International Journal of Computing and Network Technology

http://dx.doi.org/10.12785/ijcnt/010202

Network Attack Analysis and the Behaviour Engine

Anthony Benham[1], Huw Read[2], Iain Sutherland[3]

University of Glamorgan,UK
[1]Email Address: apbenham@glam.ac.uk
[2]Email Address: holread@glam.c.uk
[3]Email Address: isutherl@glam.ac.uk

Received: 12 Jan. 2013, Revised: 20 Feb 2013; Accepted: 13 Mar. 2013
Published online: 1 Apr. 2013

Abstract: Behaviour Engines allow the acquisition of tacit knowledge by using a learn-by-doing workflow and provide a direct interface between the expert user and the developing project code based on an intuitive justification-conclusion language; thus surpassing legacy policy engines by being a self developing and learning mechanism. This paper seeks to formulate the current state of the art in technology and processes and attempts to merge the application of ontological decision techniques of behaviour engines with network packet capture data, to detect data exfiltration attempts over covert channelling. The final goal of the research will be to develop a behaviour engine/intrusion detection solution for pre-emptive counter-measures to anomalous behaviour from within or without a network.speed.

Keywords: Behaviour Engines, Intrusion Detection/Prevention, Advanced Persistent Threats, Covert Channels, Data Exfiltration

I. Introduction

The complex nature of network topology and the ever progressive driving force for interconnection of networks, has led to the creation of ’attractive targets’ for malicious would be attackers. The attacker and his attack tools have developed rapidly in recent times, making computer network penetrations a serious issue for many businesses and organizations. The technology of network defence must necessarily advance to counteract this tide and strive to maintain the the continuity of the business process. Rapid and accurate identification of anomalies is critical to the efficient operation of large computer networks. Intrusion detection and intrusion prevention systems amount to a suite of techniques that are used to identify attacks against computers and network infrastructures. Anomaly detection is an essential ingredient of intrusion detection [1], where deviations from previously defined ’normal’ behaviour could suggest the presence of deliberate or unintentional, insider or outsider-activated exploits or attacks. Discovered and categorized attacks, that fall outside the ’norm’ are all fed into a policy database,where upon decisions are carried out by network administrators for mitigation of these attacks in the future. Research into Network Situational Awareness and Computer Network Defense has thrown up many areas for consideration but one critical area that has emerged is that of Covert Channel operation and in particular, the ‘stealth’ activity of Data Exfiltration.

Often in the design of a network, it is assumed that data encryption of network traffic is sufficient for maintaing the integrity and confidentiality of the data stored and transmitted.

Encryption however only prevents unauthorized persons from being able to decode and read the data being transmitted. This is where Covert Channels come into their own. Covert channels are used to hide the very existence of information transfer and often they are implemented through the manipulation of a communication protocol in a way that lies outside its specification. The vast stores of data and large number of different protocols in the Internet provides a consummate, high bandwidth medium, for concealed communication.

Currently, when a new attack is discovered, the developer submits the attack-signature code changes to the rule repository and the changes are integrated into the system. This process could take anywhere from a day to several weeks to implement depending on a large number of factors including the scale of any changes made, concurrent projects and other developers having current ownership of code blocks, not to mention changes in certain rules causing other rules to fail. This paper examines the possibility of using a Behaviour Engine for the detection of anomalous behviour on a network. A behaviour engine lets one start with no rules at all. As the user (probably not an IT developer) perform a task or takes a decision or a conclusion, the software asks them to justify their action. This justification can be made in a variety of ways, using graphical user interfaces or plain text. The uniqueness of the the behaviour engine, is that it automates the task of turning conclusions and justifications into rules, while naturally incorporating both explicit and tacit knowledge into the fast developing rule set. An example of behaviour engine implementation was the Security Cockpit project protecting European communication systems. The engine was implemented within an inference engine model, deducing whether an attack was taking place on the cockpit or not [2]. We propose a system architecture for the detection of data exfiltration, using multi vector attack profiling via the application of a behaviour engine.

RELATED WORK

I. NETWORK INTRUSION DETECTION AND PREVENTION

An intrusion detection system (IDS) inspects all inbound and outbound network or system activity and recognise suspicious patterns or events that may signify a network or system attack, intrusion or attempted compromise of a system. Network intrusion detection systems are considered an effective second line of defense, after Firewalls, against network-based attacks directed at computer systems [3], and are being employed in large-scale IT infrastructures, due to the rising asperity and tendency of such attacks. IDS’s differ from Firewalls in that a Firewall looks out for network intrusions from without (one-way facing) and limits connectivity between networks to reduce possibility of attacks taking place. An IDS on the other hand evaluates for possible intrusions from within and without the network and signals and alarm. There are several methods for differentiating IDS’s; signature-based vs anomalybased; network-based vs host-based and passive vs reactive [4]. Attacks and intrusions are developing so quickly that any IDS worth its salt must be flexible enough to be implemented at a host and network level; and reactive IDS’s have developed to become Intrusion Prevention Systems (IPS). Intrusion Prevention Systems as discussed earlier, generally implement the same or similar algorithms as IDS’s and are reactive, as well as being either signature-based or anomaly-based.

