

System-of-Systems Boundary Check in a Public Event Scenario

Bo Zhou, Oliver Drew, Abdullahi Arabo, David Llewellyn-Jones, Kashif Kifayat, Madjid Merabti, Qi Shi

School of Computing and Mathematical Sciences
Liverpool John Moores University, Liverpool, UK.
zhoubo@css.com.cn, O.J.Drew@2007.ljmu.ac.uk,
{A.Arabo, D.Llewellyn-Jones, K.Kifayat, M.Merabti,
Q.Shi}@ljmu.ac.uk

Rachel Craddock, Adrian Waller, Glyn Jones
Thales Research and Technology (UK) Limited
Worton Drive, Reading, UK.
{rachel.craddock, adrian.waller,
glyn.jones}@thaligroup.com

Abstract - *In any system-of-systems the potential exists for interactions between systems to occur that may affect the security of the overall system. We present a scenario that aims to highlight such problems, in particular that of security at a network boundary. This scenario considers cooperation and interactions between organisations and systems in the context of a major public event, such as a sporting or entertainment event. Based on this we present a modelling tool able to highlight potential access violations that might occur through transfer of data between multiple organisations and suggest ways to mitigate these vulnerabilities. The use of the modelling tool for network boundary checking is demonstrated, using the example scenario. Suggestions are made as to how security and effectiveness can be achieved by applying safeguards to vulnerable areas, while allowing the free flow of data between organisations where this is known to be safe.*

Keywords: Boundary Check, Component Composition, Crisis Management, Systems-of-Systems.

1 Introduction

In many situations, various entities need to work together in order to provide a joint service or accomplish a complex task. Where we have multiple individually autonomous systems working together in this way, with complex interactions occurring between the systems, the overall resulting construct is often referred to as a ‘system-of-systems’. The definition of a system-of-systems is often left fairly loose, with the choice of what’s meant by an individual system being broadly open (*e.g.* animals, people, organisations, devices *etc.*). However, in the context of this paper, we focus on system-of-systems comprised of devices or software communicating across a network. In this case, we can consider individual systems to be components within a larger *composed* system-of-systems.

The entities, *i.e.* the components in the composed system, could range from personal devices in a small home network to a mixture of devices in a wide-area network, such as would be available to organisations coordinating a public event, *e.g.* the emergency services. Composed

systems are becoming more common. For example they might be used in a business meeting or on military operations. In a business meeting, representatives from different companies may form an ad-hoc network to allow exchange of business reports. In a military operation, troops from multiple nations may need to share information about activities or opponents among themselves, as well as with other non-governmental organisations (NGOs) in order to achieve their shared objectives. Additionally, with the continuous development of computing and communication technologies, the reality of people’s lives is growing ever closer to the notion of ubiquitous computing. We can expect to see this kind of cooperation between system components occurring both more widely and more fluidly as time progresses.

As dynamic, heterogeneous, interoperating systems become more widespread, one of the most important and difficult challenges is to measure the security properties of the composed systems [1]. Because the components belong to different systems and organisations, they have varied requirements for information security. When composing them together problems often emerge due to inconsistent security policies. For example, the security policy of one organisation may be stricter than the policy of another, perhaps by enforcing a stronger encryption algorithm on external interfaces. At the same time a flaw in one system may result in severe consequences for the entire composed system. Besides, the security property measurement itself can be a time-consuming task, if not impossible. Further, component composition has its own unique security aspects and vulnerabilities, making the process of ensuring security more complicated [2].

In this paper we propose a novel set of security assessment tools which have been designed to address component composition security problems. These tools help to analyse different pre-deployment system-of-system scenarios and highlight the post deployment security issues which could happen in real operations. Such tools could give substantial benefits to reduce the potential damage, should these security issues become real problems. To demonstrate these tools, we have developed a scenario which has a particular focus on response to emergency

incidents at major public events, *e.g.* sporting events or festivals. This scenario provides a concrete, practical example that includes interactions between protectors and attackers. It also helps illustrate the problems which can be encountered when adopting a composition security strategy. Initial ideas about potential solutions are presented, and the demonstration applies the security assessment tools to find a solution to the security composition problems.

The remainder of the paper is split into the following sections. In Section II a brief survey of background material is presented. Section III presents our boundary check scenario and associated security analysis. Finally we discuss future work and conclusions in Section IV.

