Collaborative Anomaly-Based Attack Detection

Thomas Gamer[1], Michael Scharf[1], and Marcus Schöller[2]

[1] Institut für Telematik, Universität Karlsruhe (TH), Germany
[2] Computing Department, Lancaster University, UK

Abstract. Today networks suffer from various challenges like distributed denial of service attacks or worms. Multiple different anomaly-based detection systems try to detect and counter such challenges. Anomaly-based systems, however, often show high false negative rates. One reason for this is that detection systems work as single instances that base their decisions on local knowledge only.

In this paper we propose a collaboration of neighboring detection systems that enables receiving systems to search specifically for that attack which might have been missed by using local knowledge only. Once such attack information is received a decision process has to determine if a search for this attack should be started. The design of our system is based on several principles which guide this decision process. Finally, the attack information will be forwarded to the next neighbors increasing the area of collaborating systems.

1 Introduction

Today, the Internet is used by companies frequently since it simplifies daily work, speeds up communication, and saves money. But the more popular the Internet gets the more it suffers from various challenges that appear with increasing frequency. Challenges currently threatening networks include attacks like denial-of-service (DDoS) attacks [1] and worm propagations [2] besides others. DDoS attacks, for example, aim to overload a victim's resources like link capacity or memory by flooding the system with more traffic than it can process. The attack traffic is generated by many slave systems called zombies which an attacker has compromised prior to the attack. The attacker only has to coordinate all these slave systems to start the attack nearly at the same time. A DDoS attack is a distributed attack where zombies are located in various domains of the Internet. Every zombie generates only a small bandwidth attack flow to prevent detection of the zombies. This traffic runs on different routes through the Internet to the victim aggregating at intermediate systems the nearer it gets to the victim (see figure 1). Keeping the zombie systems undetected enables the reuse of them for a later attack.

Current efforts aim at detecting attack flows and blocking them to prevent them from reaching the victim. The detection systems are usually deployed in the access networks. Because of the small bandwidth at the zombies' detection systems close to them can hardly detect the attack flows and therefore are not able to block them in most cases. A detection close to the victim can still protect the victim's system against an attack but only if the detection system itself is not overwhelmed by the attack. There are two possibilities to bridge the gap between the point in the network where you want to block attack traffic and the place where you can detect it: on the one hand attack specific

D. Hutchison and R.H. Katz (Eds.): IWSOS 2007, LNCS 4725, pp. 280–287, 2007.

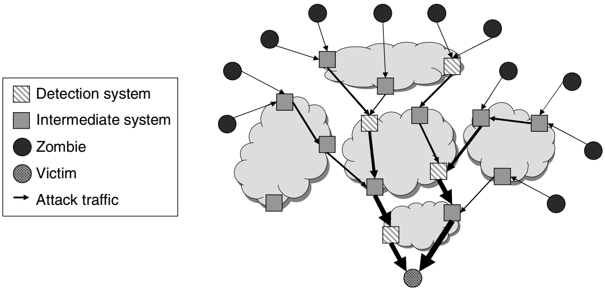

Fig. 1. Traffic aggregation during a DDoS attack

information can be exchanged in order to enable the systems close to the zombies to detect the attack traffic [3,4,5]. On the other hand, detection systems can be pushed deeper into the network if they pay special attention to the resource constraints there. The deeper the detection system is placed in the network the more attack traffic can have aggregated. A detection of this aggregated traffic is easier than a detection of the low bandwidth traffic close to a zombie.

