Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics

Overlay Network for Security Purposes

Ph.D. Thesis Booklet

Author: Zoltán Czirkos
M.Sc. electrical engineering

Advisor: Dr. Gábor Hosszú
associate professor
C.Sc. engineering

Department of Electron Devices
Budapest, 2011.

1 Introduction

Hosts connected to the Internet are frequently attacked by well-trained hackers, viruses, worm programs and other malware. Due to the increasing number of attacks and the complexity of systems to be protected, the need to create automatic intrusion detection and prevention systems appeared.

However the realization of such systems opens many theoretic and practical questions. Many of these attacks have no detectable manifestations. Also to detect those which at least do so, usually enormous amounts of data has to be processed. A significant number of these attacks affect not only single hosts, but many hosts simultaneously, maybe some of which are on a single subnetwork [1]. In this case the events which belong the same attack are detected at multiple but distinct probes of the network. Those have to be collected and analyzed for correlation.

Intrusion detection requires notable computing power even in the single host case. By discarding some of the events detected, one is able to reduce the time required for processing, but this also decreases the accuracy of detection. Some of the attacks might remain unnoticed due to incomplete input, or false positives might be generated at other times by the attack correlation procedure. When processing data generated at multiple probes, network and computational load rises quickly with the number of probes.

Research Objectives

Most of the currently deployed intrusion detection systems suffer from the weakness that data collected is kept at the probes, and is not processed or correlated globally [2]. The goal of my research was to create a novel method, which enables the probes to share this information efficiently. This way the results of detection can be reused – multiple hosts (even on different networks) are able to protect themselves

against the intrusion attempts originating from the same, recognized attacker. By sharing information about events which are not necessarily attacks by themselves, but likely part of a complex attack, it is possible to detect network scale attacks as well.

The main research goal of such a system is to ensure the stability of the distributed detection network even when failures the physical networks or the nodes obstruct its proper functioning. This is of high importance in this application, as the reliability of a node under attack can be lower than normally it is. Another goal is to enable the system to distribute the network and computational load imposed on the nodes evenly.

2 Methodological Summary

The experimental characterization of P2P overlay systems is complicated due to the high number of nodes. Testing on wide area networks requires massive resources and cooperation of research institutes [3]. Therefore algorithms and methods designed for these are usually verified by simulation or theoretical analysis.

Therefore the Kademlia overlay network [4] is examined in my Ph.D. dissertation using these as well. Due to the shortcomings of the simulators available, I have developed my own simulator application which implements the algorithms of the Kademlia protocol required for the experiments carried out. The application simulates network errors with the error distributions presented in literature, collected on real networks. These distributions are considered with justifiable simplifications to verify the model developed both by means of simulation and numerical analysis.

The broadcast algorithm developed for the Kademlia overlay could be verified theoretically with some simplifications of the network error distribution models. Due to these simplifications, some inaccuracy could observed in the results when replication was also used in the algorithm. The method was therefore to be simulated in the

aforementioned simulator, too. This was also implemented in my own application which used round trip times and network error distributions measured in real networks, presented in the literature as well.

The intrusion protection method presented in the dissertation is verified by experimental analysis. The Komondor test overlay network collected intrusion attempt data in a duration of three years. By analyzing the correlated data collected from the probes, results and efficiency of the method could be verified.

3 New Theoretical Results

3.1 Thesis 1

Thesis 1 *I have developed a network security algorithm (Komondor), which can be used to increase the security of hosts by collecting intrusion detection data from various points of the network. [J1, J4, J7, J6, J8, C4, C5, B1]*

The *Komondor* method presented here uses the network itself to increase efficiency of protection against attacks.

The essence of the algorithm is that hosts to be protected create an *application level network* automatically, which is then used by them to keep other nodes informed about detected intrusion attempts. The detection data collected by nodes are sent into the overlay network in the form of reports. Should these aggregated reports suggest an attack when analyzed by their entirety, all participating hosts are alerted about the possible danger. In this case they can take the necessary steps to protect themselves, for example by tuning their firewall rules.