2 Background

In the past, researchers studying secure component composition mainly focused on establishing the most appropriate model with the potential to formulate some property through a form of model-based analysis [3]. Some examples include Non-interference [4] and Composable Assurance [5]. Non-interference can be considered as the ‘original’ composition property; it tries to describe the flow of information through a system. More specifically, it attempts to determine the situation in which sensitive data flows to an unauthorised level through a system, in order to ascertain whether secrecy in the system is being maintained. This is particularly important to the discovery of covert channels in a system where data secrecy is paramount. Composable Assurance is also a composition property, although it takes a more generalised form compared to Non-interference, and indeed most other composition properties. Non-interference properties can be characterised as satisfying the requirement of separability, whereby the security of a system is decided by analysing each component separately [5]. Composable Assurance on the other hand, takes a different approach by considering both the security properties of individual components and the manner in which they interact. From this, we can deduce the security properties of the composed system.

Although many publications have examined securing component compositions [6-12], this has been done almost universally from a theoretical standpoint. Very little academic work can be found that attempts to apply the properties in any practical sense. This perhaps stems from the lack of a suitable practical formulation. Some relevant work with a more practical focus has grown out of the interest in service-oriented and distributed computing technologies [13-18]. Our own work has resulted in the development of an effective analysis tool called MATTS (the Mobile Agent Topology Test System) to test and demonstrate the process of secure component composition [3]. MATTS supports a Direct Code Analysis process [19] that performs a formal check of a service’s code just prior to execution, in order to first establish the properties of each individual component. It also allows a user to specify certain properties during link creation for situations in

which Direct Code Analysis is inappropriate (*e.g.* to mimic the use of a certificate attached to a component asserting certain properties). An XML script written in the *Compose* language is then used to analyse the topology of the composed system and determine its security properties. A number of security property analysis processes have been developed so far. These include a distributed access control check to discover where network data flows may result in access control violations and a distributed input validation check able to identify where inconsistent inputs and outputs might exploit buffer overflow vulnerabilities in components [19]. A further process that we believe to be particularly relevant is that of boundary checking, which we will consider in more detail in the remainder of this paper. The practical automated implementation of composition analysis which is presented here identifies boundary vulnerabilities dynamically during network reconfiguration. This would normally only be possible through careful manual analysis.

3 Boundary Property Establishment

3.1 The Boundary Checking Problem

Boundary checking is perhaps the most effective and sensible procedure in secure component composition. In general, by exploiting a weak external interface provided by one component, an attacker can get access to critical information originally well protected by other components. To prevent it, a boundary check should cover many security-related aspects along the communication channels between a composed system and the external world. As an example, the security properties that need to be checked might include the server version and its last update date and an automated scan of server-side vulnerabilities. It may then be possible to make a sound judgment based on this.

To aid with the explanations contained in this paper, two examples are presented in Figures 1 and 2. Figure 1 shows a composed system consisting of internal nodes only.

As all the nodes are trusted by each other, and there is no connectivity to the outside, the security properties of the individual nodes are assumed to be acceptable to all of the organisations involved. In Figure 2, a connection is desired between one of the internal nodes and an untrusted external node whose security properties are unknown. The internal node is now known as a boundary node and its security capabilities are all that protects the composed system from the external node. Therefore, the boundary node security properties must be at a level which complies with the prescribed security policy for the composed system in order for the connection to be allowed.

3.2 The Scenario

As an illustrative example, consider the following fictional scenario – during a public event, multiple agencies and organisations must work together to ensure that the

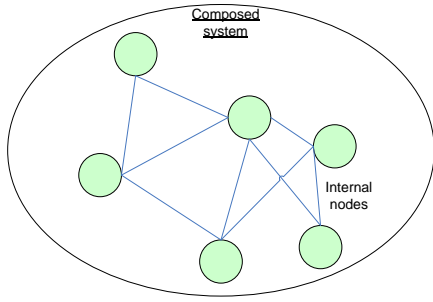


Figure 1. A composed system consisting of trusted internal nodes, with no external connectivity.

