COPnets: Cooperative Internet Security Systems

Semir Daskapan, Delft University of Technology, Delft, Netherlands, s.daskapan@tudelft.nl

Abstract

In this paper we introduce the concept of COPnets. A COPnet refers to a network of security systems (agents) that cooperate to form a collaborative defence system. COPnets give answer to the growing complexity and sophistication of attacks, like DDOS and other botnets. In a COPnet each existing security system is attributed with simple additional tasks to communicate with other allied security systems. The response to an attack is as such a collaborative effort. We will illustrate our idea by depicting a firewall COPnet with pseudo code.

1. Introduction

Many organizations are forced to widen their scope and to conduct their business on a global scale in order to keep up with the competition. At the same time they are enlarging their networks and information systems to support their new business. It becomes more and more difficult to manage the vulnerabilities of those large connected information systems. The traditional reactive behavior of security systems is expected not to be able to keep up with the complexity present in the new interconnected information systems and the emerging sophisticated collaborative attacks [1]. Instead of having a few security systems, though well-equipped, waiting passively and being surprised by each new attack or random failure, a new way of thinking about how to be security threats one step ahead is needed. Such a new way of thinking should be inspired on complex adaptive systems (CAS) and involves the idea of solving complexity by the individual constituents of the complex system itself. The constituents are the several individual security systems on the Internet that cooperate to achieve an integrated defense system. In other works there are examples of such concepts that show the strength of such a new approach [2,3,4,5].Such a web of security systems is more effective since it is able to memorize, recognize and to adapt to new types of threats. It is also expected to be more effective since each new threat is immediately dealt with by the system itself. Such a system is also more efficient as fewer resources are needed compared to traditional methods. Traditional methods require a new module or update for each new threat (IDS, virus scanner, firewall, etc), whereas with self-organization the integrated system reconfigures itself.

In this paper our aim is to emphasize the benefit of such an approach compared to traditional singular solutions. Instead of leaving this new approach to the happy few, like the references above, the goal of this paper is to elevate this approach into a new paradigm and to guide future designers with a methodology based on CAS. Cooperative defense systems based on self-organization are as such proposed as a way of thinking about security in this paper.

In section 2 we will elaborate on the upcoming threats. In section 3 we will introduce our solution methodology. In section 4 we will present an application of our proposed methodology on firewalls.

2. Threats

Malicious security attacks to information infrastructures are typically those attacks that use networks to amplify the strength of the attack, like worms and denial-of-service (DoS) or worse distributed denial-of-service (DDoS) attacks. Despite the many proposed methods and techniques for specific situations, it is commonly accepted that DDoS attacks are hard to prevent or to repulse, while their occurrences are increasing [6,7]. Hostile applications are also a direct consequence of the mobile agent paradigm, since attacks can be directed now from multiple platforms, resulting in a distributed DDoS attack [8]. The concept of mobile intelligent agents has the appearance of being a new idea, although it can be traced back to the 70's [9]. It is just with the Internet as an open network, that mobile agents attracted great interest. New possibilities to relieve mankind from his labour-intensive tasks emerge with this paradigm as these robot programs can be considered to be the revival of the master-slave concept. Things that travel everywhere, doing things at the other side of the world and come back with some virtual jewels, could not sound better for researchers…. and unfortunately also for hackers. It is the same concept that creates also new dilemmas in security: distributed DoS could evolve into volatile DoS (VDoS). In the latter, besides being distributed the malicious agents in the botnets travel also at random times between their infected hosts.

An appropriate answer to these types of attacks is not at hand. As long as information exists security has worried mankind over the history. Mankind has invented and is still inventing new ways to secure its assets. This ever-evolving process has led to many security protocols and mechanisms. It is infeasible to give an in depth overview of them all. The fact that there are so many and more are proposed each year indicates already the unsatisfactory results of security standards. Obviously, there is no silver bullet security concept that will repel any possible attack. It is for this reason that our focus in this research is not on preventing specific attacks by developing a new more refined security method, but *rather on a security methodology*.