The effectiveness of the Komondor system comes from the fact that the knowledge base created by analyzing the events is not kept at the edges of the network (i.e. at the probes), but become beneficial for all participants. This way they can increase the protection of each other in the case of attacks aimed at multiple hosts (Figure 1). The method

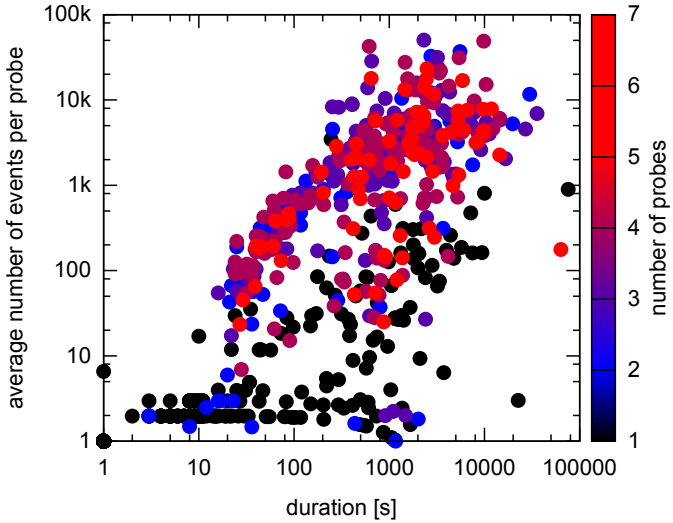

Figure 1: Login attempts of an attacker worm [5] with invalid user names and passwords

can also be used to prevent intrusions as any node can improve its protection against the recognized attacker before they become affected as well.

Subthesis 1.1 *I have developed a method to aggregate intrustion detection data from various probes, and to create a distributed database from those using an evaluation method. I have proved that in the method developed the nodes implementing the intrusion detection have to communicate on a DHT based overlay network in order to ensure efficiency. [J1, J2, J4, C1, C2, B4, B5, B6, B7, B9]*

Attacks can usually be recognized by detecting multiple *events* and *correlating* them. Several methods are described in the literature which enable one to correlate events [6, 7, 8], and those can be used

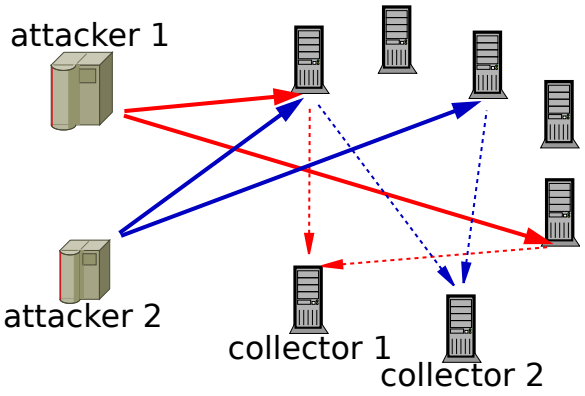

Figure 2: Intrusion detection in the Komondor method

to recognize different kinds of attacks. The Komondor method makes the *distributed* implementation of these methods possible.

Komondor nodes record every event which might be part of some attack. Recorded events are assigned an *index* and one ore more *keys*. Keys can be selected by the properties of events, and are used to correlate the attacks. Those events detected at different probes but suspectedly in some relationship with others should be assigned the same key. The key can be the IP address of the offending packet, for example. This way events that can be associated with the same attacker can be detected at different probes and still be processed together by the system.

The index of the event relates to its severity and can be assigned by examining its nature. More important events should be assigned a higher index. If the sum of these exceeds a given limit, the system treats the correlated events as a recognized attack and alerts the participants.

The application level network created by Komondor nodes is a *structured peer-to-peer (P2P) overlay*, which implements a *distributed*

hash table. DHT's store key-value pairs. Every piece of data (value) is mapped onto a node of the overlay by using its identifier (the key). In the Komondor method this is done by the key assigned to the event. This enables events with potential correlation to the same attack or attacker to be aggregated and correlated by the same node, as the same key is assigned to them. We call this the *collector node* (Figure 2), and this is the node to process events and start the alert procedure in the overlay if the correlated events indicate an attack.

This structure has several benefits. On the one hand, the P2P model provides a reliable, self-organized and stable overlay to store event data. In the case of network errors or node failures the overlay reorganizes itself to replace the missing node. The failure of the collector node is also handled by simply delegating its task to one of its neighbours. On the other hand, different keys are stored at different nodes of the overlay. Distributing network and computational load among the nodes is this way also achieved for this kind of application of the DHT, as events from different attackers are collected and processed by different nodes. Should events of the same network scale attack be detected at any node of the overlay, they are still sent to the same collector node by assigning them the same key. Processing data is therefore efficient, and the stability of P2P overlays is combined with advantages of centralized systems.