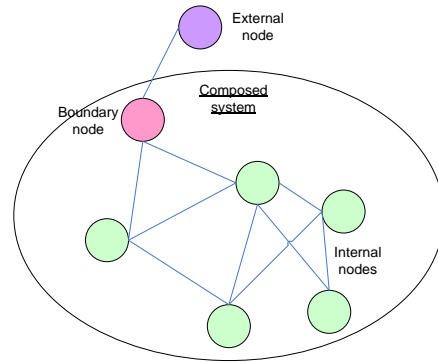


Figure 2. A composed system with a connection to the outside.

Table 1. Roles Involved in the Demonstration and their Security Properties.

Role	Sensitivity Level	Encryption Strength	Staff Skills	Firewall	IDS
Police	0	AES-256	High	Yes	Yes
Ambulance Service	3	RC2-128	Mid	Yes	No
Mobile Network Operator	3	TDES-168	Mid	Yes	Yes
Department of Transport	3	TDES-168	Low	Yes	No
Banks	0	AES-256	High	Yes	Yes
Hospitals	4	AES-256	Mid	Yes	No
Fire and Rescue Service	5	WEP-114	High	No	No
Event Organiser	5	RC2-128	Low	No	No

event proceeds smoothly, especially if an emergency situation develops. In our example, the set of emergency services and organisations shown in Figure 4 have formed a composed system in order to easily share data for the purpose of managing an event. Due to their differing areas of responsibility, the different parties will have different security standards; however, for the purposes of managing a public event, data collected by each party may be useful to the others for identifying malicious activities and to help maintain public order. As a result, each party may obtain temporary access to certain areas of the other parties' database systems. This type of system presents an ideal example for use in demonstrating the boundary checking principal.

The roles assigned to each system or component and their security properties are as shown in Table 1. Specifically, the Sensitivity Level reflects the damage that could be caused in the event that the component or data stored on it is compromised. Sensitivity levels range between 0-9, where 0 is the highest and 9 is the lowest. The Encryption Strength is represented in an "Algorithm-Key Length" format. A component may support more than one encryption algorithm with various key lengths. The algorithms and key lengths listed in Table 1 are those used by the components to establish external connections.

The Encryption Strength is broadly ordered from weak to strong as: *WEP-114*, *RC2-128*, *TDES-168*, *AES-256* [20]. The 'Staff Skills' property considers the technical knowledge and IT skills of each organisation's staff. It might be measured based on training courses that have been attended or certificates obtained. We simply categorise it to levels of *Low*, *Mid*, and *High*. Finally, use of firewalls and Intrusion Detection Systems (IDSs) are also considered.

Let us assume an attacker is trying to break into a Police computer network and steal valuable data such as information concerning criminal proceedings. However, the attempt fails because the defence mechanism adopted by the Police is too strong for the attacker to penetrate. Therefore, one of their alternatives is to attack via other systems that are connected to the Police system, for example the Event Organiser. Knowing this, the attacker shifts their attention from the Police to the Event Organiser's workstations. Due to lower security requirements, the Event Organiser is running a server suffering from a flaw that can be exploited by the attacker. The attacker can use this flaw to gain access to the Event Organiser's system, and it is then straightforward for the event organiser to gain access to the Police data through the connection between the Event Organiser's system and the Police system.

If a boundary check was performed when external devices or organisations attempted to join the composed system, such weaknesses could be exposed, and appropriate measures taken to address them. We will demonstrate how this could be done for the system described in the scenario.

3.3 The Demonstration System

In our demonstration, PCs are used to represent the various organizations, connected as shown in Figure 4. The composed system has a star topology, with the police at the centre. However, we note that this arrangement has been chosen for the sake of clarity; the techniques we have developed consider paths through the topology, ensuring that an alternative arrangement could be analysed similarly. This topology is also represented in the MATTs modelling

```

<process id="check">
  <process action="link = @ilnum[@n]" />
  <process id="link" init="0">
    <process action="link = (link-1)" />
    <process init="0" cond="(!@a[@iln[@n][link]][External]) && (!@a[@iln[@n][link]][Firewall]) &&
      (!@a[@iln[@n][link]][IDS])" action="(safe=0)" />
    <process id="ex" init="0" action="ex=@a[@iln[@n][link]][External]" />
    <process id="sl" init="0" action="sl=@a[@iln[@n][link]][SensitivityLevel]" />
    <process id="es" init="0" action="es=@a[@iln[@n][link]][EncryptionStrength]" />
    <process id="ss" init="0" action="ss=@a[@iln[@n][link]][StaffSkills]" />
    <process init="0" cond="(!ex) && (sl == 0) && ((es < 11) || (ss < 3))" action="(safe=0)" />
    <process init="0" cond="(!ex) && (sl == 1) && ((es < 11) || (ss < 3))" action="(safe=0)" />
    ...
  <process cond="link > 0" config="link" />
</process>
</process>

