Intrusion Detection Systems for Community Wireless Mesh Networks

Dwight Makaroff
Department of Computer Science,
University of Saskatchewan, Saskatoon,
Saskatchewan, Canada
makaroff@cs.usask.ca

Paul Smith, Nicholas J.P. Race and David Hutchison
Computing Department, InfoLab21
Lancaster University,
Lancaster, UK
{p.smith,race,dh}@comp.lancs.ac.uk

Abstract

Wireless mesh networks are being increasingly used to provide affordable network connectivity to communities where wired deployment strategies are either not possible or are prohibitively expensive. Unfortunately, computer networks (including mesh networks) are frequently being exploited by increasingly profit-driven and insidious attackers, which can affect their utility for legitimate use. In response to this, a number of countermeasures have been developed, including intrusion detection systems that aim to detect anomalous behaviour caused by attacks. We present a set of socio-technical challenges associated with developing an intrusion detection system for a community wireless mesh network. The attack space on a mesh network is particularly large; we motivate the need for and describe the challenges of adopting an asset-driven approach to managing this space. Finally, we present an initial design of a modular architecture for intrusion detection, highlighting how it addresses the identified challenges.

1 Introduction

Wireless Mesh Networks (WMNs) create a resilient infrastructure using a combination of wireless networking technology and ad-hoc routing protocols that together provide the ability to establish networks in locations with no prior groundwork and where a wired network would be prohibitively expensive or complex. For example, the community of Wray, situated approximately ten miles from the city of Lancaster in the north-west of England, felt strongly that the lack of broadband Internet connectivity (due to their remoteness) in their village was jeopardising local businesses, education, and the community itself.

This prompted residents to approach Lancaster University to initiate a collaboration to deploy WMN technology in the village. The network is currently managed by the residents of Wray with some technical assistance from researchers at Lancaster University. Ishmael *et al.* describe the Wray network [13], which is representative of community-driven WMNs, including Berlin roofnet [24].

The network in Wray has been operational for over three years, and is used by local residents for all manner of online tasks, including e-commerce and work-related activities. In light of this, protection of the network, its users, and their assets from malicious behaviour is now a prime concern. This concern is particularly acute as network-based malicious activity continues to rise and the nature of this activity become increasingly profit-driven and insidious. An important part of a protection (security) strategy is an intrusion detection system (IDS) – a monitoring system whose aim is to detect and characterise malicious behaviour. We have been unable to find a suitable system that is directly applicable in this socio-technically challenging environment, since most IDS platforms are targeted for use in wired enterprise networks.

In this paper, we contribute a set of socio-technical challenges that developers of an IDS for a community WMN need to consider. These are derived from our operational experience, experimentation, and consultation with the Wray community. In short, these motivating challenges, described in Section 2, are a product of the diverse range of anomalous but benign activity that can occur, the ease in which malicious entities can be introduced to a network, limited hardware resources and scarce bandwidth when a network is under attack, as well as social issues, such as the privacy concerns of users.

The potential set of attacks on a community WMN is large and is, to some degree, network-context specific. We argue that, given the potentially limited computing and monetary resources that are available in a community WMN context, it is essential to understand the locally meaningful attack vectors. We propose that an asset-driven approach

978-1-4244-2575-4/08/$20.00

to understanding likely attacks is a good way to achieve this, and highlight some of the issues with conducting such a venture in this setting. This motivation, a sample of the diverse set of assets derived from a focus group in Wray, and ideas to elicit assets in more disparate communities are presented in Section 3.

We describe our initial design of an architecture for intrusion detection in a community WMN context in Section 4. A modular architecture is proposed that enables context-specific anomaly detection components to be deployed on suitably capable devices. Meanwhile, attack characterisation components use attack profiles and network-context information to potentially determine whether anomalies relate to attacks or benign behaviour. We conclude with a description of how our architecture addresses the challenges identified and suggest activities for further work.

2 Motivating Challenges

Two related characteristics of community WMNs make intrusion detection particularly challenging. The network and end-systems are particularly *vulnerable to attack* because of the relative ease of access to the wireless medium and mesh infrastructure. No forms of attack are reduced from those that are applicable in other contexts. Related to this, the activity of *intrusion detection is in itself difficult* to perform for a number of reasons as identified in this section.

