Next Generation Firewalls for Dynamic Coalitions

Saritha Arunkumar* Stephen Pipes* Christian Makaya† Elisa Bertino‡ Erisa Karafili§ Emile Lupu§ Chris Williams¶
*IBM UK
†IBM T.J. Watson Research Center, USA
‡Purdue University, USA
§Imperial College, London, UK
¶DSTL, UK
Email: saritha.arun@uk.ibm.com

***Abstract*—Firewalls represent a critical security building block for networks as they monitor and control incoming and outgoing network traffic based on the enforcement of predetermined security rules, referred to as firewall rules. Firewalls are constantly being improved to enhance network security. From being a simple filtering device, firewall has been evolved to operate in conjunction in intrusion detection and prevention systems. This paper reviews the existing firewall policies and assesses their application in highly dynamic networks such as coalitions networks. The paper also describe the need for the next-generation firewall policies and how the generative policy model can be leveraged.**

***Index Terms*—Firewall, intrusion detection and prevention, generative policy, machine learning, coalitions networks.**

I. Introduction

Firewall systems have evolved considerably over the years with many different types of firewall in existence today that perform a variety of purposes. Typically, firewall systems are designed to protect networks with well-defined boundaries, such as in enterprise networks. Due to the nature of contemporary firewall capabilities, solutions often require a high level of computational power to analyze network traffic and to rapidly mitigate threats. Firewalls may also supplement additional security measures, such as access control to network resources as part of a broader security architecture. In most of the current firewalls, the rules are statically and manually defined. Moreover, the rules increase with the size of the network and their management might become a major issue. For example, conflicting rules may co-exist, leading to potential vulnerabilities and threats.

Whilst firewall systems are effective barriers to threats in traditional networks they are not adequate for dynamic coalition networks that may involve autonomous mobile devices (e.g., mules, drones, robots) and Internet of Things (IoT) systems. In coalition networks, firewalls are expected to operate in environments that are characterized by volatility, uncertainty, complexity, and ambiguity. The protected systems may also be highly dynamic, such as a system that applies moving target defense techniques, which results in various components of the systems to change in response to variations in its environment.

Hence, there is a need of new generation of firewalls that is able to quickly adapt to changes without requiring the manual intervention of a human operator. We notice that requiring human intervention may not be scalable when dealing with huge numbers of network devices, such as various IoT use cases, which will exceed human capacity. Apart from such a consideration, many networks device may become disconnected because of intermittent connectivity, especially when dealing with wireless and mobile ad-hoc networks (MANETs). Moreover, an autonomously adaptable firewall may be more effective in defending itself from attacks aiming at bypassing it. However, it is important to ensure that an adversary could not use the autonomous behavior of the firewall to create new attack vectors based on understanding of its behaviors and responses under threats. Some of the fragmentation issues and attacks expressed in [1] applies to firewall evasions.

Furthermore, in previous work [2], [3], [4], we considered the challenges faced during coalition operations for achieving effective dissemination of information from diverse and distributed physical sensors and decision makers across coalition partners with varying levels of trust and uncertainty. This research, which cast a risk-value trade-off as a problem of inference management in the context of sharing information led to advances in the form of an *inference management firewall* that was enabled at various edge and gateway nodes in coalition networks and that provided for trust-driven semantic-level control over the flow of sensory information, much like contemporary firewalls do at syntactic levels. These capabilities should be incorporated in our next generation firewalls and firewall policies.

As in coalition environments, it is likely that a firewall may have to enforce policies from different organizations, tools for real-time analysis of such policies are a critical component of next-generation firewall technology. Agreeing on the specific set of polices and rules to be applied during the mission brings up further challenges. The deployable policies can lead to conflicting actions, which are solved manually by sometimes bringing human errors, or by using inappropriate rules ordering. Another interesting problem is the translation of high level policies into low level ones, for example coalition mission requirements to managed devices policies. In this case, low level policies sometimes fail to represent the notions expressed by the high level ones. This problem becomes

prominent when dealing with different firewalls configurations and their high-level policies. Deciding the appropriate order of the latter, and their distribution among the various firewalls is not a trivial problem. The complexity of the problem increase when we deal with dynamic coalitions networks. In this case, the order between the firewall rules is lead by the coalitions goals and their participants.

