Edith Cowan University
Research Online

Australian Information Warfare and Security Conference

Security Research Institute Conferences

2011

Designing cyber warfare information infrastructure resilience

Semir Daskapan
Delft University of Technology, Delft, The Netherlands

Jan Van der Berg
Delft University of Technology, Delft, The Netherlands

Originally published in the proceedings of the 12th Australian Information Warfare and Security Conference, Edith Cowan University, Perth Western Australia, 5th -7th December, 2011
This Conference Proceeding is posted at Research Online.
http://ro.ecu.edu.au/isw/43

DESIGNING CYBER WARFARE INFORMATION INFRASTRUCTURE RESILIENCE

Semir Daskapan and Jan van den Berg
Delft University of Technology, Delft, The Netherlands
S.Daskapan@tudelft.nl; J.vandenBerg@tudelft.nl

Abstract

Due to many cyber attacks in the last years, governments are realizing how vulnerable they have become should there be a break out of a cyberwar. This urged them to establish a cyber warfare information infrastructure in a short time. However, this cyber warfare information infrastructure relies heavily on public infrastructures, like electricity and the Internet, which will be most likely targeted themselves. Therefore, a cyber warfare information infrastructure is by definition a vulnerable infrastructure that needs to be secured against attacks and made resilient. In this paper, we provide a method inspired by the theory of Complex Adaptive Systems to improve the resilience of cyber warfare information infrastructures. This method is applied on one specific security system as a showcase, namely, the intrusion detection system.

Keywords

Information Security design, Information Warfare, Critical Infrastructures, Resilience, Complex Adaptive systems, Intrusion Detection Systems, Self-organization

INTRODUCTION

More and more countries are facing security problems due to security attacks on their critical infrastructures, as can be illustrated by recent incidents like Diginotar in The Netherlands, Titan Rain in de US, and Stuxnet in Iran. Those cyber terrorist or cyber war attacks differ from traditional security attacks, since they do not aim at individuals or organizations. They aim at destabilizing or paralyzing the critical infrastructures of a society. It is very likely that they are instigated or at least supported by governments of 'malicious states' (Clarke, 2010).

Critical infrastructures are considered as the valuable assets of each state that should be protected against external threats (Pye, 2009). We define a critical infrastructure as the chain of subsystems, also called components, that facilitate the constant flow of information, matter, people, or energy in a society, and its failure might cause high direct and/or indirect social, economic and/or ecological damage. Some infrastructures are considered more critical than others. In (Moteff, 2003) the criteria for being critical for each infrastructure are given. Among all the types of infrastructures, energy, telecommunications information and banking infrastructures are considered to have most impact. One of the reasons is interdependency. Most of the physical infrastructures rely for example on telecommunications information infrastructures, also called critical information infrastructures (CIIs), for their operation and control. Road traffic systems, for instance, are monitored, controlled and operated from remote control and command centers. Telecommunications information infrastructure on its turn relies on the energy (electricity) infrastructure, and vice versa. Since society depends on telecommunications information infrastructure to manage other infrastructures, society is becoming more vulnerable due to their increased complexity and accessibility via the Internet. This is not just a theoretical proposition: the cyber attacks in the last years have clearly revealed this inconvenient truth about the vulnerabilities of our (critical) infrastructures.

Especially that a new (to be) established cyber warfare infrastructure will depend on those public vulnerable telecommunications information infrastructure is a painful observation (Liles, 2008). Governments are not (well) prepared to such a new type of war. They seem to lack the knowledge and experience to develop a secure and resilient cyber warfare information infrastructure (CWII). Although they are still not able to deal with it, they are becoming aware of this risk. Unlike the many IT (security) standards for the industry, there is no standard, method or guideline for governments to develop a CWII (Hathaway, 2009).

In this paper we take a first step towards designing CWII's by proposing a specific, on the theory of Complex Adaptive Systems-based, method to improve one of the important properties of CWII's: *resilience,* i.e., the ability to reduce the magnitude and/or duration of disruptive events. A CWII that comprises many (defensive and offensive) security systems should meet the highest standards for resilience to withstand the most severe conditions of a (cyber)war. To show its working, the method proposed is being applied onto an intrusion detection system, one of the many critical subsystems in a CWII (Andress, 2011).

