White Paper

What's Your Next Move? Optimizing OT Security Through Automatic Attacker Evaluation

Yehonatan Kfir

CTO, Radiflow

radiflow

TABLE OF CONTENTS

ABSTRACT

Estimating potential cyber intruder activities and what attack path they may take to access our critical assets is important in understanding how we prioritize our security measures. This paper focuses on an optimized method for automatically assessing those attack routes taking into account the challenges found due to the unique properties of industrial control systems. This model takes into consideration threats, vulnerabilities, mitigation activities and potential impact to industrial components. With this model, we can determine potential routes that an intruder could take and prioritize each path based on the potential risk to the industrial application.

INTRODUCTION

As industrial control systems grow in complexity, the ability to evaluate their vulnerability to attack becomes increasingly important to automate. Security patches and remediation are rarely deployed, leaving a greater number of assets vulnerable. Changes in the system design, even for reducing cyber risk, require long testing procedures. Safety, reliability and continuity are the highest priority for industrial systems. Thus, all changes are chosen carefully and are made after a detailed evaluation.

Labor-intensive evaluations and risk assessments are the common method used today for prioritizing security remediation measures. Those evaluation methods cover wide aspects related to industrial control system risk. Methods used include preparing security procedures, which involves people manually scanning components vulnerabilities.

Because this process can be lengthy, many turn to available automated tools to complete some of the activity.

A typical process for vulnerability analysis of a network takes the following route. First, we determine vulnerabilities of individual hosts. Using this and other information, such as connectivity between hosts, we produce attack graphs.

Each path in an attack graph is a series of exploits, which we call atomic attacks that lead to an undesirable state such as gaining unauthorized administrative access to a critical host. We then perform risk, reliability or shortest path analysis on the attack graph to assess the overall vulnerability of the network.

Creating attack graphs is critical when completing vulnerability analysis of a network of hosts. When conducted manually this process is labor intensive, error-prone, and impractical when dealing with attack graphs larger than a hundred nodes.

Automating the process of designing attack graphs also ensures that they cover every possible attack, and that they contain only those network states where the intruder is capable of reaching his goal.

We also present an automated system to generate attack graphs for industrial systems. Our design is based on passively monitoring industrial networks, and uses the following steps to produce analytical attack graphs:

1. Passively monitor industrial network traffic
2. Model the defense strategy according to the operational priorities
3. Model the technical capabilities of an industrial network attacker
4. Model the industrial communication network
5. Automatically generate the attack graph using the above parameters

The next section describes the defense strategy and what properties of the network should be preserved during a cyber-attack. It then presents a concrete definition of an industrial network attacker. This determination and its implications are subsequently shown as part of the attack graph. The paper explains how to construct the attack graph, and what parameters influence it.

The final section shows how this model for building attack vectors allows us to analyze the potential attack paths and to prioritize which devices to upgrade.

NETWORK DEFENSE STRATEGY

The central goal in designing a cyber strategy for industrial networks is to maintain the physical process in a desired condition, i.e. safety and reliability. A mandatory step in this process is to maintain devices that control the physical process in their correct operational condition.

For example, in order to ensure continuity of the physical process, we must first ensure availability and integrity of the actuators and sensors in this process.

In order to define the operational conditions of a network device, we first start with a description of the potential cyber impact that can be carried out on each device. The possibilities for cyber impacts define how an attacker can harm this device and suggests what the defender should do to protect it.

The common measures used to determine the impact on network devices are described in the Common Vulnerability Scoring System (CVSS)1 specification. The CVSS is an open framework for communicating the characteristics and severity of software vulnerabilities.

Currently, the CVSSv3 metric is the most widely accepted metric for measuring the severity of vulnerabilities. With this metric, we can determine the impact of the vulnerability using Confidentiality, Integrity and Availability (CIA):

- Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones.
- Integrity refers to the trustworthiness and veracity of information.
- Availability refers to the state of readiness for use of a specific component, such as a networked service (e.g., web, database, email).

Using CIA we can now define what properties the defenders wants to protect. We refer to this definition as a "defense strategy."

Generally, devices and zones in the industrial network may have different defense strategies. For example, a controller which is part of a safety system will have a requirement to be maintained at high availability and integrity, while a low requirement for confidentiality.