```

Figure 3. Policy enforcement script fragment.

environment as shown in Figure 8. In this figure, the modelling environment shows the various nodes of the composed system (shown as labelled circles - *P* is the Police, *E* is the Event organiser, *M* is a Mobile network operator, *B* is a Bank, *T* is the Department of Transport, *R* is the Rescue team and fire service, *A* is the Ambulance service and *+* is a Hospital). The circle in the top right corner of the display is a visual indicator of the results of boundary check system analysis. The system analysis indicator turns green to indicate an acceptable network configuration, and turns red if a boundary node/nodes do not meet the specified security policies. In such a situation, communication between the offending boundary and external nodes would be denied and the location of these offending nodes causing the security problem highlighted.

The physical network topology of the network used to model the scenario is shown in Figure 5. This topology is based on the proposed scenario shown in Figure 4, and has five PCs modelling the various organisations as nodes with appropriate properties in the software. The Ambulance Service, the Mobile Network Operator, the Bank and the Department of Transport are modelled on one of the PCs, with the other organisations each modelled on a separate PC. The Event Organiser who is acting as the boundary node for this network is modelled on the PC labelled Boundary Node PC. The PCs are connected across a standard 100BASE-TX Ethernet LAN controlled by a Smooth Wall NAT/DHCP server. The server allows the other PCs to connect in a star topology. Thus this topology simulates the different networks involved in the scenario. The MATTSS system which is used to control the demonstration and process the security checks is instantiated on a separate PC labelled the MATTSS server. This is connected to the organisational network via the central server. Each of the organisations in the network is represented as a node in the MATTSS modelled network as shown in Figure 8. In reality, the MATTSS system might be hosted by the central representative (in this example the Police).

To establish the security requirements at the boundary of the composed system, each organisation checks its own security-related properties (which may be pre-established)

and sends a report to the MATTSS server. Each report is a machine-readable XML format file describing the relevant security properties. For example, in our implementation the report specifies the encryption algorithm used by the system, the key length, whether any firewall or IDS is deployed within the organization and other pertinent information. These property reports are gathered by the central representative in order to carry out a boundary check by utilizing an appropriate algorithm such as that available in MATTSS.

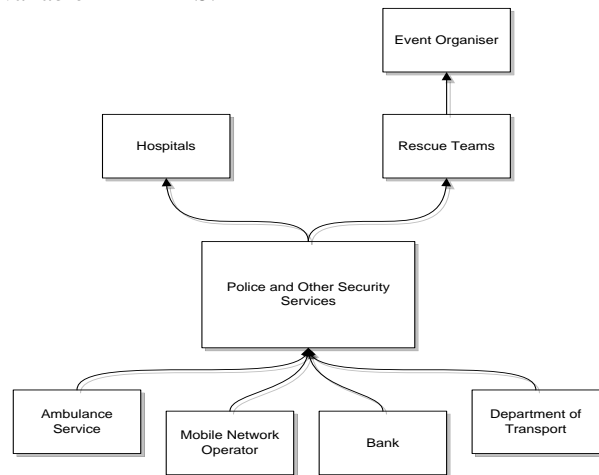


Figure 4. Overview of the composed system.

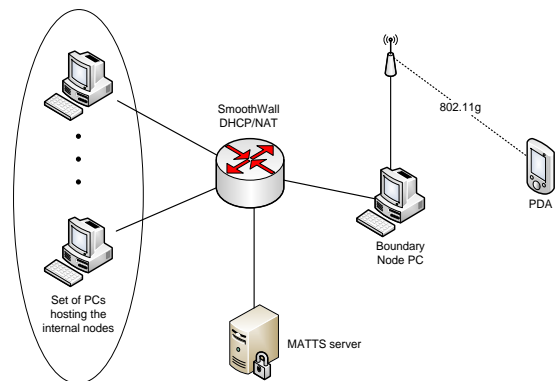


Figure 5. Physical network layout.