In [6] we proposed an anomaly-based detection system that can be deployed within the network in order to detect adverse events as early as possible. This ensures a fast reaction and therefore, an effective protection of the victim. Furthermore, the network itself and its resources can be better protected by such a detection system since detection takes place on routers within the network instead of at the victim's edge. We identified two disadvantages of such an approach: unfavorable aggregation of attack traffic and upstream activated countermeasures. If attack flows aggregate only close to the victim our approach suffers from the same effects as deployment of detection systems close to the edge of the network. An early detection is unlikely. Furthermore, if one of our systems detects an attack flow and starts blocking it detection systems downstream will experience attack traffic with a smaller bandwidth. Again, this decreases the likelihood of detection. As a consequence anomaly-based detection systems that decide on the existence of an adverse event based on local knowledge only show false negative errors, i.e., some adverse events are not detected. These detection problems could be solved or at least diminished if the knowledge of multiple detection systems can be shared and thereby detection systems collaborate in a self-organized manner. Therefore, we propose to combine in-network deployment of detection systems with information exchange in order to build an effective system that detects and prohibits adverse events.

By combining local knowledge and remote information we built a system that organizes itself and that enables each node to autonomously decide if the suspicious traffic is an adverse event or just a legal traffic anomaly. Thereby, a coordinated collaboration of independent systems is achieved. Furthermore, the detection of adverse events is improved by the fact that a false negative of one detection system is compensated by exchanging information between neighborhood detection systems. Thus, the probability of detecting adverse events increases. But such an information exchange comes with risks, too. First, an attacker can try to launch an attack on the detection system itself by injecting bogus information. Second, the scalability of the overall system must be guaranteed so that the exchange of information does not overload the detection systems.

The rest of this paper is structured as follows: after a review of related work we detail the metric-based decision algorithm in section 2. This algorithm is applied for reasonably reacting on the reception of an attack report. Additionally, we address the issues of describing adverse events detected by a detection system and of discovering neighborhood detection systems. Thereby, we consider security aspects like authentication, trust, or integrity, too. Finally, section 3 gives a short summary about this paper and mentions future work.

1.1 Related Work

Today there is a great research effort in intrusion detection systems. Common intrusion detection techniques can be divided into misuse and anomaly detection [7]. Misuse detection, e.g. snort [8], relies on signatures that define byte patterns of known attack packets. They provide a low false positive rate but are not able to detect previously unknown or protocol-conform attacks. Anomaly detection systems like NSOM [9] or NETAD [10], on the other side, monitor network traffic and search for anomalous behavior, e.g. by applying neural networks or threshold-based mechanisms. Thus, they are able to detect previously unknown attacks at the expense of a higher false positive and false negative rate. These and other similar intrusion detection systems [11,12,13] use local knowledge to form a local opinion on an observed traffic flow. So they consider a detection system just as a single instance. and do not use the possibility of exchanging information with other detection systems.

There also exist other approaches like Emerald [14], INDRA [15] or CITRA [16] that deal with the coordination of distributed detection systems and use its advantages. The INDRA-Project proposes the cooperation of different detection systems through a subscription-based group communication with a peer-to-peer architecture. CITRA-devices report detected attacks to a central Discovery Coordinator that coordinates countermeasures based on a complete view on the network. Such frameworks and infrastructures are able to detect domain-wide threats and thus, could improve their detection reliability. But they have to deal with higher communication efforts and close trust relationships between the involved entities in order to prevent an attacker from abusing the detection system. Additionally, the communication of these approaches often is based on a central control entity.

At last, there exist several other approaches to cope with adverse events like DDoS attacks. Traceback techniques like Itrace [3] or SPIE [4] allow to identify the origin of an attack even in case that spoofed source addresses are used by zombies. The Pushback [5] mechanism examines congestion situations as an indication for a DDoS attack and reacts to it with a request to preceding routers to initiate rate limiters. This could defuse the congestion situation at the victim.

Our approach has the advantage that it can detect and react to adverse events during the build-up of the event. Furthermore, different detection systems could inform each other about their recognitions and thus, enhance their detection reliability without need for a central control entity. The fact that detection systems need not trust each other absolutely reduces the requirements for security aspects.

2 Design

The self-organizing extension for our detection system can be separated into three parts: first of all, a neighbor discovery has to be performed. Afterwards, a system communicates its information to a neighborhood detection system. The receiving detection system then autonomously has to decide on how to react to this information. Therefore, we propose a metric-based decision algorithm. Third, the adverse event detected by the sending system has to be described in a comprehensible way, so that all detection systems involved in the coordination are able to process a message correctly.

