

# The application of advanced analytics for security





# Introduction

**The Telstra Advanced Security Analytics (ASA) solution applies advanced mathematical models and machine learning to security. The Telstra ASA approach to security is outlined in this paper.**

Most organisations now accept that their networks are compromised in some way. From the proliferation of devices, delivery mechanisms for targeted attacks, and the constant and evolving nature of attacks, security staff have a near impossible job of detecting sophisticated and highly evolved threats. From a security perspective, we should all accept that organisations are already compromised by unknown elements hiding within the network.

Consequently, there are no clean, uncompromised data to learn about the uncompromised behaviour of the network. This motivates the use of unsupervised anomaly detection based on stochastic machine learning models. The ASA algorithms are designed to detect the unusual; they are not designed to explicitly detect the malicious. Whilst it is true that unusual does not imply malicious, the malicious can leave unusual traces.

To improve the chance of finding the malicious amongst the unusual, the algorithms are applied to different log types or data sources, and at different levels of network resolution and time scales. This operationalises an adaptation of a familiar maxim: the malicious can hide all of its traces some of the time, or some of its traces all of the time, but it cannot hide all of its traces all of the time.

# Overview of Telstra ASA algorithms

The ASA Detection and Fusion Engines consist of a number of stochastic machine learning algorithms applied to several network log types. Figure 1 provides an overview of the three layers that make up the ASA analytics landscape - algorithms, analysis types and log types - and how the components in these layers come together.

One or more algorithms are applied to one or more log types to form an analysis type. For example, the Periodicity Algorithm is used in Periodicity Analysis on the DNS and DHCP logs; the Mixed-Feature Anomaly Detection and Pattern Mining Algorithms are used in Behaviour Analysis on the Proxy and DHCP logs; and so on.

The ASA analysis types target different stages of the attack timeline. Here, a simplified three-stage version of the attack spectrum will be used. Periodicity Analysis targets the early stage of a compromise landing on the network. Behaviour Analysis, Login Analysis and Interaction Analysis target the intermediate surveillance and reconnaissance stage. Interaction Analysis and Traffic Analysis target the final data exfiltration stage.

In addition to targeting different stages of the attack timeline, the ASA analysis types are applied to different log types and operate at different network resolutions and time scales. This improves the chance that ASA detections will include unusual traces from malicious activities.

*It must be emphasised that Figure 1 shows one possible ASA configuration involving a specific selection of algorithms applied to a particular choice of log types. ASA can be flexibly re-configured by swapping new or existing algorithms in or out, and by applying algorithms in a different way to existing logs or to new logs or even to other data sources. For clarity of exposition, this article will provide a technical description of ASA based on the configuration given in Figure 1.*

ASA algorithms are founded on mathematical models that are flexible, can handle diverse data types, and do not make unrealistic assumptions about the data. The algorithms are “stochastic” algorithms, in contrast to “deterministic” rule-based (know what “normal” looks like) or signature-based (know what “bad” looks like) algorithms. Stochastic, from the Greek “stochastikos” meaning “skilled at aiming”, describes that which cannot be pinned down precisely. A stochastic algorithm contains a random element that allows it to adapt to the variability and uncertainty that are inherent in data. Thus, ASA algorithms are not rule-based or signature-based, but are complementary to these.