In section 2 we first discuss the Internet as an almost perfect case of an existing resilient CII and then, by generalizing from that example, introduce our new method. In section 3, the new method is illustrated by applying it on a fictitious case of ants. Then, in section 4 it is applied on one of the CWII security systems. In section 5, conclusions and future work are discussed.

DERIVING A DESIGN METHOD FOR INFORMATION INFRASTRUCTURES

Design principle: resilience by self-organization

A CWII consists of many systems that support the operational cyberwar defense and offense processes that are mentioned in (Andress, 2011) and (Janczewski, 2007). Besides their core software and hardware, a CWII has also to apply perimeter security technologies to protect the core systems from being confiscated. Many security systems that are placed at the front-end of a network, such as authentication systems and firewalls have to endure many frontal cyber attacks since they form the first line of defense. Other security systems, like intrusion detection systems, face the same after that the first line of defense has been penetrated. The back end systems like the SCADA systems, hacking systems, systems for surveillance, data mining & analysis and pattern matching are the final targets. Designers usually employ common redundancy techniques to assure availability of their critical systems (Hiltunen, 2003). However, any front-end security system can be compromised by a sufficient number of consecutive (for example denial of service or DOS) attacks, since the number of redundant front-end security systems is eventually limited, whereas the number of DOS attacks is practically infinite.

Thus, the emerging problem of increasing attacks on our critical infrastructures cannot be dealt with by traditional means like prevention and fortification only. But, besides this technical deficiency there is also a practical obstacle. The fact is that we live in an open society with border crossing communication infrastructures that are accessible for practically everyone. Simply shutting them down or fortifying them directly affects the main arteries of our economy. Our approach is therefore to accept the fact that these types of attacks will occur, of which some of them will manage to penetrate our CII, and to prepare ourselves for this occasion such that when it happens our CII must be capable of surviving the blow. This is achievable if we consider a CII as complex adaptive system (CAS) (Langton, 1992) (Dooley, 1995) (Dooley, 1997), in which their components function as agents that are able to cooperate together and show self-organized behavior.

Self-organization as a mean has also been promoted by others in efforts to prevent, defend against, and respond to cyber attacks (Lawson, 2011) (Eoyang, 1999). Any distortion in the infrastructure could then be cleared, somehow, autonomously by the cooperating agents. Dependability solutions can then emerge bottom-up from the smaller constituents. We consider a complex adaptive system as a collection of interdependent rule-following agents with interactions resulting in system-wide patterns across the group. Properties of CAS are: emergent behavior, adaptation, specialization, dynamic change, decentralization and cooperation (Wilensky, 1999). The challenge is how to adopt this CAS approach to increase resilience of CWII's. To answer this question and find a method that can be applied on other CII, we look at an existing CII that already behaves as a CAS: the Internet.

A real case: Resilient Internet

The Internet is an ideal resilient CII, since it has all the ingredients to bypass any network disruption. The Internet consists of many so-called Autonomous Systems (AS). These autonomous systems are operated by a far smaller number of commercial Internet Service Providers (ISPs), non-commercial parties such as Universities and Internet Exchange Points. Connectivity between those ASs is the result of so-called peering or transit agreements, which are negotiated between ASs. Global end-to-end connectivity in the Internet is ultimately entirely dependent on these bi-lateral agreements. The ability to cope with such disruptions is considered to be high, because ASs can make new arrangements in case of planned outages of their neighbors and because routing protocols allow for "routing around the problem". This makes the Internet as an ideal resilient CII. It is as such a typical case to learn from and to derive a method by which other CII can be converted into a CAS – CII. In an earlier publication (Daskapan, 2008), we described the Internet at the level of ASs and the Border Gateway Protocol (Stewart, 1998) as it 'would' function in case of a 'pure' CAS. (The BGP is a routing protocol between ASs.) We noticed that ASs behave like autonomous agents that - based on simple assumptions and instructions while not being aware of the total Internet - are able to create together, as an emergent phenomenon of their activities, a resilient infrastructure. The observed characteristics and emergent outcomes inspired us to propose our (generalized) method in the following section.