However, an engineering station which holds all the secret logic will have a requirement to be maintained at a high level of confidentiality, while availability and integrity are far less important.

Examples for different configurations, based on the device type, can be seen in Table 1.

	Availability	**Integrity**	**Confidentiality**
PLC	High	High	Low
HMI	High	Medium	Medium
Engineering Station	Low	Low	High
Other - Server, Router, OPC, Historian	Medium	Medium	Low

Table 1: Example CIA requirements for various industrial elements

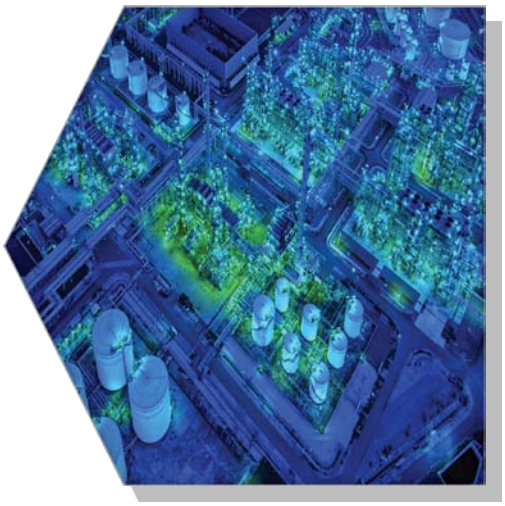
[1] https://www.first.org/cvss/

ATTACKER MODEL

Successful cyber-attacks on industrial control systems require expertise in industrial and cyber domains. A more experienced attacker is more capable of compromising network devices and pivoting throughout the network or from network to network. Therefore, modeling attacker capabilities is an essential step in estimating his/her route through the network.

In our previous paper[2] we described in detail different types of attackers who target industrial networks. We focus on two properties: expertise in exploiting protocols, and expertise in exploiting device vulnerabilities.

Expertise in exploiting protocols allows the attacker to use legitimate traffic in the industrial network to execute malicious activity. Consider an attacker that uses the legitimate control plane protocols in order to download malicious logic to a controller. In order to exploit such a control protocol, the attacker must have the knowledge (expertise) in exploiting control protocols in an industrial network. By comparison, an IT attacker may not have such knowledge, and thus is incapable of exploiting control protocols.



We distinguish between three levels of capabilities:

- Low - attackers who are capable of exploiting only IT protocols.
- Medium – attackers who care capable of exploiting IT and OT protocols. However, they are capable of exploiting only OT protocols that have open specifications. They are not capable of reverse-engineering proprietary protocols.
- High – attackers who are capable of exploiting IT and OT protocols, including those that are proprietary.

Note that for many years SCADA systems were believed to be secure because they used proprietary protocols. **In terms of our model, the misconception was that there are no attackers with a high level of expertise in exploiting protocols, and therefore, communication channels with proprietary protocol are considered to be 'secure'.**

In addition to legitimate protocols, an attacker may exploit devices through their vulnerabilities. Using vulnerabilities, an attacker would be able to affect the device's functionality. Several tools, such as Metasploit[3], are available to allow novice attackers the ability to exploit device vulnerabilities. While easy to use, those tools contain exploits only for a small portion of a wider list of known vulnerabilities. An experienced attacker can develop his own exploits. In some cases, a

[2] https://radiflow.com/download-whitepaper-meet-your-attacker-taxonomy-analysis-of-a-scada-attacker/
[3] https://www.metasploit.com/

very experienced attacker may even perform extensive research to discover new vulnerabilities, known as 'zero-day' vulnerabilities.

We distinguish among three levels of expertise capable of exploiting device vulnerabilities:

- Low – attacker is capable of exploiting only publicly known vulnerabilities with publicly available exploits.
- Medium – attacker is capable of developing his own exploits for known vulnerabilities.
- High – attacker is capable of performing extensive research to find new (unknown) vulnerabilities and is capable of exploiting them.

The following table summarizes the attacker model:

Notation	Property Description	Values
A_protocol	The level of expertise in exploiting legitimate network protocols	Low – Exploiting IT protocols Medium – Exploiting IT and OT Data-plane protocols High – Exploiting IT, OT Data-plane and OT Control-plane protocols
A_vuln	The level of expertise in exploiting device vulnerabilities	Low – using only public exploits Medium – able to develop exploits for known vulnerabilities High – able to research and exploit zero-days

Table 2: Attacker models

We define "Attacker Model" as a configuration of { A protocol , A_vuln }.