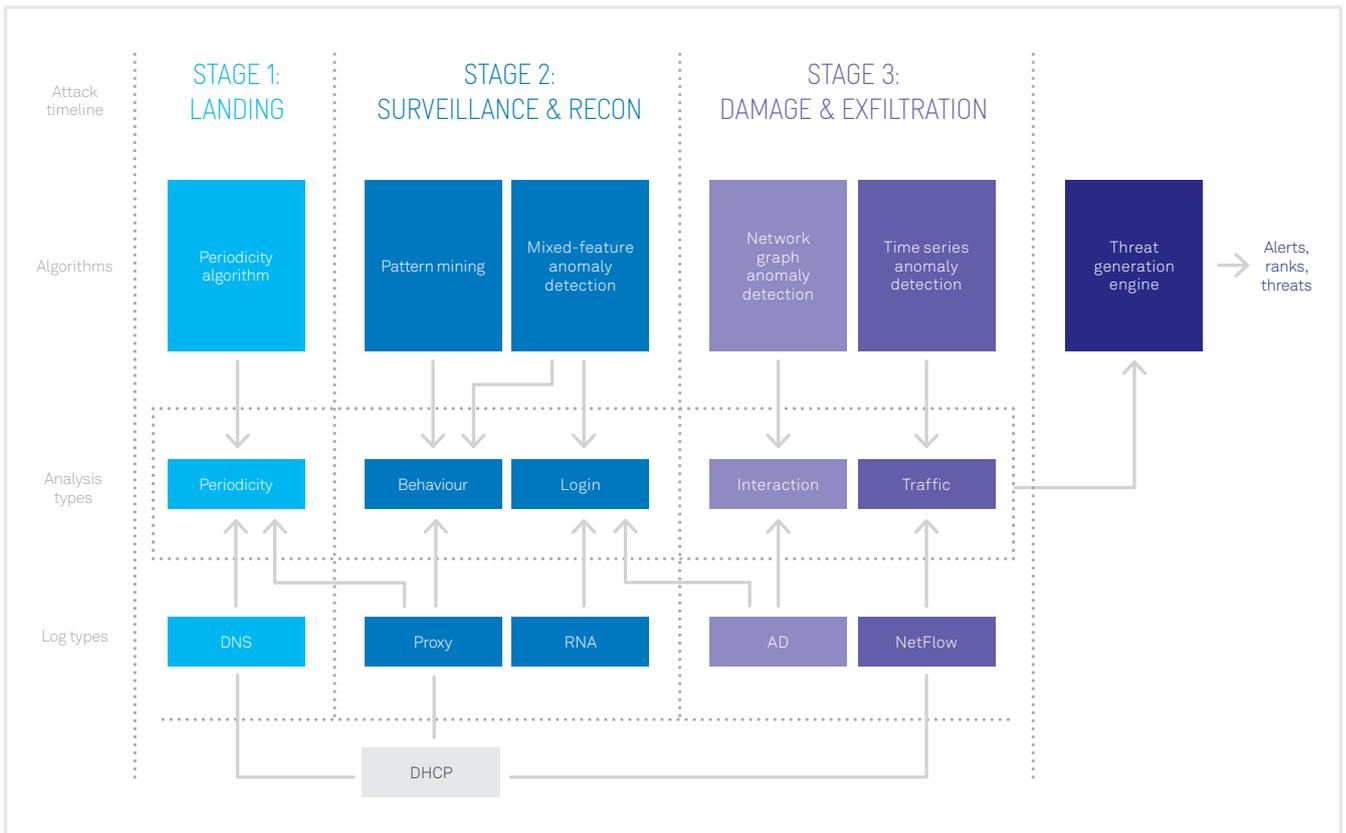


Figure 1: Overview of the ASA algorithms, analysis and log type configuration

## Generating Alerts

Except for the periodicity algorithm and the Fusion algorithm, the other ASA algorithms are anomaly detection algorithms. Each analysis type produces scored detections, which (using the Fusion Engine) are then standardised to make them comparable across analysis types. The standardised scores are thresholded to give prioritised alerts. Detections are combined to form alerts by grouping together the detections that are linked to an entity (e.g. a user, device, MAC address, IP address or domain). Alerts can then be prioritised based on

their alert scores. Figure 2 summarises the processes of score standardisation, detection formations and alert integration.. In addition to providing prioritised detections and alerts, relevant context information is available to the analyst to assist in their investigation. An additional context tool also allows the analyst to retrieve additional data deemed necessary to better understand the detections and alerts. The following sections provide more details about each algorithm/analysis type used in the configuration in Figure 1.