Within the IPS world, many attempts have been made to improve upon the detection of anomalies. Kruegel et al[5]introduce the concept of Baysian Network statistical formulization to attempt to detect anomalies.

Bayesian networks improve the aggregation process and facilitate for the input of additional information relating to an event, both of which paint a more comprehensive picture of the event being analysed. Another example of an attack detection tool is the Hugin Tool [6]- a tool for probabilistic graphical models such as Bayesian networks. A discussion of the capabilities and knowledge based functionality of this tool, in relation to the automated construction of Bayesian networks is given in . The Hugin tool achieves its ends through Bayesian Network parameter and structure learning.

The concept of Cyberspace Situation Awareness (CSA) was proposed for the first time by Tim Bass in 1999 [7], intended to introduce Situational Awareness technologies into the field of network management and network security. The aim of this concept is to organise the very complex information of largescale networks into a far more efficient and utilisable manner, hence speeding up the decision-making processes. This is for example where the Niche Theory comes into its own in the realm of network analysis and CSA. Zhuno et al[8]introduce this concept for large dynamic altering systems with time and propose Situation Niche utilisation of relational metrics such as topology change, network congestion, traffic, frequency of use, fault, attack, security, usability, information superiority, availability, disaster and emergency just to name a few; anyone of which could alter the status of the network.

Jason Shifflet[9] proposes the idea of amalgamating many areas of Intrusion Detection and intrusion prevention to produce a Cyber Situation Awareness environment in which data is fused together to create a defense-in depth system that is independent of a single technique and Jibao et al[10] show the design of a network security situation awareness model (NSAM) based on additive weight analysis and Grey theory, while Issariyapat and Fukuda[1] produced a study of the application of Principle Component Analysis and Sample Entropy to IP network anomaly detection. PCA is based on a reductive system that takes the data set and reduces it into principle components of variance.

II. Problems With Modern IDS/IPS Systems

’Normal’ profiles are either static or dynamic. Static profiles are fixed unless the IDS is altered deliberately. Dynamic profiles are adjusted constantly as additional events are observed. Due to the dynamic nature of networks where events, nodes, users, peripherals and connections are changing all the time; then a static profile will eventually become inaccurate and will need to be regenerated. Dynamic profiles however overcome this problem but they are susceptible to evasion techniques. If an attacker carries out small amounts of malicious activity, which they increase slowly over time, then provided the rate of increase in these activities remains sufficiently low enough over a period of time, the dynamic profile adds these events into the ’normal’ picture and thus the activities become false positives. In fact intrusions and malware might well be present when the normal profile is built and thus they will never be detected. Additionally, anomaly-based systems often produce a large number of false positives due to the legitimate activities that lie outside the norm being found in dynamic networks. For example, performing a whole-scale back up of the network, involving very large amounts of rapid data transfer would trigger an alert.

Signature-based detection is the simplest detection method because it simply compares the current unit of activity, such as a packet or a log entry, to a list of signatures using string comparison operations. Signature-based detection technologies do not encompass the monitoring of complex communications or network protocols, cannot detect anomalous or zero-day attacks, nor event discover unauthorised internal user connections. For example, they cannot pair requests with their responses nor can they relate together various sequential events, which in themselves seems benign but together they form an attack[11].

Anomaly IPS, designed for this purpose still falls short of achieving the required result success of detection, due to the fact that often they are composed of multiple, incongruent algorithms for various aspects of an event, requiring hugely convoluted interlinked rule-sets when implemented. Additionally, 'anomaly IPS/IDS systems rely on the subjacent concept of 'normality' within the core or edges of the network. 'Normality' is defined using a relational model of the dynamic variables affecting the network state and an event is defined as anomalous, if the variation of its characteristics from the 'normal' network behaviour is too large, for network-unique preset limits, [12]. Setting these limits and defining 'normal' behaviour is so difficult and complex, often leading to many false positives for a stringent security paradigm. These problems are compounded when attempting to detect the modern attack form of covert channeling for the purpose of data exfiltration.

The Hugin model [6] for example, is a process of fusion of both observational data and domain expert knowledge. This knowledge is translated from the business experts, by the software engineers, into knowledge bases which are usually rules that are implemented in the business network. It is vital therefore that a common business-software language is used by developer to ensure that efficient and accurate translation of knowledge takes place. However, when a series of attacks or intrusions are being experienced, the administrator still has to implement strategies for every small variation of these attacks. A behaviour engine would learn the nature and characteristic of such attacks and will eventually automate response giving the end user options for implementation.

A. Advanced Persistent Threats (APT's)

Data exfiltration from a network through covert channel operation is one form of a recent threat type known as Advanced Persistent Threats. This new category of cybercrime is directed at governments and businesses [13]. They are low-level attacks, which have been practiced by individuals previously, but are now used collectively to fire very targeted and prolonged attacks, to gain maximum access and control of an organisation's IT infrastructure and data. APTs require a high degree of stealth (hence covert channel implementation) and time to be successful [14].

