Beyond centralism: The Herold Approach to Sensor Networks and Early Warning Systems - Extended abstract -

Axel Theilmann <theilmann@pre-sense.de>
PRESENSE Technologies GmbH

January 24, 2010

Abstract

Current sensor networks for Early Warning Systems (EWS) have a simple monolithic structure where data is acquired at the network edges and then transmitted over a "dumb" infrastructure to the Early Warning Center (EWC) for analysis. This structure has proven inadequate for cross-organisational EWS and inter-EWS communication. The recently started Herold research project suggests a distributed architecture for network security systems based on independent network agents. While Herold is still in an early stage and much larger in scope, its central ideas can already be applied to alleviate the problems of monolithic EWS architectures.

1 The monolithic approach to sensor networks

The goal of an Early Warning System (EWS) is to collect all security relevant information about a network, process it and present it to its users in near realtime. Sensor networks are the main technique employed for this goal: A large number of sensor systems are deployed throughout the target network. Each in a place where network usage is somehow deemed relevant and the set of all sensors together considered "representative" for the network as a whole, much like TV ratings for a country are deduced from the TV habits of certain representative households.

Each sensor transmits the collected data to a central node, the Early Warning Center (EWC), where the data is processed and analysed. The transmission may either happen directly or through a concentrator node. A concentrator node collects the data stream of many sensors into one stream, possibly pre-processing or compressing it to reduce network load, and then forwards it to the EWC. The resulting structure is a three-level hierarchy where data collection is done at the edges and the data only flows inwards to a central point of processing. Since the sensor network's

main purpose is to forward the data to a central point of analysis, each component can be kept very simple and does not need to store any data. Everything is forwarded immediately and pre-processing is only as needed for practical purposes during transport. The types and interconnections of these data transmitting nodes make up the *data flow architecture* of the EWS. Since data flows across multiple interconnected nodes before reaching the EWC, it is a *distributed* data flow architecture.

In the described scenario, a distributed architecture is employed purely for *technical* reasons. It exists because the network data of interest is scattered across the network. If there was a place in the network where all relevant data could be observed at once, the distributed architecture would not be needed.

This EWS structure of a "dumb" sensor network and an "intelligent" center will be called the *monolithic approach*.

2 Deficiencies of the monolithic approach

2.1 Dealing with organisationally diverse structures

When trying to deploy a large sensor network in practice (i.e. beyond the limits of a research testbed), one severe limitation of the monolithic approach quickly becomes clear: It is not well suited to deal with an environment that requires *administrative distribution* and *cooperation*, i.e. where the network nodes and collected data are under different administrative control. But for an EWS to be able to truly capture a representative image of the current situation on the Internet, data from different kinds of networks (open Internet, darknet, honeypot network, DMZ) and different kinds of network operators (small companies, large companies, ISPs, housing providers, universities) has to be collected and processed.

The networks and network operators have

- different goals and motivations for network analysis,
- different privacy and data protection requirements,
- different data rates,
- different storage requirements for recording a history of network traffic, and
- different requirements about who to grant data access to (and also to revoke it again).

For an EWS to be successful in practice, its data flow architecture and overall operational model have to at least satisfy each network operator's basic requirements in the above areas.

Additionally, since operating an EWS node requires contribution of resources, there further is a question of motivating network operators to participate in an EWS. Agreeing to operate an EWS node must result in some kind of benefit for its operator.

While contributing data to an EWS for the "greater good" of an overall network view might motivate some of the network operators, practice shows that this is usually not enough. In the monolithic architecture described above, a network node (sensor or concentrator) is mainly a black box that "sucks up" network traffic and transmits it to the remote EWC without the network operator having much control or further access to it. In a scenario, where no administrational borders are crossed, this is not a problem since the network operator can be granted full access to the EWC itself and access all its visualisations and analyses. But in large-scale EWS, the network node operators can usually not be given an attractive level of access.

While pseudonymisation or anonymisation of addresses might make it possible to give operators a certain level of access, they require "dumbing down" the data and so present a trade-off between suitability for analysis and privacy requirements.

Another option would be to add multitenancy-capabilities to the EWC. Each operator would only see the data from his or her nodes when accessing the EWC. But multitenancy would either require the EWC to replicate all processing steps for each possible tenant or to process data on-the-fly for each tennants access request. Both approaches would result in a drastic increase in CPU-workload at the EWC and the first approach would additionally increase storage requirements.