In this paper we will conduct an extensive analysis of current firewalls to evaluate their capabilities with respect to coalition environments. We will then identify a number of requirements that next-generation firewalls should address and outline a novel firewall architecture based on the notion of generative policy model [5]. Different parts of a coalition are governed by their own sets of policies, which are defined as directives used to guide their actions. The vision of distributed coalition intelligence requires a dynamic, secure and resilient information infrastructure that needs to conform to the policies of each coalition member. The appropriate policy based management framework will help to attain key attributes such as autonomous operation, system composition, and control of element interaction.

The rest of this paper is organized as follows. Section II describes the key concepts of firewalls and presents a well-know open source firewall namely pfSense. Section III describe intrusion detection and prevention systems (IDS/IPS) and highlights three open source projects, that can be used alongside with firewall. Section IV highlights the issues and challenges in IDS/IPS described in previous section. while Section V discusses inference management firewall. Section VI presents the distributed firewall anomalies. Section VII highlights the need for advanced next-generation firewall and the work needed for the same for dynamic network and in coalitions environments. Finally, Section VIII concludes the paper.

II. Overview of Firewall Concepts

In this section, our goal is not to cover all existing firewalls, which is not possible due to the pages limitations constraints. We will focus on the key concepts of firewall and review a well known open source firewall that includes various packages that can be used alongside. The basic role of firewall is to analyze and filter traffic between network segments. It allows filtering packets based on their characteristics and perform actions on the packets that matches the specified policy or firewall rules. The most common application is to protect traffic between an internal network and the Internet.

Firewall rules specify the match conditions for traffic and the actions to be taken if the match conditions are satisfied. The traffic can be matched on a number of characteristics, including IP addresses and ports fields, protocol, and ICMP type. A series of firewall rules is called a ruleset. Rules are executed in numerical sequence, according to the rule number, from lowest to highest. If the traffic matches the characteristics specified by a rule, the action of the rule is executed; if not, the the next rule is executed, and so on. Firewall rules are defined as ECA (Event-Condition-Actions) rules. In ECA rules, an event triggers the automatic validation of stated conditions and actions, if the conditions hold. When multiple firewall rules are defined within a system, their interactions can be difficult to analyze, since the execution of one rule may cause an event which triggers another rule or ruleset. These rules may in turn trigger further rules and there is indeed the potential for an infinite cascade of rule firings to occur.

There are two types of firewall: *stateless* and *stateful*. A stateless firewall considers every packet in isolation. Packets can be accepted or dropped according to only basic ACL (access control list) criteria such as the source and destination fields in the IP or TCP/UDP headers. It does not store connection information and has no requirement to look up every packets relation to previous flows, both of which consume small amounts of memory and CPU. However, a stateful firewall keeps a state table of previously seen flows, and packets can be accepted or dropped according to their relation with previous packets. As a general rule, stateful firewalls are generally preferred where application traffic is prevalent. In the next section, we will review an open source firewall that comes with several security packages.

A. PfSense

pfSense [6] is a free network firewall distribution which includes additional features that are not available in some commercial solutions. pfSense is a stateful firewall with packet inspection, meaning the state table maintains information of the open network connections. It doesn't require a dedicated appliance for its deployed, since it is a software package. Hence, it is suitable for virtualized environments or in dynamic networks. While providing the basic role of a firewall, i.e., filtering traffic by source and destination IP fields (e.g., addresses and ports) and protocol, pfSense comes up with additional features. For example, it uses p0f utility for advanced passive traffic and operating system fingerprinting, e.g., allows to filter the OS behind the initiated connections. Packet normalization is enabled by default with pfSense. It offers various mechanisms for handling and optimization of the state table. Maintaining the state table is crucial for high availability (HA). Since the state table is replicated to all backup configured firewalls, the network connections are not disrupted during a failover. pfSense can be deployed and configured with several other open source security software and packages such as Snort and Suricata [7]. These packages can be used alongside with pfSense to improve security management in the network.