3. Demarcating the solution domain

Our answer to the growing complexity and sophistication of attacks, like DDoS or VDoS, and other botnets are COPnets. A COPnet refers to a network of security systems (agents) that cooperate to form a collaborative defence system. Another way of looking at this answer is that we oppose the threat with the similar weapon: complexity. Complexity, as one of the culprits that increases vulnerability and which is usually perceived as a negative tendency since it decreases manageability, can thus be positively exploited. In a COPnet each security agent is attributed with simple additional tasks to communicate and share information with other agents in the COPnet. The aimed response to an attack is as such a collaborative effort. In order not to leave the design of a COPnet to an individual effort we propose complex adaptive system (CAS) as the underlying design methodology. We consider a complex adaptive system (CAS) as a collection of interdependent rule-following agents with complex interactions resulting in system-wide patterns across the group. A characteristic is that no agent needs to be aware of the existence of the total space. Each agent knows at most what kind of capabilities it has and how it can look for relevant information in the environment.

Our positive perception of complexity, in which complexity is rather exploited to solve problems, is supported by other groups like the Santa Fe Institute [10, 11,12,13]. The reason for delimiting the scope of the research to CAS is that despite their complexity they might offer opportunities for intelligent solutions. *A complex adaptive system contains a large set of objects that interact with each other and with an external environment according to simple rules to produce overall patterns that are significantly more complex than the behaviours of the individual objects of the system.* The objects of such a system are usually called agents. Self-organization happens without any agent being in charge or consciously planning it. The agents of a complex adaptive system may follow simple rules and yet produce complex patterns. The termite, the beehive, stock market and self-healing bacteria model are examples. In the stock market all the individuals behave rather simple: sell if price exceeds some threshold of value share and buy if price reaches lowest threshold. Darwin's theory of evolution describes another complex adaptive system, in which the individual organisms compete for scarce biological resources. Computer simulations can demonstrate how separate agents independently follow rules of behaviour and collectively generate patterns at a group level.

The application of the CAS methodology happens in three steps. First, the computing entities that are relevant to the problem at hand will be identified as agents. Second, a distinction will be made between the several roles of the agents, such that the system can be subdivided by a few clusters of equal agents. Each cluster consists then of agents with equal roles, assumptions and tasks. Then, simple assumptions will be made about the behavior and knowledge of the agents and the cluster they reside in. Fourth, relatively simple similar tasks will be assigned to all agents of the same role in that system.

4. A Case: Firewall COPnet

To illustrate our claim we will depict a COPnet of firewalls. In this simple case the firewalls cooperate to improve their resilience when they are suffering from DDoS attacks. Other improvements based on this cooperation are possible, like refining each others access control policy files by sharing updated information about experienced attacks.

4.1 Approach to the specific problem

Many security systems that are placed at the front-end of a private (business) network, such as authentication systems and firewalls, have to endure many frontal attacks. The number of attacks on these systems increases and the types of attack are increasingly sophisticated. Instead of classic (multiple) unilateral attacks more distributed and adaptive attacks become more common such as (distributed) denial of service attacks. Since security of a closed information network depends on those nodes, it is therefore essential to protect these security systems. Designers employ common redundancy techniques to assure availability of the security service [14,15]. Since redundancy of the security systems by dedicated hardware is limited to the fixed number of replicas, any front-end security system can be compromised by a sufficient number of consecutive attacks. Therefore we need a more resilient solution.

The objective in this case is as such to improve the resilience of a firewall as a front end security system. The approach is to include it in a network of firewalls and applying a CAS methodology. A firewall and all computing entities that rely on such a firewall are then considered as agents in a CAS. Consequently, with the right assumptions and the right instruction set, this CAS will experience spontaneous self-organization and self-healment when it is under attack. Our strategy is to let the CAS- firewalls support each other autonomously and securely. Strategy is that when one of them is under attack and becomes unreachable for an unacceptable period of time, one of the others will continue his access control services. In fig. 1 we have depicted a COPnet of four firewalls. Each of them protects an internal (business) network of client computers. Only one network is not allied to the COPnet in this example; this network is as such considered to be malicious. When DDOS from this malicious network are directed to the internal network of AF4 the clients of AF4 are temporarily protected by one of the other remaining firewalls AF2, AF3 or AF1.