Note that we said 'would' above, because currently the Internet is gradually loosing its CAS behaviour due to concentrated ISP policy restrictions, as a consequence of peering agreements. Any human involvement or disruption of the described agent behaviour above will lead to a less resilient Internet and thus, an unreliable Internet. The maximum involvement of governments of Egypt and Syria during the Arab spring has been proven this proposition: shut down of (a large part of) the Internet. However, this net-neutrality discussion is beyond the scope of this paper.

Proposed design method

Assume now that we have another CII than the Internet, possibly a CWII, which is not resilient yet and does not behave as a CAS. In order to make it resilient like the Internet this CII should adopt the same type of behavior as the Internet. The Internet is, as was stated earlier, a CAS-CII. So, any other CII that is going to be as resilient as the Internet must first be converted into a CAS. Converting this CII into a CAS gives a semi-closed system with cooperating nodes. From the Internet case we have learned and derived the following steps to improve resilience of a CII via a CAS: the reference to the Internet is in italic.

1. Define a semi-closed system by demarcating the CII boundaries. The components/nodes of a CII that are relevant to the vulnerability problem are the constituents of this system.
   In the Internet case, the CII boundaries are the Internet infrastructure at the level of ASs.
2. Define the problem or threat that must be solved
   In the Internet case it was lack of resilience.
3. Define the components of the CII that are relevant to the problem at hand as agents; the rest is put out of scope.
   In the Internet case the AS were the agents.
4. Identify the trigger that causes the targeted problem by simulation with a minimum number of these agents, say four. This subsystem is the "lab sample" or "lab-CAS" to experiment with.
5. Define the aimed end-state of the CAS.
   In the Internet case it was a resilient CII consisting of cooperating ASs.
6. Define the required behavior of the total CAS to achieve this end-state
   In the Internet case it was rerouting packages around a failing AS.
7. Identify (by trial and error) the type of activities that have to be performed by the agents to accomplish the aimed end-state.
   In the Internet all the AS perform the same type of activities according to the Border Gateway Protocol. So, there is only one type of activity.
8. Make a distinction between agents on the basis of their different instruction sets.
   In the internet case only one type of agent was at hand, i.e. the AS.
9. Transform the activities into instructions: define by trial and error the simplest instruction set for each lab-agent, such that when executing them this lab-CAS manifests the intended behavior and achieves the aimed end-state.
   See instruction set of the Internet as a CAS in (Daskapan, 2008).
10. Define the properties, i.e. capabilities and knowledge, of each agent type in the lab sample.
    See list of properties of the agents in the Internet case in (Daskapan, 2008).
11. Subdivide the total system into a few clusters of agents based on the number of types.
    In the Internet only one type of agent was at hand, i.e. the AS. Therefore only one cluster exists.
12. Define the optimal number of agents of each type.
    Since there is just one type, all the agents should be of that type.
13. Apply exactly one instruction set to all the agents of exactly one cluster.
    See instruction set of the Internet case in (Daskapan, 2008).
14. Execute all agents in a controlled test case and go back to step 1 if intended behavior and aimed end-state is not achieved.

Each cluster consists now of agents that are harmonized, i.e. similar purpose, properties and instruction set. If done well, executing these programmed agents will lead to a self-stabilizing resilient information infrastructure. Semi-closed means that new entrants are only allowed if they are trusted and adopt the instruction set of one of the existing types of agents. It refers also to the fact that a CII is partly connected to external networks and systems for certain services, like a public key infrastructure for dealing with asymmetric key pairs and certificates.

ILLUSTRATION OF THE METHOD: A DEFENSE SYSTEM BY ANTS

Case description

In this section, we show an application of the proposed method. To show its working in a clear way, it is applied on a fictitious case of an ants-social infrastructure. In addition, this case shows that the approach has a broader application domain than improving resilience of networks alone.

