Report for CSAW ESC 2017

Chenglu Jin, Saeed Valizadeh, Mason Ginter and Marten van Dijk
University of Connecticut
Email: {chenglu.jin, saeed.val, mason.ginter, marten.van_dijk}@uconn.edu

***Abstract*—In this report, we briefly explain our defense strategy in order to secure Programmable Logic Controllers (PLCs).**

I. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[1]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

II. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[2]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

III. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components

[1] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

[2] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[3]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

IV. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[4]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

V. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[5]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

[3] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

[4] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

[5] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

VI. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[6]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

VII. Introduction

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[7]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

VIII. Implemented Defense strategy

A host-based intrusion detection (HIDS) is commonly used for monitoring specific activities and characteristics of a single device by means of software or appliance-based components known as agents. In this regard, in this project, we implemented a very lightweight HIDS module on the OpenPLC framework in order to keep track of input/output operations on the device.

To locate signs of likely security related incidents, the HIDS's agent, which we call it the *snapshotter* agent, essentially, takes a snapshot of the input/output operations happening on each PLC and sends them *in a stealthy secure way* to a server periodically. Then, the server

To locate signs of likely security related incidents, this agent, which is installed on each plc, logs all the events on the controller (more specifically, input/output operations) and sends the logs to a server periodically for the purpose of anaylsis and intrusion detection.

[6] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

[7] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure

On the other hand, the implemented server has a predefined set of acceptable input/output pairs for each device and checks the integrity of the log itself (in case, the attacker has already compromised the device) and the validity of the operations happening on the controller. If any of the previous incidents happen, i.e., whether the log's integrity check fails, or an operation is detected as invalid, a flag will be raised and an intrusion is indeed captured[8]. The server takes proper actions consequently which could be terminating the controller, revoking it from the network, or even recovering it to a previous clean/safe state.

For the log integrity checking feature, the server actually

For the operation validation purposes, the server actually traces deviations from predefined normal profile activities/behaviors of the input/output operations. Such profile activities could be generated based on the specific applications that the PLC is being used for. (see the comments in tex file)

Our defense mechanism can be summarized in security related information gathering and logging, incident identification, and taking effective actions to foil such incidents.

IX. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] Bowers, Kevin D., et al. *Pillarbox: Combating next-generation malware with fast forward-secure logging.*, 3rd ed. International Workshop on Recent Advances in Intrusion Detection. Springer, Cham, 2014.

[8] Assuming no errors in the controller functionality itself. Note that, even if there is error in the PLC functionality, our proposed method can capture it as well with same procedure