APT's are:

- Advanced - perpetrators of APT's utilise the full range of technologies and tools available as well as being prepared and able to write new software/code where necessary, to achieve their objectives.
- Persistent - attackers are not simply opportunistic or financially motivated but are driven and focused by external forces. They do not bombard networks to bring them down but rather intrude slowly and stealthily.
- Threat - attackers are well motivated, co-ordinated, skilled and funded with deliberate objectives to fulfil.

  APT attacks penetrate networks via external attack vectors such as vulnerability exploitation and internet/physical malware infection, and internal attack vectors such as insider threats and trusted connections[14]. Distinguishing characteristics of APT's are:

- Embedded email threats – Using spoofed email addresses and domains, perfectly acceptable email messages with no attachments are sent, and pass through firewalls undetected. These emails include embedded URLs that link to an infected Web page or an embedded object that upon being clicked on, drops a Remote Administration Tool (RAT) for control by an external Command & Control (C&C) server [15].

- Leverage insider threat and trusted connections, such as unprotected third parties to plant RATs system backdoors.
- Legitimate website are infected with cross-site scripting and stolen FTP credentials. Backdoor downloaders, key loggers, network scanners and password stealers may be combined for the purposes of installing malware.
- Unlike the usual botnets, APT’s remain hidden at the host level and move around the network slowly as to not be noticed by any anomaly-based IDS/IPS. APT’s can be identified, contained and disrupted at the network level, using behaviour-based technology.

B. Covert Channels and Data Exfiltration

Data exfiltration, through covert channels, as a form of APT, is the new and emerging threat today. To combat this threat, there is little purpose in looking at antivirus technologies since the vendors are well underway with their research in this area and IDS/IPS tools have also flooded the market and are well matured by now. The APT taxonomy above; the earlier explanation of the weaknesses of IDS/IPS systems to highlight behaviour,and the fact that Data Exfiltration through Covert Channels is the new and most frequently emerging form of APT; all clearly and urgently point towards this project focusing on combatting this attack form [16].

1) *Covert Channels:* Covert channels are used for the hidden and abstruse transfer of information, within the medium of legitimate communication. Covert channels manipulate a communications medium in an unexpected or unconventional way in order to transmit information in an almost undetectable fashion and thus hide the very existence of a communication [17]. This is far more more powerful than encryption, for example, since the latter only encodes the data against unauthorised observers, whereas covert channels are obscured from the prying eyes of any intrusion detection mechanisms.

Put in another way, a covert channel transfers arbitrary bytes between two points in a manner that would appear acceptable to someone monitoring the connection. Lampson introduced covert channels in 1973 [18] in the context of monolithic systems as a vehicle for one process to ’leak’ data to another process of a lower security level, which obviously could not access this data on its own [19]. Covert channels in computer network protocols have now been identified as a major modern network security threat. The huge amount of data and vast number of different protocols in the Internet seems ideal as a high-bandwidth vehicle for covert communication. Covert Channels are stealth communication channels that often violate network policies [19]. These channels use shared resources in a way that they were not intended for, [20]. The type of channel implemented by an attacker depends on the shared resource, the noisiness of the channel and the type of systems used for connection. A formal definition of the term “covert” or “covertness” in the area of computer network security would be the “concealment or stealthing of a connection between two points across single or multiple domains”. Another way of explaining it is the more ’covert and activity is, the more difficult it is to detect with specialised detection tools. Let us take the example of a user printing a document over the network to a printer. If there is no network monitoring tool for communication between this user and the printer, then the only way to know that this user is using the printer is to watch them pick up the prints. We may also find that every time we want to print, the printer is busy and this user is collecting some more print outs. Printing say once a day or week, this exfiltration process would not be noticed, where as printing reams a day or even a few pages a day, his actions might well be noticed. Abstracting this idea, we can state that covertness is related to rate of usage of medium; in fact we can state:

$$Covertness\ {}^{1}/_{\alpha}\ Rate\ of\ Medium\ Usage$$

[21] state:

$$Covertness\,\alpha\,(Capacity\,of\,medium - Transmission\,rate)$$

It is useful to observe the variation of covertness against capacity of exfiltration, for different data exfiltration techniques:

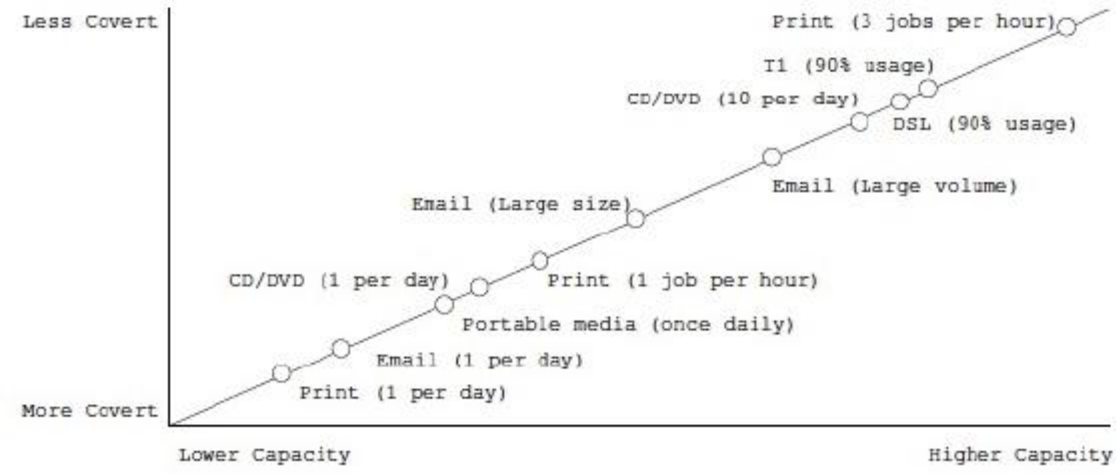

Figure 1. Covertness vs Capacity of Exfiltration [21]

