Agent-Oriented Approach to Detect Hardware Trojans

Rosinskiy Dmytro[1]
Kazmina Darina[2]
Muratov Vadym[3]

[1]Kharkiv National University of Radio-Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, dmytro.rosinskyi@nure.ua
[2]Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, daryna.kazmina@nure.ua
[3]Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv UA-61166, Ukraine, vadym.muratov@nure.ua

Abstract. *The presented work is devoted to the problems of hardware tabs and methods for their elimination using agent modeling. The work is based on the principles of building hardware and software systems using separate technologies of the Internet of things. Both software and hardware technologies are used in the work, which are combined among themselves with the help of intelligent agents acting as intermediaries, eliminators, keepers and models of software and hardware.*

Keywords: *hardware trojans, agent modeling, intelligent agents, hardware and software systems, Internet of things.*

I. Introduction and Problem statement

In recent years, new potential security threats in the field of electronics and programming based on hardware – the so-called hardware tabs or hardware Trojans, which represent a deliberate malicious modification of electronic circuits or structures, which leads to improper functioning of the electronic device. Being quite similar to a software tab, the hardware tab is a "black input" into the electronic device. This hardware Trojan has another additional advantage: it is always present at low levels of information processing, leading to conservation threats of failure or deviation from the normal operation of the system throughout the lifecycle of an electronic device, and the problem cannot be avoided by any software or hardware protection.

The range of hardware tabs (their capacities, sizes, operation mechanisms, power consumption) is huge, which together with the growing complexity of integrated circuits, both at the physical and functional levels provide ample opportunities for attackers to conceal hardware Trojans. Concerns about this hardware security issue are being expressed around the world, and it is believed that even more sophisticated and dangerous hardware tabs will be revealed in the foreseeable future [2, 5].

The purpose of the study is to model the behavior of hardware tabs and create means to eliminate and to detect them using agent-based modeling.

II. Problem solution and Results

Using preventive approaches of warning and modern methods of detection of hardware tabs (the so-called hardware Trojans) does not give full guarantee that the manufactured software-hardware system is deprived of them [1]. As security threats are a large class and have a considerable number of states for the placement of hardware tabs, this has raised the issue of ensuring the safe operation of software-hardware systems with "infected" components, as well as the issue of correct prevention of activation of the Trojans. The very approach would allow using the equipment without paying attention to possible embedded Trojans. The experimentally studied and tested mechanisms of countermeasures [3] can be divided into two main groups (Fig. 1).

The first group of the mechanisms (including the processor commands) provides for preventing activation of the hardware Trojan and/or blocking the direct access of the Trojan equipment to any vulnerable data. The security device can control the sample data stored or transmitted within or between software and hardware systems of logic modules, blocking the mechanism by which the Trojan communicates with the data [2].

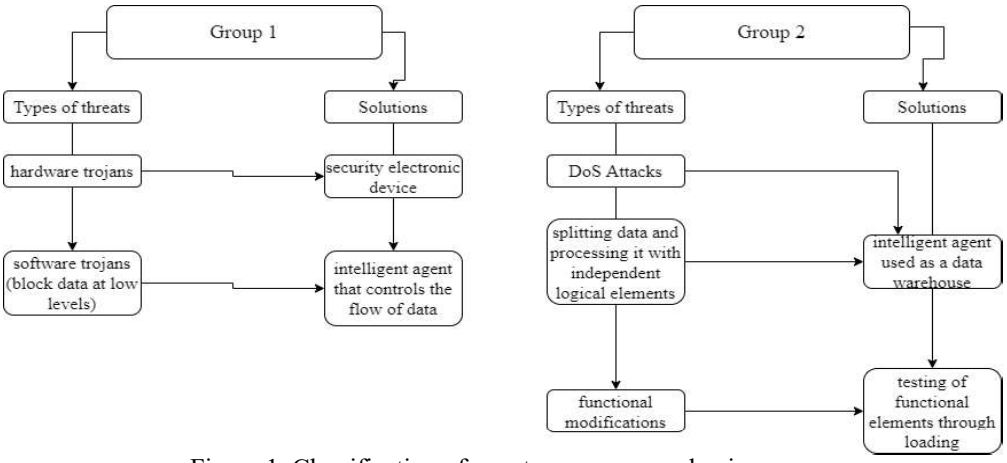

Figure 1. Classification of countermeasure mechanisms

The purpose of the intelligent agent in this group of methods is to monitor and adjust sample data to gather information about the behavior of individual logical components and send messages to the user about the "strange" behavior of the software and hardware system, or just send a list of problems caused by malfunction of logic module. To avoid getting a Trojan tab of the activation code there is another intelligent agent which holds scrambling the

performed the functions of the test elements that provided the loading. For example, in the course of the operation of a regular keyboard spy, the intelligent agent introduced by the hardware-software way simulated work with the keyboard, using it completely, during 6 hours.

