Dark Think Security: Enhancing the Security for the Autonomous Architecture over a Restricted Domain

Julião Braga[1], Rafael de Amorim Silva[2], Patricia Takado Endo[3]. Nizam Omar[4]

[1] VRS at University of Saskatchewan
PPGEEC-UPM
Sao Paulo, SP, Brazil

[2]Federal University of Alagoas (UFAL)
Maceio, AL, Brazil

[3]Pernambuco University
Recife, PE, Brazil

[4]Mackenzie Presbiterian University
Sao Paulo, SP, Brazil

juliao@braga.net.br, rafael@ufal.br, patricia@upe.br

nizam.omar@mackenzie.br

Abstract. *This paper describes a security scheme called Dark Thing Security, to be used in the autonomous architecture model over a restricted domains. Such a scheme provides strong security due to its ability to hide intelligent agents from the implementation of the model. Several intelligent agents are distributed in the environment of an Autonomous System, being accessed only through agents of the upper layer. Such upper layer agents, called controllers, are replaced by more enabled agents, over time, making it difficult for unauthorized agents to gain access from other domains.*

Resumo. *Este paper descreve um esquema de segurança chamado Dark Thing Security, para ser utilizado no modelo de arquitetura autônoma sobre um domínio restrito. Tal esquema provê segurança forte devido a sua capacidade de esconder os agentes inteligentes, da implementação do modelo. Vários agentes inteligentes são distribuídos no ambiente de um Sistema Autônomo, sendo acessado somente através de agentes da camada superior. Tais agentes da camada superior, denominados controladores são substituidos por outros mais habilitados, ao longo do tempo, dificultando tentativas de acesso por agentes não autorizados, a partir de outros domínios.*

1. Introduction

Paul Horn, in 2001, inspired by the living system physiology presented an IBM proposal for the future of computer systems [Horn 2001]. His work argued that the efforts of specialists in the maintenance, control and operation of computer systems could be minimized and consequently have their costs reduced dramatically. The community, composed mainly of researchers continued to advance in the researches of this knowledge domain becoming a paradigm named by Horn as **Autonomic Computation**.

Contributions have been expanded by multidisciplinary research groups and the results have been surprising [Movahedi et al. 2012]. A number of applications, particularly in software, have enabled, for example, the technology of space probes [Sterritt and Hinchey 2005], rather Unmanned Space Vehicles (USVs) [Insaurralde and Vassev 2015].

The interest aroused has led to the application of Intelligent Agents or Intelligent Elements (IEs) in the Infrastructure of the Internet. *An Agent is something that perceives and acts in an environment and can improve their performance trough learning* [**?**]. This research concentrates basically on the protocols and techniques like Software Defined Networking (SDN) [Shukla 2014, Nadeau and Gray 2013, Wickboldt et al. 2015]. It is important to develop case studies and experiments on the entire spectrum of applications for the Internet Infrastructure. In this sense, the renewed experience and expansion of research groups will make an effective contribution to improving and consolidating the studies that are being carried out to date, especially the principle of interdisciplinary cooperation.

This work is presented as follows. Section 2 presents essential foundations to understand autonomic computing. Section 3 describes the dark thing security model, emphasizing structural and operation aspects. Section 4 addresses the application of the proposed model into IoT scenarios, exemplifying how the agents establish communication inside a secure zone. Section 5 presents the final considerations of this paper.

2. The Application Domain and its characteristics

The Autonomous Architecture over a Restricted Domain (A2RD) model [Braga et al. 2015] is presented in Figure 1 and divided into four layers, described below. The model serves the interest of establishing an architecture of intelligent elements under the administrative domain of ASs, which is known as the designation given to the networks that form the Internet.

The model can exist in any of the 2^{32} possible ASs [Hawkinson and Bates 1996]. However, on 02/03/2017 there were 56,710 active ASs on the Internet (originating traffic), according to CIDR-report[1]. The number of an AS is unique, controlled by the Regional Internet Registers (RIRs) and / or National Internet Registers (NIRs) and is called the Autonomous System Number (ASN). Thus, the largest possible value of x is 56710, corresponding to AS56710, at the date above. There is no conflict between the model being deployed in any AS environment and being domain-restricted. In fact, the implementations are independent, but with a high degree of interoperability and, of course, intense cooperation, because ASs administrators depend on the behavior of all the others. The IANA has reserved two contiguous ranges of ASs numbers for private use [Mitchell 2013]: 64512-65534 and 4200000000-4294967294. Conveniently, these ASs numbers can be used to designate Intelligent Elements in applications that need to represent sub-domains.

