A service-centric model for intrusion detection in next-generation networks

Jimmy McGibney[a], Nikita Schmidt[b], Ahmed Patel[b]

[a] *Telecommunications Software & Systems Group, Waterford Institute of Technology, Cork Road, Waterford, Ireland*

[b] *Computer Networks & Distributed Systems Research Group, Dept. of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland*

Correspondence to: jmcgibney@tssg.org

Abstract

In this paper we argue for a new service-centric model for developing and implementing intrusion detection systems. This new approach is influenced by the convergence of computer networking and telecommunications systems, and ways that the telecoms industry manages issues of fraud. Interoperability of intrusion detection components with the rest of the system, in particular logging and billing subsystems, requires standardisation of usage data representation formats and development of a standard ontology for intrusion detection.

Keywords

Security; Intrusion Detection; Service Model; Service Creation

Acknowledgements

Part of the funding for this work is provided by the European Commission through the SEINIT (Security Expert Initiative) project. For more information, see http://www.seinit.org/.

Introduction

In this paper we argue for a new *service-centric* model for developing and implementing intrusion detection systems (IDSs). This new approach is influenced by the convergence of computer networking and telecommunications systems; in particular, the authors have considered the cross-fertilisation of intrusion detection and telecoms fraud management.

The paper is organised as follows. Firstly, we give a brief overview of intrusion detection and the key challenges faced today. We then look more specifically at the two main approaches, namely misuse detection and anomaly detection, and explore their limitations in the current environment. Next, we examine the (de facto) service model in use in telecommunications and, in the following section, we give the bones of our abstract service-oriented model. The next section presents the architectural context of service-oriented intrusion detection, which is followed by a proposal to use the IPDR's NDM-U format to represent data gathered for analysis.

Intrusion Detection: Background

Intrusion detection is one of the major pillars of any computer and network security policy. It deals with the problem of unwanted trespass into systems by users or automated software agents acting on their behalf. The potential for damage to systems from intruders is immense, and includes loss or corruption of data, release of sensitive information, theft of scarce or valuable resources and loss of availability of systems through machine crashes or resource congestion.

It is fair to say that a constant war is being waged between those who own and manage systems and those who wish to intrude into them. Access to the Internet is relatively easy and cheap, with users enjoying a high level of anonymity if desired. Even though the number and diversity of systems is increasing, as is their complexity, the level of technical sophistication required to carry out intrusions is falling [1].

From a security perspective, the main objective of computer systems managers has been to protect their own data and information systems, rather than use of the network in its own right. Thus there has been an emphasis on *perimeter security*, where a clear distinction is made between the private network containing digital assets and the (dangerous) outside world. The main engine of perimeter security is of course the firewall. From a monitoring perspective, the focus has been on *intrusion detection*. In the same way that a property owner may wish to deploy a burglar alarm even though the doors and windows have good locks, the system manager is also motivated to monitor equipment usage in an attempt to detect atypical patterns. It is often fair to assume that system perimeter security may have flaws, as software bugs and configuration errors are very common.

The biggest challenge currently faced though is one of *complexity*. Despite convergence of the core network architecture to TCP/IP [2], equipment and services in common use are becoming increasingly diverse, and configurations increasingly complex.

Increasing user desire for mobile computing and communications, with a variety of devices of appropriate scale, is encouraging this diversity. Furthermore, computing and communications capabilities are increasingly being integrated into all kinds of entertainment devices, vehicles, household electrical equipment, industrial machines, and so on. This move towards pervasive computing assumes that ultimately the urbanised world will be full of such “smart”, communicating devices.

Limitations of current models

Detection method

There are currently two major general approaches to intrusion detection, namely *misuse detection* and *anomaly detection*. A substantial literature exists that catalogues and describes several techniques that fall into one or other of these categories. A concise survey of the leading techniques is provided in [3].

The former, misuse detection, is by far the most commonly implemented in real-world systems. Also called signature detection, this method uses a pattern matching approach. The system compares collected data with a database of signatures of known attacks. If the match is positive, an intrusion is deemed to have occurred and the system reacts accordingly.

The second approach, anomaly detection, is based on modelling “normal” behaviour and observing deviations from this model. Data is collected on the behaviour of legitimate users over a period of time. Any behaviour that is inconsistent with this model is considered suspicious. Various statistical tests are used to determine what constitutes abnormal activity. A basic assumption of this model is that attack behaviour is significantly different from legitimate behaviour.

