EAI Endorsed Transactions
on Security and Safety

Research Article **EAI.EU**

Dynamic Risk Assessment and Analysis Framework for Large-Scale Cyber-Physical Systems

Adeel A. Malik*, Deepak K. Tosh

University of Texas at El Paso, El Paso, TX, USA

Abstract

Cyberspace is growing at full tilt creating an amalgamation of disparate systems. This heterogeneity leads to increased system complexity and security flaws. It is crucial to understand and identify these flaws to prevent catastrophic events. However, the current state-of-the-art solutions are threat-specific and focus on either risk, vulnerabilities, or adversary emulation. In this work, we present a scalable Cyber-threats and Vulnerability Information Analyzer (CyVIA) framework. CyVIA analyzes cyber risks and abnormalities in real-time using multi-formatted knowledge bases derived from open-source vulnerability databases. CyVIA achieves the following goals: 1) assess the target network for risk and vulnerabilities, 2) map services and policies to network nodes, 3) classify nodes based on severity, and 4) provide consequences, mitigation, and relationships for the found vulnerabilities. We use CyVIA and other tools to examine a simulated network for threats and compare the results.

Received on 06 November 2021; accepted on 13 January 2022; published on 25 January 2022

Keywords: Risk Assessment, Vulnerability Assessment, NVD, MITRE ATT&CK

doi:10.4108/eai.25-1-2022.172997

1. Introduction

Awareness of a cyber defender plays a significant role in finding the potential attack paths an intruder might choose to invade organizational security. Securing digital assets depends on what kind of security controls are in place, and the degree of protection offered by such controls. However, inside these controls or other organizational assets, the presence of vulnerabilities or weaknesses can allow a threat actor to infiltrate highly protected facilities. A defender must not only rely on and retain information relevant to security controls but would also maintain an updated vulnerability information system in order to get a clear overall picture of the organizational security posture at regular intervals. According to National Vulnerability Database (NVD), the number of reported vulnerabilities is increasing at an alarming rate. In the year 2020 alone, not only the highest number of vulnerabilities (18,352) were reported to date [1], but also 57% of the reported vulnerabilities were classified as critical or high severity [2].

To perform vulnerability assessment, cyber defenders can either manually obtain and process information about the discovered computer security vulnerabilities from the publicly available vulnerability databases (VDBs) [3–5] such as NVD, Common Vulnerabilities And Exposures (CVE), Open Source Vulnerability Database (OSVDB), etc. or use vulnerability scanning tools [6–8]. Both options have their own trade-offs [6]. In the case of VDBs, one can run into issues like data formats, data consistency and integrity, scoring systems, and metrics being used [9, 10]. The third-party vulnerability scanning tools on the other hand in most cases use the Common Vulnerability Scoring System (CVSS) [11] as a standard, however, they are still not widely adopted due to varied coverage, customization inflexibility, and the abstracted implementation details [6].

Ensuring cybersecurity is a major challenge that requires ongoing efforts for a cyber defender, especially in the case of a large scale densely-connected environment such as a CPS mainly due to the complex and heterogeneous structure [12–14]. Periodic risk assessment supports a cyber defender in quantifying risks and identifying critical areas of the infrastructure

*Corresponding author. Email: amalik@miners.utep.edu

[15]. Relevant and timely received information about potential risks, threats, and vulnerabilities aid the risk assessment process to derive more accurate and effective risk analysis on one hand, and an opportunity for a cyber defender to defend against these threats on the other. However, the current literature for cyber risk assessment is focused on risk assessment only and not considering vulnerabilities, or the proposed frameworks are only theoretical with missing implementation details, or the contextual information related to the cyber infrastructure is missing. This lack of standardized contextual information creates blind spots in the defender's analysis of systems. Furthermore, highly secured organizational infrastructure can also get compromised by socially engineered cyber-attacks [16].