Subthesis 1.2 *I have shown that in handling the traffic generated by analyzing intrusion detection data in a DHT overlay, the Kademlia topology is superior to any other known DHT topologies. [J8, C1, B1, B7, B8]*

There are several P2P overlay topologies described in literature, and those have various efficiencies when handling different workloads. The *Kademlia* overlay organizes its nodes into a binary tree, in which an iterative routing and lookup protocol is used [4]. Contrary to other DHT overlays, nodes of Kademlia do not forward data to be stored node by node to the responsible one selected by the key, but rather the node requesting the storage of the key will determine the IP address of

Overlay	**Chord**	**Kademlia**
Node lookup	0	$O(\log_2 N)$
First event	$O(\log_2 N)$	$O(1+\log_2 N)$
n **events with the same key**	$O(n \cdot \log_2 N)$	$O(n+\log_2 N)$
Average messages per event	$O(\log_2 N)$	$O(1)$

Table 1: Number of protocol messages in intrusion detection in an overlay of N nodes

the selected node by starting successive lookup requests in the overlay. After the lookup the two nodes communicate directly with each other.

In this intrusion detection application of the overlay, a usual scenario is to find that many events in correlation with the same attack are detected in a short time interval (Figure 1). By the assignment of the same key the responsible collector node is also the same. When using Kademlia, the IP address of the collector node is stored in the lookup cache of the detector node, and later they will communicate directly, thus freeing the overlay network of innecessary load. This way the average network load is only one message per event detected (see Table 1). This is also independent from the size of the overlay, therefore the choice of the Kademlia topology is reasonable for large overlays as well.

3.2 Thesis 2

Thesis 2 *I have developed a new method to determine the system wide configuration parameters of the Kademlia DHT overlay network. [J1, J2, C1, B2]*

In the iterative lookup method of Kademlia a node initiating a lookup must connect to many of its neighbours in the overlay. However it is not always possible to connect two arbitrary nodes. The main reason of this is that firewalls and network address translation inhibit nodes behind them to be connected to: they can only initiate outgoing connections, but not listen to incoming ones [9]. This results in some lookups failing. It is possible for a key-value pair stored at a node to be unreachable by others. Another possible side effect of this is keys to be stored at improper locations, not at nodes selected by the key, as the node which requests the storage cannot connect the selected node properly.

By using replication, i.e. storing the data at multiple locations, the chances of at least one node being able to be connected to storing the key will increase. But replication imposes a higher network and storage capacity demands on the overlay and its nodes. The goal of the method developed is to determine the *minimal level of replication* in such a way that the storage and retrieval of data can be guaranteed with a given probability.

Subthesis 2.1 *I have developed a model which can be used for the Kademlia overlay to determine the level of replication both analythically and numerically, in such a way that the overlay network can guarantee a required level of correctness, given the distribution of network errors as a boundary condition. [J1, C1, B2, B3]*

In the overlay the key-value pairs are assigned to the nodes by the distance of the hashed values of keys to the identifier of nodes using the SHA-1 hash function. Every pair is sent to the node which has the identifier closest to the key in question. This is usually referred to as the *closest node to the key*. When using replication, not only the closest node will store the kay, but many of the closest ones. The number of these nodes are denoted as k_r.

The node initiating the lookup will first determine IP address of the closest node to the key by means of successive lookup request sent to

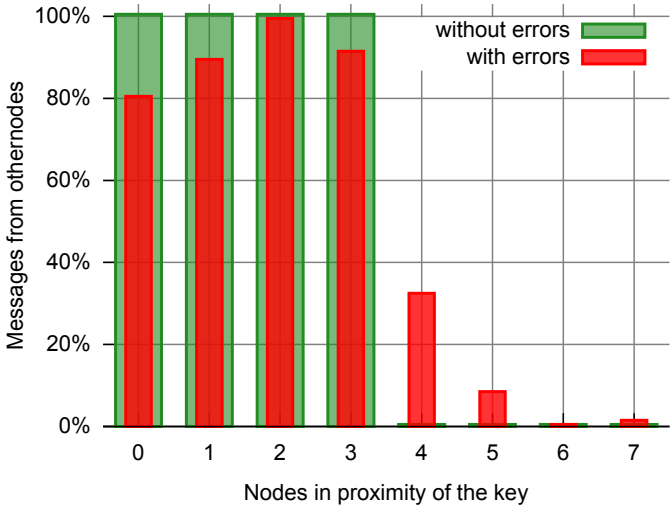

Figure 3: Messages sent to nodes in proximity of the key in the Kademlia overlay, with using k_r=4 as the replication level

neighbours. The correctness of this lookup procedure is proven in [4]. After this success of the retrieval of data depends on the initiating node being able to connect the closest node to the key. If it is unable to do so, the lookup will fail. However in order for to the lookup to be succesful it is sufficient to be able to connect to any of the nodes in proximity with the key, it is not necessary to connect to the closest node itself.