The most widely used intrusion detection systems to date in production environments have tended to focus on misuse detection, with true anomaly detection systems being reserved for research environments. *Snort*[1], for example, a leading open-source intrusion detection system is heavily rules-based and thus is mainly a type of misuse detection. *RealSecure*[2] is a popular commercial intrusion detection product that also uses signatures to analyse data. However, clever specification of rules can allow a type of anomaly detection where new attacks are caught.

Both of these leading approaches, however, have significant limitations. In the case of misuse detection, the sheer number of rules required to secure modern, complex systems means that it is difficult for system administrators to maintain an IDS that is customised to the needs of their own system set-up. This constant “arms race” between attackers and system administrators is illustrated in Figure 1. Another serious problem is its lack of *adaptivity* – i.e. a new attack pattern goes unrecognised if it does not have a corresponding signature defined in the relevant database. This requires very active management of the rule set used. In practice, systems with many thousands of rules are quite common.

Anomaly detection systems also present problems. In increasingly dynamic environments, where technology lifecycles are short and users are mobile, it is difficult to accumulate a sufficient quantity of steady-state data to train such systems. Furthermore, illegitimate behaviour can become accepted as

[1] http://www.snort.org/

[2] http://www.iss.net/

normal if it is not caught and is carried out over a period of time. A further crippling problem with anomaly detection is a typically large number of false positives. After some time, personnel tend not to take alarms seriously if most are false, making for bad security. In systems where malicious activity is only a very small percentage of total activity, it is inherently very difficult to avoid a high rate of false positives (by Bayes' theorem – [4]).

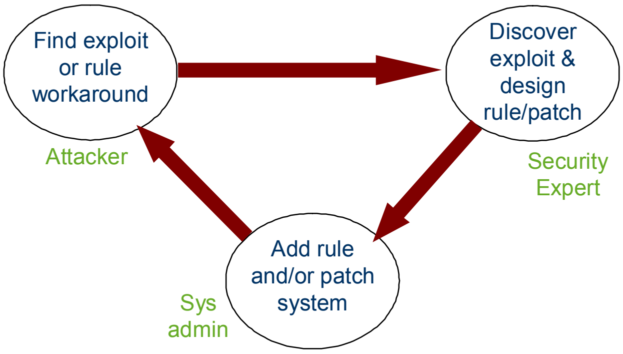
Find exploit or rule workaround
Attacker
Discover exploit & design rule/patch
Security Expert
Add rule and/or patch system
Sys admin

Figure 1: Rule maintenance cycle

Field of vision

Information systems can be utilised in a wide variety of ways at a number of different levels. Certain types of users and applications interact with systems at a very high level in such a way that underlying infrastructure (e.g. hardware, operating system, network protocols) is invisible. Others may utilise a specific operating system or communications protocol, or may operate below the operating system (OS) or high-level protocol.

The current generation of IDS is generally capable of operating in either of two ways. These are categorised as *host-based* and *network-based* intrusion detection systems, respectively.

In general, a host-based system runs on an operating system, collecting and analysing data on the usage of a specific host. The extent of its perception is usually limited to a specific machine and the applications and services running on the its OS. It normally does this by tracking logs produced by the OS (e.g. *syslog*) and applications (web servers, mail servers, etc.) and also by tracking files for modifications, and so on.

A network-based IDS has a broader field of vision, extending to an entire network segment. This type of IDS works by scrutinising packets travelling on the network, and often takes the form of a specific device with its network interface card set to promiscuous mode. The main advantage is that an attack on a host can be detected *before* the host is compromised.

The main limitation of both of these approaches is that they have a limited view. In the case of a host-based IDS, anything that happens below the level of the OS is generally undetected. In the case of a network-based IDS, low-level (sub-packet) network activity may also be undetected. Furthermore, it is difficult for these types of system to draw inferences from patterns of high-level, application-specific activity; this is further complicated of course if data is encrypted.

Another problem is with complexity and scalability. It is hard to analyse every packet on a gigabit interface, especially if a vast number of rules need to be applied. Likewise, the volume of activity on a typical server OS can be difficult to keep up with.

Furthermore, the trend towards mobile devices, and the limitations of traditional perimeter security in this environment, means that more flexible IDS strategies are required.

Lessons from telecommunications

Telecommunications networks have traditionally been public networks, with careful accounting of usage and billing of users. The concept of various *services*, provided by an operator to a user for a fee, is well understood. By contrast, early computer communications systems were designed primarily for use *within* organisations – mainly governments, universities and large businesses – before the arrival of personal computers in the 1980s, at any rate. These organisations purchased the relevant equipment and built networks for internal use without expecting to directly recoup their costs.