Table 2. Minimum Requirements on Encryption and Staff Skills for Components with Different Sensitivity Levels.

Sensitivity Level	Encryption Strength	Staff Skills
0	AES-256	High
1	AES-256	High
2	AES-256	Mid
3	TDES-168	Mid
4	TDES-168	Mid
5	RC2-128	Mid
6	RC2-128	Mid
7	RC2-128	Low
8	WEP-114	Low
9	WEP-114	Low

Consider now the case in which a new device or organisation tries to connect to the system. This is done by the addition of an external node via the boundary node, demonstrated here using a PDA, connected via a PCI/USB 802.11 wireless interface to the boundary node PC. In the scenario described above, new devices or organisations may join the system in the event of an emergency occurring, e.g. a social worker arrives at the hospital and needs to access the composed system. Such a situation also provides an ideal opportunity for the attacker to access the system. Thus, whenever an external device requests access to the composed system, the point at which it is trying to connect needs to be checked to ensure that the overall security of the system is not compromised. For the purposes of the demonstration, we have defined a sample security policy as follows.

“A component will only be allowed to have external connections when it has either Firewall or IDS running, and its Encryption Strength and Staff Skills satisfy the minimum requirements imposed in accordance with its Sensitivity Level.”

The Encryption Strength and Staff Skills minimum requirements for each Sensitivity Level are given in Table 2. Although this policy considers only connection requirements as they relate to the boundary node, the nature of the analysis means that other security requirements could be imposed, for example relating to other nodes (including the external node), to impose trust, authorisation or other security controls.

An example of this policy is shown in Figure 6. A fragment of the simple MATTS script needed to enforce this policy is as shown in Figure 3. The script works by identifying links between internal (trusted) connections and external (un-trusted) connections such as those to the PDA. All of the trusted connections that satisfy this relationship are considered to constitute the boundary and these are required to satisfy the policy. However, in the event that they do not satisfy it, the external connection is refused. An attractive feature of the system is that the security properties of the non-boundary (internal) nodes or the external nodes do not need to be checked in this scenario, thereby improving the efficiency of the overall checking process.

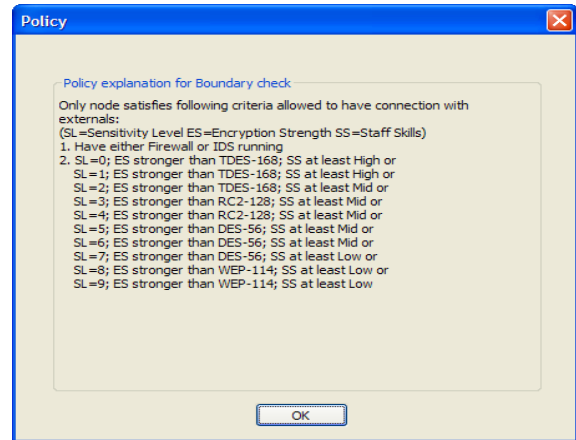


Figure 6. Example MATTS policy for boundary checking.

The user of the PDA is able to specify which of the organisations’ networks he wishes to connect to, i.e. which network will be treated as a boundary node. Initially, when the PDA connects to the network, it is only allowed to communicate with the persistent MATTS security service (on the MATTS server). An application running on the PDA negotiates with the MATTS service to finalise the connection with the boundary node.

This is done by the PDA transmitting its properties in XML format to the MATTS server over a socket connection. MATTS then determines the properties of the device. In particular, since the PDA is considered as an external device, this will effectively result in a new external interface of the composed network.

Having established details of the new device, MATTS then adds the new node with appropriate connections to the network model and uses this to perform a re-evaluation of the network properties. If the new network model is considered secure, then the new device is allowed to communicate with the boundary node and the result is used to assign appropriate rights to the connected device. A re-evaluation is triggered each time the connection structure changes, i.e. a device wishes to connect to a new boundary node.