The model presented here is based on examining them connectivity of nodes. If any of the nodes around the key is able to listen to the incoming connections of other nodes, the data stored can also be retrieved by other nodes. If there are such in the overlay, the group of nodes around the key will also contain some of these, as nodes joining the overlay choose their place in identifier space randomly in Kademlia.

If connections fail, nodes will send the data to be stored or lookup requests to nodes which are further away from the key (see Figure 3). This happens because they cannot know if their selected destinations of messages have left the overlay or their network lins are failing. The results of this is data store and data lookup requests being dispersed in a range wider than k_r. With the model presented here, this phenomenon can be observed quantitatively. In essence determining of the required k_r replication level is choosing the range to be wide enough so that there is at least one node in it, which can be connected to from most of the other nodes.

Subthesis 2.2 *I have shown that in the Kademlia overlay the retrievability of data stored depends only on the global distribution of network errors, as identifiers of nodes are chosen randomly.*
I have proved that the required level of replication needed for any given correctness is independent of the size of the overlay if the errors of network links (as the boundary condition) is the same. [J1, C1]

The keys stored in the overlay and the nodes of the overlay do not have any *ab ovo* association, but rather it is created by the overlay. A key will be assigned to a node if the node joining the overlay chose its identifier in such a way that it is closer to the hashed value of the key than any other identifiers in the overlay. As the network link of the node and its identifier also do not have any correlation, stored data and network errors also do not have such. Nodes close to the key in identifier space can be very far from each other physically. The consequence of this is that the distribution of network errors is only to be treated globally, not on a per node basis.

When the size of the overlay increases, also the identifier space becomes denser of nodes. As these nodes choose their place in identifier space randomly, the k_r closest nodes to the key will be dispersed physically regardless of the density of identifier space. If we assume that new nodes in the overlay have the same types of network links on average as older ones (for example, still 15% of them have a link with

network address translation), the observable connectivity of nodes close to the key remains unchanged. Therefore the same level of replication can be used for large overlays with more nodes, as that for smaller overlays.

Subthesis 2.3 *I have shown that by increasing the level of replication in the Kademlia overlay, data storage can be made reliable even in the presence of unreliable nodes. [J1, C1, B2, B3]*

The model developed and presented in my dissertation enables one to calculate the required level of replication.

The identifier of a node is denoted with m, and the ratio of errors (i.e. number of nodes that cannot connect to it) is denoted with the function $h(m)$. As nodes and network errors are not correlated to each other, the ratio of errors can be sorted in increasing order. If the acceptable ratio of network errors is β, node m can store data reliably if $h(m) \leq \beta$. Follows that the expression $h^{-1}(\beta)$ gives the ratio of nodes that can store data reliably, if the identifier space is mapped onto the $[0,1)$ range and approximated with a continuous distribution [10]. As hash functions cover their set of output values evenly, this expression also gives the probability of a key finally being stored at a reasonably reliable node.

The correctness of data lookups can be calculated from the above, and can be enhanced by increasing k_r, the level of replication. This is given by the following equation:

$$k_r = \left\lceil \frac{\ln(1-P)}{\ln(1-h^{-1}(\beta))} \right\rceil, \tag{1}$$

where P is the required probability of correct lookups. As replication can only be an integer number, the fraction has to be rounded up to the nearest integer value.

If the exact distribution of network errors, i.e. the function $h(m)$ is unknown, the value of $h^{-1}(\beta)$ can be approximated by nodes with

the ratio of successful connection attempts. For example if a node estimates that 15% of its connections initiated fail, then the approximation $h^{-1}(\beta) = 0.85$ can be used to determine the level of replication required. This makes the above equation easily applicable for overlay operation, as the ratio can easily be measured by any node in the overlay.

3.3 Thesis 3

Thesis 3 *I have developed an algorithm to do efficient broadcast messaging in the Kademlia overlay. [J1, J2, J3, J5, J9, C3, B1]*

In the Komondor method a collector node detecting an attack has to alert all participants in the overlay, so as to let them protect themselves against the attacker. For this a broadcast message in the overlay can be used. In P2P overlays, broadcast messaging is seldom used, but in this intrusion prevention application that is just required. The algorithm presented here uses the builtin topology of Kademlia to make the broadast as fast as possible.

Subthesis 3.1 *I have observed that any Kademlia overlay can be transformed in such a way, that a given node from the address space can move to any other address of it, while the distances in the address space of the overlay of any arbitrary two nodes are left unchanged.*
I have developed a transformation method to make the development of overlay level communication algorithms more simple.