2.1 Determining Malicious Behaviour

A network attack can result in a loss of service (e.g., as a consequence of a DDoS attack). Similarly, legitimate user behaviour can lead to a loss of service (e.g., as a consequence of a flash crowd event). We have seen on the Wray village WMN that legitimate peer-to-peer traffic can have an adverse effect on network operation [14]. Furthermore, environmental effects such as bad weather or large vehicles blocking the line-of-sight of antenna can result in poor levels of connectivity, or even none at all (a phenomenon we have also experienced).

The role of an IDS is to generate an alarm when malicious behaviour is detected. Legitimate but nonetheless anomalous (and potentially adverse) events could lead to an intrusion detection system generating false positives [19]. A sufficiently high enough false positive rate could render the system unusable. A key challenge when developing an IDS in this context is to understand to what extent is it possible to generate alarms solely in the presence of malicious behaviour, given the diverse attack space of a community WMN (see Section 3) and the range of legitimate but anomalous events that could occur.

There are two approaches in the literature used to identify malicious behaviour: misuse detection and anomaly detection [27]. Misuse detection requires significant computation, which is often not available on the hardware in WMN contexts. Thus, this paper will focus primarily on anomaly detection, but the architecture we provide allows for the addition of misuse detection in certain circumstances.

2.2 System Administration

There is a continuum of ad-hoc approaches to community WMN administration. For example, the Wray village network is managed by a consortium of local residents [6], with technical assistance from researchers at Lancaster University – whereas other networks could operate without any such consortium and be managed in an entirely decentralised manner. This is unlike typical enterprise networks, where a single entity (such as a systems services department) is chartered with administering the network and implementing a security policy, which includes services such as intrusion detection.

Furthermore, implementers of intrusion detection systems in an enterprise setting are typically qualified and motivated to do so – it represents their day job – this is not necessarily the case in a WMN context. This is not to underrate the technical expertise of community WMN providers, but there is likely to be a greater diversity of expertise, inclination, and time available for managing an intrusion detection system.

2.3 Diverse Resource Availability

The availability of hardware resources (e.g., processor, memory, and storage capacity) can be diverse on a community WMN. This is important because it is unlikely that dedicated IDS hardware will be deployed on such a network. Available devices could be made up of relatively resource abundant personal computers, through to tightly constrained systems, such as wireless access points that tend to have a smaller form factor, use little electricity, and are fan-less.

To get an indication of the ability of mesh devices to perform intrusion detection activities, we conducted experiments with a NETGEAR WG302 wireless access point [1] running OpenWrt [2], using IEEE 802.11g. Connection tracking was enabled – a necessary service if the device is performing Network Address Translation (NAT), as is the case with many deployed WMNs. A source host sent, via the mesh device, a number of different traffic loads to a sink for 60 seconds. The `top` application sampled CPU utilisation every ten seconds on the mesh device, which were averaged; the average CPU utilisation over three runs is shown Fig. 1. A low deviation was seen on these results.

The results in Fig. 1 demonstrate that under *normal* traffic loads the CPU utilisation of the mesh device is quite low, including forwarding trace traffic from the Wray village net-

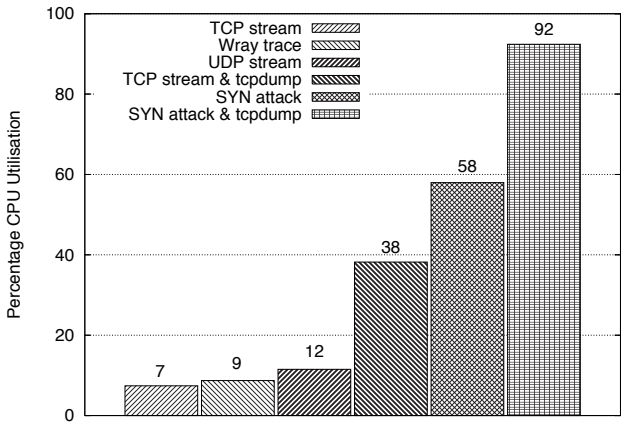

Figure 1. NETGEAR WG302 CPU utilisation under different conditions

work in real-time. The TCP and UDP stream results show a single stream being sent at full-rate; UDP packets had a 20 octet payload. By capturing a TCP stream using `tcpdump` and writing it to `/dev/null`, a significant (≈30%) increase in CPU utilisation is seen. Because of connection tracking, a TCP SYN attack [28] without source address spoofing and random source port selection has a significant impact on CPU utilisation. Combining a TCP SYN attack with packet capturing pushes the CPU utilisation to an average of 92%. These results suggest that under normal traffic loads a mesh device is well provisioned, but under attack conditions, resource availability can be significantly impaired. Thus, some intrusion detection activities can be performed on mesh devices, but care should be taken to not inhibit their performance.

