Vulnerability-Based Impact Criticality Estimation for Industrial Control Systems

*[Cyber Security 2020]

Uchenna Daniel Ani
Department of Science, Technology Engineering and Public Policy
University College London
London, United Kingdom
u.ani@ucl.ac.uk

Hongmei He
School of Aerospace, Transport, and Manufacturing,
Cranfield University
Bedfordhsire, United Kingdom
h.he@cranfield.ac.uk

Ashutosh Tiwari
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, United Kingdom
a.tiwari@sheffield.ac.uk

Abstract— **Cyber threats directly affect the critical reliability and availability of modern Industry Control Systems (ICS) in respects of operations and processes. Where there are a variety of vulnerabilities and cyber threats, it is necessary to effectively evaluate cyber security risks, and control uncertainties of cyber environments, and quantitative evaluation can be helpful. To effectively and timely control the spread and impact produced by attacks on ICS networks, a probabilistic Multi-Attribute Vulnerability Criticality Analysis (MAVCA) model for impact estimation and prioritised remediation is presented. This offer a new approach for combining three major attributes: vulnerability severities influenced by environmental factors, the attack probabilities relative to the vulnerabilities, and functional dependencies attributed to vulnerability host components. A miniature ICS testbed evaluation illustrates the usability of the model for determining the weakest link and setting security priority in the ICS. This work can help create speedy and proactive security response. The metrics derived in this work can serve as sub-metrics inputs to a larger quantitative security metrics taxonomy; and can be integrated into the security risk assessment scheme of a larger distributed system.**

Keywords—Cybersecurity, Functional Dependency, Industrial Control System (ICS), ICS Security, Security Criticality Analysis, Security Impact Analysis, Vulnerability Analysis.

I. INTRODUCTION

Industry 4.0 has given rise to the integration of modern industrial control systems (ICS) with advanced information and communication technology (ICT). However, these implementation practices of modern ICS have left open flaws in forms of security vulnerabilities [1]. For example, weakly secured IT and IoT devices and network components in ICS systems can serve as the surfaces of cyber-attacks, sensitive data breaching, and even threats to the safety of human operators. The connection to the Internet is widening the cyber security risk landscape of ICS, which were initially designed for reliability, and precision real-time operations [2], but without security considerations [3].

ICS have become prominent targets of cyber-attacks [4], and the devastating impacts tend to spread to other connected systems often before responses and remediation are conceived and initiated. These attacks target businesses where entire industry value and service chains increasingly rely on vulnerable, often interconnected and functionally dependent digital data and asset [5]. Popular ICS attack incidents, such as; the Stuxnet attack on Iranian nuclear power plant, Saudi Aramco Oil systems attack, and the German Steel mill plant network attack [6], show that the impacts of cyber-attacks on one or more components can greatly cause substantial negative effects on other connected components. ICSs are integral to critical infrastructure operations, and their successful exploitation can result in not only data corruption and exfiltration but can cause significant physical consequences including the loss of human lives.

It is crucial to develop and maintain safety and security objectives in ICS to reduce the potentials of failure [7] that breach the availability and reliability of a system. It is important to understand ICS network security requirements, identify vulnerabilities, and assess the associated impacts of exploiting discovered vulnerabilities. It is more crucial to identify those vulnerabilities and components whose exploitation can cause critical consequences in order to create effective security countermeasures. Since adversaries often search and target the weakest link – the most vulnerable functional entity in an operational chain, the weakest links could be good start points to effectively investigate the security vulnerability of a system.

The Common Vulnerability Scoring System (CVSS) is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to threat. CVSS scores can be applied as attributes to compare vulnerabilities [8], [9]. The CVSS framework can also be combined with attack graphs to derive some security metrics for measuring the impact of security attacks relative to confidentiality, integrity, and availability [10]. Considering and combining the varied sub-metrics characterised in the CVSS temporal and environmental core metrics may help yield more profound impacts severity results, to help clarify typical uncertainties and/or variations in prioritised mitigation and support improvement [8].

In this paper, a novel probabilistic Multi-Attribute Vulnerability-based Criticality Analysis (MAVCA) model to estimate impact and prioritise countermeasures is proposed abstracting from CVSS concepts. The model is based on three key attributes: *vulnerability severities*, *vulnerability exploit probabilities*, and *vulnerability host functional dependencies*. It is to quantify the potential impacts of exploiting identified security vulnerabilities in ICS networks, to support prioritizing vulnerability countermeasures and actions based on estimated scale of impacts. The rest of this paper is organized as follows: Section 2 gives a brief review of existing work in the assessment of vulnerabilities of and impact on ICS. Section 3 describes the MAVCA evaluation model. Sections 4 presents a test validation of the proposed method using a case study network covering vulnerability

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

enumerations, analysis, and discussion of the results. Finally, Section 5 presents the conclusions and future work.

II. Related Work

Several works have explored security vulnerability and impact assessments for ICS. Abraham and Nair [9] applied the CVSS exploitability metrics and attack graphs for a cyber situational awareness approach to improve the understandability of cyber-attack impacts on networks and systems. Researchers in [10]–[12] adopted experimental approaches to gain realistic details in power system simulations. They used reliability metrics to assess impacts of cyber-attacks. Testbeds and emulation systems approaches [13], [14] have also been used to explore practical scenarios to examine cyber-attack impacts and potential cascades, and to underscore the impact of channel corruption on control system functionalities. Although cascading effects can be controlled with countermeasure and contingency plans, the controls do not always consider changes in the operational modes of critical industrial systems [15].

A more profound estimation of such severities can be achieved while considering temporal and environmental factors of the known vulnerabilities and their hosts. The actual impact of security incidents tend to vary amongst different types of users, organizations, and businesses [16], [17], which necessitates different prioritised mitigation modes [18]. Exploring CVSS temporal and environmental metrics to evaluate security impacts can help clarify the variations in prioritised mitigation; since actual impact of vulnerabilities in specific organizations are better reflected [8] to support more accurate assessment.