In the Kademlia system, distance of nodes is calculated with the bitwise exclusive or (XOR) function. This is resembled by the binary tree used for illustrating the overlay. The higher the most significant bit 1 is found in the distance of two nodes, the higher the lowest subtree of the identifier space that is common for them.

If an identifier N of a node is applied with the bitwise exclusive or function to all other nodes' identifiers, a transformed overlay is created

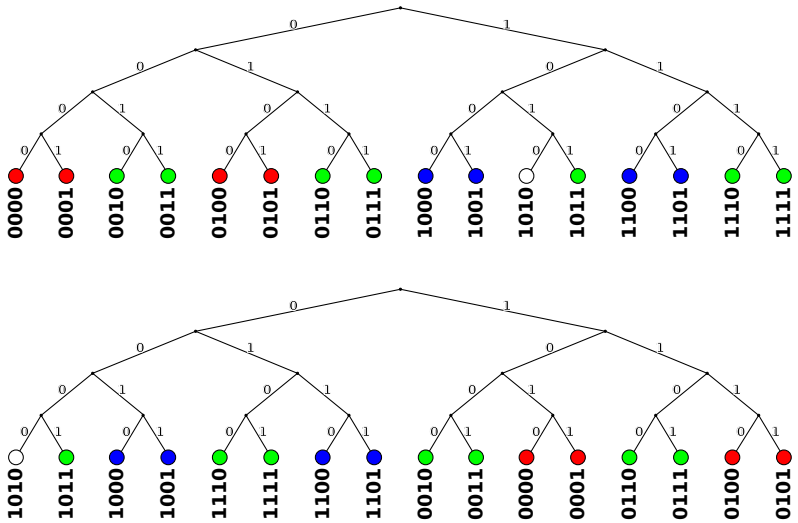

Figure 4: Transformation of the Kademlia overlay. With the transformation the white-coloured node with identifier 1010 can be moved to the 0000 position, without the distances of any arbitrary pairs of nodes changing

(Figure 4). In this overlay all identifiers are changed. As $N \otimes N \equiv 0$, the node chosen is moved in such a way that its new address is exactly zero in identifier space. Still the distances of any arbitrary pairs of nodes in the overlay are left unchanged, as it can be proven with the following equation:

$$D = A_{new} \otimes B_{new} = (A \otimes X) \otimes (B \otimes X) = A \otimes B. \tag{2}$$

Therefore the transformed overlay has exactly the same routing tables as the original one has, as those tables only depend on the distance of the nodes, not the addresses themselves.

The advantage of such transformation that it simplifies the design of algorithms of overlay communication. Any algoirthm (also the

broadcast communication presented here) can be designed as if the node executing it were in the 0 point in identifier space. Still, methods described like this are general and can be used any time, as the transformed overlay can be transformed once again back into its original form.

Subthesis 3.2 *I have verified theoretically that the number of steps required to complete the broadcast in the overlay increases only proportionally to the logarithm of the size of the overlay, and that it does not increase the maintenance cost of the overlay.*
I have verified the results of numerical analysis with simulation as well. I have run the experiments for different boundary conditions which were similar to those in real networks. [J1, J2, J3, J5, J9, C3, B1]

The main idea of this algorithm is that the node initiating the broadcast divides the binary tree of nodes of the overlay into smaller and smaller equally sized subtrees. In every step the remaining subtree is divided into two, one of which is the subtree containing the initiator and the other one which is not. In the former one it continues to send the broadcast message to other nodes, while in the latter one a responsible node is selected to do so.

This algorithm of dividing the address space in every step into two equally sized parts matches the routing algorithm of Kademlia, therefore it is possible to use the routing tables maintained by the lookup procedure. This is shown in Figure 5 with a transformed overlay, in which the node initiating the broadcast has been transformed to the 0 address of identifier space. In the distant part of the address space the responsible node will execute the algorithm, while in its own part the initiator node is itself the responsible one. Therefore it divides the remaining part of the tree again. From the distant section a responsible node is chosen, while its own section is managed by itself.