This is the only example that can be found in recent history, where someone has formally addressed covertness against capacity. However, showing how quickly technology changes, we now have to address new challenges for data exfiltration through Twitter, Facebook and VoIP. There are so many newopportunities now to create covert channels.

Centre for the Protection of National Infrastructure has published good practice guide for online social networks (OSN). The key principle on Threat, states that ’Individuals or organisations who place information on OSNs may leave themselves or the organisations that employ them at risk from a range of threats. The threats are varied and threat sources can range from individuals with a grudge through to foreign governments.’ On Vulnerabilities and risks of OSN sites, the first key principle states that ’there are three key vulnerabilities associated with OSNs - the publication of content on these sites, the social interactions between users, and technical vulnerabilities associated with their propensity to spread malware, and aid phishing attacks and spam’. It is clear therefore that OSN sites can be and are being used as media for the support of covert channel operation.

[21] in ’Data Exfiltration and Covert Channels’ go on to give a taxonomy of data exfiltration techniques and state the most commonly used methods to accomplish this: HTTP, FTP, SSH, Email, Phishing, Pharming, DNS cache poisoning, Social Engineering, Shoulder surfing, Directory traversal, Privilege escalation, Botnets, Rootkits, Spyware, Physical transfer means and Covert Channels - Timing Covert Channel and Storage Covert Channel.

An example of an APT that used covert channel methodology is Operation Aurora. Through attack vectors of social engineering and spear phishing, operation Aurora targeted intellectual property, user credentials, and source code repositories from Google[13]. It appears that Google employees who analysed this targeted attack on the infrastructure of the search engine giant discovered a second, far more comprehensive attack that not only affects Google, but more than 30 other major companies, like Adobe, Yahoo, Dow Chemical and Symantec [22].This APT implemented 18 original combinations of advanced encryption, various malware codes, and stealth programming [13]. Operation Aurora carried out the following:

1) A targeted user received a link in email or instant message from a “trusted” source.

2) The user clicked on the link which caused them to visit a website hosted in Taiwan that also contained a malicious JavaScript payload.

3) The user’s browser downloaded and executed the malicious JavaScript, which included a zero-day Internet Explorer exploit.

4) The exploit downloaded a binary disguised as an image from Taiwan servers and executed the malicious payload.

5) The payload set up a backdoor that established an encrypted, covert channel designed to look like an SSL connection, and connected to command and control servers in Taiwan.

6) As a result, attackers had complete access to internal systems. They targeted sources of intellectual property, including software configuration management (SCM) systems accessible by the compromised system. The compromised system could also be leveraged to further penetrate the network.

1) *Data Exfiltration:* Data represents an extremely important asset for any organisation [23]. Confidential data such as military secrets or intellectual property must never be disclosed outside the organisation.Data Exfiltration can be compressively defined as “unpermitted collection and acquisition of data from a source”. Decompressing this definition leads us to categorize two important aspetcs of exfiltration - one being the method and the other being the medium. The medium used by a large number of APT’s for data exfiltration is that of covert channelling, but data exfiltration has been carried out by physical methods such as printouts as well as those that utilise electromagnetic simulation and signal matching. data exfiltration from within a network has many possible avenues such as, FTP, Secure Copy, HTTP POST action (might be with SSL encrypted channel), SMTP server, SSH tunnel set up as a local proxy, IM clients, ICMP Echo Request, DNS lookup, using P2P third party connection etc. [24]

Data exfiltration has been identified as a real and severe threat to the security of classified or sensitive networks, both government or private.

The National Institute of Standards and Technology commissioned a report by BITS Financial Services, in which it states:

“The conduct of APT activities relies fundamentally on the use of malware to establish access, to maintain footholds within organizations and to exfiltrate sensitive data and/or conduct disruption of IT systems or networks.”

It also defines data exfiltration as:

“Exfiltration: method by which malware exports data from an infected host, typically refers to an unauthorized process of acquiring data from a computer system through network covert channels or unauthorized portable media” [25]

The National Institute of Standards and Technology defines a covert channel as:

“any communication channel that can be exploited by a process to transfer information in a manner that violates the system’s security policy” [26].

While the standards organisations such as the Internet Engineering Task Force (IETF) and the Institute of Electrical and Electronics Engineers (IEEE) continue in their attempts to publish new specifications and security modifications to protect the confidentiality, integrity and authenticity of the data being transmitted, [27]details in his work the simplicity of altering a published protocol for the purpose of subversive communications. These covert communications transmit exfiltrated data from a point within a target network to, generally an external target point on another network. The exfiltrated data is usually the ’sliced up’ binary sections of a targeted, access-restricted document or file.