III. Multi-Atrribute Vulnerability-Based Criticality Analysis (MAVCA) Model

The proposed Multi-Attribute Vulnerability-based Criticality Analysis (MAVCA) model is a probabilistic model that provides a novel way to address the issues of uncertainty in network vulnerability management. With MAVCA an index can be obtained to help identify the most impacting vulnerabilities in an industrial network based on network dynamic factors. This can support a priority-based approach to implementing strategic vulnerability management. The impacts of cyber-attacks on ICS can be evaluated based on: *technical vulnerabilities* in the ICS, *the likelihood* of exploiting the vulnerabilities, and *a vulnerable component's functional dependency relationship* relative to other components that make up the ICS. Impact is the influence on ICS when a vulnerability is successfully exploited. Hence, a '*criticality index*' (*CI*) value can help quantify the severity of a cyber vulnerability with respect to; (i) the probability of attracting the interests of malicious actors, and (ii) the host component functional dependency that enables a cascading impact transfer. Therefore, these attributes are combined into the MAVCA model.

Fig. 1 shows the MAVCA model, which includes two stages: (*i*) Criticality Index (*CI*) estimation and, *(ii)* prioritisation. CI is a function of two parameters: *vulnerability exploit potential (β)* and *vulnerability impact potential (γ)*. CVSS temporal score attributes: *Exploitability (Ex), Remediation Level (RL), and Report Confidence (RC) can be used to evaluate exploitation dependency*. *Ex* – the likelihood of a vulnerability being attacked and depends on the state of exploit techniques or available exploit code, with options of being *Unproven, Proof-of-concept, Functional, High*, or *Not-defined*; *RL* - the current state of ICS to reflect the urgency of remediation with options of an *Official fix, Temporary fix, Workaround, Unavailable* or *Not defined*; *RC* - the degree of confidence in the existence of the vulnerability and the credibility of available technical details with options of *Unknown, Reasonable, Confirmed* and *Not defined*. Each parameter is a numeric value in the range [0, 1]. Using the *CI* values corresponding and mapped to the vulnerabilities discovered in the ICS, a set of *CI* values can be obtained from which the most critical and weakest link can be determined based on prioritisation. This could refer to the component that has the vulnerability with the highest *CI* value. This model can support obtaining a better insight to improve the effectiveness of security measures for discovered vulnerabilities in ICS.

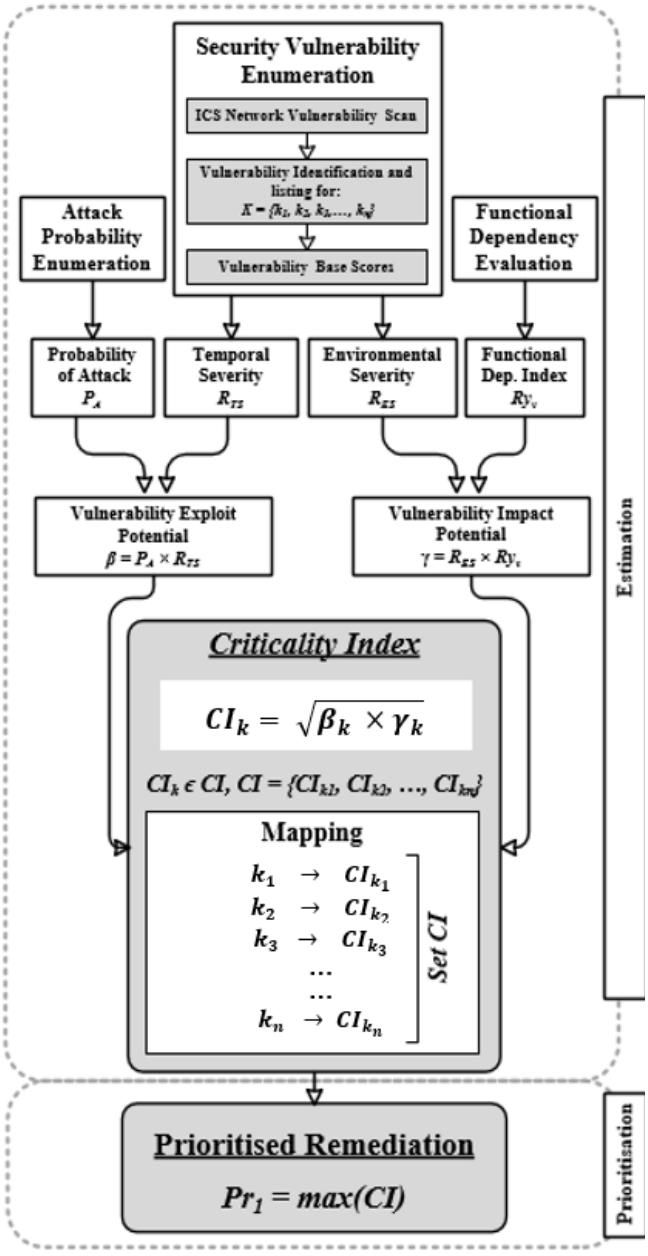

Fig. 1. The Model of Multi-Attribute Vulnerability Criticality Analysis (MAVCA)

A. Vulnerability Exploit Potential (β)

The Vulnerability Exploit Potential (β) is defined as the likelihood of exploiting a vulnerability based on its dynamic characteristics that make it attractive for exploitation comparatively with other vulnerabilities. This relates to three sources: (i) scores assigned from CVSS, (ii) functional dependency modelling structure, and (iii) probabilistic frequency of vulnerabilities evaluations. β can be derived from the probability of an attack on a vulnerability P_A given the vulnerability's temporal severity ratio R_{TS} in ICS.

A component could have multiple vulnerabilities, with each having multiple attack paths. CVSS alone may not provide enough information to ascertain the most vulnerable attack path amongst recognised options, as CVSS only offers a single machine vulnerability score. Multiple occurrences of a vulnerability can be considered into the evaluation of attack probability. The likelihood of exploiting a vulnerability is associated to the total number of attack paths to all discovered vulnerabilities on the system. The ratio of attack path occurrences to individual vulnerabilities against their accumulation in an entire network is used to represent the probability P_A of attacking a specific vulnerability in all vulnerabilities identified in the ICS. Attack paths analysis in attack graph [8] can be used, and the causal relationship amongst vulnerabilities has been modelled in [19]. The occurrence frequency of a vulnerability along an attack path can be determined by the vulnerability information in the ICS and its environment [20].