2.1 Neighbor Discovery

In section 1 we mentioned that a self-organizing collaboration of detection systems is necessary to improve an anomaly-based detection of adverse events. Such a collaboration is based on an exchange of information between neighborhood detection systems. Therefore, a detection system must be able to discover neighborhood detection systems in order to communicate the locally gathered information. Having discovered a neighborhood detection system a communication channel can be established that must have certain characteristics specified by the sender, e.g. reliable message transfer. Before exchanging the available information a security association – authentication of communicating systems or data integrity – should be established in order to prevent an exchange of wrong or forged information and DoS attacks against the detection system itself.

We believe that detection systems are sparsely distributed in the Internet and thus, two detection systems are rarely connected directly with each other. Furthermore, a detection system in our opinion has no detailed knowledge about the whole topology and the distribution of all detection systems in the Internet since only a minimal amount of long-lived state information should be maintained. Additionally, a neighbor discovery mechanism has to regard dynamics of the Internet, e.g. changing of routes or dynamically activated detection systems which cause new neighbor relationships. Thus, we need a mechanism that is able to locate neighborhood detection systems on demand. In order to discover neighborhood detection systems multiple methods are possible: expanding ring search, path-coupled mechanisms, overlay networks, and others.

With an expanding ring search [17], for example, the metric that defines the notion *neighborhood detection system* in most cases is a maximal hop count. A problem of most expanding ring search mechanisms currently deployed is that no security is provided. Another approach that provides discovery of neighborhood detection systems are path-coupled mechanisms as provided by the signaling framework NSIS [18]. This framework additionally enables a sender to specify communication requirements, e.g. reliable and confidential message exchange.

2.2 Metric-Based Decision

In our previous work as well as in many related approaches an anomaly-based detection system represents an independent and autonomous network device that detects adverse events based on local measurement. We propose to reduce the false negative rate of such an anomaly-based detection system by a self-organizing exchange of information about adverse events already detected elsewhere in the network. The system we have developed is based on the following design principles:

Verify received information. *No system should commit itself to a tight trust relationship with other detection systems but rather rely on its own recognitions.*

This ensures robustness against bogus information as well as message injection attacks due to local verification of received information. Furthermore, a close trust relationship between detection systems would constrain flexibility in dynamic environments and cause an increased overhead.

Consider current workload. *The available resources of the detection system limit the number of parallel executed detection threads.*

If the detection system is heavily loaded, i.e. it already does multiple fine-grained detections in parallel, it should reject an additional parameterized detection to prevent overload situations.

Rate report granularity. *The more fine-grained the data in the attack report is the fewer resources must be spent on its verification.*

If only few characteristics of the adverse event are known, i.e. the anomaly description is rather coarse-grained, the receiver must apply some stages of refinement. Thus, the verification of received information may waste valuable free resources.

Count duplicates. *The more systems report an ongoing attack the higher is the likelihood to detect the attack locally, too.*

If the same adverse event is reported by different detection systems the importance of the information increases dependent on the number of duplicate messages.

Check significance. *The traffic volume of the detected attack must be compared to the overall traffic on the detection system.*

If the sender of a message, compared to the receiver, only scans a rather small amount of traffic for anomalies the adverse event he reports may be negligible for the receiver. The message of a detection system that is located at the edge of the network and processes an average traffic amount of 100 Mbit/s, for example, is nonsignificant for an in-network detection system that processes 5 Gbit/s. But the information about a detected attack of this in-network system is of great significance to the edge system. If, however, multiple detection systems at the edge report the same attack, the significance of this information increases due to the principle *count duplicates*.