The responsible nodes can execute the algorithm independently in their own subtree, as they do not have to wait for each other. As the diameter of the overlay, i.e. the greatest distance of any arbitrary two

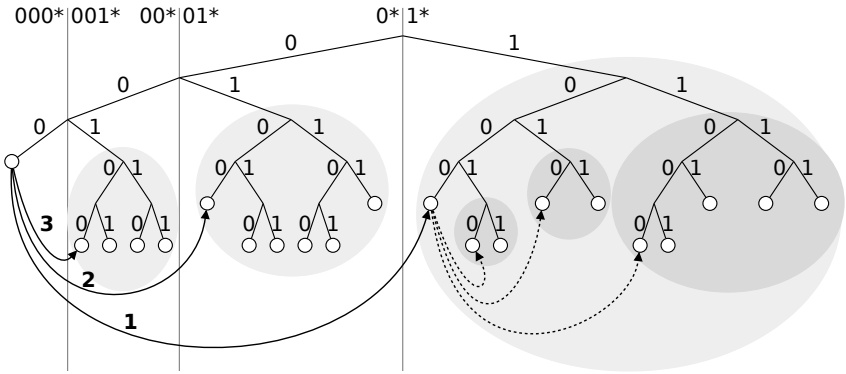

Figure 5: Broadcast algorithm in the Kademlia overlay

nodes in the overlay is proportional to $\log_2 N$, where N is the size of the overlay, the broadcast algorithm will also run in a time complexity of $O(\log_2 N)$.

The algorithm was implemented in my own Kademlia simulator, to verify the duration of the broadcast and correctness of the algorithm. The underlying physical network was modeled using data from real Kademlia networks from literature [9, 11].

Subthesis 3.3 *I have proved that by using replication the broadcast algorithm can be made reliable even when there are network errors or byzantine nodes in the overlay.*
I have determined the level of replication as the function of required correctness, the ratio of network errors and the size of the overlay. [J1, J3, J5, J9]

The nodes of the overlay receive the broadcast message with a varying number of steps, i.e. the message is forwarded by a different number of responsible nodes for every destination. Some receive it directly from the initiator, while for others the message is forwarded

many times. If these messages are lost on the network, or maybe some of the nodes are byzantine (they do not forward messages), then the reliability of the algorithm decreases. The messages lost will induce different failure ratios depending on their importance. A message which selects the responsible node for the half of the address space can even induce an error of 50%.

Reliability from the viewpoint of a node depends on the distance from the initiator. The more responsible nodes are along the path from it to the initiator, the higher the probability that the message will get lost before it arrives. As messages are forwarded to distant subtrees in every step, the address of the node it is forwarded by changes in one bit every step. Thus the number of responsible nodes along the path is equal to the Hamming distance of the two identifiers. The correctness of the broadcast can be estimated by this, with taking into account the number of bits in the address space with b, and the probability of every message arriving with P:

$$m = \frac{\sum_{i=0}^{b} \binom{b}{i} P^i}{2^b}. \tag{3}$$

The reliability of the algorithm can be increased using replication. This means that multiple responsible nodes have to be selected from every subtree. The faults of those are independent of each other, therefore the probability of not even a single responsible node receiving the message and doing its task will decrease. If the messages are lost with the probability of P_h, then the probability of a subtree receiving the message is increased to $P = 1 - P_h^{k_b}$. Using this formula the necessary level of replication (i.e. the necessary number of responsible nodes from every subtree) can be determined:

$$k_b = \left\lceil \frac{\ln\left(2\left(1 - \sqrt[b]{n}\right)\right)}{\ln P_h} \right\rceil, \tag{4}$$

where b, the virtual number of bits in the address can be estimated by N, number of nodes in the overlay ($b = \log_2 N$). Packet loss ratio is P_h,

and n is the required level of correctness. (The number of nodes in the overlay can be estimated by any node by examining the density of the address space, so this information is available to any node performing the broadcast.)

The network load compared to the size of the overlay and the replication level k_b can be determined by using simulation. The number of messages without replication is exactly $N-1$, which is the theoretical minimum possible, as every node has to receive the message at least once. When using replication, the number of messages will increase, and this also depends on the latency of messages in the network. This is caused by the broadcast message being forwarded along different paths to a single node, and the number of duplicates (innecessary messages) therefore being different, depending on which one it receives first.

4 Application of New Results

The Komondor method can generally be used by hosts connected to the Internet to detect intrusion attempts (Figure 6). By using the algorithm events correlating to attacks can be aggregated and correlated efficiently, even if they are detected at various points of the network. The DHT overlay network organized by Komondor nodes can be used to balance network and computational load on participants.

The algorithms developed for the Kademlia overlay can be used in any application which is based on Kademlia. The model of Thesis 2 enables one to determine a system wide configuration parameter, namely k_r which is the level of replication for data items. The required level of correctness can be achieved by increasing k_r, but still the network load can be kept low by using the lowest possible value. The input parameter $h^{-1}(\beta)$ of eq. 1 is the reliability of nodes in the overlay, which can be estimated by any node tracking the number of failing contacts.

