DAI-Labor
TU Berlin

CIMD– Collaborative Intrusion and Malware Detection

Technical Report: TUB-DAI 08/08-01

Rainer Bye, Sahin Albayrak

August 11, 2008

DAI-Labor der Technischen Universität Berlin
Prof. Dr.-Ing. habil. Sahin Albayrak
Technische Universität Berlin / DAI-Labor
Institut für Wirtschaftsinformatik und Quantitative Methoden
Fachgebiet Agententechnologien in betrieblichen Anwendungen und der Telekommunikation
Sekretariat TEL 14
Ernst-Reuter-Platz 7
10587 Berlin
Telefon: (030) 314 74000
Telefax: (030) 314 74003
E-mail: Sekretariat@dai-labor.de
WWW: http://www.dai-labor.de

Abstract

We present a cooperation scheme for distributed intrusion detection taking into account security-related properties of each individual participating node. This leads to a security overlay network named CIMD (Collaborative Intrusion and Malware Detection[1]) enabling participants to state objectives for cooperation and find groups for the exchange of security-related data, like monitoring or detection results, accordingly; to these groups we refer as detection groups. Our contribution is twofold: First we present and discuss a tree-oriented taxonomy for the representation of nodes within the cooperation model. Second, we introduce and evaluate an algorithm for the formation of the detection groups. These two concepts create the core of an overlay architecture dedicated to intrusion detection and response measures and show the impact of CIMD by providing two different scenarios where the collaboration is advantageous compared to the non-collaborative approach. We evaluate the benefit of CIMD in a novel packet-level simulation environment called NeSSi[2], the Network Security Simulator. Furthermore, we analyze the vulnerabilities of the system itself and possible attack scenarios against it.

[1]This work was funded by Deutsche Telekom AG.

[2]NeSSi, the Network Security Simulator is part of the project Malware Filtering funded by by Deutsche Telekom AG.

Contents

1 Introduction

Teamwork– nowadays professional life as well as private life is hardly imaginable without teamwork. Above all, complex tasks are usually managed in teams. Ideally, each participant of a team can contribute in the area of his strengths. However, teams can also be homogeneous; dependent on the task a team is to fulfill, a heterogeneous set-up might not be necessary or may even be destructive due to arising conflicts.

Intrusion detection is indisputably a complex task and there is no silver bullet coping with threats arising from malicious software or attackers. According to the 2007 Symantec Internet Security Threat Report security landscape was characterized by an "Increased professionalization and commercialization of malicious activities" [Symantec, 2007].

Computer networks are exposed to a variety of threats: Zero-day attacks leave devices connected to the Internet susceptible to attacks vulnerable because there are no appropriate signatures available during the vulnerability window. On the other hand, purely anomaly-based detection schemes capable of detecting new attacks are often of limited use due to a high false-positive rate.

Due to the shortcomings of conventional intrusion detection approaches we propose CIMD (Collaborative Intrusion & Malware Detection), a scheme for the realization of cooperative intrusion detection approaches. We argue that teams respectively groups with a common purpose for intrusion detection and prevention improve the measures against malware. By enabling participants to state their *objectives*, i.e. the aim of a detection group, and *interests* i.e. the desired properties of the team members, an intrusion detection overlay is realized. CIMD is collaborative, because for a common task, groups can be dynamically built in a heterarchical manner without pre-defined roles. In contrast, after the grouping takes place, cooperative approaches can be carried out, i.e. tasks are divided between group members and roles assigned. Nevertheless, the actual approaches after grouping can also be collaborative dependent on the used algorithm. CIMD is a part of ongoing research in the context of research activities aiming to develop automated intrusion detection and response techniques.

This work contributes a taxonomy-based data model reflecting relevant properties of the participants of the overlay. We discuss each category in the taxonomy with regard to the impact on detection groups and their value for collaborative intrusion detection. Additionally, we also provide a group formation algorithm to establish these groups. Each participating node executes this algorithm that receives as input objectives and associated interests defined as instances of the property taxonomy. Moreover, it takes maximum group sizes into account. Realization strategies for the system itself are also discussed.

Furthermore, we introduce the notion of *homogeneous* as well as *heterogeneous detection groups* analogous to the introductory example of teamwork in sociology. We consider a distributed anomaly detection approach as a scenario for homogeneous groups and discuss device similarity as a requirement for that. In the second scenario, we apply a signature mediation scheme wherein disparate *NIDS* (Network Intrusion Detection Systems) collaborate to reduce the aforementioned vulnerability window. This is an example for a heterogeneous detection group enabling exchange of signatures between the devices. We conduct simulations for the latter scenario in a novel network

simulation environment addressing the needs of security experts: NeSSi. Nevertheless, a distributed scheme like CIMD exhibits the danger of being compromised. Hence, we discuss security aspects of the system itself, provide adversary scenarios and discuss appropriate counter measures.

1.1 Organization

This paper is organized as follows: subsequently, we introduce related work, present the CIMD approach in Section 2 and offer realization alternatives in Section 3. Next, in Section 4 we conduct a vulnerability analysis of CIMD and discuss in Section 5 the merits of an intrusion detection overlay based on the aforementioned scenarios. The simulation of the "signature mediation" scenario as an example for a collaborative approach based on heterogeneous groups takes place in Section 6. Finally, we conclude in Section 7.

1.2 Related Work

We consider a scheme for collaborative intrusion detection based on group formation. Hence, we provide an overview of existing work in the area of group formation in overlay networks, collaborative intrusion detection and existing (inter operable) intrusion detection message exchange formats.

1.2.1 Semantic Group Formation in Overlay Networks

Semantic Group formation in overlay networks is not a new topic. Khambatti introduced the notion of interest-based communities in peer-to-peer networks [Khambatti et al., 2004] to reduce the communication overhead of search operations. These communities are based on common attributes. The author distinguishes between group attributes like a domain name and personal claimed attributes. Bloom filter data structures are used to represent those properties due to their efficiency in determining inclusion relations.

Loeser *et al.* have introduced the concept of semantic overlay clusters (SOC) [Loeser et al., 2004]. They use a hierarchical peer-to-peer system based on JXTA[3], where the Super Nodes, dedicated nodes within such a peer-to-peer system, realize the clustering using a pre-defined policy. Participating peers in this network match their own properties by an Information Provider Model against the policy of the Super Node. In the case of a match, the peer is added to the group administrated by the Super Peer, whereas peers can join several groups. The enhancement of search operations also motivates work.

Sripanidkulchai *et al.* have proposed interest-based shortcuts. This is an approach introducing the notion of interest-based locality, a principle expressing that if one peer has a piece of content another peer is interested in, it is very likely that the first peer has also other pieces of content that the peer is interested in [Sripanidkulchai et al., 2003]. These shortcuts are applied to a pure peer-to-peer system such as Gnutella. Here, in addition to the neighbor entries, these shortcuts are used. The purpose is to increase the performance and the scalability by providing an improved searching scheme.

[3] https://jxta.dev.java.net

The paradigm of structured peer-to-peer networks offers new opportunities for research. Thus, the application of DHTs (*Distributed Hash Tables*) enables exact mappings from resource names to peers, enabling fast and deterministic look-up operation. In this regard, Castro *et al.* realized an application-level multicast infrastructure on top of the DHT-based Pastry framework[4]. Here, participating nodes can register to a subject administrated within an overlay and in the case of notifications these are distributed to all registrants.

In summary, the related work shows that sample solutions for the grouping itself as well as approaches for the semantic clustering exist for different types of peer-to-peer networks (unstructured, structured, Supernode-based). The overall CIMD system is affected, as existing solutions can be taken and enhanced for the purpose of intrusion detection and response. In this regard, further comments on implementation challenges are discussed in Section 3, whereas related work in the context of collaborative intrusion detection is discussed in the next section.

1.2.2 Collaborative Intrusion Detection

There has been some work on collaborative intrusion detection based on overlay networks. The DOMINO system uses an overlay architecture of axis nodes exchanging intrusion-related information like black lists of IP addresses [Yegneswaran et al., 2004]. Each axis node forms the root of a hierarchy of distributed intrusion detection systems. In a retrospective analysis of the SQL-Slammer worm, the DOMINO system would have performed well for the purpose of early detection and prevention of this threat. This evaluation is based on the DSHIELD[5] data. For authentication, Yegneswaran *et al.* deem PKI mechanisms suitable for DOMINO, because the axis node overlay does not grow linear as a function of the aggregate number of nodes in the DOMINO system. No further explanations about the used peer-to- peer architecture are given and there is no cooperation scheme except the grouping of axis nodes exchanging blacklists.