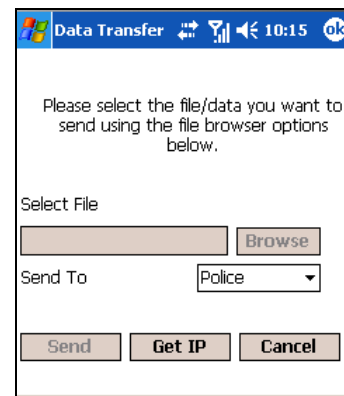


Figure 7. Sending files between organisations once connected to the network using a PDA.

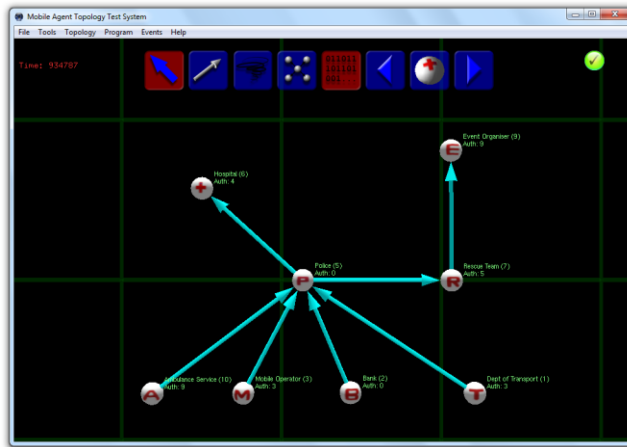


Figure 8. The initial network.

If such a connection is allowed, the user is then given the opportunity to interact with other organisations' devices (either real or modelled) that are present on the network, for example by sending a file to another organisation, as shown in Figure 7.

Returning to the scenario and the application of this policy, if the PDA were to connect to the Police, Mobile Network Operator, Bank or Hospital (organisations whose security properties meet the policy), the system analysis indicator highlighted on the network topology screen would turn green to indicate an acceptable configuration. If the PDA connects to any other organisation that does not meet the specified security policies, *e.g.* the Event Organiser, the indicator would turn red and communication between the nodes would be denied as shown in Figure 9. In addition, a visual indicator is presented in order to highlight the location of the offending boundary node or nodes causing the security problem.

Tests of the system using this configuration show it able to effectively identify potentially unsafe boundary connections based on the policy, and that it is able to achieve run-time re-evaluation as the network topology changes. Through the identification of unsafe boundary connections, it is easy to establish which nodes in the network constitute an increased risk due to their external network interfaces. Greater security can then be applied to these nodes as required.

3.4 Reconciliation

In our present implementation we have considered only the identification of potential security vulnerabilities, but clearly the question of how to resolve them once identified (other than simply refusing communications) remains an important consideration.

In the scenario, communication with the composed system depends on the security properties at the point of access being capable of providing security to the entire system. This is required because the individual systems within the composed system have different security

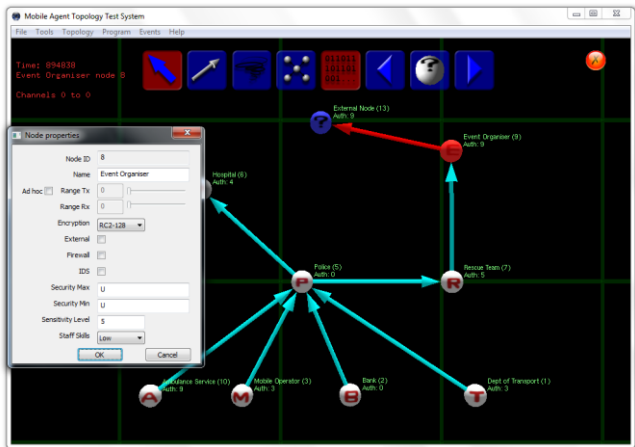


Figure 9. False analysis result.

properties, with some being weaker than others. The boundary security of the composed system needs to be appropriate to protect the highest sensitivity data contained within the system.

In general, communication between systems depends on their security properties being compatible with one another; that is, the way the systems are secured on an individual level must be similar. This has not been considered in our example, as all the organisations within the composed system are trusted. However, as we can see from Table 1, the Police and Bank systems could be said to have compatible security properties and communication between them would be fairly easy to achieve, whereas problems occur when systems such as the Event Organiser's attempt to communicate with systems such as those of the Police where properties may be incompatible.