The broadcast algorithm presented here can also be used in any

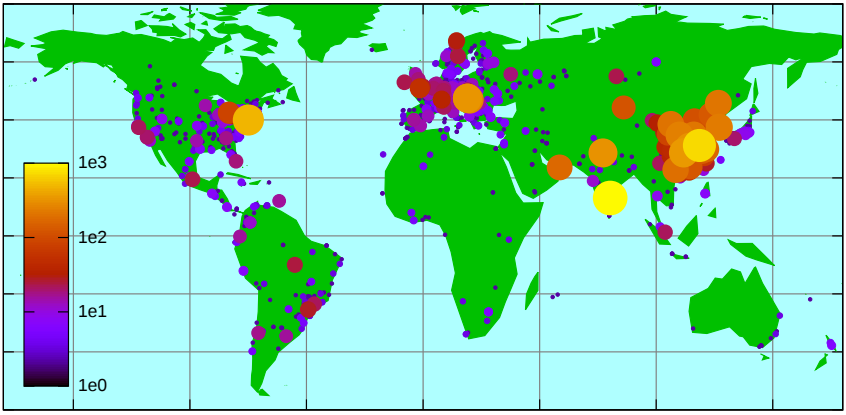

Figure 6: Attacks detected by the Komondor system on the map of the world. The size of the points show the number of attacks from every IP address

overlay network based on Kademlia. By using it, any information of global importance can be sent to nodes. Such a piece of information is for example the IP address of a recognized attacker in the Komondor intrusion detection system. The broadcast messaging algorithm also enhances the features of lookup queries in the Kademlia overlay, as it can be used to disseminate any type of query, not only the exact key match queries provided by the DHT algorithm. The level of replication for this can be determined by using eq. 4. The arguments of the formula, the number of nodes in the overlay and the ratio of network error can be estimated by nodes, and therefore no global knowledge of the overlay has to be assumed.

Journal Papers

[J1] Zoltán Czirkos and Gábor Hosszú. Peer-to-peer alapú betörésérzékelés. *Híradástechnika*, 63:29–36, 2008. Pollák–Virág díjas cikk.

[J2] Zoltán Czirkos and Gábor Hosszú. Distributed Detection of Intrusions. *Informatika – a Gábor Dénes Főiskola Közleményei*, XII(2):37–40, 2010.

[J3] Zoltán Czirkos and Gábor Hosszú. Peer-to-peer Based Intrusion Detection. *Infocommunications Journal*, LXIV(I):3–10, 2009.

[J4] Zoltán Czirkos, Loránd Lehel Tóth, Gábor Hosszú, and Ferenc Kovács. Novel Applications of the Peer-to-Peer Communication Methodology. *Journal on Information Technologies and Communications – Research, Development and Application on Electronics Telecommunications and Information Technology*, E-1(1(5)):59–70, 2009.

[J5] Zoltán Czirkos and Gábor Hosszú. Enhancing the Kademlia P2P Network. *Periodica Polytechnica*, kézirat elfogadva, megjelenés alatt.

[J6] Zoltán Czirkos and Gábor Hosszú. Műveleti rendszerek egyenrangú közlésen alapuló védelme. *Informatika – a Gábor Dénes Főiskola Közleményei*, 8(4):9–21, 2005.

[J7] Zoltán Czirkos. Operációs rendszerek egyenrangú közlésen alapuló védelme. *Linuxvilág*, VII(5):65–69, 2006.

[J8] Zoltán Czirkos and Hosszú Gábor. A Distributed Intrusion Network Based on Kademlia. *Computers & Security*. Kézirat elküldve.

[J9] Zoltán Czirkos and Hosszú Gábor. Pseudo Reliable Broadcast in the Kademlia P2P system. *Computer Networks*. Kézirat elküldve.

Conference Proceedings

[C1] Zoltán Czirkos, Lóránd Lehel Tóth, and Gábor Hosszú. Komondor – P2P Intrusion Prevention, poster. In Róbert Szabó and Attila Vidács, editors, *HSN Workshop 2009*, Balatonkenese, May 2009.

[C2] Zoltán Czirkos and Gábor Hosszú. Az elosztott betörésérzékelés hatékonysága. In *Informatika Korszerű Technikái*, Dunaújváros, 2010. Dunaújvárosi Főiskola.

[C3] Zoltán Czirkos and Gábor Hosszú. Üzenetszórás modern P2P hálózatokban. In *Informatika Korszerű Technikái*, pages 13–23, Dunaújváros, 2008. Dunaújvárosi Főiskola.