For telecommunications providers, the value has been in the networks themselves. Telecoms networks have typically been managed by a small number of licensed operators, with user access to the network tightly controlled and limited in sophistication. The connectivity services provided have typically been charged for based on usage (usually time connected, or volume of data in some cases). Thus operators have viewed any unauthorised usage of telecoms networks as theft of a charged service, and thus fraud.

Telecommunications fraud management is already quite service-oriented. Leading techniques are based on the analysis of service usage data records, typically known as call detail records or CDRs.

Service-centric model for IDS

In this approach, we propose to model *everything* as a service. The above sections highlighted some of the problems of scalability, maintainability and adaptivity that arise with present-day intrusion detection systems that are based on low-level objects like packet traces, log files, and so on. Our service-based abstraction attempts to deal with this. The main motivation for choosing a service-based abstraction is that services are at the heart of computing and communications convergence, as evidenced by the level of interest in, and rapid adoption of, *Web Services* [5].

It may be useful in some cases to distinguish between "real" and "virtual" services if the concept of a service is already in use. A service could, for example, be something like a user application or it could be a low-level communications facility. Thus dial-up Internet access is a service, and so are the DNS, DHCP, a web search engine, a music download, a currency converter, a virtual private network, and so on.

In our model, services have the following properties.

- *Composition*: a service is made up of one or more components, each of which is also a service.

  This provides scalability. The idea is that a service is fractal-like (though not to infinity of course). Intrusion detection requirements (rules) can then be defined for a service, in many cases independently of the specifics of the service's constituent components, which will have their own rules defined.

- *Inheritance*: Any defined service can be extended or specialised.

  Intrusion detection rules can thus be re-used by the specialised service. For example, "Dial-up Internet access" might inherit from "Internet access". It may be convenient in some instances to model certain entities as virtual services.

- *Lifecycle*: A service has a lifecycle, during which it can exist in a finite set of states. At any given time during its lifecycle, the service is in exactly one state. The service definition may include a matrix specifying between which states transitions are allowed or forbidden.

  IDS rules can be defined on a state-by-state basis. An example of a state transition might be a mobile worker moving from his/her employer's domain onto a public network. As another example, consider a multi-party conferencing service. Suitable states might be "pre-conference" during which invitations are issued, "in progress", and "post-conference" during which minutes are published and various documents, etc, are shared. Each state could be expected to have different security requirements.

  As well as being able to change state, services are of course capable of being created and obsoleted.

- *Contract*: A service has two actors, a provider and a user, between which there is a service specification (contract).

  This contract defines *security policy* – privacy, authentication, integrity, access control, non-repudiation, availability. It is also a mechanism for the definition of acceptable usage, "normal" usage, and intrusion detection rules.

Some contracts will be long-term and some will be short-term. For example, a user might have a long-term contract with an Internet Service Provider defining the rules for dial-up access. A separate short-term contract could be used to define the rules for a particular DHCP lease.

- Services can be *autonomic.* The state of a service may be modified by the provider or the user, or *by the service itself.*

  For example, services can be *self-securing*. It can be envisaged that some services will be sufficiently autonomous that they react to threats and attacks, patching themselves and updating aspects of their own security policy. Consider the case where the service has an associated honeypot. Any activity on the honeypot, monitored by the service, could cause the service to spontaneously modify its state.

- Services can be *monitored.* Services generate usage data for security monitoring and accounting. A later section of this paper considers suitable ways to represent this usage data.

A distinction may be made, for the purposes of discussion and analysis, between *horizontal* services and *vertical* services. Horizontal services are provided at the same layer. With vertical services, a given layer provides services to the layer above or below.

Service security

In this model, a security component is just another service. The idea is that some services will have built-in security attributes and capabilities (e.g. a music download service detecting illegal download). There will also be specialist security services that protect or monitor other services. We could view a firewall or a key issuer, for example, as a specialist security service.

Use of ontologies for service modelling

The proposed service-based abstraction allows further formalisation in descriptive logic using an ontology. The use of ontologies (which have been popularised by the Semantic Web) for intrusion detection is a relatively unexplored but promising approach [6]. A carefully selected ontology represents the knowledge about a system, its services and their relationships in a formal way so as to allow the IDS to automatically derive new rules based on this knowledge.

For this approach to be successful, all system components need to be aware of the ontology used by the IDS. The heterogeneity of modern computing and telecoms systems thus requires ontology standardisation. As it may be impossible at the standardisation stage to envisage all contexts in which intrusion detection systems will operate, a layered approach will need to be taken. Descending from more general towards more specific layers of the ontology will strengthen the reasoning ability of the IDS at the price of reduced flexibility and applicability.