The first of the four layers hosts the Intelligent Element (IE) named **Controller**. Its identification is unique and definitive: **x:0**, that is, the number **0** placed to the right side of the symbol **:**, following by the ASN that hosting the model. Sometimes, to make clear which IE is being referenced, **IE** is used before the identification, as for example,

[1]http://www.cidr-report.org/as2.0/

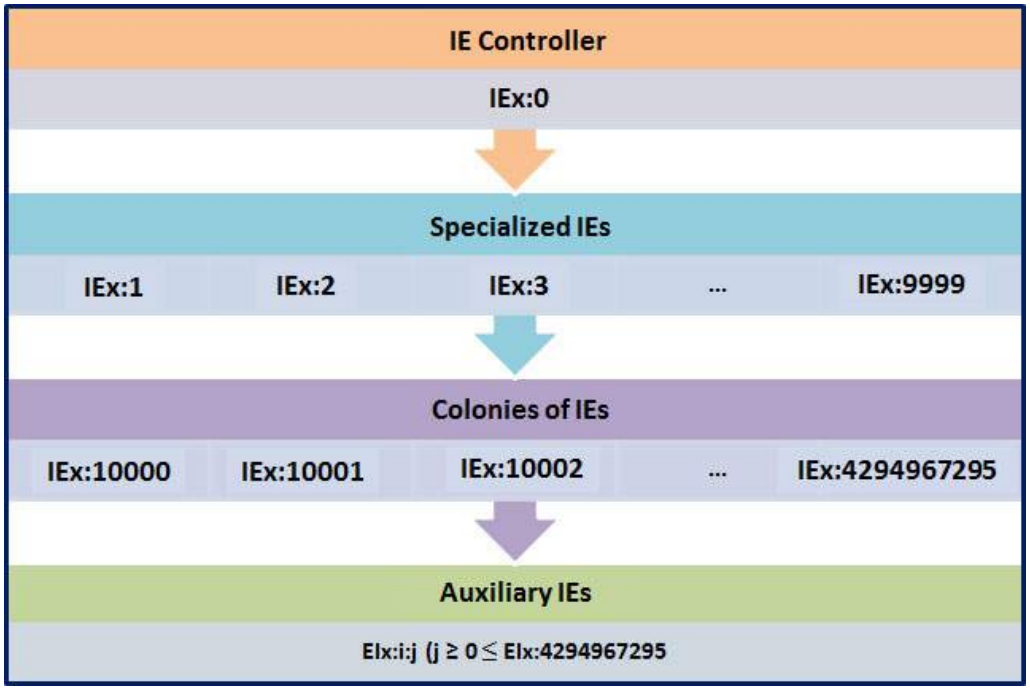

Figure 1. Four Layer Abstract Model of Autonomous Architecture for Restricted Domains (A2RD). Source: [Braga et al. 2015]

when affirming that the IE Controller is **IEx:0**. Thus, if **AS5** is the host domain of the model, then the controlling element is **IE5:0**. No IE of the lower layers can exist, without the prior consent of the IE Controller. It has the property of keeping oneself organized (self-organization) and ensuring the self-organization of any IE of the lower layers.

The second layer is represented by the so-called **Specialized IEs**. These elements are identified by suffixes that can range from **1** to **9999**. The specialized elements support the IE Controller, in specific activities and necessary to the respective functionalities. These activities range from ensuring the interoperability of the entire system of implemented IEs to the establishment of specific functionalities, such as servers with end-to-end characteristics [Saltzer et al. 1984], database access functionalities and semantic repositories, proprietary software (similar to Southern SDN APIs), features required for lower-layer IEs, and many others. However, support for the IE Controller is the primary objective of the Specialized IEs. This objective determines the functionalities of the second layer. It is assumed that some Specialized IEs may be Autonomic Elements or intelligent elements that execute automatic processes, such as proprietary software and procedures associated with legacy systems, among others. A Specialized IE can be created as a function that only concerns the IE Controller, especially when it depends on the functionalities of IEs of the third layer.