Wireless network speeds continue to increase (e.g., the IEEE 802.11n standard has purported data rates of up to 74 Mbps). In spite of this, the wireless medium is still a relatively limited resource, especially when under attack, for example, during a wireless jamming attack where adaptive rate control techniques react to poor channel quality and can significantly reduce transmission rates [11]. Stone-Gross *et al.* [26] conducted an analysis of wireless network performance in the presence of an ICMP ping sweep (Worm propagation behaviour) and a TCP SYN attack. They found that in the presence of this malicious behaviour, utilisation of the network was close to 100%, at peak times of viral activity 50% of the traffic sent were retransmissions, and the average TCP RTT climbed by over 50%. Clearly, in light of results such as these it is important to consider the potential availability of and overhead associated with control traffic that may need to be sent by a distributed IDS.

2.4 Trust Issues

It is relatively straightforward to introduce new infrastructure to a WMN that could be used for malicious purposes, for example, to instigate a phishing attack [17]. There are protocols that can be deployed to make this more difficult to achieve (e.g., WPA2 Enterprise Mode [4]), but they are rarely enabled. Furthermore, providing accountability is challenging because it is easy for a user to join or add infrastructure to the network. Even authenticated users may be difficult or prohibitively expensive to trace. This lack of accountability and traceability reduces the risk of behaving maliciously, suggesting that information from distributed sensors regarding malicious activity (or a suggested lack of activity) cannot be inherently trusted. For example, false information about the existence of an attack could be used to orchestrate a Denial of Service (DoS) attack as a result of a remediation activity (e.g., modifications to firewall rules could block access to legitimate services as a result of bogus information). In summary, a developer of an IDS for a community WMN cannot inherently trust the information it will receive for detection activities, and measures should be taken to address this problem.

2.5 Privacy Issues

One of the main operations an IDS performs is to analyse user-generated data to detect malicious behaviour, which raises privacy issues. In the context of an enterprise this is not a significant problem, as only a small number of systems administrators have access to the data that is being analysed, and measures are taken to protect such data. Furthermore, users of the enterprise network expect that their on-line activities can and will be monitored for the purpose of securing their systems and associated data (in the UK, the Lawful Business Practices Regulations Act permits such activity).

However, there are significant privacy issues that such activity raises in the context of a community WMN. For example, consider an IDS that monitors a Web cache's log files to determine if known Web sites that are being used to exploit user's computers have been visited. The information the IDS is examining is clearly sensitive, especially when a direct relationship can be trivially formed between on-line activity and an individual, as can be the case in a community WMN.

Interestingly, analysis of a focus group meeting related to security with the users of the Wray village network suggests that a spectrum of attitudes to privacy exist. We found that some users were very concerned about privacy in an on-line context (for example, only using their given name to identify themselves when using the on-line *doodle* meeting arrangement tool – a measure they cited as protecting their privacy), whereas others seemed significantly less con-

cerned; questioning why anybody would be interested in their private data at all. This diversity may arise from different perceptions of the risk associated with having private data on-line.

3 Attack Space Management – Identifying Critical Assets

The attack space of a community WMN is particularly large. In addition to the plethora of attacks that exist in the wired Internet, a WMN is susceptible to local attacks via the wireless medium (e.g., eavesdropping and jamming attacks) and to the mesh infrastructure itself (e.g., vandalism of mesh devices), both of which are in the public domain. Malicious mesh infrastructure can be introduced with relative ease, leading to potential man-in-the middle attacks, for example. Furthermore, because of their relatively static and long-lived nature, they are arguably more vulnerable to certain types of attack, e.g., brute-force attacks, than dynamic networks, such as MANETs.

We suggest that the probable attacks will, to some degree, be specific to a network's context. For example, the network's geographical location may influence how probable eavesdropping is – it is unlikely an attacker will travel to a rural village in the North of England to listen to the wireless medium; this may be a more approachable and lucrative activity in a large city. Other forms of context, could include the (regular) activities of network users and the locally provisioned services.