Measure distance. *The farther apart two neighborhood detection systems are the more likely the attack will be detected at the system receiving an attack report.*

The distance between two neighborhood detection systems in regard to IP hops recommends a parameterized detection since the probability that attack traffic aggregates between these system is the higher the longer the distance between the communicating detection systems is. This also increases the probability that a receiving detection system is able to detect the reported adverse event even if preceding systems apply countermeasures since attack traffic may be present again due to aggregation in intermediate systems.

Use verification failure history. *An attack report that has been checked unsuccessfully by the predecessors is not likely to be detected.*

If the message has passed several detection systems that scanned for the adverse event described without detecting it the message becomes less important. Additionally, the message is discarded after a maximum number of failed verifications in order to keep communication localized.

Use verification history. *If an attack report has been forwarded unchecked by the predecessors the distance to the detection instance which verified the adverse event last must be regarded.*

This ensures that communication of information about an adverse event does not run endlessly without being verified by a neighborhood detection system. Therefore, importance of a message grows the more often a verification is refused by a receiving detection system. In combination with the parameter verification failure history this parameter guarantees that a verification is done once in a while and thus, communication ends after a certain time if the adverse event reported cannot be verified at multiple systems.

Having considered all aspects described previously the detection system receiving an attack report should react in the following way:

- The system has to check if it has already detected the reported adverse event on its own. If so it can silently discard the message because it has informed its neighborhood systems before.
- If the system decides to start an anomaly detection that is parameterized by the message content and the verification succeeds it should start appropriate countermeasures. Furthermore, it communicates its own recognitions to a neighborhood detection system. If the verification, however, fails it updates the received message with its local knowledge, e.g. the verification failure history, and then forwards the message. Additionally, the content of the message is stored for a certain time in order to detect duplicate messages in the future, i.e., a soft-state approach is applied.
- If the system decides not to start an anomaly detection and the adverse event reported is not yet known to the receiving system it updates the received message with its local knowledge, e.g. the verification history, and then forwards the message. The content of the message is stored in order to detect duplicate messages.
- If the system receives a duplicate message the action depends on former decisions. If the message reports an adverse event the receiving system did not verify before it must reconsider its decision. In case that the parameters now recommend a verification and this verification is successful the system communicates its new knowledge to a neighborhood system. Otherwise – if the verification fails or if a verification was already done earlier – the message is discarded without doing something since it was forwarded before and no new knowledge is available.

In order to achieve the behavior described above a receiver needs not only the description of the adverse event but some additional information: current workload, report granularity and the number of duplicate messages can be obtained based on local knowledge. The distance between neighborhood detection systems has to be obtained by neighbor discovery (see section 2.1). Other parameters like significance of the sending detection system, verification history and verification failure history must be transmitted in addition to the description of the adverse event.

2.3 Description of Adverse Events

Having received an attack report the detection system may decide, based on the metric-based decision described in the previous section, to start a fine-grained detection itself if it is able to interpret the message correctly. At this point it must be considered that detection systems of different domains may scan for different protocol anomalies, i.e., not all systems necessarily must know the same anomalies due to different local policies or knowledge. Thus, a message possibly contains information the receiver cannot understand. Therefore, a generic and extensible message format, e.g. based on a type-length-value (TLV) structure, must be used for description of detected anomalies. In this case different data records are differentiated by the type field. The length field indicates the byte length of the following data field and thereby, enables a system to skip unknown data records. Thus, a receiving system extracts only that information it is able to understand and ignores unknown data records. Another approach that enables a flexible and extensible description of anomalies is the usage of a structured description language like XML [19].

3 Summary and Outlook

In this paper we presented a collaboration of anomaly-based detection systems which use local knowledge and measurements and combine them with remote information received by their neighbors. After establishing communication channels with detection systems in the neighborhood attack information can be freely exchanged. The reaction to such a message is determined by a metric-based decision process. The design of this decision process is guided by a set of principles. Depending on available resources, trust, and history of the message the system starts a search for the described attack locally to verify the message, forwards the message unprocessed or drops it silently.

