internal Web Server -> Internal Mail Relay Server
(b)
Attempted Information Leak
WebServer->Attacker
Attempted Information Leak
Attacker->WebServer
Attempted User Privilege Gain
Attacker->WebServer
Attempted Administrator Privilege Gain
Attacker->WebServer
Executable Code was Detected
Attacker->WebServer
Misc Attack
Attacker->WebServer
(c)

Fig. 3. Sample Alert Correlation Graphs (Meta-alerts) derived from analysis with prioritisation values 2 assigned to graph(a), 1 to graph(b) and 4 to graph(c)

rate. We validated this observation by running an additional experiment where correlation threshold value was between 0.2 and 0.3. This produced a set of alert correlation graphs which contained a high noise rate. Thus the prioritisation analysis was highly affected. A possible method to address this is to improve the correlation component [23], [31]. More statistically refined alert correlation techniques including probablistic techniques have been recently proposed[24], [25], [32].

Our research also focuses on investigating new methods for prioritising and particularly understanding intrusive behaviour. More recently, we have focussed on extracting attack patterns from alert correlation graphs and using such patterns as the basis for labelling attack classes.

VI. Summary

In this research we have presented a new prioritisation metric. We have argued and proven that (in some cases), true attacks correlate with outlier activity therefore the OutMet prioritisation metric can be useful. Our results support this claim and using a recently developed dataset to measure its performance, we achieved a TPR of 71% with less than 1% false positives. The metric is currently being further tested and integrated into an experimental Cyber Analysis tool developed at the BT Security Future Practise Research Labs.

VII. Acknowledgements

This work is being supported and funded by the British Telecommunications Security Future Practice Group and the Centre for Cyber Security Sciences(CCySS) at City University London and the Engineering and Physical Sciences Research Council (EPSRC).

Our appreciation also extends to Northrop Grumman who provided the cyberange data for the experiment.

References

[1] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," *Proceedings of the 6th ACM conference on Computer and communications security - CCS '99*, pp. 1–7, 1999. [Online]. Available: http://portal.acm.org/citation.cfm?doid=319709.319710

[2] K. Julisch and M. Dacier, "Mining intrusion detection alarms for actionable knowledge," *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, p. 366, 2002. [Online]. Available: http://portal.acm.org/citation.cfm?doid=775047.775101

[3] E. M. Hutchins, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," no. July 2005, pp. 1–14, 2008.

[4] S. Salah, G. Maciá-Fernández, and J. E. Díaz-Verdejo, "A model-based survey of alert correlation techniques," *Computer Networks*, Jan. 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1389128612004124

[5] H. Winter, "System security assessment using a cyber range," *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012*, pp. 41–41, 2012.

[6] K. Alsubhi, E. Al-Shaer, and R. Boutaba, "Alert prioritization in Intrusion Detection Systems," *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*, pp. 33–40, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4575114

[7] K. Alsubhi, I. Aib, and R. Boutaba, "FuzMet : a fuzzy-logic based alert prioritization engine for intrusion detection systems," 2011.

[8] P. A. Porras, M. W. Fong, and A. Valdes, "A Mission-Impact-Based Approach to INFOSEC Alarm Correlation," *Recent Advances in Intrusion Detection*, no. Springer Berlin Heidelberg, pp. 95–114, 2002.

[9] S. Noel and S. Jajodia, "Attack Graphs for Sensor Placement , Alert Prioritization , and Attack Response," pp. 1–8.

[10] L. Zomlot, S. C. Sundaramurthy, K. Luo, X. Ou, and S. R. Rajagopalan, "Prioritizing intrusion analysis using Dempster-Shafer theory," *Proceedings of the 4th ACM workshop on Security and artificial intelligence - AISec '11*, p. 59, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2046684.2046694

[11] S. C. Sundaramurthy, L. Zomlot, and X. Ou, "Practical IDS alert correlation in the face of dynamic threats," *International Conference on Security and Management (SAM'11)*, 2011.