Indra is a peer-to-peer system, where participants of the overlay can exchange intrusion information between each other in a decentralized manner [Janakiraman et al., 2003]. Indra proposes to use the multi cast mechanism presented in [Castro et al., 2002] to form interest-based groups with security-related topics like failed log-in attempts. The authors neither provided a scheme, how security-related topics can be organized, nor show simulation results about the benefits of that system. In the prototypical Indra version, central key servers are used for authentication. In the authorï¿½s opinion, the Web of Trust- approach is better suited for a decentralized peer-to-peer system.

Zhang *et al.* present a conceptual architecture for IDS agents on mobile devices in the context of mobile wireless networks [Zhang et al., 2003]. Such an agent also contains a module for cooperative detection that is able to interact with neighboring IDS agents and a *global response* module. The authors describe a basic majority-based, distributed intrusion detection algorithm based on exchanged anomaly status and apply a fixed scheme to detect abnormal routing table updates. Compared to CIMD, this approach follows a fixed objective and individual properties of the devices are not taken into account.

[4] http://research.microsoft.com/ antr/Pastry/
[5] http://www.dshield.org

Following the metaphor of the Biological Immune System, cooperative *AIS* (Artificial Immune System) was presented by Luther *et al.* [Luther et al., 2007]. Here, an AIS component computes the probability of an anomaly on each participating node. The data processed by the AIS is statistical in nature, e.g. traffic measurements, and obtained by a monitoring component. The probability of an anomaly constitutes the status of a client and the collaboration between the participators takes place by sharing this status levels. The cooperative aspect is realized via a hybrid, decentralized peer-to-peer system enabling the formation of a detection group and is prior work to CIMD. As a result, the false positive rate, one of the main challenges in anomaly detection, was lowered significantly in comparison to the non-cooperative scenario.

The presented collaborative schemes for intrusion detection differ from the contributions of CIMD, as they mostly aim for specialized scenarios. Indra follows a similar direction like CIMD, as the authors consider SCRIBE groups for Security related topics. But here, neither properties of the participating nodes are taken into account nor is there an evaluation showing the benefit of the approach. CIMD even makes one step beyond: it aims to offer a generic scheme to enable a collaborative approach even for distinct IDS to exchange data. For this purpose, a common data format is needed.

1.2.3 Common Exchange Formats

Because of the huge variety of IDS, there were several attempts to standardize exchange formats and communication frameworks to enable interaction between distinct IDS. The first effort was the CIDF (Common Intrusion Detection Framework)[6] funded by DARPA with the objective to enable different research projects (initially only DARPA projects) to exchange security related information.

Thus, the first outcome was the specification of the framework itself, wherein roles of the participating entities were defined; the different roles are *event generator*, *event analyzer*, *event database* and *response unit*. Second, the CISL (Common Intrusion Specification Language) was introduced basing on a prefix-based, recursive notation. This language enabled the exchange of *GIDOs* (*Generalized Intrusion Detection Objects*) that are either generated or consumed dependent on the aforementioned roles. The validation of CIDF in terms of (semantic) interoperability took place by tests in the years 1998 and 1999. Although CIDF not became a standard, it resulted in the creation of the IDWG (*Intrusion Detection Working Group*). Here, the IDMEF (*Intrusion Detection Message Exchange Format*) was developed that became experimental RFC[7].

The main intention of the IDMEF is to provide a communication standard enabling different intrusion detection analyzers from different origin (commercial, open source and research systems) to report to a managing entity (“console”) in one administrative domain. The language is XML-based and there exist two types of messages: first, there is the *Heartbeat* message sent periodically to state a component in the distributed system is still alive. Next, there is an *Alert* message sent in the case a suspicious even occurs. Those events can be associated with additional information in form of XML compound classes like the scanner type, timestamps and classifications in the case of an

[6] http://gost.isi.edu/cidf/

[7] RFC http://rfc.net/rfc4765.html, published in March 2007

alert or even self-defined attributes. Beside the language itself, there exist an experimental RFC for *IDXP* (*Intrusion Detection Exchange Protocol*) [8] providing asynchronous communication between sensors and analyzers based on BEEP, an application protocol framework[9]. By choosing an appropriate BEEP *profile* mutual authentication as well as integrity and confidentiality of the communication channels is offered.

In the wild, there exist IDMEF implementations for sensors, e.g. Snort[10], as well as for analyzers, e.g. Prelude IDS[11], with an IDMEF communication interface. The extensibility of IDMEF is given by two different approaches: one the one hand, the whole data model can be changed by inheriting existent classes, on the other hand an `AdditionalData` class enables incorporation of primitive data types as well as whole XML-Documents. But the `AdditionalData` class is only associated directly with the message class, i.e. other classes in the IDMEF data model are not extensible by it.

In contrast, the IODEF (*Incident Object Description Exchange Format*), also an XML-based format, provides a more comprehensive extension mechanism. It is an RFC draft standard [12].The main scenario for using IODEF is the exchange of incident reports between different CSERT (*Computer Security Emergency Response Teams*) in different administrative domains. To fulfill this role, there exist one type of message class, the incident message. This message must contain a global unique identifier for the sender, an assessment of the incident as well as contact information of the involved parties. Supplementary optional data, e.g. time of detection, start or end time can also be added. For the sake of interoperability, IOEDF offers additional extension strategies, because the XML schema must not be changed. First, each sub class in the IODEF is associated with the `AdditionalData` class. Second, there is a generic mechanism to add to the enumerated values of attributes; e.g. in the contact class the attribute `type` contains the values “person” or “organization” but can be extended by using an `ext-value` to integrate a type “department”. The IOEDF is compatible to IDMEF in the sense, IDMEF messages can be encapsulated and there exist common classes IOEDF uses from the IDMEF, e.g. `Impact` class or `Confidence` class.

The CIDSS (*Common Intrusion Detection Signatures Standard*)[13] defines a common, XML-based data format to share signatures. In this way, it primarily aims at IDS administrators to exchange, evaluate and criticize about signatures. Second, a future scenario is considered in which there exist independent contributors enabling the provision of signatures independent of a special product or software. Each signature message is in general divided into two parts: in the first part, possible data elements of a signature are put like source/destination addresses, protocol types or byte patterns. Second, in the `Session` class a stateful signature can be defined by using the aforementioned data and logical expressions. Nevertheless, stateless signatures can also be realized by skipping attributes of the `Session` class. This approach fits very good to the mediation scenario in Section 5.2, but the IETF-draft has not been completed and expired in November

[8] http://rfc.net/rfc4767.html, published in March 2007
[9] http://rfc.net/rfc3080.html, published in March 2001
[10] http://www.snort.org/
[11] www.prelude-ids.com
[12] http://rfc.net/rfc5070.html, published in December 2007
[13] http://xml.coverpages.org/appSecurity.html#cidss

2006.

There exist a variety of other formats that are either proprietary or have very specialized objective: the CVE (*Common Vulnerability and Exposure*)[14] represents a dictionary to name security vulnerabilities uniquely. This goal is achieved by central data base coordinated by a consortium of representatives from industry, academia and government agencies, the *CVE Editorial Board*. This widely used industry standard offers an opportunity in the case of e.g. IODEF to relate to the same vulnerability from different CSERTs. The TIDP (*Threat Information Distribution Protocol*) is a proprietary protocol from Cisco to enable static grouping among the supporting products including authentication. On top, TIMs (*Threat Information Messages*) are distributed to specify suspicious traffic characteristics and associate *Mitigation Enforcement Actions*, i.e. to block or redirect the selected traffic.