Assuming k_1 is a vulnerability in ICS, and the path to k_1 appears n times in the set of N total attack paths for all vulnerabilities discovered. The probability of exploiting k_1 among all vulnerabilities can be evaluated. As shown in Eq. (1), P_{k_1} is calculated as the ratio of the attack paths n_{k_1}, which directly link to vulnerability k_1, and the total attack paths in the network, N.

$$P_A = P_{k_1} = \frac{n_{k_1}}{N} \tag{1}$$

If vulnerability k_1, cannot be directly reached, but can be reached through another vulnerability k_2, then a conditional probability $P_{k_1|k_2}$ of the successive occurrence of k_1 and k_2 can be evaluated with Eq. (2)

$$P_A = P_{k_1} = P_{k_1|k_2} = \frac{P_{k_1 \cap k_2}}{P_{k_2}}, \tag{2}$$

where,

$$P_{k_1 \cap k_2} = \frac{Total\ attack\ paths\ with\ k_2\ and\ k_1\ in\ succession}{Total\ Number\ of\ attack\ paths} \tag{3}$$

The CVSS severity is rated in [0, 10], *0 indicating a no/low severity* and *10 indicating a critical severity,* to represent standard static *base scores* (BS) of vulnerabilities. A standard CVSS vulnerability *temporal score* (TS) is calculated with Eq. (4)[21], and a corresponding *temporal severity ratio* (R_{TS}) can be derived using Eq. (5).

$$TS = round(BS \times Ex \times RL \times RC \times 10)/10 \tag{4}$$

where, Ex = Exploitability, RL = Remediation Level, RC = Report Confidence.

$$R_{TS} = \frac{TS}{10} \tag{5}$$

Both P_A and R_{TS} are used to derive an attack severity potential, β, as shown in Eq. (6).

$$\beta = P_A \times R_{TS} \tag{6}$$

B. Vulnerability Impact Potential (γ)

The vulnerability impact potential of a component can be obtained using the device's functional dependency index and its environment metric ratio from standard CVSS ratings. Modelling node impact dependency requires to capture the relationship amongst ICS components to indicate the potential flow of adverse effects. The key quest here is to determine the devices and vulnerabilities that portends a wider scope of effects on the entire system when exploited.

Using graph theory as shown in Fig. 2, a dependency is inferred if a specific node v_3 is linked (physically or logically) to an upper-layer component v_1 and relies on the link from v_1 for the receipt or processing of signal or data streams for its own basic functionality. A directed solid arrow from v_1 to v_3 ($v_1 \rightarrow v_3$) indicates the established connection for the flow or exchange of data streams between the two nodes (components). It can be used to represent a potential transfer of attack impact from an originating component v_1 to a dependent component v_3. An impairment due to cyber-attack on component v_1 can ripple through to component v_3, thus altering its functionality or operations. A typical ICS network consists of a set of connected components, thus can be represented in a directed graph G as an ordered pair (V, E) composed of a finite set of vertices V, and a binary relation E on V. The elements of E are referred to as edges (dotted arrows) and represent the *'impact link or flow'* that cascades along successive edges. These arrows enable an ordered pair for example $e_1 = (v_1, v_3)$, of dependent nodes in the network.

In such functional dependency and impact model, every component is mapped to a *vertex,* and every dependency impact relates to an *arrow* in a graph. Thus, a graph-based structure can be used to model the topological dependencies in ICS network, and capture attack impacts. On a typical directed graph, this corresponds to the total directed edges e between a source node and the target destination node(s). The impact of an attack starts from a source node, so that determining dependency impact includes all possible impact points on the graph or network. For every vulnerability attributed to a component in the network, a corresponding functional dependency impact index $y = f(v)$ is proposed as the total dependency links across the component from the vertex v inclusive. When a component is attacked, the functional dependency index can be evaluated as the number of components that are affected along a path following the arrows from the originating component. Depending on the existence or otherwise of a functional dependency link, initial impact(s) of the attack starts at the origin and flows to any connected nodes along a path.

A logical switch function can be used to represent the conditional existence of functional dependency between any two nodes on the network. A logical 0 (*FALSE*) implies a *'non-dependency link'*, i.e., connection not configured, and a '1' (*TRUE*) implies a *'dependency link'*, i.e., connection configured. A switch function is defined as Equation 7.

$$\varphi(v) = \begin{cases} 1 \rightarrow connection\ configured \\ 0 \rightarrow connection\ not\ configured \end{cases} \tag{7}$$

For a tree network, a component's functional dependency index (denoted as y_v) is the sum of functional dependency

indices of components connected to component *v*, and it is formulated with Equation 8.

$$y_v = \sum_{u \epsilon T_v} (y_u \times \varphi(v)) \,, \tag{8}$$

where, T_v is the subset of components that can be reached directly from v.

The ratio of impact dependency may be derived in relations to the highest possible dependency, which represents the widest or worse case impact of an attack. This can be assumed to involve cases where a dependency runs through all the devices on the network, such that all are affected when a certain vulnerability is exploited. This should typically yield an impact dependency ratio of 1. A zero (0) would mean no device is affected. From Eq. (8), let the highest possible functional dependency index be represented as $max(y_v)$, so that the *impact dependency ratio,* (R_{y_v}) can be represented by the degree of dependency impact amassable from the exploit of a certain vulnerability in relations to the *widest* or *worst-case* dependency impact. It can be calculated with Eq. (9).