One example of and APT using data exfiltration is attack against RSA Security Inc [28]. The goal was to actually exfiltrate documents and files from Northrop Grumman and Lockheed Martin defence contractors. This proved to be difficult, so the attackers leveraged their attack onto a service provider of these organisation, namely RSA Security, who provided the secure ID’s and access codes to these two contractors.

This attack performed the following steps to reach its goal:

1) Spear Phishing Information gathered about low-level staff of EMC Corporation (who own RSA) – work emails.

2) An email containing an attached Excel file, was sent to one employee and cc’d to 3 others.

3) The file was opened and it contained an embedded AdobeFlash object.

4) Embedded object was immediately executed upon opening the file.

5) Object exploited CVE-2011-0609 vulnerability in Adobe Flash, by executing code and dropping a Poison Ivy Backdoor into the system.

6) Object closes attached file and infection is over

7) After this, Poison Ivy connects back to its server at good.mincesur.com. The domain mincesur.com has been used in similar espionage attacks over an extended period of time.

8) Remote attack server has full access to infected workstation

and all drives connected to it

9) The attackers travelled up the access-level through privilege escalation

10) They gained access to critical information on RSA Secure ID system and how it works

11) They exfiltrated Secure ID info on Lockheed-Martin and Northrop-Grumman.

12) Subsequently, they used this information to exfiltrate documents and information from Lockheed-Martin and Northrop-Grumman.

Giani et al [21] state in ’Data Exfiltration and Covert Channels’:

“While any comprehensive taxonomy of widelydiverse and overlapping members is bound to be subject to debate, we feel that this approach will be helpful to identifying abstractions of pathologies and weaknesses that are common to many methods. And this will allow a systematic development of defences and countermeasures independently of a specific platform or context in which specific exfiltration

phenomena take place."

This project will build upon work outlined above to create the defences and countermeasures hoped for.

THE BEHAVIOUR ENGINE PROPOSITION

III. BEHAVIOUR ENGINES

A new way of looking at the problems of detecting intrusions and attacks, such as data exfiltration over covert network channels, is the concept of the Behaviour Engine. This engine is a very different approach to the task of constructing computer systems - using test data as the primary constraints on which to bound the behaviour (or rules) for the new system. This means a developer is constructing the system within defined boundaries (which are tested there and then), rather than on a blank canvas; and performing real-time impact analysis as the system is created or modified. Test data in this context may consist of true test data for the new system or test data for an existing system. A developer then uses these constraints to create the desired behaviour within the graphical environment, where they can explore and understand the content and meaning of the data and what the system needs to achieve with this data.

The Behaviour Engine has a reversed approach to creating a rule engine, as opposed to currently implemented IPS systems. To build a typical network rule engine, the technical experts and code writers design all the rules they know based on past experience, standards such as RFC's and some requirements form the organisation they are designing the system for. They apply the IPS to the company network and spend a long time making alterations to the rule-set due to new company requirements. These alterations inevitably cause conflicts elsewhere within the rule set and thus take weeks and months to resolve. The next and major problem with current IPS systems centres around requirements capture and management, which are well documented within the IT industry. It is the defects in the requirements definition process that very often lead to failure in the successful implementation of major IT projects. We would all contend that truly successful development can only take place if we can elicit all the relevant 'Tacit Knowledge' that is contained within the experience of the domain expert and then transfer this effectively into working code.

"Tacit knowledge can be defined as knowledge that is not made explicit because it is highly personal, not easily visible or expressible, and usually requires joint or shared activities to transmit it." [29]

Knowledge communicated is either explicit or tacit. Explicit knowledge is easy to convey and turn into rules for a decision engine. Tacit knowledge however is that knowledge which cannot be transmitted in words, such as the definition of the colour green; or too complex to transmit in its entirety such as how to bring up children. This knowledge is gained thorough life experiences, subjective judgements, insight and intuition. It is thus 'learned by doing'. Tacit knowledge is highly personal and the domain expert has difficulty in expressing their tacit knowledge as they do not consciously know all the rules that they would apply to a situation. This is just the reason that requirements capture and management is inherently so difficult when building large complex computer systems, where more tacit knowledge is possessed by the user, evolving into mental models that are too large for the developer to retain and turn into explicit knowledge, then code.

The Behaviour Engine lets you start with no rules at all. As the user (unlikely to be an IT developer) performs a task, i.e. makes a conclusion, they are forced by the software to justify as to why they made this conclusion. This justification process is made either by plain text or through an graphical user interface

(GUI). The uniqueness of the this engine, in the light of what we have been considering, is that it actually automates the task of turning conclusions and justifications into rules. As it learns about the task, it begins to introduce suggestions that the user can accept or reject. I rejected, then a reason is requested so that the system learns the fine details that make the current situation different from its previous experience. Just as in human learning, the system quickly learns to handle simple situations and only needs correcting as more complex or ambiguous situations are met. Thus the engine provides good results after a remarkably short time as it learns the rules that cover common situations first. The user simply continues to work within their knowledge domain, on their usual tasks, responding to the queries made by the software. As rules are introduced in this way then the critical problem of mental modelling that is necessary in the traditional way is automated.