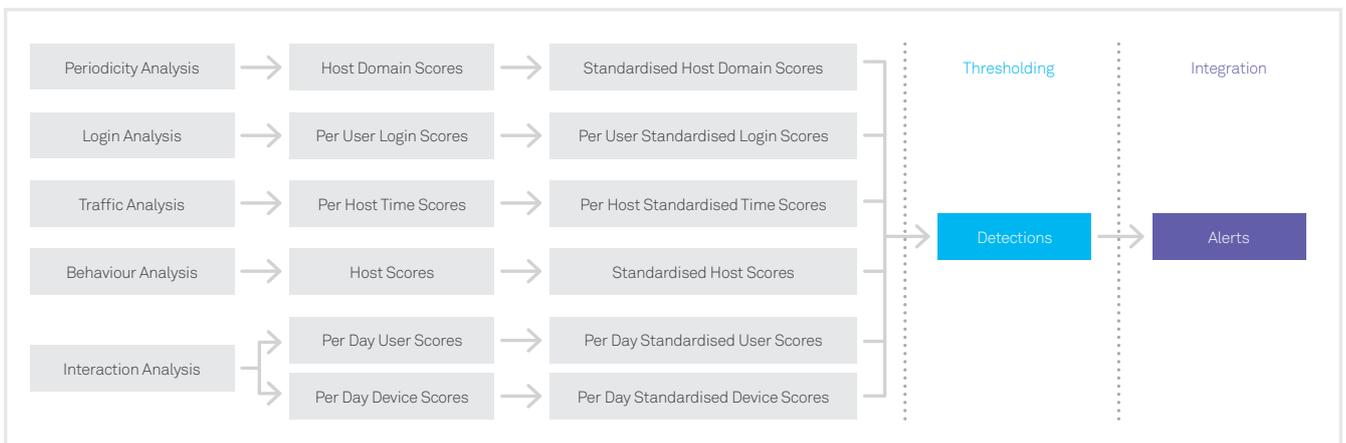


Figure 2: ASA Fusion performs score standardisation, detection formation and alert integration



# Periodicity algorithm and analysis

The Periodicity Algorithm is designed to detect a periodic pattern, i.e. a highly regular sequence of events. It identifies machines that exhibit strong periodic communication patterns, as well as the destination IPs that these machines are communicating with. When a periodic sequence occurs, it may have some events missing, and may be interspersed amongst other events that do not follow the periodic pattern. Figure 3 illustrates the detection of periodic events interspersed amongst other events. To cope with practical difficulties like this, the algorithm uses a smart selection of candidate periods, modulo arithmetic and distributional analysis.

The Periodicity Algorithm assigns a periodicity score to a sequence of events, quantifying how periodic the sequence is. A periodicity detection is raised when a periodicity score indicates that a sequence is more likely than not to be periodic. This may be caused, for example, by a benign activity such as a scheduled software update, or by malicious software beaconing back to its command and control.

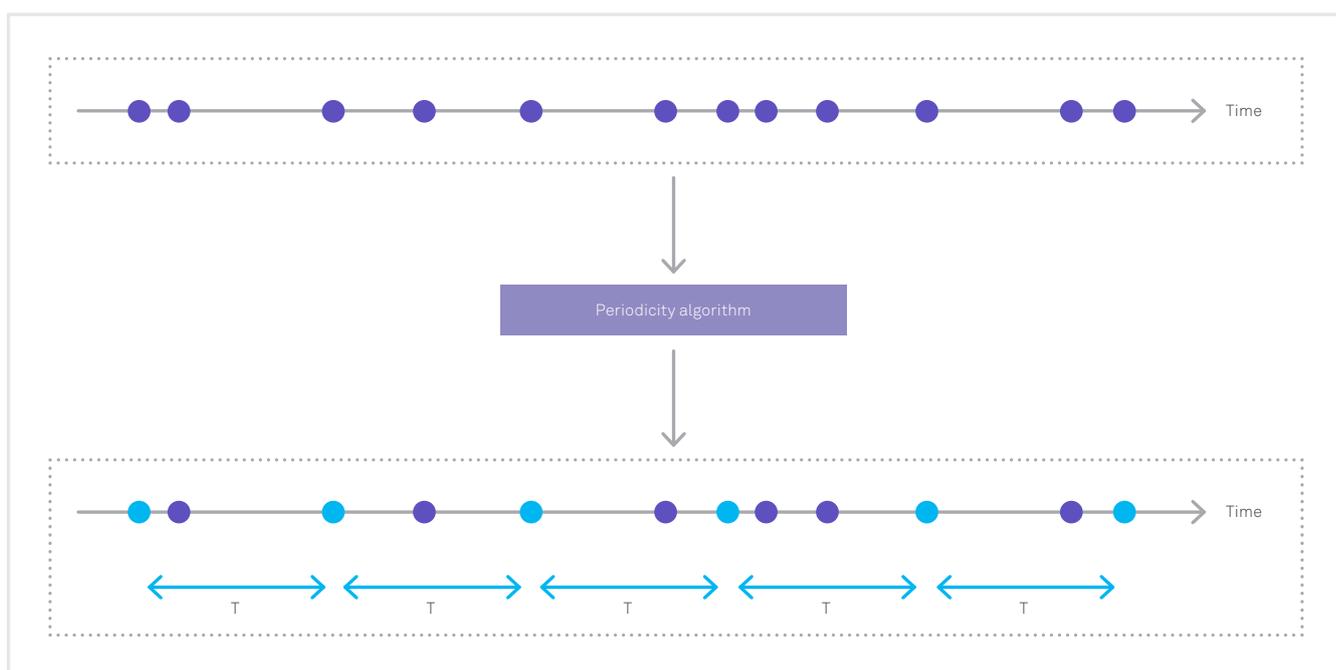


Figure 3: Detection of embedded periodic events (blue dots) by the Periodicity Algorithm