We additionally implemented the collaborative attack detection proposed in this paper. Therefore, we extended our detection system [6] with a decision and a communication engine. Neighbor detection is implemented as an external process in order to allow for transparent addition of new neighbor detection and security mechanisms. Some first evaluations in different usage scenarios show how the decision process derives a reaction to an attack information received from a neighborhood detection system. But these evaluations representing small simulated environments only describe the microscopic behavior of such a collaboration of detection systems. Thus, future work has to address the question how the collaboration behaves in a macroscopic scenario. We plan to implement our anomaly-based detection system as well as the extension we proposed in a network simulator which allows a simulation of more and larger networks. This macroscopic analysis enables an examination how each single design principle influences the metric-based decision algorithm and thus, enables the choice of an optimal decision function. Additionally, it will show which effect the collaboration has on the false negative rate. Finally, different neighbor discovery mechanisms have to be evaluated and some work has to be done regarding the description of adverse events that are only known domain-wide and have to be communicated to a detection system outside the domain of the sending system.

References

1. Hussain, A., Heidemann, J., Papadopoulos, C.: A framework for classifying denial of service attacks-extended. Technical report, USC/Information Sciences Institute (2003)
2. Shannon, C., Moore, D.: The spread of the witty worm. IEEE Security and Privacy 2(4), 46–50 (2004)
3. Bellovin, S., Leech, M., Taylor, T.: Icmp traceback messages. Internet draft, Internet Engineering Task Force, Work in Progress (2003)
4. Snoeren, A.C.: Hash-based IP traceback. In: SIGCOMM, pp. 3–14 (2001)
5. Mahajan, R., Bellovin, S.M., Floyd, S., Ioannidis, J., Paxson, V., Shenker, S.: Controlling high bandwidth aggregates in the network. SIGCOMM Computer Communication Review 32(3), 62–73 (2002)
6. Gamer, T.: A system for in-network anomaly detection. In: Kommunikation in Verteilten Systemen, February 2007, pp. 275–282. Springer, Heidelberg (2007)
7. Kumar, S.: Classification and Detection of Computer Intrusions. PhD thesis, Purdue University (1995)
8. Roesch, M.: Snort, intrusion detection system (1999), `http://www.snort.org`
9. Labib, K., Vemuri, V.R.: NSOM: A tool to detect denial of service attacks using self-organizing maps (2004)
10. Mahoney, M.V.: Network traffic anomaly detection based on packet bytes. In: Proceedings of the ACM symposium on Applied computing (SAC), pp. 346–350. ACM Press, New York (2003)
11. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), pp. 219–230 (2004)
12. Paxson, V.: Bro: a system for detecting network intruders in real-time. Compututer Networks 31(23-24), 2435–2463 (1999)
13. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004)
14. Porras, P.A., Neumann, P.G.: EMERALD: Event monitoring enabling responses to anomalous live disturbances. In: Proc. 20th NIST-NCSC National Information Systems Security Conference, October 1997, pp. 353–365 (1997)
15. Janakiraman, R., Waldvogel, M., Zhang, Q.: Indra: A peer-to-peer approach to network intrusion detection and prevention. In: Proceedings of 12th IEEE Workshops on Enabling Technologies, Infrastructure for Collaborative Enterprises (WETICE), June 2003, pp. 226–231. IEEE Computer Society Press, Los Alamitos (2003)
16. Schnackenberg, D., Holliday, H., Smith, R., Djahandari, K., Sterne, D.: Cooperative intrusion traceback and response architecture (CITRA). In: Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX), June 2001, pp. 56–68 (2001)
17. Boggs, D.R.: Internet Broadcasting. PhD thesis, Stanford University (1982)
18. Hancock, R., Karagiannis, G., Loughney, J., den Bosch, S.V.: Next steps in signaling (NSIS): Framework. RFC 4080, Internet Engineering Task Force (2005)
19. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F., Cowan, J.: Xml 1.1, 2nd edn. W3C recommendation, W3C (2006)