[C4] Zoltán Czirkos and Gábor Hosszú. P2P alapú betörésvédelem. In *Informatika Korszerű Technikái*, pages 45–52, Dunaújváros, 2007. Dunaújvárosi Főiskola.

[C5] Zoltán Czirkos. P2P alapú biztonsági szoftver fejlesztése. In *Információvédelem menedzselése XXII. Szakmai fórum*, pages 43–47, Budapest, 2006. Hétpecsét Információbiztonsági Egyesület.

Edited Books

[B1] Zoltán Czirkos and Gábor Hosszú. Usage of Broadcast Messaging in a Distributed Hash Table for Intrusion Detection. In Peyman Kabiri, editor, *Privacy, Intrusion Detection and Response: Technologies for Protecting Networks*. IGI Global, Hershey, 2011.

[B2] Zoltán Czirkos and Gábor Hosszú. Reliability Issues of the Multicast-Based Mediacommunication. In Margherita Pagani, editor, *Encyclopedia of Multimedia Technology and Networking*, pages 1215–1223. Information Science Reference, Hershey, 2009.

[B3] Zoltán Czirkos and Gábor Hosszú. On the Stability of Peer-to-Peer Networks in Real-World Environments. In Antonio Cartelli and Marco Palma, editors, *Encyclopedia of Information Communication Technology*, pages 622–630. Information Science Reference, Hershey, 2008.

[B4] Zoltán Czirkos and Gábor Hosszú. Application of the P2P Model for Adaptive Host Protection. In Margherita Pagani, editor, *Encyclopedia of Multimedia Technology and Networking*, pages 54–60. Information Science Reference, Hershey, 2009.

[B5] Zoltán Czirkos and Gábor Hosszú. Peer-to-Peer Methods for Operating System Security. In Goran D. Putnik and Maria Manuela Cunha, editors, *Encyclopedia of Networked and Virtual Organizations*, pages 1185–1191. Idea Group Inc., Hershey, 2008.

[B6] Zoltán Czirkos, Gábor Hosszú, and Kovács Ferenc. E-Collaboration Enhanced Host Security. In Ned Kock, editor, *Encyclopedia of E-Collaboration*, pages 172–177. Information Science Reference, Hershey, 2008.

[B7] Zoltán Czirkos and Gábor Hosszú. Intrusion Detection Based on P2P Software. In Mehdi Khosrow-Pour, editor, *Encyclopedia of Information Science and Technology*, pages 2232–2238. Information Science Reference, Hershey, 2008.

[B8] Zoltán Czirkos and Gábor Hosszú. A Novel Application of the P2P Technology for Intrusion Detection. In Antonio Cartelli and Marco Palma, editors, *Encyclopedia of Information Communication Technology*, pages 616–621. Information Science Reference, Hershey, 2008.

[B9] Zoltán Czirkos and Gábor Hosszú. Network-based intrusion detection. In Mário Freire and Manuela Pereira, editors, *Encyclopedia of Internet Technologies and Applications*, pages 353–359. Idea Group Inc., Hershey, 2007.

References

[1] Hervé Debar and Andreas Wespi. Aggregation and Correlation of Intrusion-Detection Alerts. In Wenke Lee, Ludovic Mé, and Andreas Wespi, editors, *Recent Advances in Intrusion Detection*, volume 2212 of *Lecture Notes in Computer Science*, pages 85–103. Springer Berlin / Heidelberg, 2001.

[2] C.V. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.

[3] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[4] Petar Maymounkov and David Mazières. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. 2002.

[5] Christian Seifert. Analyzing Malicious SSH Login Attempts. `http://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts`, November 2010.

[6] Chenfeng Vincent Zhou, Shanika Karunasekera, and Christopher Leckie. A Peer-to-Peer Collaborative Intrusion Detection System. In *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*, volume 1, page 6. IEEE, 2006.

[7] Alfonso Valdes and Keith Skinner. Probabilistic Alert Correlation. *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 54–68, October 2001.

[8] S.J. Templeton and K. Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 workshop on New security paradigms*, pages 31–38. ACM, 2001.

[9] S.A. Crosby and D.S. Wallach. An Analysis of BitTorrent's Two Kademlia-based DHTs. Technical Report TR-07-04, Department of Computer Science, Rice University, 2007.

[10] M. Naor and U. Wieder. Novel Architectures for P2P Applications: the Continuous-Discrete Approach. *ACM Transactions on Algorithms (TALG)*, 3(3):34–es, 2007.

[11] The „king" data set. `http://pdos.csail.mit.edu/p2psim/kingdata/`.