# Mixed-feature anomaly detection

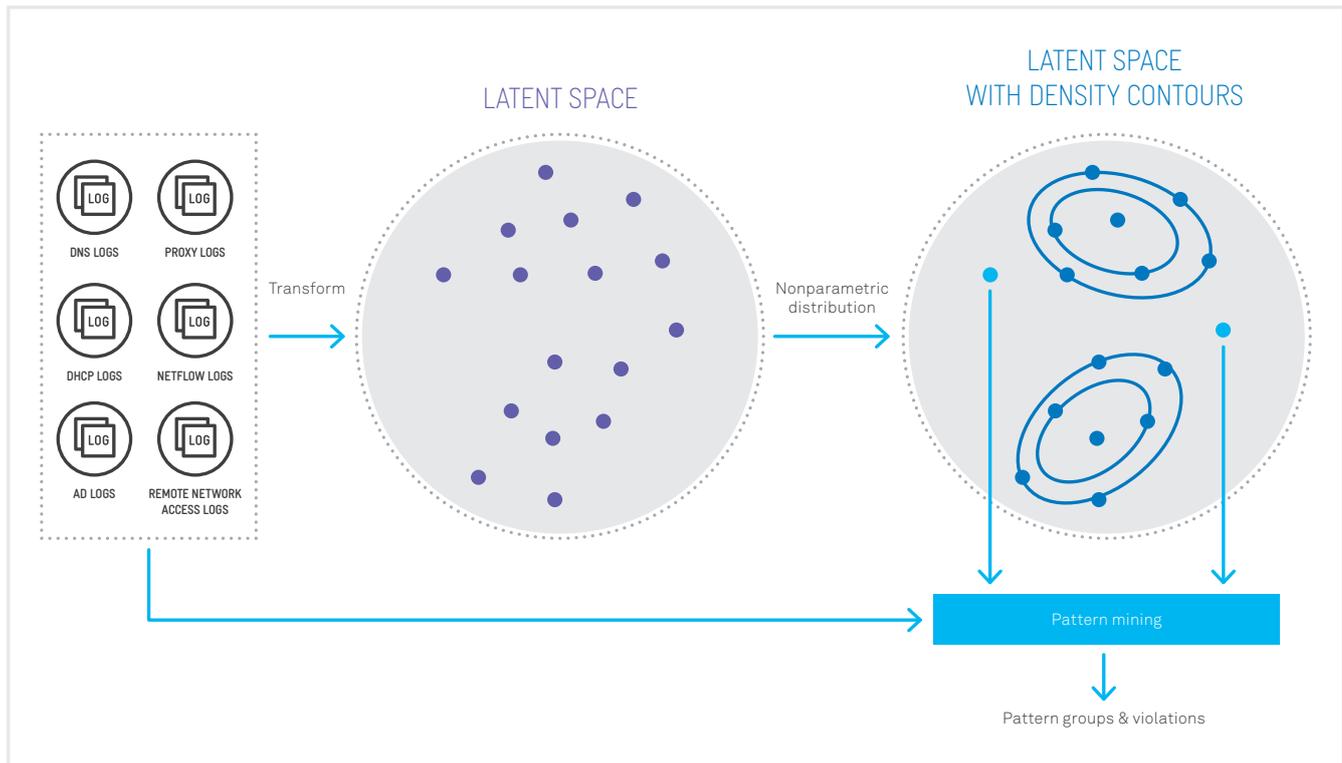


Figure 4: The mixed-feature anomaly detection algorithm

As its name suggests, Mixed-Feature Anomaly Detection is an algorithm that performs anomaly detection on data that may contain features of mixed types.