Even when access can be granted, the different views of network operators on the data comes into play. Not surprisingly, network operators are much more interested in what is happening in their own network than what is happening on the Internet in general. They are interested in sudden changes or even single events on their own network, regardless of whether other node operators see them, too. They tend to see a sensor network much more as a kind of distributed IDS, rather than an EWS. EWS operators on the other hand want to detect event-types and trends that span the whole observed network. While functions taking a node-local view and accessing single events are usually available, they are not the main focus and not automated to a useful level.

Summing up, for an EWS (or any cross-domain sensor network) to be successful in practice, it must be able to not only satisfy network operators individual requirements for data collection, processing and storage but also provide them with an attractive benefit in return for operating the network node. One such benefit is the provision of a node-specific view on the data that focusses on individual events and network characteristics rather than overall features of the sensor network. The monolithic architecture is not well-suited for this requirement.

2.2 EWS cooperation and automated analysis: Getting data back-out again

Another requirement that sooner or later shows up both from a research and from a daily-use perspective is the need to get data back out of the EWS for (possibly automated) third-party analysis and to facilitate inter-

EWS data-exchange.

As described above, in the monolithic approach data only flows from the edges of the network to the center where it gets processed and stored. Because of this hierarchical structure, the initial design of a sensor network tends to omit flexible, network-accessible interfaces to export the saved data (both raw and processed). Export interfaces get added as an afterthought in conceptually non-clean ways and for specific, rather than generic, needs, possibly introducing new data formats or access protocols. Creating export interfaces either requires a lot of internal restructuring and implementation work or leaves the interfaces unflexible and not very useful, possibly requiring even more different interfaces when the next export requirement comes along.

Because of its importance for future EWS, the issue of inter-EWS cooperation will be discussed in more detail. Maturing and growing out of their testbeds, EWS systems are proposed on different organisational levels. A large company or other network operator might consider running its own EWS and larger EWS are envisioned on national level or even international levels (e.g. by the European Union). While it might be possible to run a national and an EU-level EWS in parallel, this would require (and waste) a lot of resources. A more attractive and efficient option would be to create a federation of national EWS and have them cooperate, forming a "virtual EWS" that stores little or no original data but provides EWS information via access to the national EWS that it is based on. For this kind of cooperative approach, a much more flexible interface is needed in an EWS. Exporting a data feed in real-time is useful, but requires the importer to again store and process the data to use it. Also, once data is exported, an EWS operator loses direct control over it. Access rights cannot easily be changed or revoked and data is more difficult to delete. While the creation and operation of such federated EWS is a research topic in itself, it should be clear that again the monolithic approach is not well-suited to deal with a structure like this.

A new model is required that honors and actively supports node or EWS operators in the fulfilment of their diverse requirements for running an EWS or other sensor network. Such a model for truly distibuted network security will be suggested by the Herold research project and described in the next section.

3 The distributed Herold approach, applied to Early Warning Systems

3.1 The Herold project

Herold is a research project on distributed network security. It is a collaboration between PRESENSE Technologies GmbH, the Group "Theoretical Foundations of Computer Science" of the University of Hamburg, and n@work Internet Informationssysteme GmbH. It was started in mid-2009 as a 2-year project, partially funded by the German Fed-

eral Ministry of Education and Research (BMBF) as part of the "KMU-Innovationsoffensive Informations- und Kommunikationstechnologie".

The long term goal that is explored is the interpretation of network monitoring and policy enforcement as a collective group effort by cooperating but independent "network agents". Each agent maintains its own knowledge base about the network structure, the currently operated network policy, and the high-level security goals and best-practices. Also, each agent is equipped with technical means, so called "network security components", to enforce a network policy (e.g. by Firewalls) as well as monitor it (e.g. by IDS or Netflow probes). Agents use these security components to pursue individual security goals as well as goals that they have agreed upon with other agents. Because each agent has access to different security components, an agent may ask (or force) another agent to enforce part of its policy, thereby fulfilling security goals or monitor a specific part of the network and provide resulting event information. This results in a continuous exchange of information and services between agents.

The Herold project will first create detailed models consisting of agents' roles, actions, interactions and workflows and model these formally using the PAOSE (Petri net-based Agent-Oriented Software Engineering) techniques developed at the University of Hamburg. After that, central questions will be the modelling of global network-, policy- and event-spaces, their agent- and security-component-local meaning and the formal and practical relationships between those levels. In the second half of the project, a first implementation of the new approach will be created in the form of a network policy enforcement framework. This framework will allow the efficient and secure enforcement of a global network policy in a local context and with the security components available to a given agent.

Even though, the Herold project is still in its early stages, and the model is not yet formalised, the central ideas can already be applied to provide a new architecture for EWS and tackle some of the problems described above. At the same time, this will hopefully shed some more concrete light on the rather abstract concepts of the Herold approach.