In one sense, the behaviour engine is a case-based incremental Ripple Down Rule system. [30]argues from a situated cognition perspective that experts can never explain how they reach a conclusion, rather they justify that a conclusion is correct, and provide this justification in a particular context. Therefore all knowledge acquisition must be incremental and case-based. Hence, the behavio ur engine generates new behaviour through its conclusion-justification methodology and in doing so creates a test suite of cases that will never be broken, even as more tests are added to the suite. Also, automated system validation ensures that any new behaviour does not invalidate old behaviour relieving the pressure for doing test runs and ensuring the system never gets into an illegal state.

A. Behaviour engines are currently being used in wide variety of organisations:

- Border surveillance and control [31]
- Protection of sites and infrastructure [31]
- Safety of populations at large-scale events. [31]
- Banks - Working with Erudine, Tradocs (document exchange service) developed a ‘Just In Time Finance’ (JITF) service that enables the high volumes of supply chain trade data to be leveraged and generate revenues for banks.
- In 2009 project Security Cockpit, the outflow from the DESEREC project (DEpendability and Security by Enhanced REConfigurability), was announced by EADS. This was a groundbreaking solution for implementing network security responses in complex environments, which depended on multiple data sources and human inputs.

“Security Cockpit is an innovative concept that includes off-the-shelf tools and bespoke applications, and is highly customisable. It addresses the vulnerability of IP technologies and the growth in ‘professional’ cyber attacks, significantly improving the ability to respond to threats and attacks.” [32]

A behaviour engine was integrated by the Defense and Security body of the EADS, as a discerning technology enabling Security Cockpit to accomplish satisfactory reaction capabilities of the cockpit. The solution reduces the treatment time of security incidents and for capturing the expertise (being the tacit knowledge domain of the EBE) of responses.

The original DESEREC project was based on the following three principles:

- Modelling and simulation of critical infrastructures for improved resilience
- Various detection mechanisms integrated for detection of sever and complex incidents of seemingly unrelated events

- Response is provided by a framework of computer-aided countermeasures that mitigated threats to the dependability of the systems and rapidly thwarted any attacks. CIS reconfiguration was the highest mechanism for survivability. [33]

Incidents were defined as being caused by the sequential occurrence of certain events, be it accidental or deliberate. A CORRELATION of these events defined the existence of an incident, following the implementation of correlation rules. Coupled with this was the ONTOLOGY of the Inference Engine where EXPERT KNOWLEDGE created automated decision making. The Inference Engine can be seen as the core of the Expert System used for the detection, amalgamating a tacit knowledge model with factual network correlated data to produce possible decision or reactions to an incident.

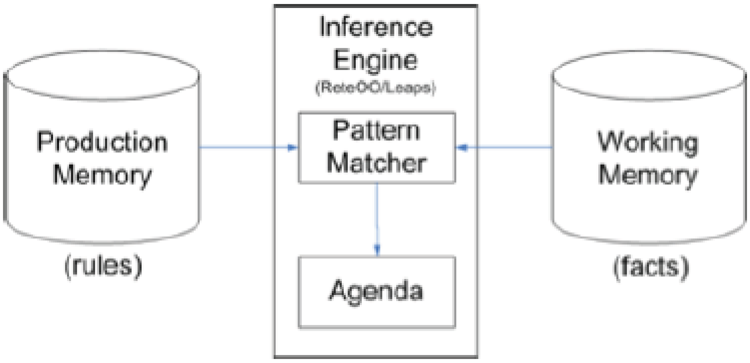

Figure 2. Inference Engine for Security Cockpit [2]

Above the layer of all IDS/IPS a behaviour engine was implemented in establishing ’normal’ network behaviour and relating this to a generated threat classification for all points within the network (which effectively is an hierarchy of machines within the network drawn up from user prescribed and tacit knowledge). The above-outlined multi-disciplinary approach allowed DESEREC to respond efficiently to the three groups of incidents which may take place on such a critical system. Upon an attack or suspected attack, a categorisation of the attack lead to varied responses. For example, if a suspected DoS attack was targeting a backup games simulation server, then the reaction would be much less severe than if the attack was targeting primary key server.

B. Behaviour Engine and attack detection

Presently, web-enabled systems are protected by sturdy firewalls and Intrusion Prevention Systems, erected to keep out directed attacks from hackers. If the system required more protection, then reinforcement is introduced into the firewall/IPS layers, sometimes putting in 20 or 30 different layers in an electronic maze [34]. The repercussions of this are two-fold. Firstly, the focus on a single type of defence, which is expected to be nearly impenetrable, means that once an attack does get through there is very little in the way of automation or rapid evaluation to ensure a quick and effective response. Worse, the response from the systems to a high-level security breach, when there is no one to manually deal with it, could be to simply shut everything down. Secondly, it largely ignores attacks occurring from within a network. In fact, most internal security breaches are not even malicious – they are the result of poor staff training, incompetence, or accident, such as unwittingly opening an email attachment infected with a worm.