	CIDF	IDMEF	IOEDF	CIDSS	TIM
Inter-domain applicability	Is discussed	Not a focus, but possible	Good	Not a focus, but possible	Not a focus, but possible
Standard	No	RFC experimental	RFC draft	Expired IETF draft	Proprietary protocol
Still in use	No	Yes	Yes	No	Yes
Extensibility with respect to compatibility	Limited	Limited	Good	Unclear	Unclear

Table 1: Evaluation of Exchange Formats with respect to key features valuable for CIMD

With respect to the CIMD scenario, the IOEDF has the advantage over the IDMEF to have more extension opportunities, without changing the entire XML-schema. Otherwise, a change of the schema would lead into interoperability. A second advantage is, that there exists an identifier for the sender in the message itself to associate it to an organization in a cross-domain scenario. In contrast, the usage scenario of IOEDF does not fit directly to the CIDM approach. Primarily, it focus on the exchange of incident information between CSERT with mandatory attributes about involved parties in terms of organizations and personnel which is not in the scope of CIMD. The CIDSS is a specialized approach focusing on the signature exchange scenario. Nevertheless in that context the approach shows his merits (see Section 5.2). Supplementary, CVE can be used to reference vulnerabilities uniquely even from different organizations, but is not a common exchange format. The results of the exchange format analysis are depicted in Table 1. The realization of a common message exchange format in CIMD is further discussed in Section 3. Next, we introduce the overall CIMD approach.

2 CIMD Approach

CIMD offers a scheme for the formation of detection groups based on an overlay network. We introduce the cooperation model considered in Section 2.1 and the decentral-

[14] http://cve.mitre.org

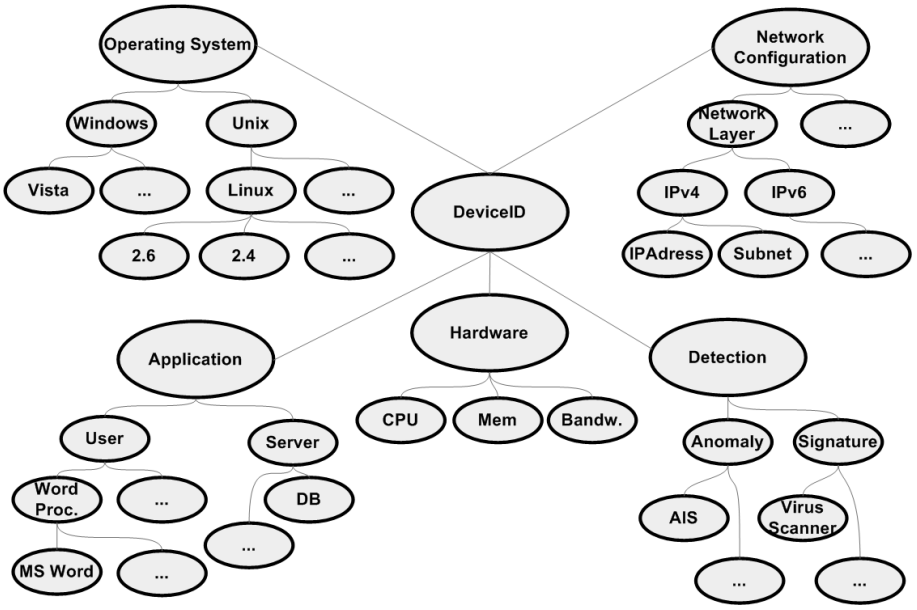

Figure 1: The example taxonomy utilized for cooperation

ized group formation algorithm in Section 2.2.

2.1 Cooperation Model

Every node in an overlay network should be able to express its interest regarding collaboration partners. In the CIMD architecture, those interests are expressed using taxonomy of properties. They are used for the specification of potential collaboration partners in the look-up phase, but also for the description of the nodes itself. The cooperation model is depicted in Figure 1.

The model is based on five main categories *OS*, *Applications*, *Network Configuration*, *Detection* and *Hardware* so far, but remains extensible also to new categories respectively is extensible in the categories itself. The first two categories are important because a lot of attacks only affect a specific OS or a particular application. Additionally, for the determination of similarity between devices, these attributes are essential (c.f. Section 5.1).

The category Operating System is modeled separately as it does not fit to the User/Server subdivision of the application branch. Each of those two categories has several sub hierarchies wherein a number of applications respectively the operating system can be explicitly specified. In the example, the OS Linux is specified with the kernel version *2.4* or *2.6*. Considering applications, for example the Microsoft *Word* program could be extended to specific versions like *2003* or *2007*. The first two categories closely follow the attack taxonomy introduced by Hansman *et al.* [Hansman and Hunt, 2005] where this structure is used to classify potential attack targets.

The third category expresses the network configuration of a device allowing to specify the protocol stack configuration. For the look-up of an interest group, IP address ranges or subnet masks can be specified. In this manner, policy constraints by a system administration entity can be reflected. On the other hand, Yegneswaran *et al.* have shown in [Yegneswaran et al., 2004] that as “closer” (in terms of IP Address proximity)

subnets are to each other the more similar attacker blacklists become. Thus, a cooperative intrusion detection approach can be improved by local clustering. Additionally, mobile device characteristics can be reflected such as mobile IP settings.

The fourth category comprises of the used detection algorithms. Basically, this can either be a signature or an anomaly based approach. The model depicted in Figure 1 shows AIS [Luther et al., 2007] as an example for anomaly detection or a virus scanner as an example for signature based approach. For simplicityï¿½s sake, this branch contains a very flat hierarchy. However in the future, if advantageous, a more granular approach like the taxonomy from Axelsson could be applied here [Axelsson, 2000].

The fifth category is the hardware properties of a device. Here, processing power and memory are relevant attributes. The formation of homogeneous groups (c.f. Section 5.1) benefits from these attributes, depending on the used feature vectors. Additionally, bandwidth capabilities can be expressed here. This category, like the fourth, can also be extended in the future based on upcoming requirements and scenarios. Furthermore, an algorithm for the formation of groups is necessary.

Input: p Property base of a device
$r_{1..n}$ Objective of group formation
$c_{1..n}$ Interests related to objectives $r_{1..n}$
$g_{1..n}$ Group related to $r_{1..n}$ of a device
$k_{1..n}$ Maximum group size for $g_{1..n}$
m Messages; contain interest m_c, objective m_r,
sender m_{sender} and type

`Receive(`*message*`)`
switch *type* **do**
  case *interest*
    foreach $i = 1$ *to* $|r|$ **do**
      if m_r==r_i **and** $|g_i| < k_i$ **and** `Matches(`m_c,p`)` **then**
        `Send(`*hit*,r_i, c_i, m_{sender}`)`
      end
    end
  end
  case *confirm*
    Add m_{sender} to $groupg_x$ related to m_r
  end
  case *hit*
    if `Matches(`m_c,p`)` **and** $|g_x$ *related to* $m_r|$ $< k_x$ **then**
      `Send(`*confirm*, $|r_x|$, m_{sender}`)`
      Add m_{sender} to group g_x
    end
  end
end

Algorithm 1: Message Handling in the CIMD

Input: $r_{1..n}$ Objective of group formation
$c_{1..n}$ Interests related to the objective $r_{1..n}$

`Propagate()`
foreach $i = 1$ *to* $|c_i|$ **do**
| `Search(`r_i,c_i`)`
end

Algorithm 2: Propagation method in CIMD; here, look-up for collaboration partners is triggered. In the case, interests are not satisfied by enough other peers, process is repeated.

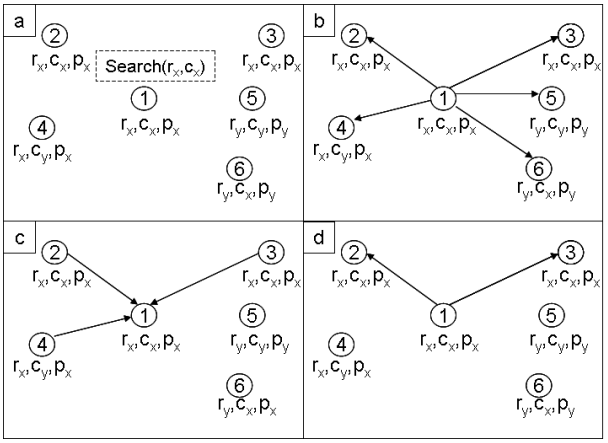

Figure 2: Group Formation Process divided in four phases a-d. The tuple entries below each node denote, which objectives (r) and interests (c) are pursued (for the sake of simplicity here, each node shown to have only one interest and associated objective) and p represents the property base of the node, where c_x matches p_x and c_y matches p_y. Node 1 propagates his objectives via the underlying search functionality (a) and sends interest messages. In the second step (b), the nodes 2-6 receive an interest message, where in (c) 2,3 and 4 respond with a hit message, as their objective and their property base match to the stated objective as well as the stated interest. Finally, in (d) node 1 sends a confirmation message to the nodes 2 and 3 but it does not send to 4 as the interest of c_y can not be fulfilled by the property base p_x of node 1.