III. Intrusion Detection and Prevention System

An Intrusion Detection System (IDS) is a device that monitors a network or systems for policy violations or malicious activities. These policy violations are typically reported either to an administrator or collected centrally to an event management system for further analysis. When an IDS focuses on monitoring and analyzing the network traffic, it is called a network IDS (NIDS). An NIDS is a passive system that scans traffic and reports back on threats. On the other hand, an Intrusion Prevention System (IPS) is a network security

and threat prevention technology that examines network traffic flows to detect and prevent vulnerability exploits. Vulnerability exploits usually come in the form of malicious inputs to a target application or service that attackers use to interrupt and gain control of an application or machine. An IPS often sits directly behind the firewall and provides a complementary layer of analysis and security. An IPS is deployed in the direct communication path between source and destination (inline mode), while actively analyzing and taking automated actions on all traffic flows that enter the network segments. The below section details about the Network IDS/IPS called Bro and Suricata.

A. Bro NIDS

Bro [8] is a network security platform with enhanced features than those supported in typical IDS tools. It supports both signature and anomaly-based IDS and its extensible architecture provides the ability to write custom policy analyzers. Bro is based on three-tier layered architecture [9]: Tap, Platform, and Applications. The tap network link sends up a copy of the traffic to the packet processing module which filters down the high-volume stream via standard libpcap [10] packet capture library. The platform layer is composed of two main modules: the *Event Engine* converts the captured traffic to a series of high-level events reflecting underlying networks activities in policy-neutral terms while the *Policy Script Interpreter* executes a set of event handlers written in Bro's custom scripting language. The script can incorporates the policies and context from the past and takes actions (e.g., generate alerts, record to disk, executes response programs, etc.). Bro can be deployed in standalone or cluster mode.

The scripting language being an event-driven, can be used to express arbitrary analysis tasks and customize policies or define actions to be taken given an event. Bro's scripting language facilitates a much broader spectrum of very different approaches to finding malicious activity, including semantic misuse detection, anomaly detection, and behavioral analysis. Bro can act as a dynamic and intelligent firewall when used in conjunction with blocking gateway (e.g., firewall). For example, it can block the access from offending IP addresses, known hostile activity, terminate connections and/or sends alarms, locates site policy violations. Furthermore, the dynamic application detection feature allows port selection rather than specifying which protocol analyzer to use for a given port. In fact, using the NetControl framework, Bro can connect with various network devices and equipments such as switches, firewalls, and routers through their specified API. The NetControl framework provides a flexible, unified interface for active response and hides the complexity of heterogeneous network equipment behind a simple task-oriented API, which is easily usable via Bro scripts [9].

The type of application and use cases, also called Bro Frameworks, to be supported with Bro are very broad, ranging from intrusion detection, vulnerabilities management, file analysis, traffic analysis and measurement, compliance monitoring, etc. Bro comes with many pre-written scripts and analyzers for many protocols that are highly customizable to support traffic analysis for specific environment and needs. Virtually with Bro scripting language, various type of policies can be defined and Bro can interface with other network equipments and applications for real-time exchange of information. For more details, the reader can refer to [9].

B. Snort

Snort [11] is a rule-based NIDS and IPS capable of performing traffic analysis (e.g., protocol analysis, content searching and matching), detecting various attacks and probes, and packet logging in real-time. In fact, Snort combines the benefits of protocol, signature, and anomaly-based inspection methods to perform flexible and efficient protection against security threats. Snort's rule is composed of the rule header and options. The rule header contains the rule's action, protocol, source and destination IP addresses (or subnet), source and destination port(s), and the direction operator (specifies in which way the signature has to match). When a packet matches the rule criteria, the rule's action tells Snort what to do. The rule options form the heart of Snorts intrusion detection engine combining ease of use with power and flexibility. Data Acquisition (DAQ) concept has been introduced in Snort 2.9 to replace the direct calls to libpcap functions.