The size and context-specific nature of the attack space suggests it is essential that measures are taken to determine the most probable and damaging attacks for a particular network. By doing this, an IDS can be developed that makes the best use of the limited monetary, time, and computing resources that are available in a community WMN context.

An approach to identifying the most important attacks to detect involves discovering critical assets (those assets that, if exploited, would have the highest impact and also have the highest probability of being attacked) and the vulnerabilities associated with them. Attempts to exploit the identified vulnerabilities can then be detected. (This activity will, of course, inform appropriate prevention mechanisms to employ, further helping to manage the attack space.)

Within an enterprise setting, processes such as the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) process [3] can be employed to ascertain an enterprise's critical assets[1] and the technical vulnerabilities associated with them. A process such as this cannot be trivially transposed to a community WMN context. For instance, people may be less willing to share details of their assets with others (for privacy reasons). The set of assets to protect in a community WMN context may also be quite different.

We carried out a focus group in Wray with the aim of eliciting assets from a subset of the network's user group. We found the assets identified to be quite diverse. Users described such as things as digital media, a community Web portal, credit card details, personal identity and reputation, and their children as 'assets'. There was also a surprising amount of work-related use of the network, which contains sensitive assets.

Currently, we are investigating ways that an activity such as OCTAVE could be carried out in community WMNs. Organising such an activity in more socially and organisationally disparate communities could be problematic. With this in mind, we are investigating the suitability of forms of cultural probes [9] and on-line questionnaires that require less contact time to identify assets.

4 Community WMN IDS Architecture

With an understanding of the assets and associated vulnerabilities in a given context, an IDS can be appropriately configured. We propose a modular architecture for community WMN intrusion detection, which is depicted in Fig. 2. The primary motivation for a modular design is the heterogeneity of the systems that participate in detection. For example, it enables lightweight anomaly detection to be carried out on resource constrained mesh devices and more resource intensive characterisation on more powerful end-systems. Another motivation for modularity is that the deployment context affects the priorities of detection and protection, as discussed in Section 3, as the primary user group determines from their asset valuation. In these different contexts, some modules may become irrelevant if the users do not value the type of asset that the modules can protect. We will discuss the components of the architecture in the following sections.

4.1 Anomaly Detection Components

The first phase of intrusion detection is to highlight anomalous behaviour that could be caused by attacks. The system features to monitor for anomalies should relate directly to the critical assets being protected – for example, log-in attempts to a server that hosts a community Web site. Detection components may execute on a range of devices, such as mesh devices and end-systems, monitoring a number of pertinent features, similar to those identified by Kandula *et al.* [15].

We envisage that mesh devices will perform anomaly detection based upon relatively cheap metrics, for example, looking at packet headers to determine anomalous rates of

[1] Example assets in an enterprise setting could include an employee's email or a customer database.

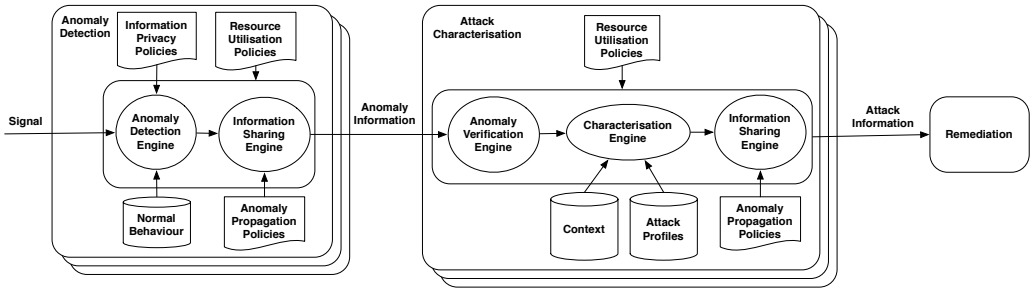

Figure 2. Community WMN IDS Architecture

certain types of traffic from hosts. More expensive payload-based detection is also possible, but should be performed only on more powerful end-systems. Another task for an end-system might be to host a *honeypot* [21] – a machine that should receive very little legitimate traffic, and thus all traffic observed at a honeypot is potentially malicious.