Using the attacker model, we can define the attacker level to be:
attackerlevel = A_protocol * A_ vuln (see Table 3).

We will use those two definitions when building the attack graph.

		Abusing protocols		
		Low	Medium	High
Abusing device vulnerabilites	Low			
	Medium			
	High			

Table 3: Attacker level scoring

ATTACK GRAPH

The attack graph describes the methods by which an attacker can move throughout the network. In this graph, every node represents a device, and every edge (connection between devices) represents a route that can be used to move from one device to another.

We can see that using the attacker capabilities allows us to build all the possible routes for an attacker to move in the network. Combining defense strategy allows us to prioritize the paths and vulnerabilities with respect to the industrial priorities.

EDGES AND NODES

An attacker's goal is to move throughout the network to violate the defense strategy of devices.

We assume a simple monitoring system exists in the network that monitors hosts and connections between them. This system alerts when new hosts or new connections are established between hosts. Upon alert, the attacker's activity is blocked and he will not be able to achieve his goal. Therefore, in order to stay undetected, the attacker will have to use only legitimate connections between devices.

Based on the attacker model, the attacker can move throughout the network by exploiting protocols or by exploiting device vulnerabilities. Using legitimate protocols in the network allows the attacker to better hide his activities. For example, an attacker that uses default credentials for an allowed SSH connection between devices will not be detected by the simple monitoring system.

We modeled the communication network as a directional graph G=(V,E), where:

- V – nodes represent all the devices in the network.
- E: V x V– are ordered pairs (x,y) of nodes in V, that represent a communication link from x to y.

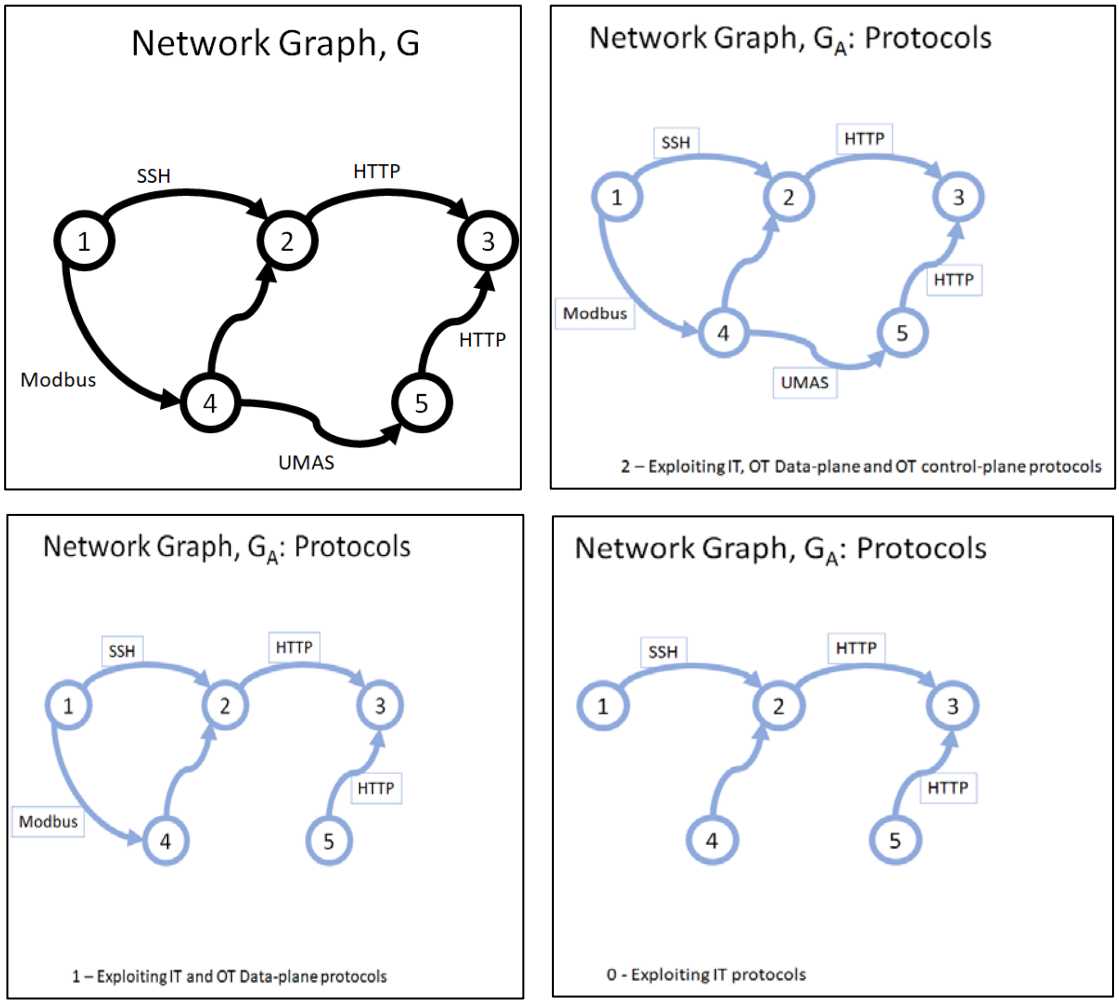
Attack Graph is defined as a directed and weighted multi-graph with exploitability weights on edges (x,y), which represent how easy it is for an attacker to use that edge for moving from device x to device y.