2.2 Group formation

In this algorithm, we assume to have an overlay network providing search capabilities (in Algorithm 2 `Search` is used). The Algorithm 1 performs the grouping of the devices connected to the overlay. A device contains a property source p as described in Section 2.1. Additionally, each device has several group formation objectives $r_{1..n}$ and interests $c_{1..n}$ associated with them. An objective is the purpose of the collaboration like "Blacklist IPs" or "Signature Exchange"; in contrast, interests are the desired properties of a group associated with the objectives. Each device has a notion of his groups $g_{1..n}$ related to an objective and contains a maximum size constraint $k_{1..n}$.

Without loss of generality, we assume only one interest is associated with the objective. In this manner, the readability of the algorithm is increased; a *1 to n* relationship between objective and interest as well as group size is realized by verifying for each objective the corresponding interests and group sizes. An objective associated with more than one interest is necessary in the case of heterogeneous groups (c.f. Section 5.1).

The *Propagate()* function in Algorithm 2 enables a node A to state his search requests via the underlying overlay architecture and results in *interest* messages. A message can be processed by a node B if m_{sender} is compliant to an established security policy. This is then already realized before the message is processed by the algorithm (c.f. Section 4 for *Access Control* respectively security policies).

Upon the receipt of a message, we distinct between three cases: First, a node B may receive an *interest* message m of a node A containing an objective m_r and the interest m_c reflecting the desired properties. If the objective fits and the corresponding group has not reached the maximum member size, the interest m_c is matched against the own property base. We discuss the matching in Section 3. In the case of a match, a *hit* message is sent to the requester A wherein the objective and the interest(s) of B regarding the objective are contained. When A receives this message, it is matched against its own property base and checked whether the corresponding group is still not complete. In this case, a *confirm* message is sent and A adds B to the appropriate group. Accordingly, after receipt of the confirmation, B also adds A to his group.

As a consequence, when each participating node in CIMD looks up its own groups, the resulting groups are not common i.e. non-equal sets for each node. This may be desirable in some cases, but for the sake of finding common groups, one peer in a group can take a leading role for the formation of the group. Additionally, this measure reduces communication overhead in the overlay. An illustrative example for the group formation is given in Figure 2.

The concrete implementation of CIMD is still work in progress. Hence, in the following section we give an overview, which technologies we consider promising for achieving this aim.

3 Realizing CIMD

The main aspects of CIMD are the taxonomy describing the device properties, the grouping algorithm and the matching function(s). Above all, an overlay network providing search functionality and grouping support is necessary. Additionally, we argue that a common language to communicate between the participating nodes is beneficial for applied detection schemes.

The prototype realized in NeSSi (see Section 6) is based on an extension to the hybrid decentralized peer-to-peer protocol first presented in [Luther et al., 2007]. This protocol is based on the Supernode concept; i.e. in general all nodes in the system are equal in their opportunities, but after an election process a sub set of peers is chosen to perform the role of the Super Node. Furthermore, theses nodes often have other beneficial properties like long uptime, high bandwidths or a public IP address.

The Super Nodes fulfill special tasks like in the case of CIMD the formation of the groups as realization of the aforementioned grouping algorithm. Additionally, sample ontologies can be realized as EMF[15] (Eclipse Modeling Framework) data model for the description of the participants. In the following, we will give two alternative realization strategies for CIMD and discuss their characteristics with respect to possible usage scenarios.

[15] http://www.eclipse.org/modeling/emf/

3.1 Structured Overlay with Distributed Knowledge Base

We consider a structured overlay network as an interesting realization alternative for CIMD. These are peer-to-peer networks, where the nodes are not arbitrarily connected but in a deterministic manner and resources can be looked-up fast, e.g. logarithmic time. Often, the underlying technology is a DHT-based (Distributed Hash Table) approach. Here, one or more hash functions are used to arrange the participating nodes in an overlay to the co domain of the used hash function(s). In this manner, each node is responsible for a fraction of the co domain and administrates it. Next, a resource to be stored respectively their location is also hashed and put to the node responsible for that fraction of the co domain this value belongs to. Important representatives of this concept are *CAN* (Scalable Content Addressable Network), *Chord* and *Pastry* [Ratnasamy et al., 2001, Stoica et al., 2001, Rowstron and Druschel, 2001]. The structured approach offers the advantage of fast and deterministic search, i.e. if the resource can not be looked up in the overlay, it is not available. This is a big difference to unstructured networks, where popular resources often can be looked up easily, but rare resources are hardly found although they are available. On top of the structured overlay, we consider peer-to-peer based RDF (Resource Description Framework) stores (http://www.w3.org/RDF/). The RDF is a formal language to provide meta data in the context of Semantic Web and is based on statements about resources. Such statements are comprised of a subject, predicate and an object and are noted as triples, e.g. in the case of: "the car has the color pink", the car is the subject, has the color the predicate and pink the object. In this way, also the CIMD data model can be mapped to RDF: an illustrative example is "Device x has Operating System Linux" and additionally to take also the kernel version into account "Device x has Kernel Version 2.6". The RDF stores are used to store this triples in a DHT-based peer-to-peer overlay network. Therefore, each triple is stored three times in the hash table for each of the keys subject, predicate and object. Battré *et al.* as well as Heine *et al.* present an approach for the distributed querying of semantic information in the RDF stores present an approach for the distributed querying of semantic information [Battré et al., 2006, Heine et al., 2005]. For the sake of group formation, the aforementioned SCRIBE protocol offers an application-level multicast offering publish-subscribe mechanism similar to grouping concepts [Castro et al., 2002]. In the case, the CIMD data model is stored in RDF triples and is put to a DHT, even uncommon objectives and interests can be satisfied.

3.2 Unstructured Overlay with Local Knowledge Base

In contrast, links in unstructured overlay networks are established arbitrarily. Here, different overlay networks exist, that can be classified into purely decentralized, partially centralized, and hybrid-decentralized ([Androutsellis-Theotokis and Spinellis, 2004]). In the introduction of the main section it was already mentioned, that a hybrid-decentralized approach is realized in the NeSSi. To show the overall applicability of the CIMD approach to overlay networks, we will discuss the system to be based on a purely decentralized scheme: Gnutella. We consider the first available version 0.4 of the system (http://rfc-gnutella.sourceforge.net/developer/stable/index.html). Here, participating nodes are randomly connected to each other. In the case of a search, `Query`

messages are flooded to all neighbors. This flooding is limited by application-level hop count (this is not TTL of Internet Protocol). In the case a node fulfills a query, it sends a `QueryHit` message to the originator of the search. In contrast to the aforementioned approach, the instance of the cooperation model for every node is stored on the node, i.e. matching is carried out via the node itself. One limitation is obvious: the search does not include the whole overlay network, but only the nodes contacted via flooding. The notion of interest-based shortcuts, already introduced in the Related Work enables semantic group formation to bring nodes with similar interests in the overlay together [Sripanidkulchai et al., 2003]. This should enhance the look-up of new peers for collaboration, whereas the groups itself can be administrated via the algorithm presented in "Group Formation". As an example matching technique, we adapt the approach used in Bauckhage *et al.* to CIMD [Bauckhage et al., 2007]. In that work, a fast algorithm for expert peering in web communities is presented. Here, a large taxonomy reflecting different domains and their sub domains is constructed, basing on the inherent structure of the Open Directory Project (http://www.dmoz.org/). The whole taxonomy is converted into a large binary vector representation from the top to bottom and left to right. Then, each expert in the system can be identified by an instance of such a vector relating to his areas of expertise. In the next step, users can formulate requests for an expert. Those requests are also transformed to a binary vector representation and the scalar product of experts and requests is calculated as a measure of similarity. Additionally, weights for the entries in the taxonomy can be defined, e.g. as closer the leaves in the taxonomy tree are to the root node, the higher the associated weights. The CIMD approach can be directly transferred to such a system. The cooperation model can be encoded from top to bottom and left to right. Example: a peer has the interest to form a homogeneous detection group. The characterizing properties are encoded by the peer and submitted as `Query` via the underlying overlay. Each peer contacted, in case it has the same objective, calculates the scalar product of his property base against the interest and if a threshold is crossed, the peer responds with a `QueryHit`. On the one hand, the approach has the disadvantage, that the underlying data model is hardly extensible. It is not sensible or possible to match instances of different versions of cooperation models against each other. Second, because binary vectors are used, values can only be encoded with difficulties. One approach is to convert a string to the bit representation, but then maximum length must be fixed. On the other hand, this approach is very fast due to using scalar product and also scales well.