In the third layer lies the largest IEs agglomeration, which is why it is called the **Colonies of IEs**. Elements of this layer can be autonomous, autonomic or automatic, except legacy and are directly responsible for the most important activities of the application, including software reuse. They act under the influence of a high degree of interoperability and cooperation between them and between IEs of other layers and other domains / subdomains. They do not directly participate in interconnections or exchange messages with

other IEs outside the domain, but they do so through the IEs of the upper layers. There is intense semantic interoperability activity by these IEs, which have a high capacity for self-learning due to continuous interactions with the domain environment, and produce improvement effects on the knowledge of other IEs of the colony itself and the IEs of the upper layers, IE Controller. In other words, these IEs favor the learning of the entire cluster of IEs of the layer model, which hour is being described. The IEs of the colonies receive an identification with numeric suffixes, ranging from **10000** to **4294967295**.

The fourth layer is composed of **Auxiliary IEs**. This layer exists to allow the transfer of computing demands to a new set of IEs (successiveness of the model). It reproduces, successively, the first, second, third and a new fourth layers. This new IEs sequence has an additional suffix **:j:0** for a new IE Controller responsible for the following four new layers. In the new second, third and fourth layers, the IDs of the IEs are post fixed with **:j:id**, where **j** is a colony IE number that originated the new fourth layer and the **id** is a number with the above specifications. A typical application for the fourth layer are sub domains, such as home networks (home net).

The Figure 2 is the A2RD implementation model, where the small and colored rectangles are IEs. The IEs are arranged and distributed among the layers, similar to the abstract model. As an example, IEs are implemented in the domain of an AS whose number is **x**. In the same figure, one can observe that the IEs functionally is important for the inter-domain operations reside into the upper layers.