Fig. 2 shows the fragment of code that is responsible for the operation of the keylogger. The result of this test was that a purposely created keylogger had run out of memory, and

```
public static void Logger (object sender, EventArgs e)
{
        if (!former.StolsClip.ToString().Contains(MyProject.Computer.Clipboard.GetText().
        Replace(''. '', ''<.> '').Replace       (''http '', ''<http> '')))
        {
                (!former.StolsClip - Operators.AddObject(former.StolsClip, MyProject.Computer.Clipboard.GetText().
                Replace(''.             '', ''<.> '').Replace(''http '', ''<http> '')
                + MyProject.Computer.Clipboard.GetText          ().Replace(''. '', ''<.> '').
                Replace(''http '', ''<http> '') + ''\r\n'');
        }
}
```

Figure 2. An example of shielding links used during keylogger

information channel [6-8]. Scrambling is used for processing data blocks that are not involved in the calculations. The principle of the agent is to encrypt selected data in a short time.

The second group is based on replication, fragmentation, and majority sampling strategies. This method is effective to protect against the DoS-attacks. The role of the intelligent agent in this method is to prevent the DoS-attacks by setting redundancy elements working in the project. Logic elements are subset into small pieces with little information. The intelligent agent or group of agents can be used as a data as a repository or their handler [3, 4, 5]. Accordingly, the data are grouped as fragments for storage and fragments for processing. At the same time there takes place a replication of selected pieces to ensure system reliability.

It is important to note that all the existing approaches to identify hardware tabs have its own unique features, but at the same time, there are some limitations. There is no method capable of detecting any class of malicious modifications with a high degree of certainty. The best way to improve the reliability of tests is to use the set of different ways to detect malicious software implementations and to provide comprehensive protection [1, 2, 4].

Hardware tab detection methods are divided into non-destructive and destructive ones. When destructive methods are used, the integrated circuits are demetallized to extract a layer-by-layer image of the chip using a scanning electron microscope.

Non-destructive methods can be divided into the system monitoring and testing prior to the system startup. In turn, the testing prior to the system startup includes two categories: the functional testing and the third-party channel analysis.

The system monitoring during its work is performed during critical calculations to detect specific harmful behavior that may occur during long-hour work. For example, a tab used to collect confidential information through wireless channels can cause large power surges during downtime.

Only non-destructive methods were considered in the article as they relate to both software and hardware. The intelligent agents involved in these methods performed work of "smart" observers (while monitoring the system performance, the intelligent agents observed the behavior of intentionally implemented hardware Trojan, making records stored on Google Drive, and sending work reports to the user), and work of "blockers" (i.e. they programmatically disabled the element that was affected by the Trojan).

During the testing prior to the system startup, the intelligent agents were involved in the functional testing. There, they

strange messages indicating the location of the problematic component (the component was introduced into the system registry) began to arrive at the message center in the software that was created to manage the intelligent agents).

III. CONCLUSIONS

Since there is no solution that can provide comprehensive protection to the whole range of threats and mechanisms of activating hardware Trojans during hardware and software systems working, the combination of the existing classical methods (such as monitoring work of the system and functional testing) with the related fields (IoT, Cloud technology, machine learning) and new methods gives the highest efficiency. Using a combination of the methods presented can cover a wider area to detect and eliminate hardware threats. The latest technologies will allow not only conducting analysis by the standard tools in the usual places of damage, but will also be able to help detect hidden hardware tabs and create new methods for their elimination.

REFERENCES

[1] S. Bhasin, F. Regazzoni. A survey on hardware trojan detection techniques, In IEEE International Symposium on Circuits and Systems (ISCAS), 2015.

[2] H. Li, Q. Liu, J. Zhang. A survey of hardware Trojan threat and defense, 2016.

[3] Q. Sui, Z. K. Wu, J. Li, S. Q. Li. A detection method of Hardware Trojan based on two-dimension calibration. 2nd IEEE International Conference on Computer and Communications, 2016.

[4] J. He, Y. Zhao, X. Guo, Y. Jin. Hardware Trojan Detection Through Chip Free Electromagnetic Side Channel Statistical Analysis. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017.

[5] C. B. Bao, D. Forte, A. Srivastava. Temperature Tracking: Toward Robust Run Time Detection of Hardware Trojans. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2015.

[6] A. N. Nowroz, K. Q. Hu, F. Koushanfar. Novel Techniques for High Sensitivity Hardware Trojan Detection Using Thermal and Power Maps. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2014.

[7] G. T. Becker, F. Regazzoni, C. Paar, W. P. Burleson. Stealthy Dopant level Hardware Trojans. International Conference on Cryptographic Hardware and Embedded Systems, ser. CHES. Berlin, Heidelberg: Springer Verlag, 2013.

[8] Martovytskyi V. O. Arkhitektura multyahentnoi systemy monitorynhu rozpodilenykh informatsiinykh system / V.O. Martovytskyi, K. R. Lokotetska // tezy dopovidei KhXVII mizhnarodnoi naukovo-praktychnoi konferentsii MicroCAD-2019, 15-17 travnia 2019 r.: u 4 ch. Ch. IV «Informatsiini tekhnolohii: nauka, tekhnika, tekhnolohiia, osvita, zdorovia». – Kh. : NTU «KhPI», 2019. – S. 164.