The algorithm combines a latent space model and a nonparametric distribution model as illustrated in Figure 4. Each data point is represented by a feature vector, and the feature vectors for a set of data points are transformed into points in a latent vector space (Euclidean space). The resulting latent points are illustrated as blue dots in the middle part of Figure 4. This latent space approach provides flexibility by allowing anomaly detection to be performed in a vector space (Euclidean space), thus avoiding constraints that may exist in the original data space or feature space. A latent space model

can provide noise reduction for better detection capabilities and, depending on the transformation method, may also provide dimensionality reduction. After the feature vectors have been transformed to a latent space, a nonparametric distribution is fitted to the latent points. The fitted distribution is used to assign an anomaly score to each point, indicating how unusual a point is relative to other points, in terms of information contained in the feature vectors. This is illustrated in the right side of Figure 4, where unusual points with high scores are depicted as red dots, and other points as green dots.

The latent space model and distribution model are both nonparametric models, which avoid unrealistic assumptions about the data thus allowing the data to speak for itself. Login Analysis and Behaviour Analysis provide two examples of the use of Mixed-Feature Anomaly Detection.

## Login Analysis

Login Analysis uses Mixed-Feature Anomaly Detection to detect unusual login events for individual users. In this case, the dots in Figure 4 represent login events for an individual user, providing a user-specific login profile that is used to identify logins at times that are unusual for that user. This analysis is directed at unauthorised access that is characteristic of the second stage of a cyber attack, during which the attacker conducts surveillance of the compromised network. The anomaly score assigned to a login event indicates how unusual that event is relative to other login events for that user.

A login alert is a login event that has a score indicating that it is more likely than not to be unusual, due to the login event occurring at an unusual time for that user. Figure 5 provides an example of results produced by Login Analysis for a user. Each vertical line in the figure represents the occurrence of a login event – the value on the x-axis is the event time expressed as time-of-day, and the height is its anomaly score. Green lines are workday events while red lines are non-workday events. Two high-scoring unusual non-workday events are indicated in the figure.

## Behaviour Analysis

Behaviour Analysis uses Mixed-Feature Anomaly Detection to detect entities with unusual behaviour within a group of entities. It creates a group behaviour profile and uses it to identify entities whose behaviours deviate from the group behaviour. Behaviour Analysis detects the departure of the behaviour of an individual entity from group behaviour, which may be caused by surveillance activities orchestrated by an attacker. An anomaly score assigned to an entity indicates how unusual that entity is relative to other entities in the group. A behaviour alert points to

an entity whose behaviour is more likely than not to be unusual.

In Figure 4, each dot now represents an entity's behaviour, and entities with similar behaviours are located closer together in the latent space. The example in the figure shows two behaviour alerts depicted as red dots. These, together with the raw logs, are then further processed by the Pattern Mining Algorithm to determine how those alerted entities' behaviours differ from the group behaviour.

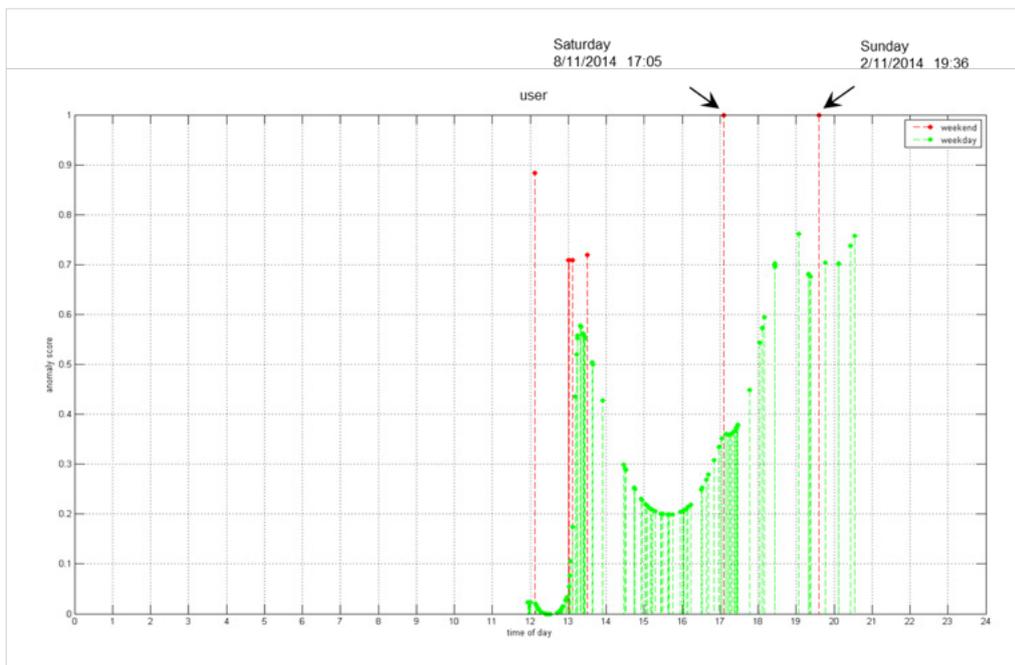


Figure 5: Login analysis example

# Network graph anomaly detection & interaction analysis

Network Graph Anomaly Detection produces anomaly scores for dynamic relational data represented as a time-sequence of network graphs.