[12] F. Cuppens and R. Ortalo, “LAMBDA: A Language to Model a Database for Detection of Attacks,” pp. 197–216, Oct. 2000. [Online]. Available: http://dl.acm.org/citation.cfm?id=645838.670728

[13] S. Cheung, M. W. Fong, R. Ave, and M. Park, “Modeling Multistep Cyber Attacks for Scenario Recognition,” *In DARPA Information Survivability Conference and Exposition (DISCEX III)*, no. DISCEX III, pp. 284–292, 2003.

[14] G. V. Steven Eckmann , “STATL: An Attack Language for State-based Intrusion Detection.” [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.7004

[15] L. M. Cédric Michel, “Adele: An Attack Description Language For Knowledge-Based Intrusion Detection.” [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.8821

[16] A. Valdes and K. Skinner, “Probabilistic Alert Correlation,” *In Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pp. 54–68, 2001.

[17] O. Dain and R. K. Cunningham, “Fusing a Heterogeneous Alert Stream into Scenarios,” *In Proceedings of the 2001 ACM workshop on Data Mining for Security Applications*, pp. 1–13, 2001.

[18] A. Hofmann and B. Sick, “Online Intrusion Alert Aggregation with Generative Data Stream Modeling,” *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, vol. 8, no. 2, pp. 282–294, 2011.

[19] P. Ning, D. S. Reeves, and Y. Cui, “Correlating Alerts Using Prerequisites of Intrusions,” Tech. Rep.

[20] H. Debar and A. Wespi, “Aggregation and Correlation of Intrusion-Detection Alerts,” *Recent Advances in Intrusion Detection.*, pp. 85–103, 2001.

[21] X. Qin, “A Probabilistic-Based Framework for INFOSEC Alert Correlation,” Ph.D. dissertation, Georgia Institute of Technology, 2005.

[22] S. H. Ahmadinejad and S. Jalili, “Alert Correlation Using Correlation Probability Estimation and Time Windows,” *2009 International Conference on Computer Technology and Development*, no. 1, pp. 170–175, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5360130

[23] H. Ren, N. Stakhanova, and A. A. Ghorbani, “An Online Adaptive Approach to Alert Correlation,” *Proceedings of the 7th international conference on Detection of Intrusions and malware, and vulnerability assessment (DIMVA)*, pp. 153–172, 2010.

[24] M. Marchetti, M. Colajanni, and F. Manganiello, “Identification of correlated network intrusion alerts Pseudo-Bayesian,” *Cyberspace Safety and Security (CSS), 2011 Third International Workshop on*, pp. 15–20, 2011.

[25] S. Benferhat, A. Boudjelida, K. Tabia, and H. Drias, “An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge,” pp. 520–540, 2013.

[26] P. Ning and D. Xu, “Learning attack strategies from intrusion alerts,” *Proceedings of the 10th ACM conference on Computer and communication security - CCS ’03*, p. 200, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=948109.948137

[27] L. G. Shapiro and R. M. Haralick, “Structural descriptions and inexact matching.” *IEEE transactions on pattern analysis and machine intelligence*, vol. 3, no. 5, pp. 504–19, May 1981. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21868971

[28] M. M. Breunig, H.-p. Kriegel, R. T. Ng, and J. Sander, “LOF : Identifying Density-Based Local Outliers,” *Proceedings Of The 2000 Acm Sigmod International Conference On Management Of Data*, pp. 1–12, 2000.

[29] M. Bateni and A. Baraani, “Time Window Management for Alert Correlation using Context Information and Classification,” no. September, pp. 9–16, 2013.

[30] D. Pokrajac and E. Hartford, “Incremental Local Outlier Detection for Data Streams,” no. April, 2007.

[31] M. Bateni, A. Baraani, A. Ghorbani, and A. Rezaei, “An ais-inspired architecture for alert correlation,” vol. 9, no. 1, pp. 231–255, 2013.

[32] M. Bateni, A. Baraani, and A. A. Ghorbani, “Using Artificial Immune System and Fuzzy Logic for Alert Correlation,” vol. 15, no. 1, pp. 160–174, 2013.