The *anomaly detection engine* yields a description of an anomaly; a format such as Intrusion Detection Message Exchange Format (IDMEF) [8] could be used to describe anomalies to be distributed to characterisation modules. The *information sharing engine* is responsible for distributing anomaly information to attack characterisation components as well as other anomaly detection engines. Detection components will be configured with a set of characterisation components, based upon the privacy policies described in Section 4.4.

4.2 Attack Characterisation Components

Low-level alerts must be correlated with one another to establish confidence that observed behaviour is consistent with a threat and not just noise [5]. These alerts may be from multiple sources, enabling a network-wide perspective. Heuristics have been used extensively to identify malware, based on anomalies in traffic patterns [10, 26]. By just examining data from packet headers, the overhead of such procedures can be kept low. However, characterisation is easier with access to packet payloads. Pattern-matching algorithms (e.g., CREST [22]) with efficient data structures can detect malicious traffic with reduced complexity.

The *attack profiles* repository contains descriptions of anomalous behaviour that relate to attacks that are primarily associated with the locally relevant assets. The aim being to take the anomaly information given by the anomaly detection components and relate this to specific attacks. Alongside this, *context* about the network, e.g., current radio signal strength, meteorological information, geographical location of systems, and normal usage patterns, can be used by the characterisation engine to determine whether reported anomalous behaviour relates to the current benign network context, and can therefore be negated when determining the existence of an attack. For example, poor TCP throughput witnessed by a system could relate to rain, rather than an attack by a malicious mesh device that is processing its data traffic and adding delay.

To address the trust issues brought about by the ease in which bogus infrastructure can potentially be introduced into a mesh network, before characterisation of anomalies takes place, the alert is passed through a *verification engine*. The result of this is a measure of confidence in the accuracy of the anomaly identified by a system. The verification engine could be tied to a reputation system [12], for example, where confidence in the alert is related to the source system's reputation. A confederation of verification engines can exchange model information amongst themselves in order to build confidence in the conclusions drawn [7].

To simplify the initial implementation of the characterisation engine, we expect verification to yield a boolean value – trusted, wherein anomaly information is used for characterisation, and untrusted, wherein the information is discarded. The application of continuous values and how they could be applied to give a measure of confidence in the alert generated by the IDS will be investigated later.

4.3 Resource Utilisation Policies

The potential lack of dedicated IDS hardware will mean that the systems used for intrusion detection will have some other function, such as a home office computer or a wireless hop in the network. Therefore, in a similar fashion to techniques suggested by Sharma and Byers [23], the resources an IDS uses on a system should be tuneable (e.g., only use resources when the system is otherwise idle), or have a small enough footprint that it does not adversely effect the normal operation of the system. How the resources of a system can be used for intrusion detection should be expressed through policies.

The goal of small footprints has been achieved by some

intrusion detection systems and filtering mechanisms [7, 18, 25], for implementation in FPGAs or similarly sized embedded systems. Selecticast [10] and Partial Completion Filters [18] both make specific attempts to be lightweight and scalable.

In Section 2, we highlighted how the wireless medium can be a limited resource under attack conditions. Techniques to ensure alert traffic has higher levels of QoS than data traffic, for example, in attack conditions could aid the delivery of messages during attacks. How traffic should be prioritised and the volume of control traffic sent by a system can be expressed through policies that will affect the way a system handles all its traffic, including data traffic.

4.4 Privacy Policies

Privacy policies relate to two classes of information – the raw data used for anomaly detection (e.g., log-in attempts, data traffic, and log files) and output from the IDS (anomaly and attack information). It should be possible for users of the community WMN to express how and with whom this information is used and shared. For example, users may be comfortable with an IDS looking at packet headers, but averse to deeper packet inspection. (It is unlikely that users will express concerns in these terms, but the implications of certain strategies (e.g., deep packet inspection) could be explained and the acceptability of these determined.) Also, users may wish to only share information with a subset of the entire community, for example, only sharing raw data with systems that are operated by family and friends. Bloom filters allow patterns (in data packets) to be correlated while protecting sensitive information [20].

4.5 Remediation

Having detected the existence of an attack, the goal is to remediate. Attack information exported by the IDS should help to configure and inform an appropriate remediation strategy. Details such as addresses of malicious and targeted systems, the severity of an attack, and the confidence the IDS has in an alert (a measure derived from the results of the verification engine and the likelihood of anomalies being a manifestation of benign behaviour) will be used. We anticipate remediation may itself cause anomalous behaviour; understanding and closing this control loop is a matter for future work, and is being considered in the EU FP7 INTERSECTION project.