In order to build the attack graph, we need to discover each method for moving between devices. For each method, and for each pair of devices, we create an edge representing an attacker that can

use this method. We distinguish between edges created based on protocol exploits and those created based on vulnerability exploits.

For the protocol exploit method, we discover all the legitimate communication links in the network. Based on the attacker capabilities on protocols, we then decide whether to add this link to the attack graph or not. For example, if the attacker does not have the expertise to exploit control protocols, then control links will not be added to the graph. Note that the attacker model can generate a significant difference attack graph, as can be seen in figure 1.

For modeling exploit vulnerability edges, we discover all the devices with vulnerabilities in the network. We add an edge to vulnerable devices then check each of their neighbors' devices. Using the same method as we used for protocols, we subsequently choose vulnerabilities with respect to attacker capabilities. For example, an attacker that uses only public exploits will have a small number of weapons at his disposal, as he is excluding the much more effective zero day vulnerabilities that are much harder to detect.



EDGES WEIGHTS (EXPLOITABILITY)

The weight on edge (x,y) represents how easy it is for an attacker to use that edge to move from device x to device y. For edges based on protocol exploits, we estimate weights based on evaluating protocols available in a large number of attacks

For edges based on vulnerability exploits, we use a 'context-aware scoring' method. This method is based on the CVSSv3 scoring system. In CVSSv3, each vulnerability score has three metrics but most published vulnerabilities are ranked only using the base metric.

Based on the defense strategy, we set confidentiality, integrity and availability requirements in the environmental part of CVSSv3. As such, we receive a more accurate vulnerability score which is relevant to the context of the device.

ATTACKER ROUTES AND PRIORITIZING PATCHES

Attacker propagation in the network is done in steps. In the first step, the attacker gains control over a random device.

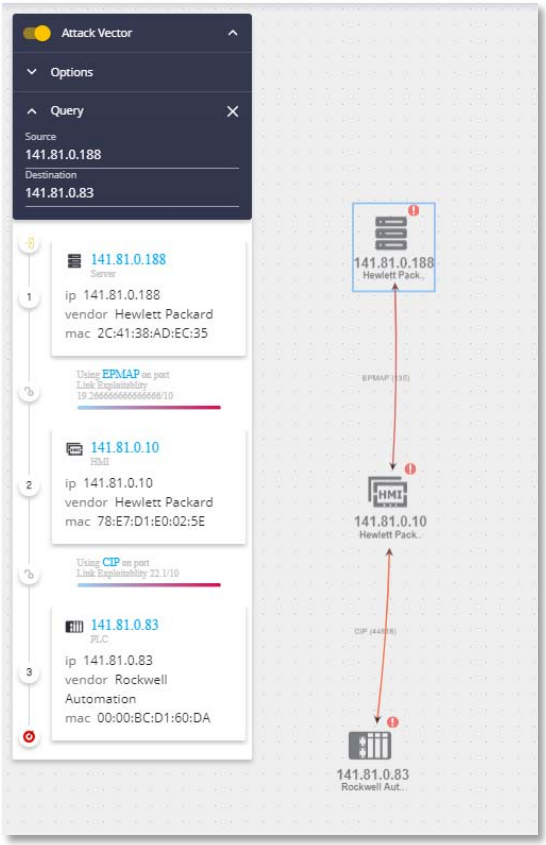
From there, the attacker continues to spread throughout the network, device by device. During each step, the attacker chooses the easiest way to move through the network, either by using vulnerabilities or protocols.