$$R_{y_v} = \frac{y_v}{max(y_v)} \tag{9}$$

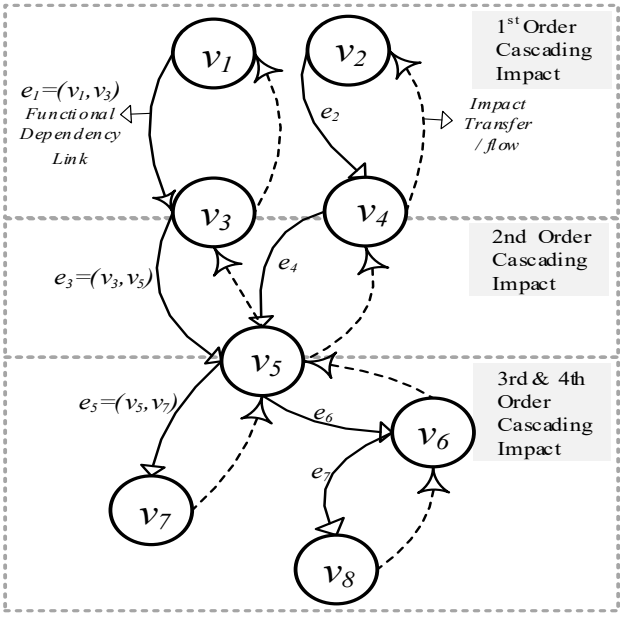

Fig. 2. Functional dependency graph

However, for a non-tree network, one component could be reachable by multiple components. Therefore, Eq. (9) may not be suitable for this case. The algorithm for determining reachable vertices from a vertex *v* in a digraph *G* can be applied to search components that a component *v* can reach to. Both depth-first and breadth-first search algorithms for digraph can be used for this purpose. Addressing these algorithms is not within the scope this work.

In a multi-order dependency structure, the impact could cascade from a single component to others in multiple layers as represented in Fig. 2. To clearly articulate the dependency status of a given component in the network, for each component $v \epsilon V$, the list of components that the component *v* depends on, and the list of components that depend on the component *v* need to be known.

Environmental severity ratio (R_{ES}) is used to standardise environment severity scores. Since varied environment setups can yield varied severity potentials, the environment metric is the modified equivalence of the base metric, considering the status dynamics within the environment. CVSS *environmental scores* (ES) are often considered optional measures when evaluating vulnerability severities. ES describes the *proportion of vulnerable systems affected*, and the value of ES is defined as a function of the *Adjusted Temporal (AT)* score, the *collateral damage potential* (CDP) of a vulnerability, and its *target distribution* (TD), calculated with Eq. (10) [22][23]. The *Environmental Severity Ratio* (R_{ES}) can be calculated with Eq. (11).

The *Adjusted Impact (AI)* of a vulnerability is obtained using the *confidentiality impact (C)*, *confidentiality requirement (CR)*, *integrity impact (I)*, *integrity requirement (IR)*, *availability impact (A)* and *availability requirement (AR)* metrics of a vulnerability. *AI* contributes to the *Adjusted Temporal (AT)* score of the vulnerability in the standard CVSS computation [21]. To preserve the consistency of CVSS 2.0, the values of all relevant formula are rounded to 1 decimal.

$$ES = round\big((AT + (10 - AT) \times CDP) \times TD \times 10\big)/10; \tag{10}$$

$$R_{ES} = \frac{ES}{10} \tag{11}$$

where: $AT = \frac{round(AI \times Ex \times RL \times RC \times 10)}{10}$;
$AI = min(10, 10.41(1 - (1 - C \times CR)(1 - I \times IR)(1 - A \times AR)))$
AT = Adjusted Temporal, CDP = Collateral Damage Potential,
TD = Target Distribution, AI = Adjusted Impact,
Ex = Exploitability, RL = Remediation Level,
RC = Report Confidence, C = Confidentiality Impact,
I = Integrity Impact, A = Availability Impact,
CR = Confidentiality Requirements, IR = Integrity Requirements,
AR = Availability Requirements

Quantitative values for device functional dependency and environmental impact ratio can be combined to yield a more reflective *dependency impact potential* (γ) for a device *v* with a known vulnerability and its environmental severity. γ can be calculated with Eq. (12).

$$\gamma = R_{ES} \times R_{y_v} \tag{12}$$

Typically, the initial values for *BS* are auto-obtained from vulnerability scanning tools such as NESSUS [22]. The status of other associated variables – *temporal(TS) and environmental (ES)* – are described in scanning tool results and used to obtain equivalent quantitative values as prescribed in the CVSS scoring system [21] in use.

C. Estimating Criticality Index (CI)

A *criticality index (CI)* attribute should mirror the severity of a vulnerability – what harm possibilities and the level a vulnerability can allow. First, *CI* needs to mirror the likelihood of attracting the interests of malicious actors – how easy (including availability and usability of tangible and intangible resources) it can be for malicious actors to accomplish the harm with success. This is represented by *vulnerability's exploit potential* (β). Second, *CI* needs to consider the relationship between a vulnerable component and other components that connect to it. It is crucial to consider how impairing such vulnerable components can affect others across a chain of connectivity. This is represented *the vulnerability's impact potential* (γ).

The geometric mean is an average value that indicates the central tendency (typical value) of a set of numbers by using the product of their values. Hence, it is proposed that '*criticality index*' (*CI*) value of a vulnerability (k) can be evaluated as the geometric mean of the *vulnerability's exploit potential* (β) and *the vulnerability's impact potential* (γ), as shown in Eq. (13).

$$CI_k = f(\beta_k, \gamma_k) = \sqrt{\beta_k \times \gamma_k} \quad (13)$$

In an ICS network with multiple vulnerabilities, Equation 13 can be used to obtain a set of Criticality Indices, $CI = \{CI_1, CI_2, \dots CI_n\}$, mapping to the set of discovered vulnerabilities, $K = \{k_1, k_2, k_3, \dots, k_n\}$, n is the total number of vulnerabilities. A *larger-values-first* rule is applied to create priority queue of control measure, a decreasing order of criticality indices per remediation time. The highest value in M takes the 'first priority' (Eq. (14)) and the associated vulnerability in K should be investigated first. This could be considered the weakest link.

$$Pr_1 = max(CI) \quad (14)$$

IV. Model Testing

To demonstrate the use of the proposed model and evaluate its feasibility, a production line ICS emulator testbed to simulate basic ICS functionalities is used. The network architecture is shown in Fig. 3. The criticality indices of vulnerabilities in the ICS are calculated, from which, the weakest link v can be identified. Assume that an inside attacker has gained the access to the network via an access point on an IP-enabled router.