A network graph consists of a set of nodes representing entities, and connecting edges between nodes representing relationships between entities. The algorithm uses a sequence of latent space models to assign anomaly scores to nodes at each time-step in the sequence.

Figure 6 provides an example of how the algorithm is used for Interaction Analysis, where it detects change in network

interaction patterns between two groups of entities – users and machines in this case. The daily interactions between users and machines are represented by a network graph for each day. Each network graph is then transformed to points in a latent space. To obtain anomaly scores for entities at the current day (day d in the figure), a reference profile is created based on a number of days (3 days in the

figure) leading up to the current day. The latent points for the current day are then compared against the reference profile to identify changes in interactions; for example, a user connecting to machines not usually connected to, which may be due to unauthorised access to those machines using stolen credentials.

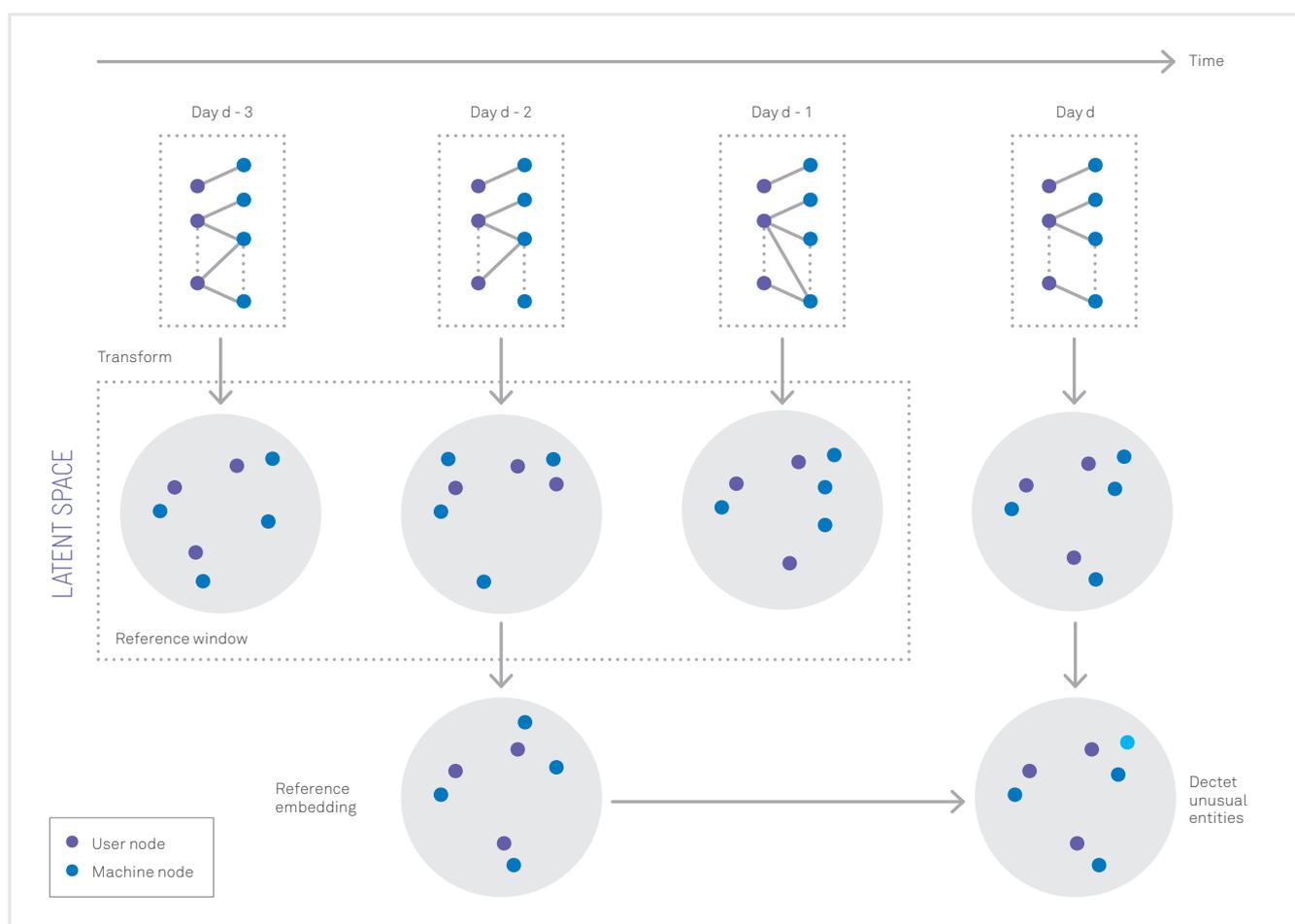


Figure 6: Network graph anomaly detection and interaction analysis



# Time series anomaly detection & traffic analysis

Time Series Anomaly Detection uses a nonparametric time series model to detect anomalies in a time-sequence of dynamic events. An example of its use is in Traffic Analysis to detect unusual data traffic patterns for individual devices. It creates a device-specific data traffic profile that is used to identify unusual traffic events and their times of occurrence. The goal is to detect unusual traffic events, especially outbound events that may be indicative of data exfiltration.

The algorithm takes a time-sequence of traffic events and transforms them into points in a reduced-dimension latent space. In Figure 7, the observed traffic events are depicted as vertical blue lines in the top panel. The heights of the lines represent the volumes of data transmitted. The blue dots in the middle part of the figure represent the transformed latent points. These are used to obtain a smoothed version of the time series as well as estimates of local variabilities within the time series. In the bottom panel of the figure, the smoothed series

is overlaid as a red curve on the observed time-sequence. This red curve is surrounded by a pair of green curves, one above and another below the red curve. The vertical separation between the pair of green curves illustrates how quickly data volumes are changing locally in time – the wider the separation, the quicker the change. The observed time-sequence, smoothed version and local variabilities are used to obtain an anomaly score for each traffic event.

For each device, the anomaly score assigned to a traffic event indicates how unusual that event is relative to other traffic events within a time neighbourhood around that event. A traffic alert is a traffic event whose score indicates that it is more likely than not to be unusual. This may be caused, for example, by an event containing an unusually large volume of outbound data.