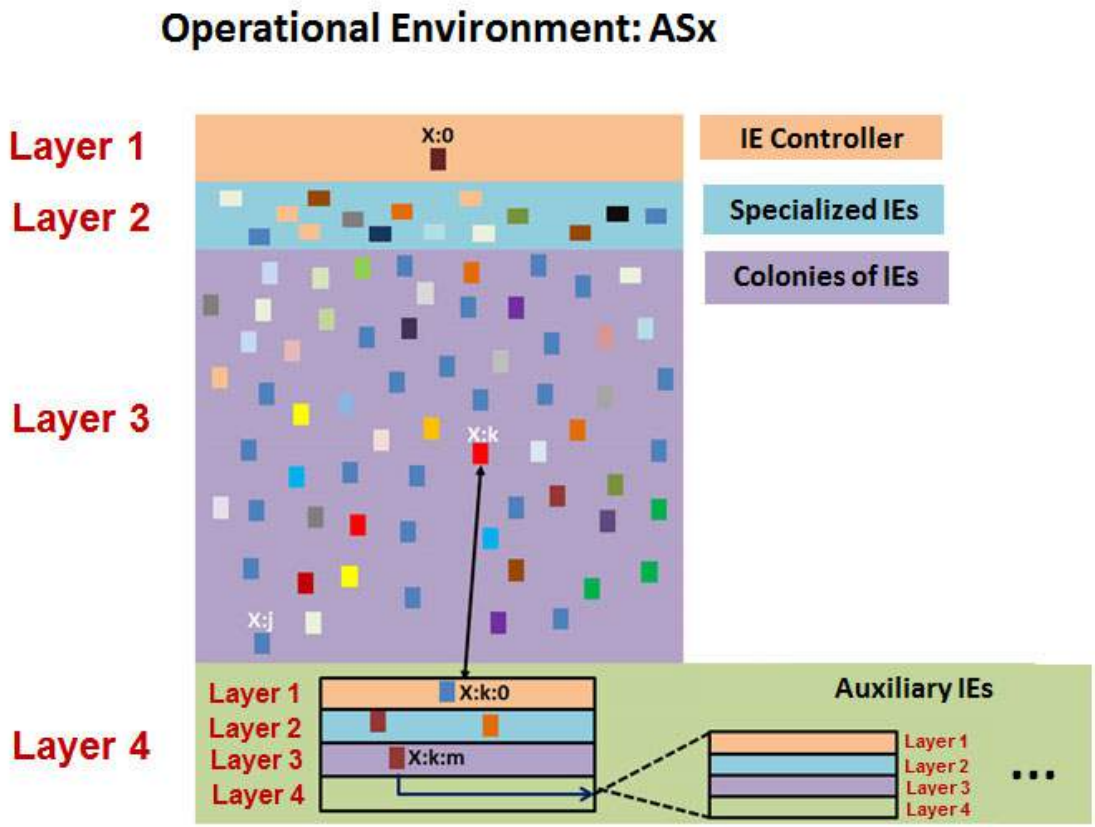

Figure 2. A2RD Implementation Model. Source: [Braga 2015a] [Braga et al. 2015]

The implementation model that a classification of relevance is the intensity of aggregation that an IE has in relation to the **self-*** properties. If an IE, however, has some self-organizing capability, it must participate directly linked to the IE Controller. Even if you participate in the layer of Auxiliary IEs there may be a new IE Controller that logically builds a new layer architecture. And so on.

On the other hand, the representation of the model is logical (abstraction of the physical implementation). Physically, locating an IE in the domain environment is essential. The best alternative is IP addressing, preferably IPv6, for availability reasons. The IE Controller must maintain a table associating the logic reference with the IP designated by the IE Controller itself, from the premise that an IPv6 block must be available at the beginning of the implementation. However, this is not a fundamental issue, because as will be seen in next section, in the name of security an IP relation as the IE ID will be available in a primitive Domain Name System (DNS), the hosts file allocated internally and with direct link to the IE Controller.

3. Dark Thing Security

The DTS model is a security scheme that protects the majority of intelligent elements from an A2RD architecture, hiding their IP addresses from the external access. This model only guarantees the IP of the IE Controller is externally visible and can be the interface of its host. In this case, the access or the interconnection to the IE Controller by other domains must have a security mechanism such as the Resource Public Key Infrastructure[2], with certification servers available in different RIRs (e.g. LACNIC[3], responsible for covering all the Latin American and Caribe regions). Figure 3 illustrates this unique visibility through a black box model, which protects all the IEs from the external attacks by hackers.

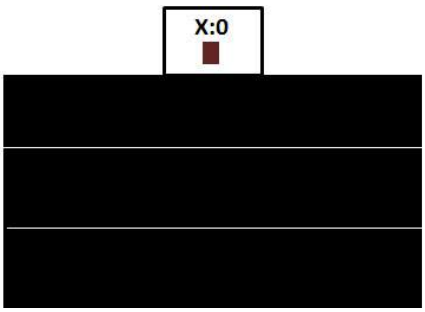

Figure 3. The Dark Thing Model for the A2RD

The IE **X:0**, as well as any IE can replicate itself in the dark environment of the figure. At any time, **X:0** can replace itself or be replaced by some other IE whenever necessary, such as if a Cyclic Redundancy Check (CRC) is used on the IE code and, in the event of an undesirable interference.

To illustrate the behavior of DTS, we will assume a scenario in which IE **X:0** must deal with the problem of source validation. In other words, IE **X:0** should ensure that an IE from another domain wants, for some reason, to bring an IE enabled in the creation

[2]http://www.lacnic.net/en/web/lacnic/informacion-general-rpki
[3]http://www.lacnic.net/en/web/lacnic/certificacion-de-recursos-rpki

of the **route**[4] object to any Internet Routing Registry (IRR) server. Some considerations should be taken into consideration for understanding what will be said:

- RPKI is one of the alternatives to the origin validation problem, but not the only one.
- When an A2RD model is implemented it announces the IP block (preferably IPv6) for all existing A2RD implementations.
- For the purpose of the example that illustrates the DTS operation, the local ASN is the AS64512 and the remote ASN is the AS64513, both private AS numbers.
- When used the notation **64512:n** or **64513:n** for any **n**, it is the same as **IE 64512:n** or **IE 64513:n**.
- The word **bootstrapping** can refer to the development of successively more complex, faster programming environments[5]. So, the word will be used in the sense of self-replacing the code of an IE by other specialized IE code. The **bootstrapping** activity happens systematically into the DTS model.