Snort can operate in *passive* (tap) and *inline* modes. In passive mode, Snort acts as an IDS. However, with the inline mode, Snort acts as an IPS allowing drop rules to trigger. In inline mode, Snort creates a transparent bridge between two network segments, and is responsible for passing traffic between the two segments. Snort inspects the traffic based on the specified rules, then either drop the suspicious traffic or pass it out to the other interface without any modification. For more details about Snort, the reader can refer to [12].

C. Suricata

Suricata [13] is another open source capable of real-time network IDS, IPS and network security monitoring (NSM). It inspects the network traffic using a powerful and extensive rules and signature language, and has a scripting support for detection of complex threats, policy violations and malicious behavior. Suricata can also detect many anomalies in the traffic it inspects. Although it has similar functionality as Snort, its modularity and automatic protocols recognition are the key advantages. For example, it will automatically detect protocols such as HTTP on any port (i.e., it is port agnostic) and apply the proper detection and logging logic. Suricata is based on rule/signature which consist of action, header and rule-options.

IV. Challenges and Issues of IDS/IPS

Although the previously described IDS and IPS present higher advantages and benefits, they still have shortcomings. In fact, they are often static and require manual configuration. For example, upon huge volume of alerts and notifications, a network administrator should manually sort out the issues using visualization or reporting tools to identify the alerts that pose legitimate risks. Nowadays, network are fairly dynamic

and keeping systems up to date with human-in-the loop (HIL) can be challenging and prone to errors. These IDS/IPS usually are unaware of the context and hinder the full potential of network security automation. In fact, without context-aware capability, fast threats assessment and mitigation as well as efficient and reliable automation become challenging. Often, the rules are not automatically updated. Furthermore, relying on external repository that updates and writes rules at a given frequency (e.g., every hour) to combat new and evolving threats might not be efficient to protect against unseen attacks. For example, Emerging Threats [14] is modified daily, Talos [15] is updated weekly or multiple times a week.

With the advances in machine learning, IDS and IPS can be extended with efficient predictive mechanisms for writing rules application to new traffic based on insights from history. The predictive model can help to detect unseen traffic or abnormal behaviors. This falls in the big umbrella of generative policy model [5], since each IDS or IPS device has the ability to generate in real-time new policies and rules based on the context. ECA policies are then created in real-time to adapt to the context and the dynamicity of the networks. This is crucial for automated and reliable policy-based management systems for example in distributed and coalitions networks.

Firewalls are designed to block or accept different types of traffic based on the 5-tuple (source and destination IP addresses, source and destination ports, and protocol) instead of detecting or blocking attacks. Firewall aim is not to analyze intrusion inside a network. Hence, efficient and dynamic defense systems such as IDS/IPS are deployed to detect attacks and improve security management capability. IDS/IPS can catch the attacks that the firewall didn't see or allowed traffic and detect mis-configured firewall.

V. Inference Management Firewall

Recent research activities into firewall design for context-aware information masking led to the initial realization of an *inference management firewall* (IMF) capability [16], [3], [4]. The focus of this work is based around the notion of inferences that can be learned from shared information. Each possible inference is classified and assigned to either a *whitelist* (which represents inferences that are permitted by policy) or a *blacklist* (of those inferences that are not permissible). These lists are then used as factors in determining appropriate access control policies (and mechanisms) over a shareable data set. To complement the theoretical work, a practical implementation was developed that enables inference management controls at various points in the end-to-end flow from the information publisher to the consumer. As part of this practical exploration, an architecture was defined that classifies the IMF into three general components: the first component comprises a network policy enforcement and a decision-making system that operates at the core of the information network; the second component is the end-point policy evaluation and enforcement system that enforces policy on low-capacity mobile devices operating at the edge of the information network (typically at source and sink points); the final component is the communication and associated systems that integrate the different inference management tiers into a logically single firewall.