3.3 Summary

We presented two realization alternatives for CIMD and briefly presented their advantages respectively disadvantages. Comparing the approaches, the Extensibility of the first approach is much better than the second one, but it is not possible to look up directly similar devices. Here, the unstructured scheme enables a measure of similarity. This is a good fit to the homogeneous detection group scenario. In contrast, if exactly searched, every (rare) property can be found in the structured overlay network scenario. Additionally, the communication expenses for the first scenario should be less than in the unstructured scheme. Only in the case, queries become to complex, e.g. joint queries of RDF-triples, communication overhead increases. The unstructured scenario

has the advantage to be very robust against failures. Thus, we consider the structured scenario a good fit for fixed, large networks, as the approach is scalable and has logarithmic look-up time. In contrast, the unstructured approach is a good fit for mobile nodes or Ad-Hoc networks. The overlay is robust and the matching not computational expensive. Here, the property scheme needs to be fixed, but often mobiles have similar configurations compared to a desktop pc, e.g. similar hardware, similar set of application etc. Independent of the used approach, two more topics need to be mentioned: First, as common exchange format, we consider the IOEDF a good alternative due to is inherent, already presented characteristics. Furthermore, additional exchange formats can be incorporated as additional data like CIDSS or IDMEF. Second, we regard Trust Management an important topic in CIMD. Here, we refer to [Artz and Gil, 2007]. We foresee a static approach comprised of a priori trusted or non-trusted parties is not applicable for CIMD. Exemplarily, a pre-trusted host may be compromised and attack the system. There must also be a dynamic component, based on feedback as presented by Kamvar et al. [Kamvar et al., 2003]. Regardless of the wide variety of implementation options, the choice for CIMD highly depends on the value of the system for the purpose of intrusion detection.

4 Vulnerability analysis

Above all, the application of an omnipresent overlay dedicated to intrusion detection and prevention enforces concerns about the security of the system itself. Important security topics when considering overlay structures respectively peer-to-peer networks are *Availability*, *Access Control*, *Anonymity* and the *Authenticity* of stored "documents", i.e. in this case the device defining properties. In the following, we will briefly discuss each topic, demonstrate two adversary scenarios and offer possible counter measures.

First, *Access Control* is an important topic as CIMD provides on the one hand knowledge about contained nodes but on the other hand also enables peers to participate in a (possible) variety of intrusion detection measures. Here, a central login server respectively some central login servers like e.g. in the proprietary Skype[16] system can be used. For the case of the simulated scenario (c.f. Section 6.2), the NSP can provide the login functionality for his private or business customers.

Second, preserving the *Authenticity* of stored documents is not as challenging as it is in file-sharing peer-to-peer networks where it is difficult to determine whether a document a existed before a document b and to decide which is "original". Here, the creator respectively originator of the device description is well-known: the device itself. In this regard, whether the properties are stored or not stored on the device itself –but e.g. in a DHT– it is sufficient to sign the properties by the device itself. In the case of a look up, the authenticity of those properties can be verified by comparison of the device public key.

The *Availability* is highly affected by *DDoS* attacks or exploitation of protocol flaws. Fiat *et al.* present a censor resistant peer-to-peer network that sustains the breakdown of up to 50% of the participating nodes [Fiat and Saia, 2002]. Generally, availability depends on the underlying peer-to-peer overlay and as there are different implementation

[16] www.skype.com

strategies for CIMD, we abstract from it here. We now consider two sample adversary scenarios:

In the first scenario, we assume a malicious peer managed to access the CIMD overlay and searches for devices exposing vulnerabilities. Due to the fact, that vulnerabilities of e.g. a frequently used software or firmware are publicly known, an attacker may look up exploitable device configurations.

At first, the attacker needs to enter the system and secondly must have the permission to read such information. Hence, data needs to be associated with an authorization level. Thus, the formation algorithm (c.f. Algorithm 1) can be extended to include security policies. Based on the implementation, the properties can directly be extended by a privacy value. Then, the sender needs to provide the necessary authorization level to read the classified information.

In the second scenario, we regard in special the pattern exchange scheme as an application of *CIMD*. Here, the generator devices distribute patterns resulting in a DoS-attack. For instance in the case of a signature, the string "HTTP/1.1 200 OK" would result in blocking web server responses.

There exist two reasons for this scenario: (I) the used detection scheme in a device may result in a false positive so that a wrong signature is created. This is a general problem that especially affects anomaly-based detection schemes. Prior work from Luther *et al.* confronted this problem by enabling a cooperative anomaly status exchange affecting all participating detection units [Luther et al., 2007]. This scheme enabled a significant reduction of false positives. (II) The second reason for the distribution of wrong signatures is that such a device is compromised by an attacker with the clear intension to commit a DoS attack against the system. In both occasions, not relying on one, but at least m devices reporting a pattern is an option. Alternatively, (human) supervisors can be in charge of verifying signatures transmitted by the pattern generating machines and are the only entities "regular" devices accept signatures from.

5 Scenarios

A global detection overlay systems like CIMD enables a variety of scenarios improving state of the art approaches as well as allowing the development of new detection schemes. As a result, we present here two sample scenarios: in this manner, in the first scenario we are considering a homogeneous detection group enabling collaborative anomaly detection and in the second, a heterogeneous group of NIDS is applied exchanging signatures. The latter scenario is also simulated (c.f. Section 6).

5.1 Homogeneous Detection Group

As described in Section 1.2, there is ongoing research in cooperative AIS. The AIS is, like the Biological Immune System, based on the distinction between self and non-self [Forrest et al., 1994]. Initially, an n-dimensional feature space is covered by detectors (i.e., n-dimensional vectors of features: CPU utilization, memory usage, number of TCP connections...). In a training phase, these detectors are compared to feature

vectors describing the self. In the case of a match the detectors are eliminated, while the remaining than describe the non-self and are used for the detection of anomalies.

There are two challenges arising when dealing with anomaly detection schemes in general respectively AIS in special: on the one hand, anomaly detection often suffers from high false positive rates. Hence, we applied in [Luther et al., 2007] (c.f. Section 1.2) a cooperative intrusion detection approach to lower the false positive rate. On the other hand, anomaly detection can become computational expensive based on the number of detectors used. Essentially, in the training phase, computational costs depend on the covered feature space and the aimed density of detectors. In the detection phase, costs directly depend on the number of detectors to compare a feature vector with. Accordingly, a solution to lower computational costs is to partition the overall feature space and distribute different portions to several AIS nodes. In this way, each participating node is receiving a portion of the feature space and conducts the training. It is obvious, that just preserving a detector on one node comes with the danger of missing attacks. Accordingly, this results in a trade-off between desired redundancy on the one hand and performance constraints on the other. With combinatorial methods, a specified level of redundancy can be carried out deterministically in a decentralized manner. For further details we refer to [Bye et al., 2008a] and in Figure a3 an illustrative example is given.

The general assumption for such scenarios is that participating nodes have a common understanding of "normality". The nodes must, depending on the measured feature vector, be similar, i.e. have a common behavior, similar hardware etc. Otherwise an exchanged detector build by one AIS node is not suitable for another AIS node. As an example: in the case of measuring network statistics as input for the AIS, a web server would most probably offer a different behavior than an "ordinary" client computer. Here, the CIMD approach allows the formation of homogeneous groups by the specification of a similar node configuration, e.g. using the same Operating system, having similar hardware resources or even fulfilling a server application role like SMTP or HTTP.