A. Network Structure and Vulnerability Scan

The production line network emulator consists of an industrial-grade controller with extended input/out modules, HMI device, router gateway, a programming and control workstation, as well as some miniature production line equipment: conveyor and punching machine as Field Machine 1 (FM1), and a robotic arm machine as Field Machine 2 (FM2). FM1 and FM2, equipped with some sensors and actuators, are controlled by a master RTU controller and a slave extended module unit. As shown in Fig. 3, the controller, HMI, and workstation are connected via the router gateway, which serves as the central hub for the production line network. The case study will demonstrate how to use the proposed model to estimate critical indices and prioritise potential impacts of inherent vulnerabilities.

To achieve this, network vulnerability analysis was performed using *Nexpose* vulnerability analysis tool to scan the network for existing vulnerabilities. For each discovered vulnerability on the network, its corresponding severity BS is obtained from CVSS.

B. Graph Strucrure for Functional Dependencies

A graph-based structure is used to model the functional dependencies amongst the network components as described in Section 2. Functional dependency is used to represent the influence of a component's functions on the functions of other components in the network. For example, from Fig. 2, the functions of FM2 solely depends on the normal functions of the master RTU controller that controls FM2. Similarly, based on connections and configurations, all master, slave controllers and the HMI link to the router, and there exist data exchange among these components. Hence, the unimpaired functions of these three components depend on an unimpaired function of the router. Impairing the functions of a router can cause a malfunction in the controllers, which in turn can cause the malfunctions of FM1 and FM2. These type of functional relationship amongst network components is presented in Fig. 2, using a functional dependency directed graph.

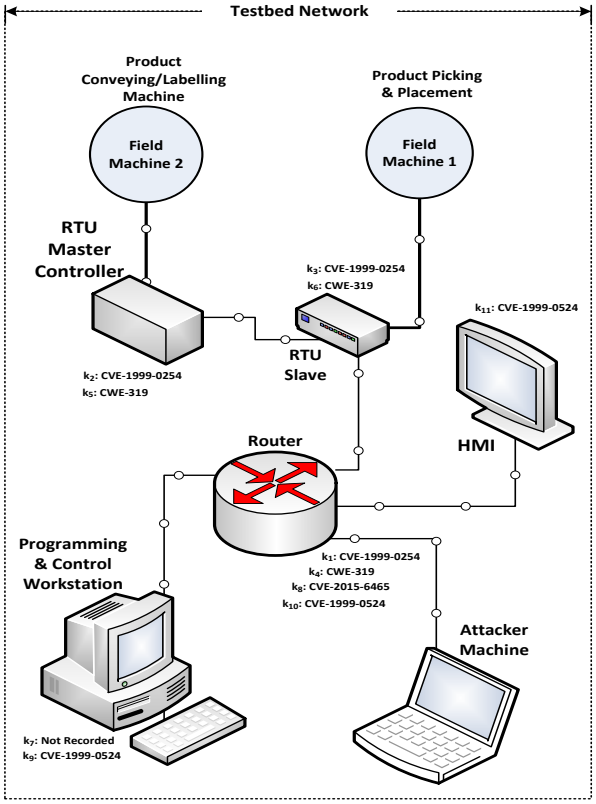

Fig. 3. Production Line ICS Emulator Testbed Network Architecture

C. Results

As shown in Table 1, 11 vulnerabilities were discovered in the network components. Some of the components had multiple vulnerabilities, and each vulnerability had a different base score. For example, vulnerabilities: k_2 and k_5 were found in the master RTU controller with severity CVSS 2.0 base scores of 10.0 and 8.0 respectively. Similarly, the gateway router had four vulnerabilities k_1, k_4, k_8, and k_{10}. The slave RTU had two vulnerabilities k_3 and k_6, the programming/control workstation had two vulnerabilities; k_7 and k_9, and the HMI had one vulnerability k_{11}.

a) MAVCA Attribute Evaluations: Applying the impact estimation and prioritisation method on the results of initial vulnerability scan yielded some discrete values for both the temporal and environmental scores. Functional dependency indices were also evaluated for each component that had an inherent vulnerability, and the probability of attack determined accordingly. This was done in line with the attack path analysis concept earlier discussed.

Following the controller example, vulnerability k_2 in the controller indicated a "*Default or guessable SNP community names: public*" [23] discovered in 1999 as indicated by the vulnerability number. This vulnerability is expressed as a weak authentication mechanism in the controller as a network device through the use of unencrypted 'community string'

[15]. Attackers can exploit this to acquire sufficient details about the network including system information, routing table and *tcp* connections, and enable remote access, reconfiguration, and device shut down. A CVSS BS of 10.0 is assigned to this vulnerability, indicating a significant magnitude of damage possible if exploited by an attacker. However, the score only accounts for the intrinsic features of the vulnerability. The dynamic feature variables, such as *temporal severity* (*TS:* Eq. (4)), *temporal severity ratio* (R_{TS}: Eq. (5)), *Environmental Score* (*ES*: Eq. (10)), *Environmental Severity Ratio* (R_{ES}: Eq. (11)), *Functional Dependency Index* (y_v: Eq. (8)), *Impact Dependency Ratio* (R_{yv}: Eq. (9)), *Vulnerability Exploitation Probability* (P_A: Eq. (1)) are calculated. Table 2 presents the values of all these dynamic variables. Then, the attack severity potential (β: Eq. (6)), dependency impact potential, and *Criticality Index* (*m*: Eq. (13)) can be calculated. Fig. (4) shows the values of the attack severity potential (β) and dependency impact potential (γ), and Fig. 5 shows Criticality Index (*CI*) for the 11 vulnerabilities in the system.