5 Related Work

Lightweight detection will be necessary on mesh devices. A lightweight and tuneable method of identifying traffic without access to payload data is provided by BLINC [16]. It operates on patterns identified regarding inter-host traffic, as well as examining the functions performed by a host and associated flow characteristics.

End-systems are capable of more powerful processing, and could be used to store data captured at the routers in times of high alert. Some of the analysis could be done offline, such as is currently done in the UCSB Meshnet Project [26]. For our system to be effective, the work done at the mesh devices must be done in real-time.

The FLIPS system adds extra feedback into an anomaly detection mechanism to reduce the rate of false positives [19] and to recognise injected code to prevent intrusions. Our system has similar goals, and approaches detection with a defense in depth attitude, but needs to be even more lightweight in the normal case.

6 Conclusion

We have identified a set of socio-technical challenges that a developer of an intrusion detection system needs to consider in a community wireless mesh networking context. An initial design of a modular architecture for an intrusion detection system, which addresses these challenges has been presented.

Modularity helps to address the challenge of diverse resource availability in this context, enabling anomaly detection and attack characterisation activities to be conducted on suitability capable devices (e.g., lightweight detection on mesh devices). Furthermore, modularity enables the application of context-specific anomaly detection components that relate to the vulnerabilities associated with local assets.

Privacy concerns are realised by policies that determine which components are privy to both user-generated data and information derived by the detection system. We envisage these policies to reflect the diverse set of privacy concerns elicited through consultation with a subset of the Wray network user community, and follow normal social groupings.

We propose an anomaly verification engine that is used to address the trust issues brought about by a lack of accountability in a community WMN context and the ease in which bogus devices can be introduced into the system. A number of approaches to implementing this engine are suggested, which in the first instance we propose should yield a boolean result.

A key challenge in this space is determining whether anomalous behaviour is benign or malicious. We propose that attack characterisation components could use attack profiles and context about the local network to resolve this issue. Further work in understanding how this information can be collected, managed, and related is necessary.

Our approach relies upon an activity whose aim is to identify critical assets, their vulnerabilities, and associated attacks. The benefits of performing an activity of this sort

are well understood; however, we feel conducting an activity such as this is *essential* in a community WMN context because of the diverse and context-specific attack space, and the highly constrained resources that are available.

Our architecture does not explicitly address the challenges associated with system administration. We propose to conduct further consultation with the Wray user group to understand their willingness and ability to conduct these activities. This will influence which aspects and to what extent we will aim to make the system self-organising and how we can simplify some activities. Issues described by West [29] will be considered that may help us address the inertia often exhibited by users when carrying out security-related tasks.

Acknowledgements

Paul Smith's research is supported by Telekom Austria AG. Dwight Makaroff was partly supported by a UK EPSRC research grant (EP/F038496/1) when this research was conducted. The authors are grateful to the members of the Wray village community that took part in the focus group, and to Sara Bury, Fabian Hugelshofer and Johnathan Ishmael for their contributions to this research.

References

[1] NETGEAR ProSafe 802.11g Wireless Access Point WG302. http://www.netgear.com/Products /APsWirelessControllers/AccessPoints/WG302.aspx.

[2] OpenWrt. http://openwrt.org/.

[3] C. Alberts, S. Behrens, R. Pethia, and W. Wilson. Operationally Critical Threat, Asset, and Vulnerability Evaluation Framework, version 1.0. Technical Report CMU/SEI-99-TR-017, Carnegie Mellon University, June 1999.

[4] W.-F. Alliance. Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks, Apr. 2003.

[5] S. Cheung, U. Lindqvist, and M. Fong. Modeling Multistep Cyber Attacks for Scenario Recognition. In *Information Survivability Conference and Expo.*, pages 284–292, Washington, DC, Apr. 2003.

[6] W. C. Communications. Wray community communications. http://www.wraycomcom.org.uk/.

[7] G. Cretu, J. Parekh, K. Wang, and S. Stolfo. Intrustion and Anomaly Detection Model Exchange for Mobile Ad-Hoc Networks. In *Consumer Communications and Networking Conference*, pages 635–639, Las Vegas, NV, Jan. 2006.