Ant colonies are one of the most disciplined communities. Although each of them is not smart enough and is not aware of the collaborative goal, together they accomplish most difficult and complex tasks to survive. The colony can be considered as a social-technical infrastructure, since they are themselves the components of the infrastructure when they form bridges, logistical chains, etc with their body to carry and harvest food.

In our case we assume an ant colony of *n* ants that have to protect their valuable food against other bugs. In the figures below the food heap is represented by a pentagon and one bug is trying to reach it. In figure 1a, the attacking bug has clearly enough opportunities to reach the food heap, since the ants do not form an optimal collaborative defense infrastructure.

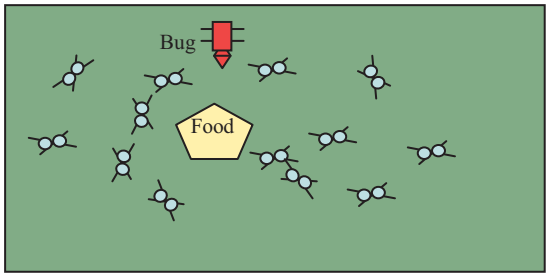

Fig. 1a Start state: no defense

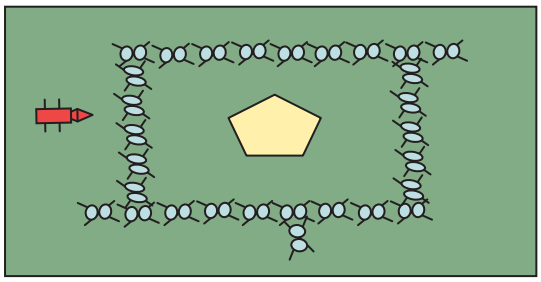

Fig. 1b End state: rectangular defense

In order to protect the food heap and thus to prevent the bug to reach the heap they have to form an ant-barrier as a collaborative defense infrastructure as shown in figure 1b. In order to form a collaborative defense infrastructure the sample of ants in the start state must be converted first into a CAS. Each ant is an individual entity that can be instructed to form a CAS. We now follow the steps of the proposed CAS method.

Application of method

1. Define a semi-closed system by demarcating the CII boundaries.
   All the ants of this colony are within the scope of a closed system.
2. Define the problem/threat that must be solved
   Weak defense of food against attacking bugs.
3. Define the components of the CII that are relevant to the problem at hand as agents; the rest is outside the scope.
   All the ants of this colony are agents. The bug and food heap are not.
4. Identify the trigger that causes the targeted problem by simulation with a minimum number of these agents, say four. This subsystem is the "lab sample" or "lab-CAS" to experiment with.
   The trigger is a near contact with a haunting bug.
5. Define the aimed end-state of the CAS
   The end-state is depicted in fig.1b.
6. Define the required behavior of the total CAS to achieve this end-state
   The ants should form a rectangular with their bodies around the heap of food.
7. Identify (by trial and error) the least number of types of activities that have to be performed by the ants to accomplish the aimed end-state.
   There are activities for
   - *1. ants to form the straight lines of a rectangular*
   - *2. ants to form the corners of a rectangular*
8. Make a distinction between agents on the basis of their different activities sets.
   - *Two types of agents are needed, since two types of activity sets are required.*
9. Translate the activities in instructions: define by trial and error the simplest instruction set for each lab-agent, such that when executing them this lab-CAS manifests the intended behavior and achieves the aimed end-state. A first trial gives the following two instruction sets:
   - For agent type 1:
     i. *DO until an ant grabs you in the middle*
        walk random
     ii. *IF you encounter an uncoupled ant*
        THEN grab the encountered ant at the back AND follow it
     iii. *ELSE IF one ant is already coupled to your back AND encountering ant grabs you in the middle*
        THEN stop walking AND let go the ant in front of you ENDIF
        ENDIF
   - For agent type 2:
     i. *DO until you grab an ant in the middle*
        walk random
     ii. *IF you encounter an uncoupled ant AND one other ant is already coupled to your back*
        THEN grab the encountered ant at the middle AND stop walking ENDIF .
10. Define the properties, i.e capabilities and knowledge, of each type of agent in the lab sample.
    - For agent type 1:

- *It can walk straight forward*
- *It can recognize the back of another ant*
- *It sees/perceives one ant a head*
- *It is not aware of the overall goal*
- *It can grab the back of other ant*
- *It distinguishes ants, food and bugs*

- For agent type 2: Same as agent type 1, but also
  - *It can grab the middle of another ant instead of the back*

11. Subdivide the total system into a few clusters of agents based on the number of types.
    Two types of agents are at hand, i.e. the agent type1 and type 2. Therefore two clusters exists.
12. Define the optimal number of agents of each type.
    There should be minimal 4 agents of type 2 in order to form four corners.
    The rest of agents should be of type 1.
13. Apply exactly one instruction set to all the agents of exactly one cluster.
    See instruction set above.
14. Execute all agents in a controlled test case and go back to step 1 if intended behavior and aimed end-state is not achieved.
    Going back to step 1 is required because of the following test results.
    - *the ants form rectangulars, but they are not necessarily centered around the food heap. This would require a special type of agent to centralize the rectangular.*
    - *the ants occasionally create unclosed rectangulars. Also here we need a special type of agent.*
    - *creating a circle is also possible.*

In this case we have collected evidence that, by applying it on ants, our method is applicable to other critical information infrastructures in order to increase their dependability.

APPLICATION OF METHOD IN CWII: RESILIENT SECURITY

Intrusion detection systems

Besides their tailored software and hardware, a CWII has also to apply security technologies to protect the core systems from being compromised. Now that we have explained and illustrated the method, we can apply it on one specific security system that is used in CWIIs (Andress, 2011): the intrusion detection system.

An *intrusion detection system* (IDS) is software that monitors the events occurring in a computer system or network and analyzing them for signs of possible *incidents* that are violations of computer security policies. An IDS can be placed on the network to monitor network traffic for particular network segments/devices or on a host to monitor the characteristics of a single host and the events occurring within that host for suspicious activity. An IDS can be based on *signatures* when types of attacks are known or on statistical *anomalies* for unknown attack (Scarfone, 2007).

Like many other security systems in a CII, IDSs are a target themselves before hackers aim at the operational control systems. In case of a CWII, an IDS is not just perimeter software, it is also the core (operational) software: it is one of the means that will be used against the enemy. Although it is not easy to attack these guardians, IDSs suffer from some known attack strategies (Tan, 2002). One of them is the blinding attack: the purpose here is to deceive the IDS with multiple or sophisticated false attacks, that hold up the IDS, while at the same time conducting another real attack. Especially false attacks that simulate new anomalies are likely to have more success, since they require more processing time. The result is that the generated excessive false alarms masquerades the real attack. It is difficult to withstand such attacks since an IDS is just doing what it should do: detecting new anomalies or signatures. Another type of attack is to change the reference (training) data model of statistical anomaly-based IDS, such that after some time the attack behavior will fall within the bandwidth of the normal activities. This happens slowly by training the IDS by simulating gradually the eventually needed behavior for an attack. In the following section we will show how a CAS approach can provide a solution to both types of attacks.

An intrusion detection systems network as a CAS

In order to characterize a group of IDS as a CAS we need to define the assumptions and the instruction set for each IDS-agent. Following the subsequent steps of the method gives the next (high level) application.

1. Define a semi-closed system by demarcating the CII boundaries.
   The CII is in this case the CWII with all its systems.
2. Define the problem/threat that must be solved.
   - *Blinding attack: fake anomalies*