The Figure 4 displays the activities within the DTS. The only IE that accepts a request for external interconnection to the domain or even from some component of the local domain is 64152:0. Any IE is highly specialized and contains the minimum required code to meet the demand of your expertise.

B0: 64512:0 is waiting for some activity.
B1: 64513:n try to connect 64512:0.
B2: 64512:0 bootstrapping by 64512:1 which has the ability to identify whether the source IP belongs to the AS64513 domain.
B3: If the IP does not belong to AS64513 then bootstrapping the 64512:0 .
B4: 64512:1 bootstrapping 64512:101 whose ability walks over the BGP path attribute to confirm the presence of AS64512.
B5: If the AS64512 has no presence in BGP path atribute, them bootstrapping 64512:0.
B6: 64512:101 bootstrapping 64512:102 that has the ability to verify the RPKI for AS64512 based in IP captured.
B7: If RPKI is denied then bootstrapping 64512:0
...
Bn: 64512:5900 bootstraping 64512:0.

Figure 4. IE bootstrapping

This algorithm refers to Figure 5, which illustrates the relationship of the bootstrapping process of the involved IEs.

4. Concluding Remarks

There are many challenges to the progress of the project. Among them is the construction of the vocabulary (set of words and their meanings) that can meet the demand of the construction of ontologies in the area of Internet Infrastructure. The development of this

[4]http://bit.ly/2nRP9IH
[5]https://en.wikipedia.org/wiki/Bootstrapping

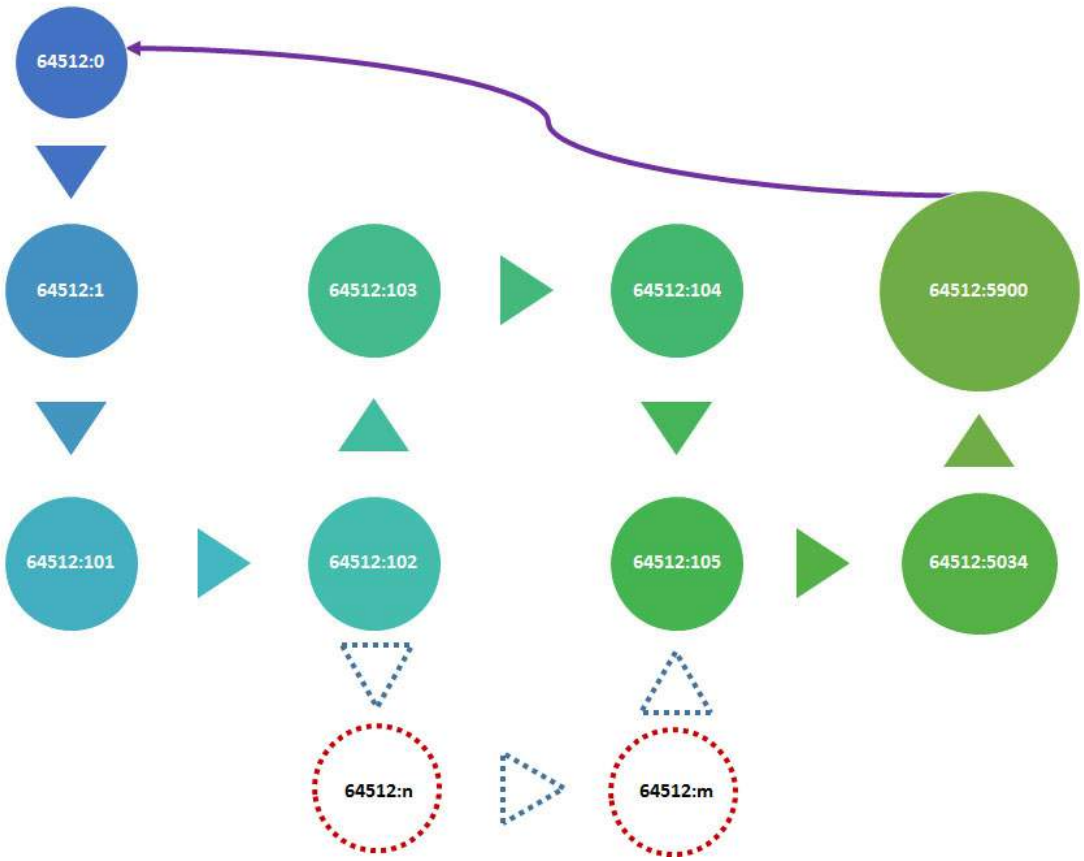

Figure 5. Bootstrapping $IE \quad X : 0$ execution model, where $X = 64512$, a private AS number