In terms of our model, the attacker chooses the edge with the highest exploitable score in the attack graph that is connected to the devices under his control.

Using the attack graph, we can now answer the following question:

For a given attacker model, and assuming device 's' is compromised, what is the potential attacker path to any given device 'd'?

For a given attack model, we can build an attack graph, and find the most exploitable path.



Furthermore, we can now answer a more important question:

How to prioritize which devices to patch / upgrade on the route from 's' to 'd'?

It is clear that the probability to compromise a device depends on the attacker level and the device vulnerabilities and protocols. However, the device exploitability also depends on the device location in the network. This means that a device located on every possible attack route has a higher probability to be compromised than a device which has only one route.

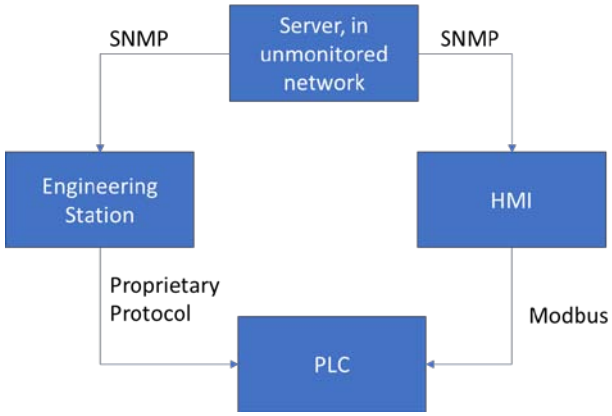
In order to prioritize device patches, we build attack graphs for all possible attack options. We can then, in each graph, calculate the most exploitable route between the two devices. This gives us the possible routes for all types of attackers. Finally, we prioritize the devices based on the number of routes they are part of, and the level of attacker that could exploit them. Devices that are located at most of the routes, and exploitable by lower attackers, should be patched first.

USE-CASE EXAMPLE

This network diagram describes a very common scenario.

A server on an unmonitored network is assumed to be used as the point of penetration for an attacker attempting to gain access into the monitored network.

However, in order to preserve the *safety* of the physical process the operator must preserve the *availability* of the PLC.



We know that:

- The PLC's firmware has a high vulnerability score and a high impact on availability.
- Both the engineering station and the HMI using unpatched Windows.
- Upgrading the PLC is more complicated than patching the engineering station or the HMI.

What should the operator do in order to maintain the safety of the process? Patch the engineering station? Or upgrade the the PLC's firmware?

Using the Attacker Model:

1. In the case of a low-level attacker that uses only open IT protocols, the operator should patch the HMI.
2. A medium-level attacker with expertise in proprietary protocols will likely also have the capability to utilize the engineering station to change the the PLC's logic. Therefore, the operator must fix both the HMI and the Engineering station.
3. In the case of a very high-level attacker capable of developing a zero-day attack, both actions above would be futile. The only practical action the operator could take is to install a firewall between the PLC and the network.

As demonstrated above, the Attacker Model dramatically alters the operator's "normal" course of action. On one hand, patching the HMI is useless in the case of a skilled attacker; on the other hand, in the case of a low-level attacker, there is no reason to install a firewall between the PLCs and the network.

SUMMARY

Many companies have limited time and limited resources for deploying and maintaining security measures on their industrial networks. Thus, anticipating and estimating potential cyber intruder activities and what attack path they may take to access critical assets is important in understanding how to prioritize security measures. Attack graphs are an essential method for predicting which routes an attacker will take in the network. The proposed model in this white paper takes into account the industrial characteristics when ranking the attack vectors. This methodology can resolve two big challenges: finding the highest probability attacker path and prioritizing patches in large networks.