Whitepaper

The check bloc

v.1 May 2022

Editorial

Blocmount, LLC. is a technology startup that provides early detection of sophisticated cyber-attacks that aim to take over control systems. We focus on protecting industrial control systems (ICS) that are present in critical infrastructure and military applications. Such systems have been targets to various state-actors and APT groups.

Our framework solution is composed of a comprehensive and extensible library of checks that inspect signals using methods from industrial control charting, statistical outlier detection, change-point detection, time-series analysis, and machine learning prediction models, and a proprietary state-of-the-art artificial intelligence (AI) Defense Agent that dynamically assigns a subset of checks to inspect various signals.

The Blocmount solution is unique in few important aspects:

- We adopt a game-theoretic construction in providing our technology. Our solution does not rely on any "security through obscurity". Our framework is built under the assumption that the adversary knows all the checks that we have in our library along with their configuration. This enables us to provide stronger security guarantees.

- Our solution runs as a cloud service (private or public) and does not require the installation of any special-purpose hardware. Access to the signals is obtained through means already present in most modern industrial control system (e.g., historian servers, OPC UA servers, network captures, etc.). Thus, our technology has a minimal footprint on the existing infrastructure and can also be configured within other security solutions.

- Finally, our technology is grounded in research and the latest advancements in machine learning and artificial intelligence. Due to our game-theoretic construction from the ground up, we use such advancements as attack vectors in testing our solution.

Ensuring the security and safety of ICS requires many efforts on different fronts. The Blocmount technology provides an important piece in this large puzzle.

I would like to thank you for your interest in Blocmount and invite you to read this whitepaper and reach out with any questions or additional information.

Mina Guirguis,
Founder and Research Scientist

Introduction

Traditionally, Industrial Control Systems (ICS) were air-gapped -- designed to operate on their separate and dedicated networks that are isolated from the Internet. With the emergence of Industry 4.0 and the deployment of Industrial Internet of Things (IIoT), new trends have emerged in which various components in the ICS can now be accessed over the Internet. Accessing ICS remotely allowed for off-site administration and management of the devices and their configurations. Moreover, the ability to mine data gathered from the "floor" at real-time has been instrumental in opening-up new business cases for optimizing operations focusing on efficiency, energy, and agility. This has been fueling innovations across many sectors.

Cyber-attacks on ICS have been a real concern since the Stuxnet worm -- which targeted the Iranian nuclear facilities -- was uncovered in 2010 and over the past few years, we have been witnessing two concerning trends: (1) an increase in the frequency and sophistication of cyber-attacks on ICS and (2) targeting smaller entities that play important roles in supply chains. Nowadays, it is common to hear about cyber-attacks that rely on social engineering to infiltrate companies and organizations and create havoc. Moreover, such attacks have grown in sophistication to encompass recent advances in Artificial Intelligence and Machine Learning (AI/ML) techniques.

On the defense side, there have been many efforts from various government agencies, private and public companies in developing models and standards to guide the implementation of cyber-security solutions such as the Cybersecurity Maturity Model Certification (CMMC). The academic research community has also developed numerous checks that can detect current (and potential) attacks. These checks go beyond the traditional threshold and statistical ones to include advanced data analytics and AI/ML techniques.

Blocmount, LLC is set to advance the state-of-the-art in defending ICS by providing a novel framework that detects subtle changes in the values of the control and measurement signals that are propagated within the control loop. The framework runs as a cloud service (private or public) and uses the data streams that are typically collected by the ICS at run-time (e.g., data historian and network captures). The framework allows for developed checks (and custom-built ones) to be dynamically integrated, configured and orchestrated in a provably secure manner that minimizes risk.