For example, using the k_2 vulnerability, the process for calculating the metrics is shown as follows: From NEXPOSE vulnerability scanning and analysis tool report, vulnerability k_2 on the PLC has BS=10. Ex status shows 'functional'= 0.95, RL shows 'workaround'=0.95, RC shows 'confirmed'=1.0, Computed TS = 9.1 (Eq. (4)) and R_{TS} = 0.91(Eq. (5)). C shows 'complete'=0.66, CR shows 'medium'=1.0, I show 'complete'= 0.66, IR shows 'high'= 1.51, A shows 'complete'= 0.66, AR shows 'high'= 1.51, Computed AI= 9.99 (Eqn. in box), Computed AT= 9. (Eqn. in box), Computed ES (Eq. (10)) = 7.1 and R_{ES} = 0.71 (Eqn. (11)). (Eq. (8), as the master controller is connected to 4 devices); $R(y_v)$ = 0.57 (Eq. (9), as the maximal dependency in the network is 7); P_A = 0.53 (Eq. (1), as all the 11 vulnerabilities on the system could yield total 15 attack paths, of which, 8 attack paths cover k_2); β=0.53×0.91 = 0.585 (Eq. (6)) and CI: = 0.487 (Eq. (13))

TABLE I. TESTBED VULNERABILITY RESULTS

Vulnerability Number	Description	Label	Devices Affected	Base Score
CVE-1999-0254	Default or guessable SNMP community names: public	k_1	Router	10.0
		k_2	RTU Master Controller	10.0
		k_3	I/O Module	10.0
CWE-319	SNMP credentials transmitted in clear text	k_4	Router	8.0
		k_5	RTU Master Controller	8.0
		k_6	I/O Module (RTU Slave)	8.0
Unnumbered	Reset Password Backdoor Vulnerability in Windows 7	k_7	Control Workstation	7.9
CVE-2015-6465	Resource Exhaustion: authenticated users to cause a denial of service (reboot) - Port 80	k_8	Router	6.8
CVE-1999-0524	ICMP timestamp response	k_9	Control Workstation	0.0
		k_{10}	Router	0.0
		k_{11}	HMI	0.0

TABLE II. TABLE 1: SEVERITY ESTIMATION RESULTS

Vul. Lab	Devices	TS	R_{TS}	ES	R_{ES}	y_v	R_{y_v}	P_A
k_1	Router	9.1	0.91	9.5	0.95	7	1.00	0.80
k_2	RTU-MC	9.1	0.91	7.1	0.71	4	0.57	0.53
k_3	RTU- Slave	9.1	0.91	2.4	0.24	2	0.29	0.27
k_4	Router	7.6	0.76	9.7	0.97	7	1.00	0.80
k_5	RTU MC	7.6	0.76	7.3	0.73	4	0.57	0.53
k_6	RTU- Slave	7.6	0.76	2.5	0.25	2	0.29	0.27
k_7	CW	6.5	0.65	9.1	0.91	6	0.86	0.67
k_8	Router	6.0	0.60	9.4	0.94	7	1.00	0.80
k_9	CW	0.0	0.00	Null	Null	6	0.86	0.67
k_{10}	Router	0.0	0.00	Null	Null	7	1.00	0.80
k_{11}	HMI	0.0	0.00	Null	Null	1	0.14	0.07

where, Master Control (MC), Control Workstation (CW).

D. Analysis

Although initial scanning revealed 11 vulnerabilities as shown in Table 2, some of the vulnerabilities appear multiple times. CVE-1999-0254 was found in the router, RTU master controller, and RTU slave. CWE-319 also exist in the router, master controller, and slave. CVE-1999-0524 exists in the control workstation, router, and HMI device. Multiple but different vulnerabilities also exist in single components, for example, vulnerabilities k_1, k_4, k_8, and k_{10} in the router. k_2 and k_5 in the master controller, k_3 and k_6 in the I/O module, k_7 and k_9 in the Control Workstation. Because similar vulnerabilities mean similar BS, it is directly unclear which vulnerability can have the greatest impact on the ICS network.

Lower *TS* values are observed compared to associated *BS* values of vulnerabilities. Although the values for *ES* are slightly higher than *TS*, they are still lower than the *BS* values. For example, k_1, k_2, and k_3 have the same severity BS of 10.0, which were all reduced to a uniform *TS* of 9.1, and temporal score ratios of 0.91. However, they had dissimilar severity ES of 9.5, 7.1, and 2.4, and environmental score ratios of 0.95, 0.71, and 0.24 respectively (See Fig. 4). The temporal and environmental scores and ratios imply further considerations of dynamic features and criteria more specific to each vulnerability relative to its immediate host component and network. These initial results indicate that standard vulnerability BS represents severities from global perspective (i.e. in relations to external factors). However, a more realistic measure of severity and corresponding magnitude ratios can be estimated with using the temporal score, and the environment score. Although a vulnerability can have a global severity scale, *(i) its frequency in a network*, *(ii) the network positioning its host component within a localized system*, *(iii) the availability of a known remediation measure*, and (*iv) the availability of exploits at varied modification requirements*, all contribute to a more profound measure of severity.

Based on these attributes' combination in the scenario setup, the vulnerability with the highest severity environmental score is k_4, with *ES* = 9.7, and R_{ES} = 0.97 *(SNMP credentials transmitted in clear text)* on the router. No functional dependency attribute consideration yet. The impact of the change magnitude on other network components dependent on the router is yet to be considered, which makes the current severity consideration weak for resolving functional impact prioritising vulnerabilities. More so, three vulnerabilities: k_9, k_{10}, and k_{11} do not have ES because their BSs are 0.0 (Null, i.e., no severity). Since both temporal and environmental scores are considered better representations of vulnerability severities (base scores) and depend on the base scores in relations environmental changes in the local

network, it is reasonable to arrive at null scores for the two dynamic quantities. Fig. 4 Vulnerability Exploit & Impact Potentials