A demonstrable prototype was developed as part of the study, consisting of the three components described above. This prototype applies access control policy over a publish-subscribe messaging pattern and is based on the ITA Information Fabric [17]. The prototype supports practical exploration of the inference management principle on data traversing a network of participating nodes. At the edges of network are mobile devices that are capable of sensing their environment and publishing sensed data to the network. Inference management is exploited at the edge by utilizing ipShield [2] running on Android devices.

Whilst this prior work builds a foundational basis for further practical exploration, it also assisted with realizing a number of initial challenges. Firstly, the data publisher is required to configure privacy policy in a fashion that enables enforcement as a shared responsibility between network participants. This requires the definition and application of a suitable policy schema. Secondly, bidirectional exchange of control information must be established between participants in the network core and those operating at the network edge. For instance, the edge nodes are required to provide the network core with release policies in accordance to the information owner's privacy preferences. Similarly, participants operating in the network core must provide policy applicable to information consumers at the network edge. The nature of this scheme for policy expression and management, which was beyond the scope of the original research, must operate in a distributed fashion and be robust to changes in operational dynamics, such as when a network node is removed, or when power availability or network capacity is abruptly terminated. A further area of investigation considers the challenges of achieving information security with the goal of mitigating appropriate vulnerabilities.

VI. Discussion over the Different Anomalies in Firewall and Distributed Firewalls Systems

An important aspect of firewalls that should be taken into consideration for designing the next-generation of firewalls systems are the anomalies [18] that are created between rules. These anomalies can be created because of an incorrect ordering or representation of firewall rules or redundancies and conflicts between rules of different firewalls. Some of the most common anomalies are when a packet matches different firewall rules, or when we are in a distributed firewalls environment and for the same packet different firewalls that are on the same path performs different actions. Some of the anomalies of centralized firewall system are due to the bad ordering of the firewall rules. Below, we present some of the firewall systems anomalies, where we denote by r_i, r_j the firewall rules, with $<$ the relation of precedence between them, e.g., $r_i < r_j$ means that r_j has a higher ordering respect to rule r_i, so if the rule ordering for r_i is i and for r_j is j, then j is smaller i.

- Shadowing anomaly: rule r_i is shadowed by rule r_j, when r_j matches all the packets matched by r_i, and because $r_i < r_j$, rule r_i is never applied to these packets, instead r_j applies. The shadowing problem is a crucial anomaly because the shadowed rules never applies, thus a packet that should be blocked is permitted and vice versa.
- Correlation anomaly: rule r_i is correlated with a rule r_j, if they perform different actions and r_i matches some packets where r_j can be applied, and r_j matches some packets where r_i can be applied. These rules can be seen as partially redundant for their spectrum of action but have different actions.
- Generalization anomaly: rule r_i is a generalization of rule r_j, if they have different actions and if r_j is able to match all the packets matched by r_i. In this case, we are dealing with a redundancy, where r_i is included in r_j, but these rules apply different actions.
- Redundancy anomaly: two rules are redundant if they match the same packets and they perform the same actions. In this case, one of them can be removed.
- Irrelevance anomaly: a rule r_i that does not match any traffic is irrelevant. This rule can be removed from the firewall rules.

Nowadays, we often find systems that use different firewalls. In this case, the anomalies created are not only the one of the firewalls themselves, but also what can be created by the use of different ones. It is important for our future work to understand and analyze the created anomalies, especially dealing with coalitions, where every coalition can have their own sets of firewalls or rules, with their appropriate ordering, thus it is common that conflicts and anomalies arise between the various firewalls' rules. Below, we present some of the distributed firewalls systems anomalies, where the firewalls' rules are denote by r_i, r_j, in this case we use $\prec$ to denote that a firewall rule is more close to the destination of the packet than another one, thus $r_i \prec r_j$ means that rule r_j is part of a firewall that is more close to the destination then the firewall of rule r_i.