In order to facilitate composition between these systems a middle ground needs to be found. On the part of the police, sensitivity levels could be decreased while on the part of the Event Organiser's system sensitivity levels could be increased. This causes potential problems such as data leakage. The Police system is highly secure, but allowing sensitive data to flow to the Event Organiser's system, which is less secure, could mean that the Police data is then available to the wrong people. This could be through incompatibilities between sensitivity levels of data and devices; it could be that the Event Organiser's system is compromised (so data is being accessed by unauthorised entities); or it could even be through another system to which the Event Organiser is connected or exposed.

Some kind of reconciliation therefore needs to be performed in order to resolve any incompatibilities. This could be done through the use of automatic policy reconciliation using a reconciliation engine, or by meetings between system administrators deciding the most appropriate course of action. However, it may be that there is no way to reconcile the policies using the existing configuration. In this case, changes to software or hardware may be needed in order to increase the security of the system and allow a connection to be made. In other words,

in the case of the scenario presented here, this would involve targeting appropriate additional boundary check protection where it's needed.

Instead of reducing overall security, it could be maintained by adding better security functionality to the lower security systems, either through Commercial Off-the-Shelf (COTS) Software or third-party components. Adding COTS software to a system does pose security threats of its own; often COTS software is tailored for a general case, rather than the specific cases you might well find in highly critical systems such as a Bank's. Also there are unknown factors such as the potential for bugs in applications, which could be exploited to gain access to the system. Use of third-party components, often added as a quick fix, such as a component to make the encryption technologies available to a system, may also cause potential problems. For example the component developer and security characteristics may not be known, unless it has been evaluated and appropriately certified.

Clearly care must be taken when incorporating additional components into the system in order not to compromise the security of the composition further. This must be weighed against the dangers associated with a lowering of the security levels of the highly secure system to allow data transfer.

4 Conclusions and Future Work

Even in non-time-critical situations, a secure system can be hard to achieve. It follows that in crisis situations involving dynamic interactions between multiple public and commercial organisations, the difficulty of providing adequate protection is substantially increased. It is important however, that we do not impede the flow of data where the need for that data is especially acute. We have presented a network modelling tool that can incorporate dynamic updates based on real network changes and which is able to highlight vulnerable areas of an inter-organisation data network. The intention has been to show how such a tool can be used to centre attention on the areas that are of most importance, allowing adequate safeguards to be put in place, and which we demonstrated using the example of a public event.

At present, only a relatively small collection of data flow related vulnerabilities can be highlighted using our tool, *e.g.* boundary checks and cascade vulnerability detection [21-23], and in our example we used a somewhat simplistic set of data access policies. In future work we intend to consider a wider variety of policies, ideally incorporating real systems commonly used within organisations for the purposes of access control. We believe the extensible nature of the tool will allow us to do this, while at the same time highlighting new problems that might be identified by the system. In the longer term, we hope to assess the effectiveness of the tool using real-world case studies, and tackle the important question of security reconciliation.

While the work remains at a relatively early stage of development, we know of no other tools available for assessing real-world inter-organisation data flows in this way. Moreover, we believe our current implementation provides a good foundation to build on, allowing the application of useful techniques that go beyond the security and effectiveness improvements described here. Other applications include securing data access, allowing organisations with otherwise incompatible security policies to interact, and removing the need for complex security negotiation in advance of cooperation. With growing use of communication between dynamic coalitions of organisations, the issue of secure composition will become increasingly important.

References

- [1] J. McLean, "Trustworthy software: why we need it, why we don't have it, how we can get it," in 30th Annual International Computer Software and Applications Conference COMPSAC 2006, Chicago, IL, USA, 17-21 September 2006.
- [2] L. Gong and X. Qian, "The complexity and composability of secure interoperation," in Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 16-18 May 1994.
- [3] M. Merabti, Q. Shi, B. Askwith, and D. Llewellyn-Jones, "Secure Component Composition for Personal Ubiquitous Computing: Project Summary," Liverpool John Moores University, Liverpool, December 2005.
- [4] P. Ryan, C. Mellon, J. McLean, J. Millen, and V. Gligor, "Non-interference, who needs it?," in 14th IEEE Computer Security Foundations Workshop (CSFW-14), Cape Breton, NS, 11-13 June 2001.
- [5] Q. Shi and N. Zhang, "An effective model for composition of secure systems," *Journal of Systems and Software*, vol. 43(3), pp. 233-44, November 1998.
- [6] R. Focardi and S. Rossi, "Information flow security in dynamic contexts," *Journal of Computer Security*, vol. 14(1), pp. 65-110, 2006.
- [7] A. Bossi, R. Focardi, C. Piazza, and S. Rossi, "Verifying persistent security properties," *Computer Languages, Systems & Structures*, vol. 30(3-4), pp. 231-58, October 2004.
- [8] J. Jurjens, "Composability of secrecy," in *Information Assurance in Computer Networks. Methods, Models and Architectures for Network Security*. International Workshop MMM-ACNS 2001, St. Petersburg, Russia, 21-23 May 2001.