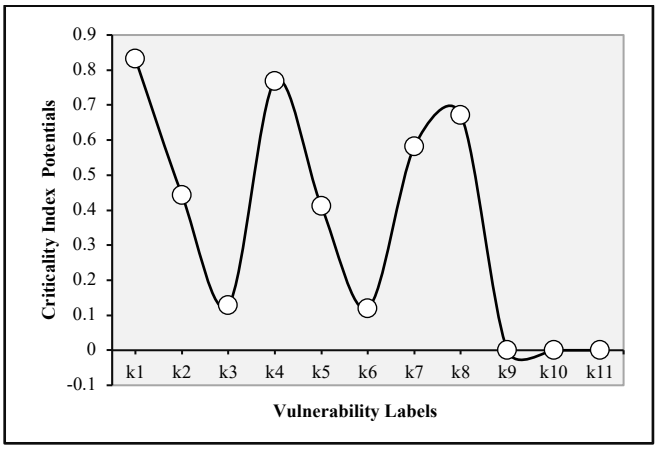

Fig. 5. Criticality Index Estimation

With varied functional dependency ratios and attack path probabilities, each vulnerability in the list yields a different vulnerability attack potential value. Although varied vulnerabilities in a single component can have similar attack path probabilities, overall individual vulnerability attack potential can vary depending on the exploit modification requirement of each vulnerability, and the available remediation capabilities. These are integrated into a temporal metric score. These account for the varied vulnerability exploits potentials *β;* ($\beta_{k1} = 0.728$, $\beta_{k4} = 0.608$, $\beta_{k8} = 0.480$, *and* $\beta_{k10} = 0.000$) for each vulnerability in the router. Higher values indicate a higher likelihood of a vulnerability being attacked, and lower values indicate otherwise.

A component can have multiple vulnerabilities with each bearing different attack potential ratio. This informs of each vulnerability's likelihood of attracting exploitation interests comparatively to other vulnerabilities in the system. In the test setup, the vulnerability with the highest exploit potential is k_1, $\beta_{k1} = 0.728$ (*Default or guessable SNMP community names: public*) on the router. Estimations for vulnerability impact potential is evaluated, and the least quantitative impact potential is 0.070 associated to vulnerability k_3 on the RTU slave. The highest impact potential ratio is 0.970 for vulnerability k_4 on the router (See Fig. 4). These values suggest the scale of possible cascading effects that can occur from exploiting the identified vulnerabilities in the specified host components.

The *Criticality index (CI)* (Fig. 5) can be obtained by combining the vulnerability exploit potential *(β)* and the vulnerability impacts potential as shown in Eq. (13). Prioritisation of vulnerability control is performed on the set *M* of criticality indices obtained (Eq. (14)). In determining the vulnerability to take *'first priority'* response, k_1 with the highest criticality index of 0.8316 is identified and eligible for the Pr_1 position, thus, should be resolved first. This is followed by k_4 (CI_{k_4}= 0.7679), k_8 (CI_{k_8}= 0.6717), k_7 (CI_{k_7}= 0.5813), etc. Analytically, the k_1 vulnerability resides in the router which serves as the central communication medium for all other IP-enabled components (Controller, Extended module, Control Workstation, and HMI) on the test network. All other components seem to depend directly or indirectly on the normal functioning of the router. Thus, any malfunction in the router can affect the appropriate functions of the other components. CVSS description of vulnerability k_1 in the router shows it can allow for unauthorized disclosure of information; unauthorized modification and disruption of services. These can be perpetrated via a network and does not require any form of authentication to be accomplished. The attack paths from this vulnerable component to other components are more numerous than for any other component vulnerability in the network.

An ability to illegally obtain and use router access credential implies that all other connected devices stand the risk of being compromised using various attack forms, including Man-in-the-middle, denial-of-service, session hijack, etc. These types of attacks influence process manipulations into forms other than the required, e.g., disrupting industrial processes from the controller or control workstation points. Accessing information on operating components means that they can easily be misconfigured or shutdown illegally. Conversely, similar attacks on the same vulnerability in the controller would only affect the target device and the equipment connected to it, and not all other components. In this case, impact would be much lower than if it was on the router because fewer network components connect to, and depend on the controller.

Similarly, no other vulnerability on any other devices expresses as much damage characteristics like the k_1 vulnerability in the router, which is why its criticality ratio is the highest amongst all. Applying patch or any countermeasure on k_1 vulnerability can help prevent the possibility to reach or attack all other components connected, thus, avoiding the largest possible impact described. If not in use, the SNMP service in the router can be disabled to help deter the exploitation of this vulnerability. When in use, the default community strings can be changed to private, or a filter can be applied to the incoming UDP packets that exploit the port (41028), thus, the access is denied to the information that can empower an attacker to exploit this vulnerability.

V. Conclusion And Future Work

To maintain continuous understanding of security states of ICS networks in the face of several vulnerabilities and improve the security of the components, it is crucial to have an effective approach for estimating susceptibility impacts and their criticalities. Non-static and environmental characteristics of vulnerabilities can provide a more profound view and episteme about cascading impacts and attack potentials.

We propose a new model for Multi-Attribute Vulnerability Criticality Analysis (MAVCA) which combines CVSS temporal and environmental attributes with attack graph probabilities and functional dependency attributes. MAVCA provides a methodology for assessing the security state (e.g. severities and impacts) in an ICS network, creates a practical priority-based security strategy to prevent cyber-attacks. This can support timely response to prevent cyber incidents and reduce potential physical and economic losses. Also, the reliability and availability of ICSs can be ensured.

The novelty of the proposed MAVCA model lies in its combining of static, temporary, environmental, and dependency attributes to evaluate relevant vulnerability and host-based criticality indices. The indices can be used to drive prioritisation of countermeasures. Apparently, a vulnerability can have different exploitation and impact potentials in

different network environment, hence, it can yield different criticality indices. A range of vulnerability criteria: *functional dependency ratio of host components*, *attack path probability*, *exploit techniques or code availability*, *vulnerability remediation level, degree of confidence*, *potentials for loss of life or physical asset from vulnerability exploit*, *proportion of vulnerable system*, and *value of the affected asset*, can contribute to a more profound measure of security impacts and criticality in ICS an environment.