- Inter-firewall shadowing: when r_i blocks packets that are permitted by r_j, where $r_i \prec r_j$. This anomaly is important as traffic that should arrive to the destination is blocked.
- Spurious traffic: when r_i permits packets that are denied by r_j, where $r_i \prec r_j$. This anomaly is critical as non wanted traffic is getting close to the destination.
- Redundant anomaly: when r_j denies packets that are denied by r_i, where $r_i \prec r_j$. This anomaly effects the efficiency of the firewalls system, as traffic that was already blocked by the firewalls that are more far from the destination is blocked again by firewalls more closer to the destination.
- Correlation anomaly: when r_i and r_j have different actions and part of the packets matched by r_i, are matched by r_j, and vice-versa.

There are different techniques for solving the above anomalies. In [18], the authors capture these anomalies by constructing policies trees. The latter represent the firewall rules, where every node represent a network fields and every branch a possible value associated to that field. The graphical representation offered by the policies trees helps identifying the various anomalies. In [19], the authors introduce a dynamic rule-ordering technique, that uses Internet traffic characteristics, for firewall filtering. Other techniques are introduced for dealing firewall anomalies and their rule ordering. An interesting technique used in [20] is argumentation, where an innovative firewall configuration management is introduced that performs the automatic firewall rules ordering, by avoiding the creation of anomalies.

VII. Challenges for Next Generation Firewalls

Recent IoT-based botnets [21] have shown that many types of the device can be easily compromised and recruited into a botnet. In dynamic environments where devices can move in-out from networks, we cannot certainly exclude the possibility that compromised devices could move into a system externally protected by a firewall. Such devices can then start executing actions, such as sending requests to a target destination as part of distributed denial of service (DDoS) attack. Preventing such malicious use of devices requires that firewalls be able to filter not only the traffic incoming toward the protected system but also the traffic outgoing from the protected system in order to make sure that the traffic is directed towards legitimate destination and according to the specific missions being carried out by the protected system.

An initial approach to build filtering capabilities to prevent IoT devices from being used as bots by a botnet has been recently proposed by Habibi *et al.* [22]. Such a firewall takes advantage of the fact that communication patterns for several categories of IoT devices are quite predictable as these devices have often very specialized functions and usually only communicate with specific applications located at a predefined set of IP addresses. Extensive testing has shown that such a simple approach is effective. However research is needed for developing techniques for profiling devices characterized by more complex communication patterns and correlate such patterns with the input received by the devices and the current context of the devices. An approach along such lines has been developed in the context of data protection from insider threat [23]. The specific approach aims at creating profiles of SQL application programs. Such profiles record the specific SQL queries executed by the application programs based on the input parameters. At run-time, queries issued by each application are matched against the query profile of the application and if there is mismatch the query is flagged as anomalous. Such an approach is quite complex as it uses concolic testing techniques and also the use of a log system to capture application input and SQL queries issued by applications. However perhaps a simpler approach along those lines could be developed for profiling communications of IoT devices.

In the current firewall, the matching criteria is based on execution of regular expression against IP packet headers. Moreover, the network IDS and IPS have capability to extract insights from the monitored traffic. With next-generation firewall, the advances in machine learning can be used to analyze packets and build predictive models to anticipate abnormal behaviors of unseen traffic. Such technology can be designed as plug-in to IDS/IPS that can be used to enhance and update firewall rules dynamically. This would allow it to learn actual context in order to refine the policies for potential unseen attacks. With the increasing adoption of network function virtualization (NFV) and software-defined network (SDN), virtual firewalls or network security devices can be easily deployed and setup throughout distributed network and point-of-entry.