With respect to intrusion detection and discovering possible data exfiltration taking place within a network, then one of the most vital elements in this process is the ability for a fully automated or semi- automated (decision-aided) response following an alert detection. It is unreasonable to expect a large system to be manned at all hours, or to have a human being examine and respond to every threat. Equally, it is not good enough to have an automated response that is too limited, simply quarantining a virus or shutting down the system is too generic a response. In a world full of asymmetric, rapidly changing threats, the behaviour of system security must be more complex if it is to swiftly handle a wide range of incidents and respond appropriately. Utilising Behaviour Engine technology, what is needed is rapid reaction- decision components that can be integrated within existing systems or packaged with its own security solutions. The Behaviour Engine allows the rapid capture of the complex decision logic used to respond to incidents, contextualised to the customer's specific environment. Once captured, this behaviour is used to determine the impact of attacks on business services, evaluate the relevant actions in response to an alert, and establish whether manual authorisation is required for the relevant response. In this way, the response component allows complex decision support and fully autonomous incident resolution. In the face of frequently changing threats and increasingly dangerous computer viruses, the Behaviour Engine can learn or be shown the responses to new types of incident. As soon as a new threat is identified and the correct response established, it is a quick and simple process to add the new behaviour and ensure systems are protected against the latest dangers.

It is also clear that the Erudine Behviour Engine played a major role within the outlined Inference Engine mode. Building upon this success and facing the new challenge of data exfiltration, the EBE will be used to successfully and substantially improve upon current technologies for the detection of this attack form. This project will collect and correlate data and network information from all the sensors in the network, classifying all possible attack vectors, and using the tacit knowledge model of the EBE, profile the results to generate immediate possible decision reactions for the user. The overall process will be called Multi Vector Attack Profiling (MVAP).

IV. PROPOSED ARCHITECTURE

The Behaviour Engine is the analytical core of this system Packet data, network data flow and other network information is captured from various sources on the network that sniff data packets going in and out of the network. Initially this information is captured as PCAP files and then turned into xml format to be input into the engine. The diagram below shows an architecture prototype.

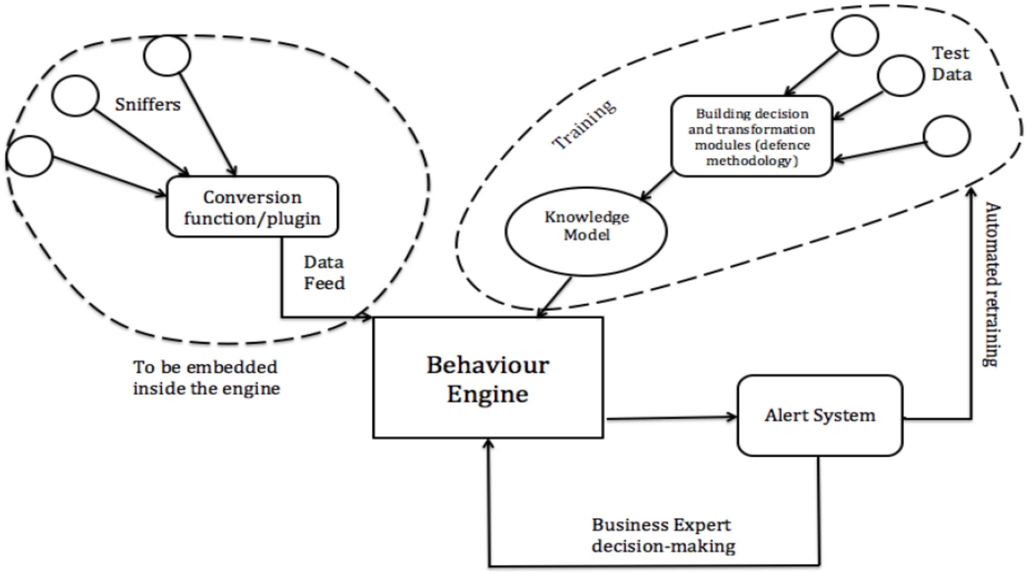
Sniffers
Conversion function/plugin
Data Feed
To be embedded inside the engine
Training
Test Data
Building decision and transformation modules (defence methodology)
Knowledge Model
Automated retraining
Behaviour Engine
Alert System
Business Expert decision-making

Figure 3. Proposed System Architecture

Firstly, the behviour engine will be trained with test data, to create the tacit knowledge rule-model. It is then envisaged that all data input and preparation processes for analysis to be embedded within the behaviour engine to enable fast live analysis and the ability for the business expert to make an immediate decision upon an unknown attack or network event.

V. Summary

This paper has demonstrated that the current state of the art in anomaly network intrusion detection and prevention technology possess deficiencies in satisfactorily detecting anomalous intrusions, more specifically those of the advanced persistent threat category. Due to the rapid rise in APT's and especially data exfiltration through covert channeling, there is a definite need for mitigation of this threat. This paper has presented the technology of Behaviour Engines, which facilitate the acquisition of tacit knowledge through justification conclusion decision knowledge-model creation. This system is an automated, self developing and learning mechanism. The proposed method towards a solution is to merge the application of ontological decision techniques of behaviour engines with network packet capture data, to detect data exfiltration attempts over covert channelling, through the principle of multi-vector attack profiling. The final goal of the research will be to develop a behaviour engine/intrusion detection solution for preemptive counter-measures to anomalous behaviour from within or without a network.