- *Reference (training) data attack: change the normal*

3. Define the components of the CII that are relevant to the problem at hand as agents.
   All the (network-based and host-based) IDSs of the CWII are agents.
4. Identify the trigger that causes the targeted problem by simulation with a minimum number of these agents. This subsystem is the "lab sample" or "lab-CAS" to experiment with.
   There are two triggers:
   - *a delayed response from an active IDS or*
   - *an IDS with a different definition of the normal activities*
5. Define the aimed end-state of the CAS.
   The end-state is a logical network of cooperating IDS.
6. Define the required behavior of the total CAS to achieve this end-state.
   The IDSs cooperate together to filter out the infected IDS and to operate on its behalf when one of the previous triggers manifests.
7. Identify (by trial and error) the least number of types of activities that have to be performed by the agents to accomplish the aimed end-state.
   There are activities for agents
   - *to detect a delayed response or a deviant reference data model from an active IDS. Decision is made by majority voting.*
   - *to take over the service provisioning of a suffering IDS*
8. Make a distinction between agents on the basis of their different activities sets.
   Three types of agents are needed, since two types of activity sets are required
   - *A master IDSm*
   - *A successor IDSs*
   - *A history database HD*
9. Transform the activities into instructions: define by trial and error the simplest instruction set for each lab-agent, such that when executing them this lab-CAS manifests the intended behavior and achieves the aimed end-state.
   See instruction set below.
10. Define the properties, i.e capabilities and knowledge, of each type of agent in the lab sample.
    See assumptions below.
11. Subdivide the total system into a few clusters of agents based on the number of types.
    In this case three types of agents are required. Therefore three clusters exist.
12. Define the optimal number of agents of each type.
    Detection should happen by at least two agents in order to have majority voting. For a take over only one agent is sufficient. Therefore, including the suffering agent three agents are required.
13. Apply exactly one instruction set to all the agents of exactly one cluster.
    See instruction set below.
14. Execute all agents in a controlled test case and go back to step 1 if intended behavior and aimed end-state is not achieved.

Pseudo code and interpretation

- ***Properties***
  - An agent IDS detects behavior anomaly.
  - An agent IDS maintains a list SL with ranked preferred successors.
  - An agent HD stores network/host activities in a few history databases HD.
  - A local agent HD is a HD that has knowledge of the activities of a certain requesting IDS.
  - A global agent HD is a HD that has no knowledge of the activities of a certain requesting IDS.
  - An agent sees a limited set of agents in the total space.
  - An agent is connected to at least two other agents.
  - An agent is not aware of the consequences of his actions for the overall system.
  - 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.

- ***Instruction set***

  A. Master IDS: IDSm
  1. IDSm: read activities on local network or host.
  2. IDSm: retrieve definition of normal behavior (DefNorm) from multiple local HD.
  3. IDSm: compare new activities with DefNorm of majority.
     - *IF majority decision= abnormal THEN send alarm to administrator*
  4. IDSm: send new activities to multiple local HD.
  5. IDSm: create frequently a tuple T of the successor list *SL* with identities of other IDSs.
  6. IDSm: send frequently secret shares *t* of this tuple T to all IDSs in SL ([Shamir, 1979]).

B. Local/global HD

1. local HD: receive request for DefNorm from IDSm.
2. local HD: send DefNorm to IDSm.
3. local HD: receive new activities from IDSm and adjust DefNorm.
4. local HD: send DefNorm to global HD.
5. Global HD: compare new DefNorm with their previous DefNorm.
   - *IF majority decision = abnormal THEN send alarm to administrator.*

C. Successor IDS: IDSs

1. IDSs: check frequently availability of IDSm by frequently receiving tuple *t*.
   IF t is not received in time from IDSm (it is insufficiently available) DO:
2. IDSs: send declaration of death of suffering IDSm to other IDSs.
   IF majority of IDSs *agree on death IDSm DO:*
3. Majority of IDSs: send *t* to DSs that is the first ranked on the successor list SL.
4. First ranked IDSs: send alarm to administrator AND
   send command to the switch to (temporarily) redirect data traffic
5. GO TO 2 with IDSm = IDSs.

Note that it would go beyond the purpose of this paper, but all the messages should be considered as being encrypted including the message authentication code, timestamp and id's by using Kerberos or a public key infrastructure that also has applied CAS.

The interpretation is as follows. In A1-A3 the IDS performs its normal detection activity but then after consulting a few other IDS. In A4-A6 the IDSm sends some messages to other HD, so that they can adjust their definition of what normal behavior is in B1-B3. The IDSm sends also some messages to other IDSs to inform them that it is "alive" and to inform them which of them is allowed to succeed him first if when he might be compromised. In B4 and B5 the local HD also checks this new definition with other non-local HD, which are not serving this IDSm. It is not likely that they are infected, when IDSm is compromised by a *Reference (training) data attack*. In C the other IDSs are frequently expecting an alive message from the IDSm. Once this is delayed a *Blinding attack with fake anomalies* is assumed by the majority of the IDSs and the administrator is alrmed and network/hoist data is rerouted to another IDSs that is the first ranked on the successor list SL.