- [9] H. Mantel, H. Sudbrock, and T. Krausser, "Combining different proof techniques for verifying information flow security," in 16th International Symposium on Logic-Based Program Synthesis and Transformation, LOPSTR 2006, Venice, Italy, 12-14 July 2006.
- [10] J. McLean, "A general theory of composition for a class of "possibilistic" properties," IEEE Transactions on Software Engineering, vol. 22(1), pp. 53-67, January 1996.
- [11] S. Tini, "Rule formats for compositional non-interference properties," Journal of Logic and Algebraic Programming, vol. 60-61, pp. 353-400, July 2004.
- [12] D. v. Oheimb, "Information flow control revisited: noninfluence = noninterference + nonleakage," in Computer Security - ESORICS 2004. 9th European Symposium on Research in Computer Security. Proceedings, Sophia Antipolis, France, 13-15 September 2004.
- [13] J. Buford, R. Kumar, and G. Perkins, "Composition trust bindings in pervasive computing service composition," in Proceedings. Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshop-PerCom Workshop 2006, Pisa, Italy, 13-17 March 2006.
- [14] I. Djordjevic, T. Dimitrakos, N. Romano, D. Mac Randal, and P. Ritrovato, "Dynamic security perimeters for inter-enterprise service integration," Future Generation Computer Systems, vol. 23(4), pp. 633-657, May 2007.
- [15] C. Farkas and S. Jajodia, "The inference problem: a survey," ACM SIGKDD Explorations Newsletter, vol. 4(2), December 2002.
- [16] M. Deubler, J. Grunbauer, J. Jurjens, and G. Wimmel, "Sound development of secure service-based systems," in ICSOC '04: Proceedings of the Second International Conference on Service Oriented Computing, New York City, NY, United States, 15-19 November 2004.
- [17] M. Kolberg, E. Magill, D. Marples, and S. Tsang, "Feature interactions in services for Internet personal appliances," in 2002 International Conference on Communications (ICC 2002), New York, NY, 28 April-2 May 2002.
- [18] M. Braem, N. Joncheere, W. Vanderperren, R. Van Der Straeten, and V. Jonckers, "Guiding service composition in a visual service creation environment," in 2006 4th IEEE European Conference on Web Services, Zurich, Switzerland, 4-6 December 2006.
- [19] B. Zhou, A. Arabo, O. Drew, D. Llewellyn-Jones, M. Merabti, Q. Shi, A. Waller, R. Craddock, G. Jones, and K. L. Y. Arnold, "Data Flow Security Analysis for System-of-Systems in a Public Security Incident," in The 3rd Conference on Advances in Computer Security and Forensics (ACSF 2008), Liverpool, UK, 10-11 July 2008.
- [20] S. Garfinkel, Spafford, G., and Schwartz, A, Practical Unix and Internet Security, 3rd ed: O'Reilly Media, Inc., 2003.
- [21] D. Fotakis and S. Gritzalis, "Efficient Heuristic Algorithms for Correcting the Cascade Vulnerability Problem for Interconnected Networks," Computer Communications, vol. 29(11), pp. 2109-2122, 26 July 2006.
- [22] C. Servin, M. Ceverio, E. Freudenthal, and S. Bistarelli, "An Optimization Approach using Soft Constraints for the Cascade Vulnerability Problem," in 2007 Annual Meeting of the North American Fuzzy Information Processing Society, San Diego, CA, United States, Jun 24-27, 2007.
- [23] J.-W. Wang and L.-L. Rong, "Cascade-based attack vulnerability on the US power grid," Elsevier, vol. 47(10), pp. 1332-1336, December 2009.