We propose a threat intelligence system specifically tailored for large-scale environments that covers security for both, cyber and physical aspects of a CPS to provide contextual analyses. For the cyber aspect, CyVIA provides insights into risks related to found vulnerabilities within the installed operating systems and applications, whereas, for the physical aspect, CyVIA considers the employed security controls and related policies, applicable adversarial actors with their capabilities, and network/service dependencies among network nodes. CyVIA dynamically collects the vulnerability information from major VDBs, the infrastructure information from the evaluating network, subject matter expert input from the defender for fine-tuning where needed, and generates various real-time analyses of the infrastructural security on the fly. We present a cyber threat intelligence system that generates a comprehensive breakdown of the target computing environment by producing:

1. Network and dependency maps,
2. Control-based and vulnerability-based risk scores, and
3. identifying critical infrastructure elements.

For conducting vulnerability information, CyVIA uses the most popular VDBs and provides a detailed featured dataset that describes the environmental and vulnerability-specific data. Furthermore, the cyber defenders are educated by CyVIA on the consequences, mitigation, and relationships of the discovered vulnerabilities. We also compare CyVIA with the state-of-the-art tools and discuss findings.

The rest of the paper is organized as follows: Section 2 discusses the related works, Section 3 presents the system model, Section 4 evaluates CyVIA in comparison with other state of the art systems, Section 5 concludes the paper.

2. Related Works

Traditional computer networks have transformed into Cyber-Physical Systems (CPS) with an ever-growing number of connected devices and increased numbers of various applications and services. Internet of things (IoT) and Industrial Internet of Things (IIoT) on the other hand are also reshaping our traditional networks to highly convoluted infrastructures introducing several uncertainties. Identification of cyber and physical aspects is extremely important to evaluate network security. Authors in [17] propose a novel method that helps in solving the network structure identification problem by comparing various classical sparse recovery methods on noisy observed data. Similarly, authors in [18] use a similar approach to identify the bottlenecks within the given network. On the other hand, securing such a wide range of integration has become a major challenge of recent times where cyber defenders either have limited awareness or limited resources [19]. On average, organizations spend $18.4 million annually on cybersecurity tools [20] where 58% are willing to increase the budget by an average of 14% for the following years. However, 53% of information technology experts are unsure whether the cybersecurity tools are working as expected, and only 39% admit they are confident in the investment [21]. Global spending on cybersecurity products and services is expected to exceed $1 trillion in 2021 [22].

Vulnerability scanning tools provide insights into cyber aspect of any network and proactive defense against application threats and are still not widely used as compared with malware or antivirus software. Authors in [6] provide a comparative evaluation of different tools and provide guidelines to practitioners for selecting the right tool. Authors in [11] evaluate nine different cybersecurity risk assessment tools. The study shows that most of these tools use the Common Vulnerability Scoring System (CVSS) as a standard and can integrate with other commercial technology partners for enhanced vulnerability management. Similarly, authors in [6–8, 23] propose many other vulnerability scanning tools. However, the main issue with vulnerability scanning tools is that they do not offer insights about the overall infrastructural risk, and the implementation details on the other hand are generally abstracted.

Cybersecurity is an ongoing effort and organizations can not afford to look away in order to manage their cyber risk effectively. A cybersecurity evaluation tool (CET) is proposed in [24]. CET consists of 35 self-rate question survey that identifies organizational vulnerabilities based on a set of standard measures. CET helps in identifying the fundamental post-breach efforts that can proactively secure sensitive data. Romilla Syed proposes a cyber intelligence alert (CIA) system that informs common users about vulnerabilities and their potential countermeasures [5]. CIA collects vulnerability from Twitter, CVE, NVD, vendor websites, and uses a machine-learning approach to reason if the alert should be raised for a vulnerability or not. Evaluating cybersecurity has also become a challenge with the increased number of cyber threats. Authors in [25] propose a cybersecurity audit model (CSAM) that implements the cybersecurity awareness training model (CATRAM). Similar to CET, CSAM also presents an ontology that can be used to evaluate cybersecurity assurance, however, the main challenge with these ontological schemes or tools is that they are subjective and carried out by individuals based on their perceptions of the risk.