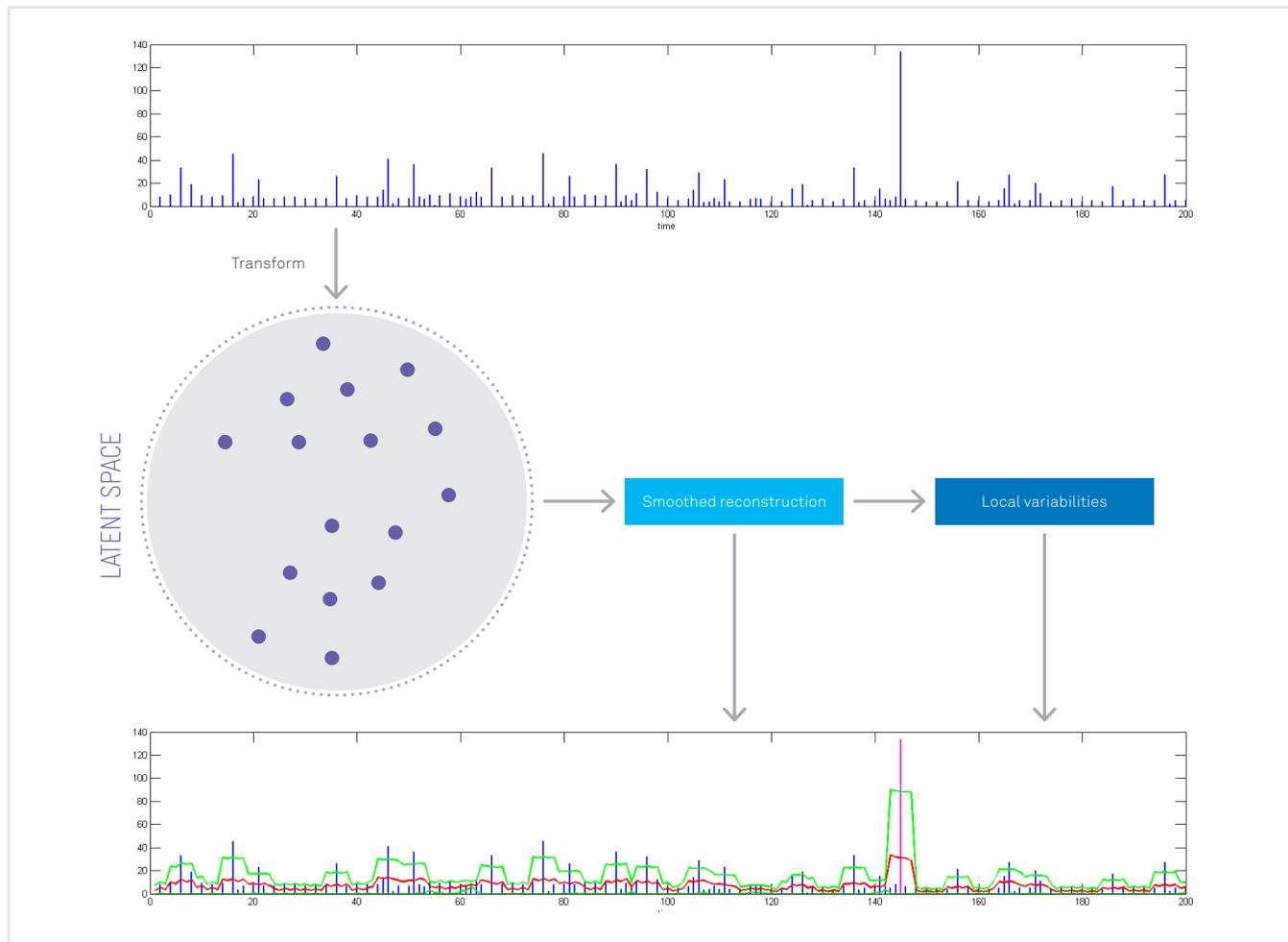


Figure 7. Time series anomaly detection and traffic analysis

# The ASA fusion engine

Periodicity Analysis produces periodicity scores while the other analysis types produce anomaly scores.

Depending on the algorithms used, some of the scores produced are probabilities whilst others are not; hence, the resulting scores are not immediately comparable. To make them comparable, the scores must be standardised. The ASA Fusion Engine standardises scores across detections and then integrates those scores to a relevant entity.

Score standardisation is based on the automatic control of false discovery rates, defined as the conditional probability that an entity/event is not unusual given that an anomaly detection algorithm decides that it is. Thus, controlling false discovery rates helps to reduce false positives. Given a set of scores from an anomaly detection algorithm, score standardisation uses a multiple testing approach that considers the number of scores and their distribution.

The standardised scores are thresholded to give detections. A detected entity or event indicates that it is more likely than not to be unusual, or periodic in the case of a periodicity detection.

Detections are combined to form alerts by grouping together detections that are linked to an entity (e.g. a user, device, MAC address, IP address or domain). To combine detections into alerts, ASA uses the maximum operation, which assigns the largest standardised score as the integrated alerts score. This equates to picking the strongest signal to be the representative combined signal.