[8] H. Debar, D. Curry, and B. Feinstein. The Intrusion Detection Message Exchange Format (IDMEF). http://www.ietf.org/rfc/rfc4765.txt, Mar. 2007.

[9] W. Gaver, A. Dunne, and E. Pacenti. Design: cultural probes. *Interactions*, 6(1):21–29, 1999.

[10] P. Gross, J. Parekh, and G. Kaiser. Secure "Selecticast" for Collaborative Intrusion Detection Systems. In *Workshop on Distributed Event-Based Systems*, pages 50–55, Edinburgh, Scotland, May 2004.

[11] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and Mitigating the Impact of RF Interference on 802.11 Networks. In *SIGCOMM*, pages 385–396, Kyoto, Japan, Aug. 2007.

[12] M. Gupta, P. Judge, and M. Ammar. A Reputation System for Peer-to-Peer Networks. In *NOSSDAV*, pages 144–152, Monterey, CA, USA, June 2003.

[13] J. Ishmael, S. Bury, D. P. Pezaros, and N. J. Race. Deploying Rural Community Wireless Mesh Networks. *IEEE Internet Computing Magazine*, 12(4):22–29, July–August 2008.

[14] J. Ishmael and N. Race. Routing challenges: Peer to peer applications on a community mesh network. In *WiMesh-Nets'06*, Waterloo, Canada, Aug. 2006.

[15] S. Kandula, S. Singh, and D. Sanghi. Argus - A Distributed Network Intrusion Detection System. In *SANE*, pages 333–350, Maastricht, The Netherlands, May 2002.

[16] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark. In *SIGCOMM*, pages 229–240, Philadelphia, PA, Aug. 2005.

[17] R. Kay. Quickstudy: Phishing. *Computerworld*, Jan. 2004.

[18] R. Kompella, S. Singh, and G. Varghese. On Scalable Attack Detection in the Network. In *IMC*, pages 187–200, Taormina, Sicily, Italy, Oct. 2004.

[19] M. Locasto, J. Parekh, A. Keromytis, and S. Stolfo. FLIPS: Hybrid Adaptive Intrusion Prevention. In *RAID*, pages 82–101, Seattle, WA, Sept. 2005.

[20] J. Parekh, K. Wang, and S. Stolfo. Privacy-Preserving Payload-Based Correlation for Accurate Malicious Traffic Detection. In *SIGCOMM Workshop on Large-Scale Attack Defense*, pages 99–106, Pisa, Italy, Aug. 2006.

[21] G. Portokalidis and H. Bos. SweetBait: Zero-hour Worm Detection and Ccontainment using Low- and High-interaction Honeypots. *Computer Networks Journal*, 51(5):1256–1274, 2007.

[22] G. Portokalidis, A. Slowinska, and H. Bos. Argos: an Emulator for Fingerprinting Zero-Day Attacks for Advertised Honeypots with Automatic Signature Generation. In *Eurosys*, pages 15–27, Leuven, Belgium, Apr. 2006.

[23] M. Sharma and J. Byers. Scalable Coordination Techniques for Distributed Network Monitoring. In *PAM*, pages 349–352, Boston, MA, Mar. 2005.

[24] R. Sombrutzki, A. Zubow, M. Kurth, and J.-P. Redich. Self-Organization in Community Mesh Networks The Berlin Roofnet. In *Operator-Assisted (Wireless Mesh) Community Networks*, pages 1–11, Berlin, Germany, Sept. 2006.

[25] H. Song and J. Lockwood. Efficient Packet Classification for Network Intrusion Detection using FPGA. In *FPGA*, pages 238–245, Monterey, California, USA, Feb. 2005. ACM.

[26] B. Stone-Gross, C. Wilson, K. Almeroth, E. Belding, H. Zheng, and K. Papagiannaki. Malware in IEEE 802.11 Wireless Networks. In *PAM*, pages 222–231, Cleveland, OH, Apr. 2008.

[27] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer. A Comprehensive Approach to Intrusion Detection Alert Correlation. *IEEE Transactions on Dependable and Secure Computing*, 1(3):146–169, 2004.

[28] H. Wang, D. Zhang, and K. G. Shin. Detecting SYN Flooding Attacks. In *IEEE INFOCOM*, pages 1530–1539, New York, NY, USA, June 2002.

[29] R. West. The psychology of security. *Communications of the ACM*, 51(4):34–40, 2008.