Understanding the potential threats in CPS itself is challenging [26], authors in [14] present a security framework that studies the four main security concerns of CPS, i.e. threats, vulnerabilities, attacks, and controls. The proposed framework can be used to develop effective controls for CPS. The main challenge in CPS security is the increasing number of IoT devices that leads to a rise in the number of vulnerabilities, and eventually leading to successful exploitation [27]. Unlike [14], authors in [12] focus on the impact of cyber attacks on authenticity, confidentiality, reliability, resilience, and integrity. Similar to [14], the main challenges with CPS are raised in [12] and a tree of potential attacks on CPS is proposed. The difference between CPS, IoT, and Industry 4.0 is still very ill-defined, defining layers for each can help security researchers and professionals to develop more concrete security frameworks. Authors in [13] try to differentiate CPS from IoT and traditional information technology systems. The authors also present security issues at various layers of CPS, the affected security parameters, and the associated countermeasures to address these issues. Authors in [28] propose and implement a risk-informed approach that identifies critical CPS assets and the impact of affecting vulnerabilities on a smart grid system and plan to develop a tool to automate the process.

Cyber threat intelligence (CTI) sharing is another risk-informed approach that provides evidence-based knowledge about cyber threats that may exist within any cyber infrastructure. Utilizing such knowledge can be very beneficial in aiding the decision-making process to detect and prevent catastrophic events. However, how and what type of information to share still remains unclear since there is no common definition or ontology available for CTI sharing [29, 30]. Most of the current CTI platforms operate manually and the slow sharing process becomes an obstacle for CTI sharing [31]. On the other hand, certain organizational risks such as free-riding, trust violation, negative publicity, reputational damage, etc. also prevent CTI sharing [32, 33]. Authors in [34, 35] stress the need for rules and regulations for CTI sharing in the existing policies.

Researchers at MITRE took a different approach for CTI. At first, they introduced Common Attack Pattern Enumeration and Classification (CAPEC) in 2007 that provides a range of commonly used attack patterns [36]. Later in 2015 MITRE introduced the Adversarial Tactics Techniques & Common Knowledge (ATT&CK) framework. ATT&CK is a behavioral model that provides specific information on adversary tactics, techniques, and procedures as observed by the community for known actors. ATT&CK can be used for adversary emulation, red teaming, behavioral analytics development, defensive gap assessment, and cyber threat intelligence. The ATT&CK model consists of a set of techniques and sub-techniques that an adversary can take to accomplish their objectives which is represented in the ATT&CK Matrix as shown in [37]. ATT&CK also provides mitigation techniques for preventing the listed adversary techniques and sub-techniques. ATT&CK is further extended to focus on industrial control systems with additional use cases [38].

The aforementioned studies either do not satisfy the evolving security needs of CPS, or highlight the security concerns related to CPS, and propose theoretical concepts to address the same. MITRE ATT&CK on the other hand is a community-based knowledge base with the focal point on adversary emulation and provides threat-actor-based information. A proactive cyber threat intelligence system specifically tailored for CPS to provide contextual information is critically needed. To ensure CPS or any infrastructural security it is vital to understand and identify the 1) various layers and the integrated devices in each layer as seen in Figure. 2, 2) assets that need protection, 3) controls protecting the assets and integrated devices, 4) threats, vulnerabilities, and VDBs, and finally, 5) users and other environmental variables such as running applications, open ports, processes, etc. We provide a context-aware framework that considers all of the above and can be used to mitigate malicious and harmful threats. We discuss the various characteristics of the proposed framework in the following Section.