The use of the maximum operation also prevents the dilution of strong detections which can lead to a diminished sense of urgency. The resulting alerts can then be prioritised based on their alert scores. When a list of high-value network assets or high-privilege users is available, a separate high-priority alert list can be highlighted to users using this scoring information.

# Conclusion

Three words characterise ASA and, together, they differentiate the ASA solution from others: *unsupervised, multiplicity, re-configurable*.

ASA uses unsupervised anomaly detection based on stochastic machine learning algorithms to detect new and unknown security alerts. To improve the chance of detection, ASA uses multiple algorithms applied to multiple data sources, at multiple levels of network resolution, and at multiple time scales.

The task of detecting new and unknown security threats has been likened to finding a needle in a haystack. ASA is a magnet for this task. ASA can be flexibly re-configured by swapping in new algorithms and swapping out existing algorithms. These algorithms can be applied in a different way to existing logs or to new logs or to other non-log data sources, such as organisational groupings and network infrastructure data, building access data, outputs from other security tools, and so on.

The flexibility provided by the ASA architecture to constantly add new algorithms and utilise new data sources will allow ASA to respond to specific new requirements from customers.



-  contact your Telstra account executive
-  1300 835 787
-  [www.telstra.com/enterprisesecurity](http://www.telstra.com/enterprisesecurity)