3.2 Early Warning Systems from a network agent point of view

In the distributed architecture suggested by Herold, each EWS is considered an independent network agent. Each network node in a given EWS can either be modelled as a security component or a network agent itself. Simple sensors might adequately be modelled as security components, while concentrator nodes that collect data from all sensors in a given administrative domain would be considered network agents. A rule of thumb would be that every time the EWS structure crosses an organisational boundary, the connection should be modelled as a connection between agents, not between agent and security component.

Each network agent then functions according to an operational model similar to the one that was used for the central EWC node at the mono-

lithic approach.

Each agent individually

- maintains a structural and attribute description of the network it is "interested in" or responsible for,
- maintains a description of the network parts that it can directly (via security components) or indirectly (via subordinate or cooperating agents) control or monitor,
- maintains a network policy in the form of algorithms that evaluate a network status according to early warning principles (or similar network monitoring principles),
- analyses and evaluates incoming network events according to the policy and provides results and visualisations in near realtime,
- records network events as a history for further reference and analysis, and
- offers an "agent interface" that other agents can use to request data or services.

The agent interface is used to structure the agents into a topology. This topology can be strictly hierarchical as in the monolithic approach, or it can be less strictly structured, allowing agents to exchange data and services on a more even level.

3.3 Hierarchical network agents

A hierarchical structure consists of multiple network agents, connected as a tree like the network nodes in the monolithic approach. Each agent has (at most) one superior and a number of subordinate agents. Each agent, together with its corresponding subtree of subordinate agents, behaves like, and possibly is, a separate EWS or network monitoring system. This kind of recursive structuring can either be used to allow lower-level network agents to generate and maintain an EWS-like view of their own network or it can be used to connect existing EWS into larger structures. Many company EWS can be connected into a national EWS and a number of national EWS could be connected into an international EWS.

In this hierarchical structure, the agent model can solve or at least alleviate many of the problems or requirements described in section 2.1. Because of the individual processing and storage of sensor data, individual algorithms and visualisations can be used to monitor, analyse and evaluate the network situation. Network operators can use their network node to monitor their own network according to their own principles and motivations while controlling, what gets saved and what gets transmitted further up the tree.

If an agent does not want to directly transmit a certain part of the data because of access restrictions or network load, it can either keep it to itself or just advertise to the higher-level agents, that the data is available on explicit request. An agent that needs advertised data in raw or processed form can send a request down the tree. When the request reaches the node

that has (at least part of) the data, it can decide whether the requested access is permitted. If yes, the data gets passed back up the tree, possibly after some processing.

For example, if the root agent wants to determine the number of times that a given source IP adress was seen across the network in a given time interval, it would send a request "*give the frequency for source IP X in time interval Y*" (using some formal query language, of course) down to each of its subordinate agents. Each receiving agent decides whether its local authorisation settings allow that request. If yes, the event data about the requested IP address is retrieved from storage, and the frequency is determined. Then, the request is sent further down the tree with each agent reacting similary. Once the answer comes back, it is combined with the local data and sent up the tree. All answers are received by the root agent, combined and used as planned. Of course, not only the root agent but any agent in the tree can send such requests down its available subtree to determine data of interest.

Transmitting queries and evaluating them in the tree is more efficient than sending all data to the root agent and do all processing there. The sensor network behaves like a distributed database evaluating queries on the collective set of event data.

3.4 Network agents in non-hierarchical structures

While EWS have traditionally been structured hierarchically, this is not a requirement for the network agent model. The hierarchical structure of the monolithic approach mainly stems from the technical goal of easily and quickly transmitting all data to the center for processing. "Up" was, where the data was sent. But since a network agent can transmit its data to any interested agent, the hierarchical structure can easily be discarded and replaced with a model where agent connections are arbitrarily structured. Each agent can request data from other agents and offer data to others, either as a continuous data stream of realtime event data or as the evaluation of specific requests on stored data. Of course, the same interface can also be used by automated analysis tools or to export data out of an EWS as it was described in Section 2.2.

Together with agent-internal data storage and processing, a general data access interface can be used to form many different kinds of useful structures and service relationships, both static and dynamic. Network agents can be connected together to form a virtual EWS (see Section 2.2) for a designated purpose, allowing an agent to contribute its data to multiple EWS in different contexts or incorporate data from multiple special EWS or other sensor networks in its own analyses.

The Herold project will create a software infrastructure to create and run network agents and have them exchange information and services for a truly cooperative approach to network monitoring. Especially non-hierarchical interaction of network agents will hopefully provide a large and interesting field for research experimentation and use in everyday practical network security. Further results of the Herold project will be published as the project continues.