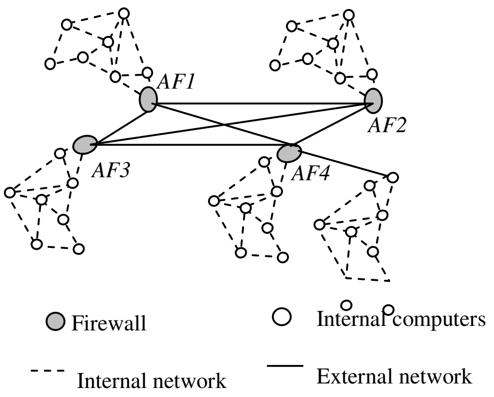

Fig 1. A simple COPnet of four firewalls AF1 to AF4

4.2 Firewalls as a CAS

In order to characterize a firewall as a CAS we need to define the assumptions and the instruction set of each agent. A firewall is a computer system that takes care of access control to other computers in a network. It analyse incoming data traffic and based on an access policy file it determines to grant access or not to the internal network.

Assumptions:

- A computing entity is an agent with identity *id*.
- An agent is either a client agent (*AC*) or a firewall agent (*AF*).
- An AF can deliver access control services to *ACs*.
- An *AC* has an individual access control policy file *ACP*
- An AF maintains a tuple *t* of the CPs of the *ACs*.
- A client agent is called internal when it is protected the access control services of an AF else it is external.
- An agent sees a limited set of agents in the total space.
- An AF is connected to and has with two ore more other AFs a trust relationship.
- An agent is not aware of the consequences of his actions for the overall system.
- An agent perceives only availability and trustworthiness of other agents.
- An agent has a memory system, i.e. it can store basic information about other agents.
- An agent is able to send and receive messages.
- An agent is either triggered by an explicit message or by a faulty expected message from other agents.
- An external trust authority (TA) is ad hoc present.
- An AF has a signed certificate to verify their asymmetric key pair and their trustworthiness [xx16].
- Each AF is frequently suffering from denial of service attacks, but not all at the same time.

The next autonomous actions for each agent separately in the CAS enable the firewall service to be resilient by continuously hopping away from the AF under attack.

Instruction set

```
0. ACs: subscribe by sending their ACPs and ids to one of the AFs (using AFs public key) based on his trustworthiness.
1. AF: sends to and receives frequently from other AFs updates of experienced attacks and other anomalies
2. AF: gives consent to incoming requests to access the internal ACs based on their ACPs.
3. AF: frequently creates a tuple t of those ACPs and a tuple SL of his ranked Preferred Successor AFs (PSAFs).
4. AF: frequently sends shares of t [17] and replicates of SL to all PSAFs.
5. PSAFs: check frequently availability of suffering AF.
  If AF is insufficiently available do:
   6.PAFs: send declaration of death of suffering AF to each other.
   If majority of PSAFs agrees on death AF:
   7.Majority of PSAFs: send their t to first ranked PSAF on SL (FAFs).
    If FAF is sufficiently available do:
      8. FAF: reconstruct from several t the ACPs of the clients.
    Else Go to 6 with AF = FAF
9. Go to 2 with AF = successor AF.
```

Note on security: besides the asymmetric key pair the firewalls also share symmetric keys with other allied firewalls and their clients via Kerberos [18]. All the messages are encrypted and the message authentication code, timestamp, id's are also part of the message.