3. CyVIA System Architecture

This Section introduces the CyVIA architecture and discusses the different integrated components that dynamically interact with each other to create an effective cyber threat intelligence system. CyVIA inputs data from three sources: 1) multiple VDBs, 2) network nodes (configurations, services, running processes, open ports, and so on), and 3) the security policies keeping the network nodes secure on the network such as the applied security controls and other administrative policies. CyVIA produces two types of

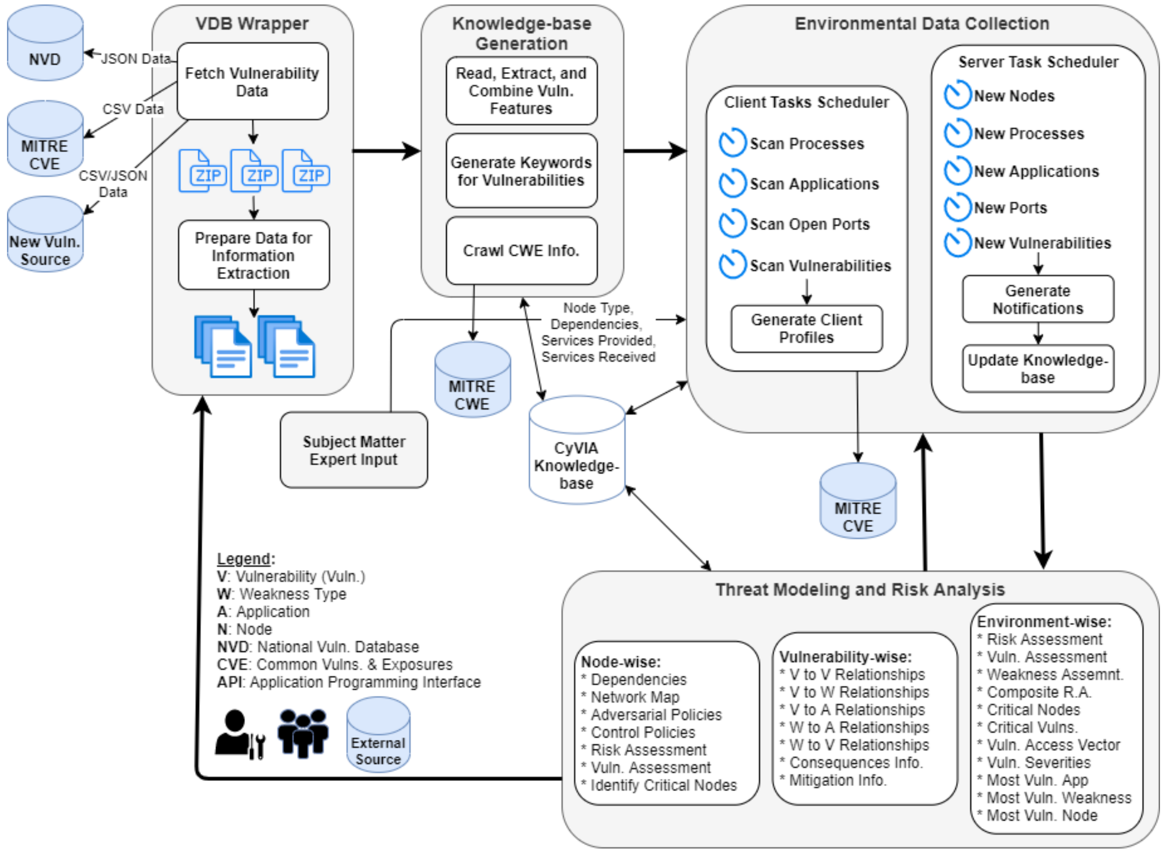

Figure 1. CyVIA System Architecture

output: 1) dynamic informed analysis of changing network configurations and vulnerabilities, and 2) comprehensive analysis of network infrastructure based on the applied security controls and discovered vulnerabilities. In the following subsections, we first go over each component and then describe different phases from the CyVIA architecture as seen in Figure. 1.

3.1. VDB Wrapper

CyVIA is capable of collecting vulnerability data from multiple sources and multiple formats. At present, we collect data from NVD and MITRE, however, CyVIA is capable of integrating data from other sources. As of October 2021, the NVD database contains 172,427 publicly known vulnerability reports. These reports are bundled together in yearly JSON compressed files starting from the year 2001 to date. MITRE on the other hand provides vulnerability groups by weakness types and other attributes such as weakness type description, applicable platforms, modes of introduction, and more in a CSV file format. During this phase, CyVIA collects the multi-formatted datasets from NVD and MITRE and prepares data for extraction during the next phase.