An architecture for service-centric intrusion detection

With our model, intrusion detection is viewed as the real-time or non-real-time observation of indicators from service usage data, and determination of whether an intrusion is taking place or has taken place. This should then trigger some action, such as blocking access to the service or generation of an alert for appropriate personnel.

Implementation of standard security facilities like reliable authentication of identity and policy-based access control is needed as a first line of defence. The reader is referred to work on securing access to charged services that has been undertaken by the Authentication, Authorization and Accounting Working Group of the Internet Engineering Task Force [7].

Our central focus is on *a posteriori* processing in order to detect intrusions – accepting that, despite the enforcement of strict controls, there remains the possibility of intrusions. This processing takes place either while a service is being used or some time afterwards. Figure 2 presents a reference architecture for intrusion detection.

Inputs to this intrusion detection process include usage data gathered from the currently active service, historical usage data, consumer profile information and a set of intrusion detection rules that are to be applied to this data. These intrusion detection rules might be tailored to individual services, individual consumers, or both.

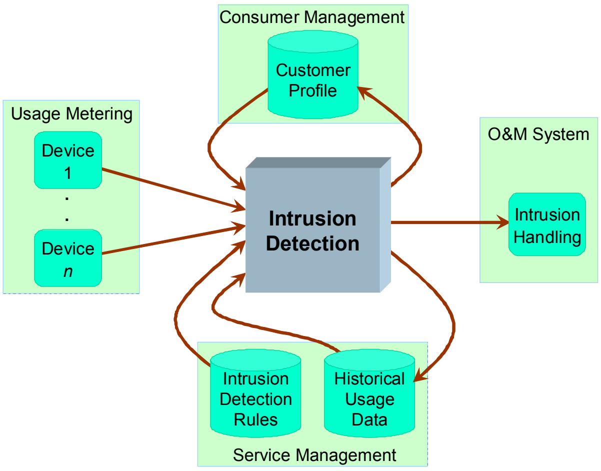

Figure 2: Intrusion Detection in context

There are many possible outputs from the intrusion detection process. Examples include:

- Events or alerts to be processed by the service provider's existing operations and management (O&M) system. These events or alerts could be Boolean (i.e. a determination that intrusion has occurred) or a quantitative measure indicating the likelihood of intrusion.
- An updated consumer profile, to be used as input to future intrusion detection activity.
- Updates to historical data that is maintained on service usage.

Common usage data representation

A standard multi-service usage monitoring data format is needed for our service-centric model to be useful in practice. This usage format can then provide a basis for IDS rule specification, wherein rules are specified on a per-service basis as functions of service usage data elements.

The records most commonly monitored for fraud in telecommunications have been CDRs. A CDR usually contains information about a completed telephone call or call attempt and is used for billing purposes. There is no single CDR standard and billing and fraud detection systems have typically had to be tailored to specific formats. Typically each vendor of telecoms equipment has it its own CDR format, with CDR formats even further varying between different generations of technology from the same vendor.

One multi-service format that shows potential for use for all kinds of services is the Internet Protocol Detail Record (IPDR) organisation's Network Data Management – Usage (NDM-U) specification [8]. The IPDR organisation is an industry consortium, founded by several prominent vendors of management solutions for IP-based networks and services. The main objective of the IPDR organisation is "to define the essential attributes of information exchange between network elements and services, operation support systems and business support systems". The specification is based on the core functional roles and interfaces of the TeleManagement Forum's Telecom Operations Map [9].

There are several reasons for adopting the NDM-U specification for intrusion detection in a service-oriented environment:

- The IPDR structure specifies a generic, flexible record format for exchanging usage information in a multi-service environment.
- IPDR provides an extension mechanism so that additional, optional, usage metrics may be exchanged for a particular service, or even a particular service usage instance.
- IPDRs can be used for exchanging any kind of usage data. For example, IPDRs can be generated periodically while a service is being used, enabling near real-time intrusion detection.
- IPDRs are self-descriptive and human-readable, based on eXtensible Markup Language (XML), allowing for more straightforward integration into diverse systems. Figure 3 shows a sample IPDR for an email service usage instance.

```
<IPDR>
<seqNum>1</seqNum>
<IPDRCreationTime>2004-11-01T07:00:00Z</IPDRCreationTime>
<userLoginName>Joe Verbose</userLoginName>
<userLoginLocation>152.168.1.10</userLoginLocation>
<providerName>Acme ISP</providerName>
<providerLocation>208.99.88.99</providerLocation>
<eventType> storage</eventType>
<size> 83100</size>
<startTime> 2004-11-01T07:00:00Z</startTime>
<endTime> 2004-11-01T07:00:00Z</endTime>
</IPDR>
```

Figure 3: Sample IPDR for an Email service

A potential drawback with the IPDR model is that, in common with most self-describing text-based specifications, it is not the most efficient way to represent data. Efficiency can be improved however with the use of native XML databases as well as XML compression.