4.3 Interpretation

The resilience of front end security systems of collaborating security systems is improved by applying a CAS approach. This instruction set takes care of continuously replicating the tuple of access control files and letting it to resurrect on another AF each time this service is under attack, for example, due to a DDOS attack. The successor AF functions then as a temporary carriage, i.e. execution platform, until he too is attacked and so on. This mechanism lets therefore the access control service to be independent from the resources of a particular AF. The clients are not necessarily aware of this host transition, since the trust relationship is based on the shared secret s and not on the identity of the AF

5. Discussions and future work

Given the increased complexity of attacks we expect that solutions to protect information infrastructures cannot be found in the traditional singular security concepts. Our aim in this paper was therefore to propose a new way of thinking about security development and to provide designers of security systems a methodology. As such, we proposed the application of complex adaptive systems theory to design cooperative internet security systems, i.e. COPnets. In COPnets several internet security systems collaborate to repel sophisticated attacks at one or more of them.

In future work, we aim at further developing and testing the proposed firewall COPnet and a new COPnet of intrusion detection systems. We will empirically show that they are more capable to withstand security attacks than singular security systems

6. References

[1] Sachin Katti, Balachander Krishnamurthy, and Dina Katabi, "Collaborating Against Common Enemies", *ACM Internet Measurement Conference*, USA, 2005.

[2] Benny Wong, Michael E. Locasto, Angelos D. Keromytis, “PalProtect: A Collaborative Security Approach to Comment Spam”, *IEEE Workshop on Information Assurance,* NY, 2006.

[3] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, Towards Collaborative Security and P2P Intrusion Detection, *IEEE Workshop on Information Assurance*, USA, 2005.

[4] Hai Jin, Feng Xian, Zongfen Han, Shengli Li, "A Distributed Dynamic Firewall Architecture with Mobile Agents and KeyNote Trust Management System", *4th International Conference on Information and Communications Security*, Singapore, LNCS 2513, Springer-Verlag, 2002.

[5] Semir Daskapan, Willem G. Vree and Rene W. Wagenaar, Emergent information security in critical infrastructures, *Int. Journal of Critical Infrastructures*, Inderscience 2(2/3), pp. 247-260, 2006.

[6] Wang, C. *A Security Architecture for Survivability Mechanisms*, University of Virginia, 2000.

[7] Chang, R. K. C. "Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial." *IEEE communications magazine* 40(2), pp. 42-51, 2002.

[8] Ferber, J. *Multi-agent systems*, Addison-Wesley, 1999.

[9] Hewitt, C. "Viewing control structures as patterns of parsing messages." *Artificial Intelligence*, 8(3) , pp. 323-364, 1977.

[10] Langton, C. G., C. Taylor, et al., Eds. Santa Fe Institute Studies in the Sciences of Complexity, *Artificial life 2*, Addison-Wesely, 1992.

[11] Dooley, K., T. Johnson, et al, "TQM , Chaos and Complexity", *Human Systems Management* 14(4), pp. 1-16, 1995.

[12] Dooley, K. "A Complex Adaptive Systems Model of Organization Change." *Nonlinear Dynamics, Psychology and Life Science* 1(1), pp. 69-97, 1997.

[13] Fromm, J. *The Emergence of Complexity*, Kassel university press, 2004.

[14] A. E. Barbour and A. S. Wojcik , "A General Constructive Approach to Fault-Tolerant Design Using Redundancy," *IEEE Transactions on Computers* 38 (1), pp. 15-29, 1989.

[15] M. A. Hiltunen, R.D. Schlichting, et al.,"Building Survivable Services Using Redundancy and Adaptation," *IEEE Transactions on Computers* 52(2), pp.181-194, 2003.

[16] Semir Daskapan, Ana Cristina Costa, Reengineering Trust in Global Information Systems, in Kautonen, T. and H. Karjaluoto (Eds), *Trust and New Technologies: Marketing and Management on the Internet and Mobile Media*. Cheltenham, UK and Lyme, US: Edward Elgar, 2008.

[17] A. Shamir, "How to Share a Secret," Communications of the ACM 22(11), pp. 612-613, 1979.

[18] B. Clifford Neuman and Theodore Ts'o, Kerberos: An Authentication Service for Computer Networks, IEEE Communications 32(9), pp. 33-38, 1994.