vocabulary should be cooperative and, although there are initiatives of researchers in the area of Internet of Things (IoT) [Hachem et al. 2011], and by those interested in the Internetware paradigm of the Context Aware Supporting Environment for Internetware[6] (CASEi), the authors find to have a central coordination, such as the Internet Research Task Force (IRTF), through a multi-stakeholder group. As future work the authors recommend analyzing the performance of the proposed methodology.

Without exhausting other proposals it is necessary to develop methodologies for the construction of IEs, in this case, identified as **Intelligent Objects** (IOs). IO is an intelligent element built with quality, reusable and preserving the knowledge of its life cycle, in the context of A2RD model applications, strengthening the security environment proposed by the DTS model and other related models. Construction of IOs, in an adequate and standardized manner induces interdisciplinary cooperation and rapid development of IEs. One proposal would be to use the methodology INTERA [Braga 2015b], adapted to the construction of IOs. INTERA is a successful methodology used in Learning Objects.

References

Braga, J. (2015a). Modelo para Implementação de Elementos Inteligentes em Domínios Restritos da Infraestrutura da Internet. Master's thesis, Universidade Presbiteriana

[6]https://code.google.com/p/casei/

Mackenzie, São Paulo, SP.

Braga, J. (2015b). *Objetos de Aprendizagem: Metodologia de Desenvolvimento*. Editora da UFABC, São Paulo, 1 edition.

Braga, J., Omar, N., and Granville, L. Z. (2015). Uma proposta para o uso de elementos inteligentes em domínios restritos da infraestrutura da internet. In *Anais CSBC 2015 - WPIETFIRTF*, Recife, Pernambuco, Brasil.

Hachem, S., Teixeira, T., and Issarny, V. (2011). Ontologies for the internet of things. In *Proceedings of the 8th Middleware Doctoral Symposium*, page 3. ACM.

Hawkinson, J. and Bates, T. (March 1996). Report on MD5 Performance . Technical report, RFC Editor. RFC1930. `https://tools.ietf.org/rfc/rfc1930.txt`.

Horn, P. (2001). Autonomic computing: Ibm's perspective on the state of information technology.

Insaurralde, C. C. and Vassev, E. (2015). Autonomic computing software for autonomous space vehicles. In *Nature of Computation and Communication*, pages 33–41. Springer.

Mitchell, J. (July 2013). Autonomous System (AS) Reservation for Private Use. Technical report, RFC Editor. RFC6996. `https://tools.ietf.org/rfc/rfc6996.txt`.

Movahedi, Z., Ayari, M., Langar, R., and Pujolle, G. (2012). A survey of autonomic network architectures and evaluation criteria. *Communications Surveys & Tutorials, IEEE*, 14(2):464–490.

Nadeau, T. D. and Gray, K. (2013). *SDN: Software Defined Networks*. O'Reilly, USA, 1 edition.

Saltzer, J. H., Reed, D. P., and Clark, D. D. (1984). End-to-end arguments in system design. *ACM Transactions on Computer Systems (TOCS)*, 2(4):277–288.

Shukla, V. (2014). *Introduction to Software Defined Networking*. Amazon, USA, 1 edition.

Sterritt, R. and Hinchey, M. (2005). Engineering ultimate self-protection in autonomic agents for space exploration missions. In *Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the*, pages 506–511. IEEE.

Wickboldt, J. A., De Jesus, W. P., Isolani, P. H., Bonato Both, C., Rochol, J., and Zambenedetti Granville, L. (2015). Software-defined networking: management requirements and challenges. *Communications Magazine, IEEE*, 53(1):278–285.