No matter what format is used, not all data of interest for intrusion detection will be available in a single record and in the correct format. Thus, one of the required subsystems of any intrusion detection system is a *mediation* component that is responsible for pre-processing data with appropriate correlation and aggregation, and then presenting it in the correct format to the rules engine.

Impact on standards

Standardisation is key to good performance of a general purpose intrusion detection system. The better an IDS interacts with its environment, the more knowledge it has at its disposal to make decisions. Two aspects of this standardisation have been discussed in this paper.

- Data representation standards are to provide the IDS with an understanding of the data circulating in the system. Of particular importance are usage data, which lack standards at present. IPDR is suggested to be used for such data.
- Service specification standards are to provide a common shared view among the IDS, other system components, and system administrators of the system in general, its services and their relationships. Various policy specification languages, description logic languages and ontology representations need to be carefully considered.

Conclusion and future directions

This paper presents the case for a service-centric approach to intrusion detection, whereby all system activity is modelled in terms of a service abstraction.

Limitations of the current generation of intrusion detection systems are described. In particular we discuss the difficulty in solving the trade-off between IDS adaptivity and tendency to produce excessive false positives, as well as the limited field of view of the systems that are generally focused on network layer packets and/or host OS activity. These limitations provide a motivation for considering a fresh approach, and such an approach, based on a service model, is proposed here.

Notable aspects of our outline service-centric model are the contract between service provider and consumer, as well as the ability to specify autonomic capabilities to allow a service to secure itself. Also, an important aspect of the model is that services can be configured to be monitored by other services. The IPDR service usage data format is proposed as a means of representing activity that is being monitored in a way that is consistent with this approach.

The next steps include:

- Defining a syntax (or use an existing one) for service specification in accordance with the above properties. Candidates include various policy specification languages like *eXtensible Access Control*

Markup Language (XACML)¸ *Ponder*, and the IETF/DMTF *Policy Core Information Model*, a general language like the Object Management Group's *Model Driven Architecture* (MDA), or perhaps the W3C's *Web Service Definition Language* (WSDL) or the *Directory Enabled Networks New Generation* (DEN-ng) approach that has found favour in telecommunications management. Further refinement can be achieved through ontology specification languages, such as W3C's *Resource Description Framework* (RDF) and *Web Ontology Language* (WOL).

- Writing mappings for a set of real-world scenarios
- Implementing intrusion detection for these scenarios
- Evaluating the effectiveness of this approach, with respect to reduction in complexity, performance, robustness, manageability and the ability to reduce the incidence of false positives and false negatives.

Initial experimentation with this service approach and use of the IPDR representation for service usage data has produced promising results. A prototype system for telecoms fraud detection based on the IPDR is described in [10].

References

[1] C. Manikopoulos, S. Papavassiliou, "Network Intrusion and Fault Detection: A Statistical Anomaly Approach", *IEEE Communications Magazine*, October 2002

[2] L. Mathy, C. Edwards, D. Hutchison, "The Internet: A Global Telecommunications Solution", *IEEE Network*, July/August 2000.

[3] T. Verwoerd & R. Hunt, "Intrusion detection techniques and approaches", *Computer Communications*, vol. 25, pp 1356-1365, September 2002

[4] S. Axelsson, "The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection", *ACM Transactions on Information and System Security*, August 2000.

[5] World Wide Web Consortium, Web Services Activity. Information available at http://www.w3.org/2002/ws/

[6] J. Undercoffer, A. Joshi & J. Pinkston, "Modeling computer attacks: An ontology for intrusion detection", *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, pp 113–135, Pittsburgh, PA, USA. Lecture Notes in Computer Science 2820, September 2003.

[7] Internet Engineering Task Force, Working Group on Authentication, Authorization and Accounting, http://www.ietf.org/ html.charters/aaa-charter.html

[8] IPDR Organization, 2002, *Network Data Management – Usage For IP-Based Services, Version 3.1.1*, October 2002. Available at http://www.ipdr.org/

[9] TeleManagement Forum, *Telecom Operations Map*, GB910, Version 2.1, March 2000.

[10] S. Hearne, J. McGibney, A. Patel, "Addressing Fraud Detection and Management in Next-Generation Telecommunications Networks", *Proceedings of SCI 2004*, Volume VIII, pp. 29–33, Orlando, Florida, USA, July 2004,