References

[1] C. Issariyapat and K. Fukuda, Anomaly detection in ip networks with principal component analysis, in *Communications and Information Technology, ISCIT 2009. 9th International Symposium on*, 1229 –1234.

[2] M. Avelino and F. Torre, Advanced rule-based techniques in mission critical systems, in *Emerging Security Information, Systems and Technologies, SECURWARE '08. Second International Conference on*, (2008), 375 –380.

[3] R. Bace, *Intrusion Detecion*, (2007), Macmillan Publishing Co., 2000.P. Engelbrecht, Computational Intelligence: An Introduction, 2 ed.: John Wiley and Sons.

[4] V. K. A. Lazarevic and J. Srivastava, Intrusion Detction: *A Survey*. Springer, 2005.M. Clerc, Standard Particle Optimisation From 2006 to 2011 version 13-July-2011.

[5] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, Bayesian event classification for intrusion detection, Proceedings of the *19th Annual Computer Security Applications Conference*, (2003).

[6] M. L. U. B. K. Anders Madsen and F. Jensen, The hugin tool for learning baysian networks, *Lecture Notes in Artificial Intelligence 2711*, (2003), 594 - 605.

[7] T. Bass, Multisensor data fusion for next generation distributed intrusion detecion sys, April (1999).

[8] Y. Zhuo, Q. Zhang, and Z. Gong, Cyberspace situation representation based on niche theory, *Proceedings of the 2008 IEEE International Conference on Information and Automation*, (2008), 20-23.

[9] J. Shifflet, A technique independent fusion model for network intrusion detection, *Proceedings of the Midstates Conference on Undergraduate Research In Computer Science and Mathematics*, (2005).

[10] W. H. Lai Jibao and Z. Liang, Study of network security situation awareness model based on simple additive weight and grey theory, *IEEE Computational Intelligence and Security*, (2006).

[11] K. Scarfone and P. Mell, Guide to intrusion detection and prevention systems (idps), tech. rep., *National Insitute of Standards and Technology*, (2007).

[12] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, Anomaly detection methods in wired networks: a survey and taxonomy, *Computer Communications*, **27**(16), 1569–1584 (2004).

[13] J. Bourquin, The evolution of cyber espionage: A case for an offensive u.s. counterintelligence strategy, tech. rep., *Utica College*, (2011).

[14] Damballa, Advanced persistent threats, (2010).

[15] M. Security, Advanced persistent threats, (2010).

[16] M. C. Eric Hutchins and R. Amin, Intelligence-driven computer networkdefense informed by analysis of adversary campaigns and intrusion kill chains, in *6th Annual International Conference on Information Warfare and Security*, (2011).

[17] J. Thyer, Covert data storage channel using ip packet headers, *SANS Institute*, (2008).

[18] B. W. Lampson, A note on the confinement problem, *Commun. ACM*, **16**, (1978), 613–615.

[19] S. Zander, G. Armitage, and P. Branch, A survey of covert channels and countermeasures in computer network protocols, *Communications Surveys Tutorials*, *IEEE*, **9**, (2007), 44 –57.

[20] S. Cabuk, C. E. Brodley, and C. Shields, Ip covert channel detection, *ACM Trans. Inf. Syst. Secur*., **12**, 22:1–22:29, (2009).

[21] A. Giani, V. H. Berk, and G. V. Cybenko, Data exfiltration and covert channels, in *in Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, (2006).

[22] McAfeeLabs, Protecting your critical assets, (2010).

[23] E. Bertino and G. Ghinita, Towards mechanisms for detection and prevention of data exfiltration by insiders: keynote talk paper, in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, (New York, NY, USA), 10– 19 (2011), ACM.

[24] R. C. V. Antwerp, Exfiltration techniques: An examination and emulation, Master's thesis, *University of Delaware*, (2011).

[25] B. F. Roundtable, Malware risks and mitigation report, tech. rep., *National Institute of Standards and Technology*, (2011).

[26] *N. I. of Standards and Technology*, Trusted computer system evaluation criteria, (1983).

[27] R. Sbrusch, Network covert channels: Subversive secrecy, *SANS Institute*, (2006).

[28] M. M. P. C. T. Research, A technical analysis on the cve-2011-0609 adobe flash player vulnerability, tech. rep., *Microsoft*, (2011).

[29] Erudine, Unlocking tacit knowledge by conclusion and justification, *Erudine White Paper*, (2006).

[30] P. Compton and R. Jansen, A philosophical basis for knowledge acquisition, in *3rd european knowledge acquisition for knowledge based systems workshop*, (1989).

[31] Erudine, Erudine 1, **1**, 6(2011).

[32] EADS and E. B. Engine, Security cockpit project, (2009).

[33] T. Hartog and G. Kleinhuis, Security analysis of the dependability, security reconfigurability framework, in *Risks and Security of Internet and Systems, 2008. CRiSIS '08. Third International Conference on*, (2008), 93–100.

[34] T. K. B. Calvert L. Bowen and R. W. Thomas, A plan for scada security to deter dos attacks, in *Proceedings of the Department of Homeland Security: R&D Partnering Conference*, (2005).