Using Declarative Networking Rules to Secure Overlay Networks

Maroun Touma, Jorge Lobo, Seraphin Calo
IBM Research
touma, jlobo, scalo@us.ibm.com

Alessendra Russo
Imperial College London
a.russo@imperial.ac.uk

Abstract—We propose the use of a Declarative Networking proxy, called a universal proxy, for securing coalition overlay networks. The primary objective of our approach is to define a framework that provides a high level of flexibility and adaptability to meet stringent security requirements in ad-hoc joint missions.

Keywords—declarative programming; declarative networking; overlay network; universal proxy; access control

I. INTRODUCTION

An overlay network is a set of interconnected objects that rely on an existing network to run high-level protocols for establishing logical links and enabling data exchange between its nodes. In the case of military operations, overlay networks can provide a highly distributed environment where transient links are established to enable the flow of information between the various coalition assets, in the context of a given mission. In this environment, traditional solutions for security and network management tend to break down due to the highly dynamic nature of the network, where assets may join or leave the network at any time.

In what follows we will limit our scope to overlay networks that are built on top of Internet Protocol (IP) networks such as those using Distributed Hash Tables (DHT) [1], the ITA Sensor Fabric [2], the GAIAN database [3] and Peer-to-Peer (P2P) Networks [4]. The control flow of information among nodes in an overlay network follows a well-defined pattern modeled by a state machine. A network proxy can track the state of each asset and, as required by the state, enforce security and access control rules defined for a particular mission. In order to provide the flexibility and extensibility needed to secure a large base of overlay network protocols with diverse mission security requirements, we propose to use a declarative programming framework where states and state evolution are maintained by means of Datalog rules [5], an extension of SQL that supports recursion. A network proxy can then be defined as a programmable entity, called a universal proxy, whereby changes in the protocols and security requirements can be captured by changes in the Datalog rules. This is the first step of a wider research agenda where we aim to develop a declarative networking infrastructure for the design, analysis, implementation and evaluation of integrated security and network management protocols, at different network layers (IP, overlay and service level networks).

II. USE CASE REQUIREMENT AND IMPLEMENTATION

A. Secure Information Dissemination

Ongoing research in the field of Declarative Networking [6] has demonstrated the viability of a declarative approach for the programming and execution of network protocols that take into account link layer constraints and soft-state semantics. It also defines a verifiable framework for constructing overlay networks with proper handling of message delivery, acknowledgements and failure detection.

In a coalition setting, we require an additional layer of security to be implemented as an extension to the existing network protocol in order to provide the soldier in the battlefield with sensitive information (e.g., the latest intelligence on enemy positions gathered by other members of the coalition). We further require that security controls must be programmable and adapted to each mission, taking into account established policies among all coalition members for information dissemination. It is well understood that such policies may vary as the mission progresses toward accomplishing its final objective. Security requirements may, for example, require a preferred routing scheme that is meant to obfuscate the identity of the sender or receiver, based on a number of parameters, including: the recipient affiliation and rank; the volume, quality and timeliness of the information; and, the capabilities of the deployed asset used for decryption.

B. Universal Proxy

The universal proxy is a programmable entity that is deployed at various access points in the network and encodes all security controls for specific segments of the network. This preferred configuration provides for a clear separation between the application logic, implemented by the various nodes in the overlay network, from the security enforcement logic that is built into the proxy. In a hybrid network, the preferred location
of the proxy could be set at every wireless access point which guarantees that all ingress and egress traffic follows the stated security constraints for the mission. Proxy location can be driven by mission requirements and the network structure.

By analyzing the traffic flow between various nodes of the overlay network, the universal proxy tracks the state of each end of a connection using a state machine and enforces specific security rules associated with each state. In our first implementation of the proxy, we consider security rules as they apply to five security dimensions, including: who can see what information, who can provide what information, what information can be shared, what information can be trusted, and how the information is made secure. In addition, the universal proxy can provide traceability to who has what information, and how and when it was obtained.

### III. STATE MACHINE EXECUTION MODEL

Each node in an overlay network follows a well-defined state machine schema starting with an IDLE state. As it discovers other nodes in the network, it transitions to an AWARE state. After sending a search query to other nodes in the network, it may transition to an INFORMED state once it has acquired knowledge of the resource(s) available in one or more nodes in the network. At this point, it may initiate a transfer where it is sending or receiving blocks of data. Once the transfer is complete, the node is expected to revert back to the AWARE state or disconnect from the network. A STALE state is defined to force the node to refresh its knowledge of what resources are available in the network on a regular basis.

![Figure 2. Overlay Network State Machine](image)

This high level state machine provides the underpinning for programming generic security rules into the proxy. The proxy, by monitoring the traffic, only needs to identify the transition between these states and enforce the appropriate policy as shown in the following (simplified) examples.

#### A. Node Authorization Rule

This rule augments an existing protocol with checks to determine that both the source and destination are authorized to participate in a secure overlay network. The main trigger for the rule evaluation is the insertion of the “announce” event when the proxy detects that the source node is attempting to join the overlay network. If any of the rule conditions fail, the traffic between source and destination is dropped (i.e., the aware state for the source is not reached by the destination).

The declarative rule below defines a join between three distributed queries and uses the result set to confirm the transition from the IDLE to the AWARE state.

\[
\text{aware@DstIp(SessionId, SrcIp, DstIp, DstPort, Proto)}::\text{announce@SrcIp(SessionId, SrcIp, DstIp, DstPort, Proto),}
\]

\[
\text{authorize@proxy(SrcIp, SrcPort, Proto),}
\]

\[
\text{authorize@proxy(DstIp, DstPort, Proto),}
\]

\[
\text{authenticate@SrcIp(SrcIp, User)}
\]

Figure 3. Overlay Network State Machine

#### B. User Authentication Rule

The rule above can be further enhanced to add user authentication. Given the distributed nature of the network, this rule assumes that the relevant query will be executed at the source node. In cases where the overlay network protocol does not provide native support for user authentication, an out-of-band remote procedure call can be used to complete the user authentication.

\[
\text{aware@DstIp(SessionId, SrcIp, DstIp, DstPort, Proto)}::\text{announce@SrcIp(SessionId, SrcIp, DstIp, SrcPort, DstPort),}
\]

\[
\text{authorize@proxy(SrcIp, SrcPort, Proto),}
\]

\[
\text{authorize@proxy(DstIp, SrcPort, Proto),}
\]

\[
\text{authenticate@SrcIp(SrcIp, User)}
\]

Figure 4. State Machine With Authenticate Rule

### IV. FINAL REMARKS

We are currently working on the detailed design and coding of the proxy to demonstrate its specific uses and capabilities. We are planning to work with at least two overlay network protocols. We will consider first proxies that act in isolation, and then experiment with security policies that require the coordination of several proxies, so exploiting the declarative networking capability of maintaining distributed state information.

### REFERENCES