5.2 Heterogeneous Detection Group

In the first scenario, we show how the CIMD approach enables the cooperation between different intrusion detection systems: we assume to have three different IDS manufacturers A, B, C selling NIDS appliances d_a, d_b, d_c. These systems are capable of detecting known malware by stored signatures provided centrally by their corresponding manufacturers. Accordingly, exclusively detecting known threats leaves the customer vulnerable to unknown threats especially zero-day attacks. As a result, the *vulnerability window*, i.e. the time between a threat emerging and patch being released needs to be minimized. The companies A, B and C provide updates about new attacks independently of each other. Each individual appliance d_{ij} ($i \in \{a, b, c\} \wedge j \in \{1..n\}$) connects during a fixed update interval (e.g. every hour) to its manufacturer checking whether new patterns are available.

Furthermore, we consider a large network service provider T connecting a set E of companies respectively business customers to the Internet, whereby each customer uses one of the aforementioned IDS appliances. In the first scenario (I) the appliances a, b

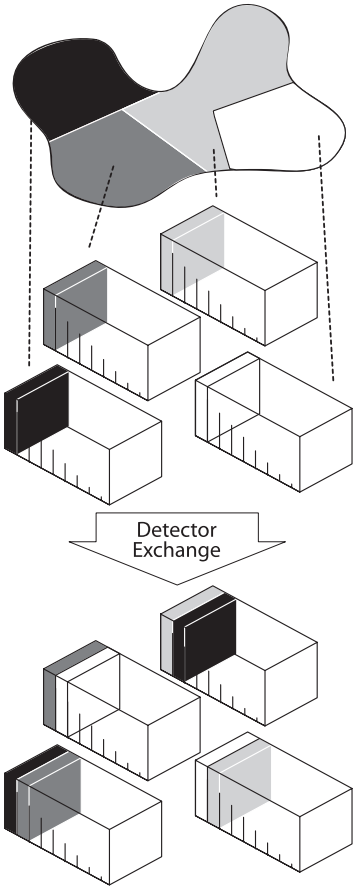

Figure 3: In the first step, we divide the common feature space among the similar devices and each device trains, based on this portion of feature space, detectors for abnormal states. Afterwards, combinatorial design techniques are used for detector exchange to guarantee a defined level of redundancy (*here*: every detector exists two times).

and c are used to protect each customer in the NSP network independently of each other.

Secondly, T applies mediators in the network capable of converting signatures between distinct formats, e.g. T has a contract with the different IDS manufacturers permitting this conversion. This can be realized by a distinct device that is capable of transforming the signatures to the according formats or as an extension hardware respectively software module to each appliance itself (*see also* CIDSS, c.f. Section 3). Similarly, IDS vendors may have bilateral contracts for signature conversion to improve their position on the market.

Hence, in addition to the update of signatures from the manufacturer of each device type, the mediator devices are checked. This cooperation can be realized via the CIMD detection overlay. The group objective is the mediation of signatures and the desired properties incorporate the different appliances. Hence, in scenario (II) the mediators are used to supply contracted devices with new signatures.

Last (III), the NSP applies devices capable of generating signatures based on suspicious traffic patterns. Hence, a device can also update the mediator and in this way deliver the self-generated signature to the other devices. Arising challenges regarding the specificity of the detection scheme respectively exploitation scenarios for this mech-

anism are discussed in Section 4. Concerning CIMD, this is an extension to scenario (II) incorporating the signature-generators in the groups. The introduced variables are further used in the simulation part (c.f. Section 6.2). There, we evaluate the benefit of CIMD for the here specified scenario.

6 Simulation

After motivating the application of CIMD, we define the simulation set-up for the scenario introduced in Section 5.1. At first, a novel network simulation environment tailored to security-related scenarios is presented: NeSSi.

6.1 NeSSi

The NeSSi is an agent-based network simulation environment built upon the JIAC (Java Intelligent Agent Componentware) framework [Fricke et al., 2001]. Thus, a discrete, event-based, packet-level simulation is realized; where each device contains a network layer enabling IPv4 or IPv6 packet transmission. Above the network layer, end devices additionally contain a transport layer offering TCP and UDP as well as an application layer providing SMTP, HTTP and IRC. The discrete steps in NeSSi are denoted as "ticks".

Foremost, NeSSi provides an API for the deployment and evaluation of detection units. These detection units can be well-known security solutions as standard virus scanners or new tools developed in scientific research projects. In NeSSi, both can be incorporated as long as they are adapted to a specified interface, and their performance can be compared for different traffic scenarios.

Furthermore, when a security framework composed of several detection units is to be tested, profiles can be used in NeSSi to simulate attacker behavior and attack patterns as well as user (E-Mail, HTTP) or system-inherent behavior. Thus, the profiles express characteristic traffic behavior that can be customized via port ranges, mean interval lengths and other distribution function dependent parameters. The Cooperative AIS presented in [Luther et al., 2007] was evaluated in the NeSSi environment. For further details about NeSSi we refer to [Bye et al., 2008b].

6.2 Simulation Set-Up

Here, we define the simulation set-up for the aforementioned scenario "Heterogeneous Detection Group" (cf. Section 5.1). We consider the network of T providing Internet access to a set E of customers. Each customer network e_j is protected by a device d_{ij} with $i \in \{a, b, c\}$ monitoring all by-passing traffic on the gateway connected to T.

The simulated networkï¿½s topology is based on X-Win, the backbone of Germany's National Research and Education Network (DFN[17]). Originally, this backbone connects more than fifty research institutes all over Germany, whereby in this scenario a smaller set of 29 locations is used. The core network is depicted in Figure 4.

[17] www.dfn.de

Parameter	Value
Susceptible nodes	726
Web Server nodes	322
Customer Networks	58
Average Susceptiple	12.5
Average Web Server	5.5
Pattern Generator Detection Threshold	4
Scanner Update Interval in ticks	100
Minimum Update Time in ticks	600
Maximum Update Time in ticks	2000
Mean Request Interval in Ticks	100

Table 2: Simulation parameters

In addition, each e_j is modeled as an *Access Networks* in NeSSi. In this regard, a core location is connected to an average of two customers resulting in 58 *Access Networks* and therefore 58 used scanners. The different types of scanners a,b and c are equally distributed among the customers. Each customer is represented by an average of 12.5 clients and 5.5 servers, i.e. there are in total 726 susceptible clients and 322 web servers. The constant simulation settings are depicted in Table 2.

The *attack vector* is based on drive-by downloads, i.e. exploiting vulnerabilities in a users client software like a web browser to install malicious code. According to the active Symantec Internet Security Threat Report [Symantec, 2007] the attack gained a considerable significance. Hence, we use the drive-by download for the infection of clients in this scenario.

Thus, the simulation variable p denotes the portion of malicious web servers. In this regard, the susceptible nodes select randomly, when initiating a request, an existing server IP address. Due to the random selection, a client might choose also a server from his "home" network. In the case of a malicious node, the server tries to install malware on the client node. The simulated malware is always unknown to the applied IDS appliances at the beginning of a simulation, but the appropriate signatures become available at runtime. Every device type has a different update time randomly (uniformly distributed) selected out of the interval between a fixed *Minimum Update Time* and *Maximum Update Time*. Hence, every device d_{ij} tries to update its threat data-base in a fixed *Scanner Update Interval* from a central server, administrated e.g. by T or the manufacturer.