This approach offers the benefits of a way to address uncertainties arising from cases of multiple vulnerabilities with the same severity ratings. It can support gaining insight to the effectiveness and efficiency of any prior vulnerability control or management measures. It can be used to determine and fix component '*weakest link*s' in networks. Most importantly, it provides a basis for prioritising security responses. The vulnerability exploitation and impact potential metrics can be part of larger security metrics taxonomy and can be used in a larger security risk assessment scheme. Using the evaluation model, infrastructure owners; especially the non-technical users, can make proactive decisions to improve the cybersecurity of ICS infrastructures.

This represents a part of a body of research for developing an automated system to analyse vulnerabilities, characterise their impact criticalities on overall systems, and prioritise vulnerabilities to address. Future works include implementing a tool that can automate the evaluation and prioritisation process based on the model

ACKNOWLEDGMENT

The authors acknowledge the support of the Royal Academy of Engineering (RAEng) and Airbus under the Research Chairs and Senior Research Fellowships scheme (RCSRF1718\5\41) and PETRAS National Centre of Excellence for IoT Systems Cybersecurity, STEaPP, UCL. Dr Uchenna Ani is a Research Fellow at the PETRAS Hub. Professor Ashutosh Tiwari is Airbus/RAEng Research Chair in Digitisation for Manufacturing at the University of Sheffield. The authors also acknowledge the funding provided by Research England's Connecting Capability Fund (CCF) through the Pitch-In project.

REFERENCES

[1] Q. S. Qassim, N. Jamil, M. Daud, A. Patel, and N. Ja'affar, "A review of security assessment methodologies in industrial control systems," *Inf. Comput. Secur.*, pp. 1–15, 2019.

[2] Z. Drias, A. Serhrouchni, and O. Vogel, "Analysis of Cyber Security for Industrial Control Systems," in *International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, 2015, pp. 1–8.

[3] U. D. Ani, N. Daniel, F. Oladipo, and S. E. Adewumi, "Securing industrial control system environments: the missing piece," *J. Cyber Secur. Technol.*, vol. 2, no. 3–4, pp. 131–163, 2018.

[4] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, "Cyber-physical security challenges in manufacturing systems," *Manuf. Lett.*, vol. 2, no. 2, pp. 74–77, Apr. 2014.

[5] Roland-Berger, "Think Act: Cyber-Security, Managing threat Scenarios in manufacturing companies," Munich, 2015.

[6] U. P. D. Ani, H. He, and A. Tiwari, "Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective," *J. Cyber Secur. Technol.*, pp. 1–43, 2016.

[7] S. Delamare, A. A. Diallo, and C. Chaudet, "High-level modelling of critical infrastructures' interdependencies," *Int. J. Crit. Infrastructures*, vol. 5, no. 1/2, pp. 100–119, 2009.

[8] C. Frühwirth and T. Männistö, "Improving CVSS-based vulnerability prioritization and response with context information," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009*, 2009, pp. 535–544.

[9] S. Abraham and S. Nair, "Cyber security analytics: A stochastic model for security quantification using absorbing markov chains," *J. Commun.*, vol. 9, no. 12, pp. 899–907, 2014.

[10] D. D. Dudenhoeffer *et al.*, "Interdependency modeling and emergency response," in *Summer Computer Simulation Conference*, 2007, vol. 2, no. August, pp. 1230–1237.

[11] B. Rozel, M. Viziteu, R. Caire, N. Hadjsaid, and J. P. Rognon, "Towards a common model for studying critical infrastructure interdependencies," in *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES*, 2008, pp. 1–6.

[12] N. Hadjsaid, C. Tranchita, B. Rozel, M. G. Viziteu, and R. Caire, "Modeling cyber and physical interdependencies - Application in ICT and power grids," in *2009 IEEE/PES Power Systems Conference and Exposition, PSCE 2009*, 2009, pp. 1–6.

[13] G. Dondossola, F. Garrone, and J. Szanto, "Supporting cyber risk assessment of Power Control Systems with experimental data," in *2009 IEEE/PES Power Systems Conference and Exposition*, 2009, pp. 1–3.

[14] A. Tesfahun and D. L. Bhaskari, "A SCADA testbed for investigating cyber security vulnerabilities in critical infrastructures," *Autom. Control Comput. Sci.*, vol. 50, no. 1, pp. 54–62, 2016.

[15] A. Nieuwenhuijs, E. Luiijf, and M. Klaver, "Modeling Dependencies in Critical Infrastructures," *Crit. Infrastruct. Prot.*, pp. 205–213, 2008.

[16] M. Ishiguro, H. Tanaka, K. Matsuura, and I. Murase, "The Effect of Information Security Incidents on Corporate Values in the Japanese Stock Market," in *The Workshop on the Economics of Securing the Information Infrastructure*, 2006, pp. 1–15.

[17] R. Telang and S. Wattal, "An Empirical Analysis of the Impact of Software Vulnerability Announcements on Firm Stock Price," *IEEE Trans. Softw. Eng.*, vol. 33, no. 8, pp. 544–557, 2007.

[18] Y.-P. Lai and P.-L. Hsia, "Using the vulnerability information of computer systems to improve the network security," *Comput. Commun.*, vol. 30, no. 9, pp. 2032–2047, 2007.

[19] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An Attack Graph-Based Probabilistic Security Metric," in *Data and Applications Security XXII: 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security London, UK, July 13-16, 2008 Proceedings*, vol. 5094 LNCS, V. Atluri, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 283–296.

[20] V. Shandilya, C. B. Simmons, and S. Shiva, "Use of attack graphs in security systems," *J. Comput. Networks Commun.*, vol. 2014, 2014.

[21] FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," *Forum of Incident Response and Security Teams (FIRST)*. Forum of Incident Response and Security Teams (FIRST), Morrisville, North Carolina, U.S.A., pp. 1–21, 2015.

[22] I. Tenable, "Nessus Professional | Tenable™," *Tenable Nessus Webiste*, 2017. [Online]. Available: http://www.tenable.com/products/nessus-vulnerability-scanner/nessus-professional. [Accessed: 11-Oct-2017].

[23] R. Gary A, "Global Information Assurance Certificateion Paper: SNMP Community Strings," 2000.