Consequently, with the application of the method above with the right assumptions and the right instruction set, this CWII experiences self-organization and shows resilience when an IDS is under attack. When one of the IDS is under attack and becomes unreachable for an unacceptable period of time, one of the other agents will temporarily continue his services.

CONCLUSION AND FUTURE WORK

Now that cyber terrorist or cyber war attacks aim at destabilizing or paralyzing the critical infrastructures of a society the (to be) developed cyber warfare information infrastructure must survive any attack. In this paper we have presented a method to develop resilient cyber warfare information infrastructures that is based on self-organization and emergent behavior. This method is not to replace, but to complement the traditional resilience methods that use redundancy techniques. By this unconventional method defense solutions emerge from collaborating subsystems or components such that they are able to withstand the more sophisticated distributed type of attacks from enemy states. To apply this method however an information infrastructure should first be converted into a CAS; each computer system in the information infrastructure is embedded with additional properties and instruction sets to perform simple tasks. We have shown how this method works in case of a fictitious infrastructure of ants to prove that it can be applied on practically any components of a CII and as such also on SCADA systems. But we have also applied it in a specific case to increase resilience of intrusion detection systems in a CWII.

It is however difficult to test our work in real CWII. Therefore, in future work, we aim at developing a lab environment with a virtual CWII where we can validate and visualize our claim.

REFERENCES

Andress, Jason & Winterfeld, Steve. (2011). Cyber warfare. Syngress. 187.

Clarke, R.A & Knake, R. (2010). Cyber War: The Next Threat to National Security and What to Do About it. HarperCollins.

Daskapan, S, Ubacht, J. & Vree, W.G. (2008). Information Infrastructure Protection by Technology stimulating Policy. Critical Information Infrastructure Security Third International Workshop. CRITIS. Italy.

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

Dooley, K., T. Johnson, et al. (1995). TQM , Chaos and Complexity. Human Systems Management. 14(4). 1-16.

Eoyang, Glenda & Doris Jane Conway. (1999). Conditions That Support Self-Organization in A Complex Adaptive System. International Association of Facilitators. USA.

Hathaway, Melissa. (2009). Cyberspace Policy Review: Assuring a Trusted and Resilient Information and Communications Infrastructure. Report. United States Office of the White House.

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

Janczewski, Lech J. & Colarik, Andrew M. (2007). Cyber Warfare and Cyber Terrorism. Information Science Reference.

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

Lawson, Sean. (2011). Beyond cyber-doom: Cyberattack Scenarios and the Evidence of History. working paper. George Mason University.

Liles, S. (2008). Our Crumbling Infrastructure: How the Aging of America's Infrastructure is a Homeland Security Concern.

Moteff, John, Copeland, Claudia, Fischer, John. (2003). Critical Infrastructures: What Makes an Infrastructure Critical?". Report for Congress. Resources Science and Industry Division.

Pye, Graeme & Warren, Matthew (2009). An emergent security risk : critical infrastructures and information warfare. Journal of information warfare. 8(3). 14-26.

Scarfone, Karen & Mell, Peter. (2007). Guide to Intrusion Detection and Prevention Systems. Special Publication 800-94. Recommendations of the National Institute of Standards and Technology.

Shamir. A. (1979). How to Share a Secret. Communications of the ACM. 22(11). 612-613.

Stewart, John W. (1998). III. BGP4: Inter-Domain Routing in the Internet. Addison-Wesley Longman Publishing.

Tan, Kymie M. C., Killourhy, Kevin S. & Maxion, Roy A. (2002). Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits. In Fifth Intern. Symposium on Recent Advances in Intrusion Detection. Lecture Notes in Computer Science 2516. Springer-Verlag. Zurich. 54-73.

Wilensky, U. & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. Journal of Science Education and technology. 8(1). 3-18.