Accordingly, if the scanner d_{ij} protecting e_j possesses already the signature for the attack, it prevents the infection of the client node; in case the malicious server is inside of the network e_j or the attack is still unknown, the node becomes infected. Additionally, if the cooperation is enabled, a detection device requests at the same time updates from the group members. We apply a grouping strategy building heterogeneous groups comprised of three members from different customer networks incorporating the disparate device types a,b and c. In addition, the generators monitor the network traffic and are capable of generating a signature for a new attack. In this regard, we model this functionality in NeSSi that a signature can be generated after observing it for a number

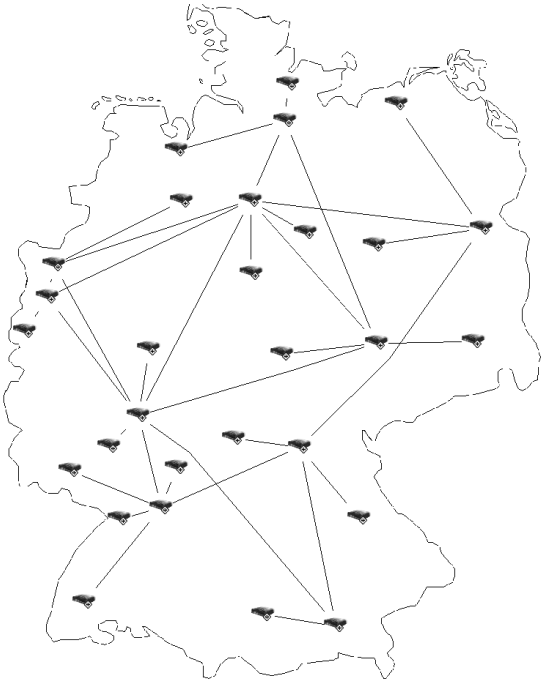

Figure 4: Backbone of simulated network analogous to X-Win

of times denoted by *Signature Generator Detection Threshold*. This functionality can be attached to a device, whereby from each group randomly one device is chosen.

Finally, this results in three different simulation options: first, there is the non-cooperative scenario (I) without using signature generators. Second, we apply cooperation but no generators (II); in the last scenario we apply in addition to the cooperative aspect signature generators (III). The scenarios were simulated with four different web server infection probabilities $p = 0.005, 0.01, 0.025$ and 0.05, where each scenario-infection probability combination was ran 40 times, i.e. a total of 480 simulation runs. Each run ends after completion of the *Maximum Update Time* plus two times the *Scanner Update Interval* because then every scanner must have had received a pattern update.

6.3 Results

The results are depicted in two different types of charts: Figure 5 provides an overview of the simulation showing the total number of infections for all scenario-infection probability combinations.

One trend is already visible in this chart: the signature generator approach "benefits" from a higher number of infected web servers, as the *Signature Generator Detection Threshold* is a constant value and a higher number of infected Web Servers results in a faster generation of a signature. We show the total number of infections over time in detail for the infection probabilities 0.005 and 0.05 in Figure 6 and Figure 7 accordingly. The time units here are intervals of 100 ticks. These charts show the common behavior of both series, non-cooperative and cooperative till approximately interval 7. Afterward, both series diverge.

The *Minimum Update Time* leads to an equal behavior in the beginning. In the scenario depicted in Figure 6 an amount equal to 80% of the non-cooperative scenario

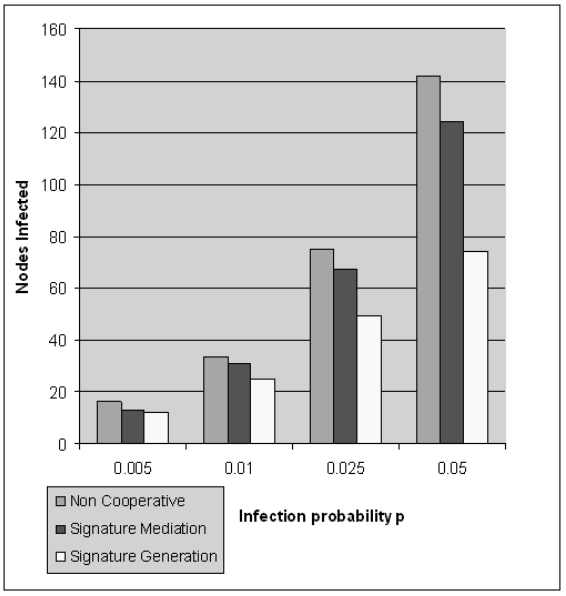

Figure 5: Total number of infections for each scenario with respect to the different infection probabilities

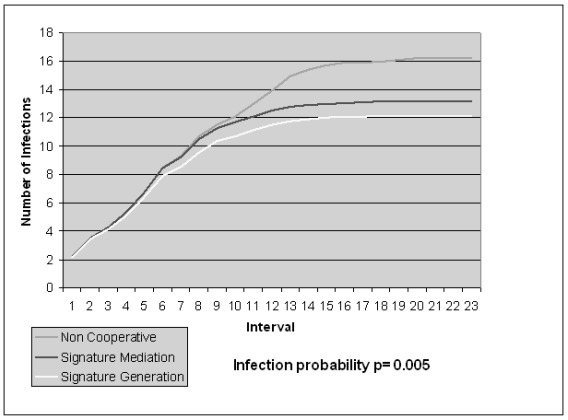

Figure 6: Cumulated infections over time; 0.5% WebServers infected

nodes is infected in the mediated case and an amount equal to 75% in the signature generation case. Analogous, these are 89% respectively 53% in the other scenario.

The impact of the *Minimum Update Time* is also shown in Figure 8, where the occurred infections within that interval are neglected. The average benefit in terms of less infections compared to the non-collaborative scenario is here 32 percent.

6.4 Analysis

The simulation results indicate already the merits of the collaborative approach. In the following, we give a formal analysis for the *Heterogeneous Detection Group* scenario. Here we compare the cooperative scheme ("signature mediation") with the non-cooperative approach. First, we show that a signature update is helpful in every case, i.e. in the worst-case there exist a reasonable probability a device remains uninfected during the vulnerability interval independent of the used approach. Second, we will

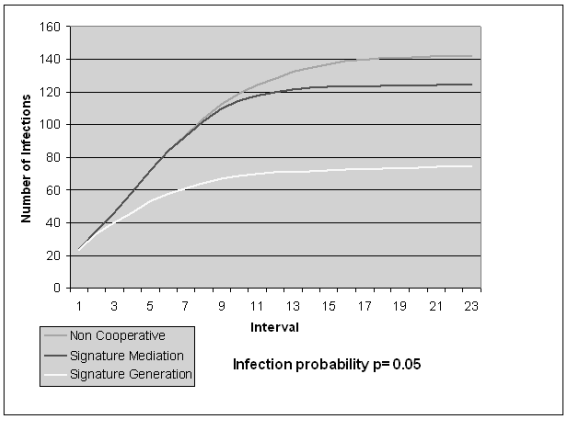

Figure 7: Cumulated infections over time; 5% WebServers infected

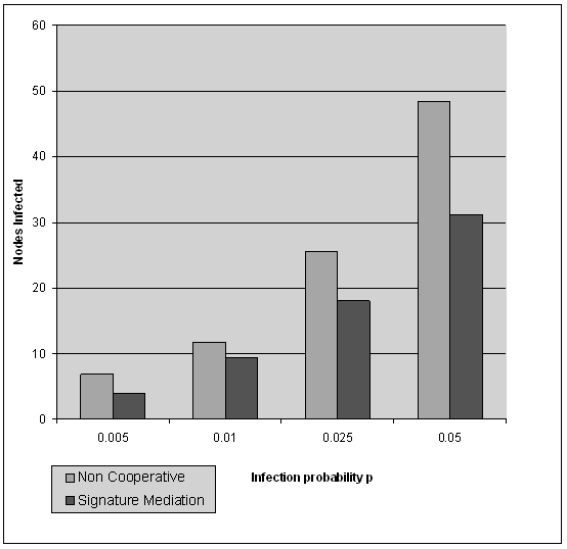

Figure 8: Total number of infections for signature mediation and non-cooperative approach with respect to the different infection probabilities. In this chart, infections occurring in the Minimum Update Time are neglected.

focus on the correlation between the total number of different scanning devices and the decreasing effectiveness of the non-cooperative scenario.