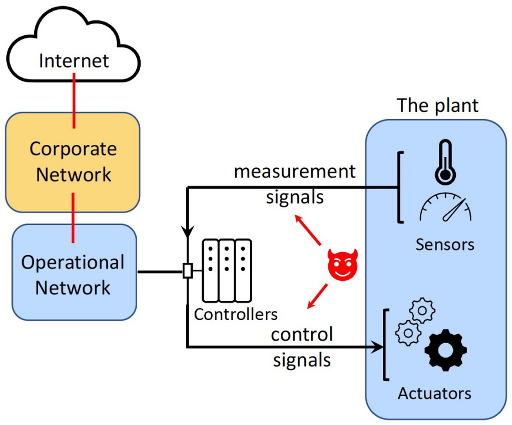

Fig. 1. A description of an ICS

The Blomount Check Library

Our framework is composed of three main components:

The Blocmount Check Library: This library offers a comprehensive array of methods to check and inspect signals. The methods encompass industrial control charting, statistical outlier detection, change-point detection, time-series analysis, and machine learning models. We refer to a single check as a "check bloc". The library is extensible and allows for custom checks to be integrated through the use of a common interface.

The Blocmount Training Agent (TA): This TA is responsible for baselining. The agent uses clean baseline data or online stream data to configure the checks for various signals and obtain performance metrics to aid the defense agent in making assignment decisions.

The Blocmount Defense Agent (DA): This DA judiciously and dynamically assigns the proper checks to inspect various signals to detect attacks and anomalies.

This document gives an overview of the check bloc component, -- a basic building block in our Blocmount Check Library.

What is a check bloc?

In a nutshell, a check bloc is an object that receives a multivariate signal and raises a flag if any of the signal components is suspicious. Thus, its input is a multivariate signal that comes from one or more assets (e.g., sensors) and its output is a binary flag. A check bloc has 3 main components: an algorithm, an internal state, and a set of configuration parameters.

The algorithm describes the method used in checking a signal. A wide variety of algorithms are used that encompass the following domains:

- industrial control charting
- statistical outlier detection
- change-point detection
- time-series analysis
- clustering analysis
- machine learning

The internal state is a collection of programming variables that are maintained and updated by the check bloc every time a new signal arrives. For example, comparing a new signal to an older one requires storing the previous value. Check blocs that do not maintain an internal state are called stateless. Stateful blocs are those that need to maintain a state to operate.

The Check Bloc

The configuration parameters are programming variables that affect the behavior of the check bloc. They are used to configure the algorithm, specify how sensitive the check bloc reacts to potentially suspicious activity and when an alert should be generated. They are typically generated during the baselining process by the training agent. Some of the configuration parameters are intentionally decoupled from the algorithms to make the check blocs more versatile. For example, the same check bloc can be configured to inspect many types of signals with various sensitivity levels.

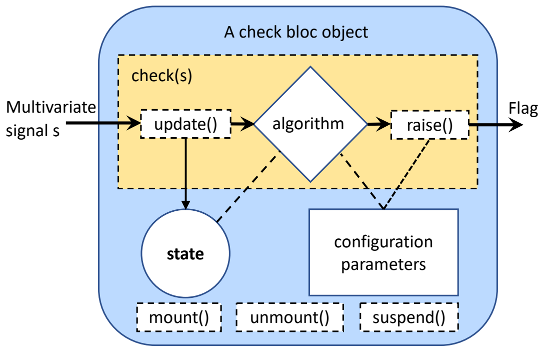

Fig.2. A description of A check bloc

How does a check bloc operate?

The check bloc uses the value of the incoming signal, its internal state (if stateful) and the configuration parameters to flag the signal based on the algorithm. Figure 2 shows the relationship between components and the basic operation of the check bloc.

The signal is first incorporated into the internal state through the update() function. The algorithm is implemented inside the function check(). The algorithm executes the checking method and raises a flag if the signal is suspicious. The algorithm has access to the internal state and the configuration parameters.

Additional components of the check bloc such as mount(), unmount() and suspend() functions are utilized by the DA in managing check blocs. mount() and unmount() refer to assigning and unassigning check blocs on assets, respectively. suspend() is used by the DA to keep the state updated without executing the algorithm.

How is a check bloc used in defense?