VIII. Conclusion

This paper mainly addresses various existing firewall technologies and their policies. It also highlights the need for next generation firewall giving an indication of the challenges in the existing firewalls. Some of the key areas for future work includes using machine learning, virtualization, autonomic systems, and knowledge base that would help to design these next-generation firewalls for highly dynamic networks such as coalitions environments. Our future work will be around the challenges and issues identified in this position paper.

Acknowledgment

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon. Supported by EPSRC Project CIPART grant no. EP/L022729/1.

References

[1] A. Atlasis, "Attacking IPv6 implementation using fragmentation," *Black-Hat Europe*, pp. 14–16, 2012.

[2] S. Chakraborty, K. R. Raghavan, M. P. Johnson, and M. B. Srivastava, "A framework for context-aware privacy of sensor data on mobile systems," in *Proc. of the 14th Workshop on Mobile Computing Systems and Applications*. ACM, 2013, p. 11.

[3] S. Pipes, S. Chakraborty, and F. Cerutti. (2014) Inference management in the experimentation framework. [Online]. Available: http://nis-ita.org/science-library/paper/doc-2775

[4] S. Pipes, B. Hardill, C. Gibson, M. Srivastava, and C. Bisdikian, "Exploitation of distributed, uncertain and obfuscated information."

[5] E. Bertino, S. Calo, M. Touma, D. Veram, C. Williams, and B. Rivera, "A cognitive policy framework for next-generation distributed federated systems - concepts and research directions," in *Proc. of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017)*, 2017.

[6] pfSense. https://www.pfsense.org/.

[7] https://doc.pfsense.org/index.php/package_list.

[8] V. Paxson, "Bro: a System for Detecting Network Intruders in Real-Time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999.

[9] B. NIDS, "The bro network security monitor, https://www.bro.org/index.html."

[10] Libpcap. http://www.tcpdump.org/.

[11] Snort. https://www.snort.org/.

[12] *SNORT Users Manual*.

[13] Suricata. Suricata: Open source IDS/IPS/NSM engine, https://suricata-ids.org/.

[14] Emerging Threats. https://www.emergingthreats.net/.

[15] Snort Talos. https://www.snort.org/talos.

[16] S. Pipes and S. Chakraborty, "Multitiered inference management architecture for participatory sensing," in *Pervasive Computing and Communications Workshop (PERCOM), 2014 IEEE International Conference on*, 2014, pp. 74–79.

[17] J. Wright, C. Gibson, F. Bergamaschi, K. Marcus, R. Pressley, G. Verma, and G. Whipps, "A dynamic infrastructure for interconnecting disparate ISR/ISTAR assets (the ITA sensor fabric)," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*, 2009, pp. 1393–1400.

[18] E. Al-Shaer, H. H. Hamed, R. Boutaba, and M. Hasan, "Conflict classification and analysis of distributed firewall policies," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 10, pp. 2069–2084, 2005.

[19] H. H. Hamed, A. El-Atawy, and E. Al-Shaer, "Adaptive statistical optimization techniques for firewall packet filtering," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 23-29 April 2006, Barcelona, Catalunya, Spain*, 2006.

[20] A. K. Bandara, A. C. Kakas, E. C. Lupu, and A. Russo, "Using argumentation logic for firewall configuration management," in *Integrated Network Management, IM 2009. 11th IFIP/IEEE International Symposium on Integrated Network Management, Hofstra University, Long Island, NY, USA, June 1-5, 2009*, 2009, pp. 180–187.

[21] E. Bertino and N. Islam, "Botnets and internet of things security," *IEEE Computer.*, vol. 50, no. 2, pp. 76–79, 2017.

[22] J. Habibi, D. Midi, A. Mudgerikar, and E. Bertino, "Mitigating the internet of insecure things," *IEEE Internet of Things Journal*.

[23] L. Bossi, E. Bertino, and S.-R. Hussain, "A system for profiling and monitoring database access patterns by application programs for anomaly detection," *IEEE Trans. Software Eng.*, vol. 43, no. 5, pp. 415–431, 2017.