Let p_{inf} be the probability a susceptible node v_x requests a web site from an infected web server and n be the number of total requests a node conducts before a signature update is available. Hence, a node remains uninfected with the probability $p(v_x = not\_inf) = (1 - p_{inf})^n$. In the worst-case scenario, signature updates are available at the *Maximum Update Time*. As each node requests in a regular interval web sites, the maximal number of requests is

$$n_{max} = T_{max\_update}/T_{mean\_interval}$$

In this manner, Equation 1 expresses the probability for each susceptible node in the network not being infected till the *Maximum Update Time*. All these variables are known in the scope of the simulation (cf. Table 2). Thus, the worst-case probability $p(v_x = not\_inf)$ ranges from 0.90 (0.5 percent infected web servers) to 0.36 (5 percent

infected web servers).

$$p(v_x = not\_inf) = (1 - p_{inf})^{T_{max\_update}/T_{mean\_interval}} \quad (1)$$

These result shows, that even in the most unlikely case that signatures are available at the *Maximum Update Time* and five percent of the web servers are infected, at least an average of one third of the susceptible nodes remain uninfected.

Second, we want to show the impact of the signature mediation scheme. In the simulation, we equally distribute the three different IDS appliances d_a, d_b, d_c over the network, whereas each network is protected by exactly one device. Besides the case, the requested web server is inside the same network as the requesting client, the response passes two IDS. In the following, we assume to have n distinct appliances. In each network the probability a specific device is installed on a path is $1/n$. Thus, the probability an IDS is not installed one one gateway is $1 - 1/n$ and the probability it is not installed on a path between two different networks $(1 - 1/n)^2$. In this way, we receive Equation 2 denoting the probability a specific device type exists on a path between client and server.

$$p(d_x\_exists) = 1 - (1 - 1/n)^2 \quad (2)$$

In the case of the simulated scenario $p(d_x\_exists)$ equals 0.56, whereas in the mediated one all appliances receive signatures. Considering equation 2 it is obvious, that the advantage of mediation compared to the non-cooperative scenario becomes bigger with the increasing number of distinct device types. We neglect the scenario where a client sends a request to a server inside the same network as these just results in a constant factor $1/$*Customer Networks* for both schemes.

7 Conclusions and Future Work

We presented CIMD, a cooperation scheme for distributed intrusion detection approaches. Above all, we presented a taxonomy reflecting security-related device properties as well as an algorithm enabling participating nodes to form groups based on their aims, the *objectives* and associated *interests*. We also introduced the notion of detection groups and presented example scenarios where heterogeneous as well as homogeneous "teams" are beneficial. Additionally, the security of the system itself was discussed. Furthermore, we simulated a cooperative signature mediation scheme in NeSSi, a new simulation environment suited especially to security related scenarios. The mediation scheme showed a better performance as the non-cooperative approach, although the third scenario, applying both signature generators and mediation, outperforms the others. Subsequently we gave a formal analysis for the scenario where we showed that the value of cooperation grows with the increasing number of distinct, collaborating devices.

The results indicate that collaborative security schemes and the CIMD approach are promising. In this regard, the next step will be a comparison of ontology matching techniques for the matching function used in the grouping algorithm. We believe CIMD should support not only one but a variety of techniques. Nodes in CIMD may be interested on the one hand in concrete parameter values, but on the other hand, more abstract

notions of similarity can be beneficial for e.g. the homogeneous detection group scenario.

Implementation-wise, a standardized interface description will enable different implementations of CIMD respectively components of it. Further, the automatic gathering of the device defining properties is also an important task, as this can be, if done by hand, a time consuming activity. Finally, we plan to carry out a more detailed vulnerability analysis.

8 Acknowledgement

The authors like to thank their fellow colleagues at the Competence Center Security at DAI-Labor.

References

[Androutsellis-Theotokis and Spinellis, 2004] Androutsellis-Theotokis, S. and Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371.

[Artz and Gil, 2007] Artz, D. and Gil, Y. (2007). A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71.

[Axelsson, 2000] Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Department of Computer Engineering Chalmers University of Technology Göteborg, Sweden.

[Battré et al., 2006] Battré, D., Heine, F., and Kao, O. (2006). Top rdf query evaluation in structured p2p networks. In *Euro-Par*, pages 995–1004.

[Bauckhage et al., 2007] Bauckhage, C., Alpcan, T., Agarwal, S., Metze, F., Wetzker, R., Ilic, M., and Albayrak, S. (2007). An intelligent knowledge sharing system for web communities. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 2007*. IEEE Computer Society Press.

[Bye et al., 2008a] Bye, R., Luther, K., Camtepe, S. A., Alpcan, T., Albayrak, S., and Yener, B. (2008a). Decentralized Detector Generation in Cooperative Intrusion Detection Systems. In Masuzawa, Toshimitsu; Tixeuil, S., editor, *Stabilization, Safety, and Security of Distributed Systems 9th International Symposium, SSS 2007 Paris, France, November 14-16, 2007 Proceedings*, Lecture Notes in Computer Science , Vol. 4838. Springer.

[Bye et al., 2008b] Bye, R., Schmidt, S., Luther, K., and Albayrak, S. (2008b). Application-level simulation for network security. In *Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems*.

[Castro et al., 2002] Castro, M., Druschel, P., Kermarrec, A. M., and Rowstron, A. I. T. (2002). Scribe: a large-scale and decentralized application-level multicast infrastructure. *Selected Areas in Communications, IEEE Journal on*, 20(8):1489–1499.

[Fiat and Saia, 2002] Fiat, A. and Saia, J. (2002). Censorship resistant peer-to-peer content addressable networks. In *Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA.

[Forrest et al., 1994] Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994). Self-nonself Discrimination in a Computer. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 202–212. IEEE Computer Society Press.

[Fricke et al., 2001] Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sesseler, R., and Albayrak, S. (2001). Agent-based telematic services and telecom applications. *Communications of the ACM*, 44(4):43–48.

[Hansman and Hunt, 2005] Hansman, S. and Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31–43.

[Heine et al., 2005] Heine, F., Hovestadt, M., and Kao, O. (2005). Processing complex rdf queries over p2p networks. In *P2PIR '05: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*, pages 41–48, New York, NY, USA. ACM.

[Janakiraman et al., 2003] Janakiraman, R., Waldvogel, M., and Zhang, Q. (2003). Indra: A peer-to-peer approach to network intrusion detection and prevention. In *WETICE '03: Proceedings of the Twelfth International Workshop on Enabling Technologies*, page 226, Washington, DC, USA. IEEE Computer Society.

[Kamvar et al., 2003] Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA. ACM.

[Khambatti et al., 2004] Khambatti, M., Ryu, K., and Dasgupta, P. (2004). Structuring Peer-to-Peer Networks Using Interest-Based Communities. In Aberer, K., Koubarakis, M., and Kalogeraki, V., editors, *Databases, Information Systems, and Peer-to-Peer Computing – First International Workshop, DBISP2P 2003 Berlin, Germany, September 7 - 8, 2003 Revised Papers*, volume 2944 of *Lecture Notes in Computer Science (LNCS)*, pages 48–63. Springer.

[Loeser et al., 2004] Loeser, A., Naumann, F., Siberski, W., Nejdl, W., and Thaden, U. (2004). Semantic overlay clusters within super-peer networks. In Aberer, K., Kalogeraki, V., and Koubarakis, M., editors, *Databases, Information Systems, and Peer-to-Peer Computing*, volume 3367 of *Lecture Notes in Computer Science (LNCS)*, pages 33–47. Springer-Verlag.

[Luther et al., 2007] Luther, K., Bye, R., Alpcan, T., Albayrak, S., and Müller, A. (2007). A Cooperative AIS Framework for Intrusion Detection. In *Proceedings of the IEEE International Conference on Communications (ICC 2007)*.

[Ratnasamy et al., 2001] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Schenker, S. (2001). A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, volume 31, pages 161–172. ACM Press.

[Rowstron and Druschel, 2001] Rowstron, A. I. T. and Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware*, pages 329–350.

[Sripanidkulchai et al., 2003] Sripanidkulchai, K., Maggs, B. M., and Zhang, H. (2003). Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM*, volume 3, pages 2166–2176.

[Stoica et al., 2001] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California.

[Symantec, 2007] Symantec (2007). Symantec Internet Security Threat Report. Technical Report Volume XII, Symantec Corporation.

[Yegneswaran et al., 2004] Yegneswaran, V., Barford, P., and Jha, S. (2004). Global intrusion detection in the DOMINO overlay system. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2004*.

[Zhang et al., 2003] Zhang, Y., Lee, W., and Huang, Y.-A. (2003). Intrusion detection techniques for mobile wireless networks. *Wireless Networks*, 9(5):545–556.