A check bloc is a unit of defense that is available to the DA. In a way, this is similar to a piece in a chess game except that the DA has access to hundreds of different blocs with various capabilities and associated performance metrics – an arsenal of check blocs. A common interface is used to define all the check blocs to enable the orchestration of checks unto various assets in the control system. Our proprietary state-of-the-art artificial intelligence DA strategically mounts and unmounts blocs at various points in time to inspect signals from various assets in the control system. Figure 3 shows a sample strategy in which the DA mounts various blocs on assets over time.

The Check Bloc

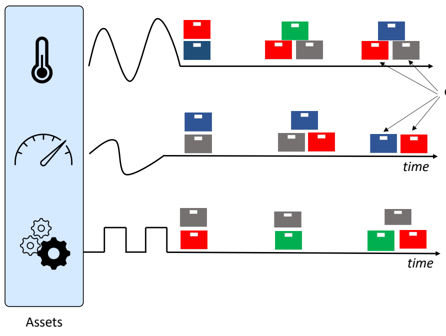

Fig.3. The DA mounting various check blocs on signals

We are now ready to introduce two terms: a general schema and an active schema. A general schema is a description of all the check blocs that are available in the Blocmount library. An active schema is an *instantiation* of the check blocs for a particular asset. An active schema is generated from the general schema during training which occurs offline (with baseline data) or online.

We provide three different examples of various check blocs that differ in their complexity.

An example of a Stateless Bloc:

The simplest check bloc is a threshold bloc which is a stateless bloc with 2 configuration parameters. The 2 parameters are the minimum threshold and the maximum threshold. During training on an asset, these parameters are obtained and stored in the active schema. At run-time, each value will be inspected, and a signal is flagged if it is below the minimum threshold or above the maximum threshold.

Examples of Stateful Blocs:

A cumulative sum check bloc is another threshold bloc which is stateful. In addition to the 2 configuration parameters, it maintains 2 state variables that track the cumulative sum. At run-time, the state is updated based on new values and a signal is flagged if the current cumulative sum is below the minimum threshold or above the maximum threshold.

Another example of a stateful bloc is a Long-Short-Term-Memory model. The configuration parameters are the hyper-parameters of the model (e.g., number of layers, number neurons in each layer, loss function, thresholds, etc.). The state is the multivariate window of signal values. During training, a model is obtained and stored. At run-time, a prediction is obtained from the model and is compared to the new signal received. A flag is raised if the difference is above a parameter threshold.

Custom-built check Blocs

The Blocmount framework allows for custom checks blocs to be created for specific applications using the bloc Application Programming Interface (API). These include checks that are not readily available in the Blocmount library. Examples of these can be assertions for values collected from the same asset at different collection points, physics-based models on a subset of assets, as well as checks that deals with transient behavior, etc. These will integrate directly into the library and will be made available to the defense agent to use when inspecting signals.

Benefits and vision

One of our goals is to expand the Blocmount library to incorporate hundreds of check blocs that can be accessed through a common API.

Here we outline some of the benefits of the Blocmount check bloc library:

- Every asset is unique in its operation, and it may not be clear which check bloc is best suited to inspect signals from that asset. The TA during training collects performance metrics on the effectiveness and the overhead of mounting every check bloc in the library on every asset. Given a comprehensive library of checks blocs that span multiple fields, we can identify the most effective ones for any asset.

- Due to our game-theoretic construction, a check bloc that is the most effective in detecting suspicious signals may not be the best defense since a clever cyber-attack can craft the signals to bypass detection (e.g., using Generative Adversarial Networks). The Blocmount DA takes this into account and may even mount several less effective check blocs to minimize risk. It becomes much harder for a cyber-attack to craft the signals to bypass many mounted check blocs consistently, over time.

We are currently working on providing the community with a tutorial on how to incorporate check blocs in the library as well as write custom ones. Furthermore, we plan to offer an online bloc recommendation service in which the client submits a time-series data from any subset of assets, and we would shortlist the most effective blocs to use against different cyber-attacks.

Blocmount

www.blocmount.com

v.1 May 2022