3.2. Knowledge-Base Generation

This phase is responsible for generating a knowledge-base from the collected datasets. This knowledge-base is used by all other components of CyVIA. During this phase, each report item is analyzed and categorized, vulnerability features are extracted, and keywords for each vulnerability are generated. Various information pieces are combined into a comprehensive knowledge-base based on the found relationships in the data points, irrespective of the different data formats. This phase also crawls additional related information from the MITRE website such as parent and child relationships among weakness types. Once the dataset is prepared, the environmental data is collected during the next phase.

3.3. Environmental Data Collection

In this phase, the computing environment or digital assets information is collected. This process has two sub-components (schedulers), a server component that runs on any of the administrator servers, and a client component that runs on all clients. The client and server scheduler communicate and exchange information with each other. The components and sub-components of this phase are discussed in detail as follows:

Schedulers. Providing up-to-date analysis strictly depends on the following factors: 1) how updated the obtained vulnerability information is, and, 2) how updated the network node profiles are. To ensure the up-to-date analysis, CyVIA integrates a scheduler module that has two sub-components:

(i) **Client Task Scheduler**: For command and control, adversaries employ a variety of tactics and protocols after a successful attack to maintain persistence within the target environment. In such cases, most of the related processes execute in the background without user awareness. CyVIA monitors running processes in real-time to alert administrators of any newly detected processes on any of the network nodes. The recorded information for each process includes but is not limited to process id, executing file path, process owner, number of threads, CPU, memory used by the process, etc. Similarly, processes using high memory and CPU are also highlighted during this process for the administrators to take necessary actions if required. Furthermore, any newly installed application, open port, or a vulnerability associated with any of the installed applications is also reported. A client-side scheduler is responsible to keep track of processes, applications, open ports, and vulnerabilities to ensure updated client/node profiles and informed administrator.

(ii) **Server Task Scheduler**: The server-side scheduler captures the changes in information between the server and clients, validates the information, and generates notifications for the administrator about the newly discovered nodes on the network, processes, applications, ports, and vulnerabilities on the network nodes. The server-side scheduler is also responsible to keep the knowledge-base up to date with the latest vulnerability information.

Node Profiling. Any cyber threat intelligence system must collect environmental data specific to the computing environment in order to generate contextual analysis. CyVIA can not only capture changing network configurations on the go, but it can also notify administrators of the changes so that they can take appropriate actions where needed. With the help of a remote agent, CyVIA initially captures the active nodes on the network and their associated information. And, with a local agent running on the detected nodes, this information is refined even further. This process captures and generates node profiles and the IT administrators can fine-tune the profiles as needed. Based on the acquired node information, a node profile contains information such as hostname, IP address, gateway, installed OS, installed apps, open ports, and running processes.

3.4. Subject Matter Expert Input

CyVIA allows the subject matter experts or the administrators to fine-tune various elements where needed. For example, assigning security controls and adversarial risks to nodes on the network, changing the control and adversary weights, overriding the final risk values to get more realistic scores. Once the node profiles are generated, the administrator can define the following information:

(i) **Asset Type**: whether the node is a computer (server, workstation, etc.), a network device (firewall, router, etc.), etc.

(ii) **Control Policy**: states the defensive mechanisms or controls such as technical, physical, or administrative, that are applied on the current node.

(iii) **Adversarial Policy**: defines which types of adversarial risks are applicable on this particular node.

(iv) **Services Provided**: lists the number of services offered by the current node to other nodes on the network.

(v) **Services Received**: if the current node is receiving any services from other nodes on the network, it must be recorded in the node profile.

In the next Section, we discuss controls and policies in detail.

3.5. Control and Adversary Mapping

To protect digital assets and mitigate associated risk factors, cyber defenders deploy several cutting-edge security controls. It is critical to consider these controls while performing cyber risk analysis. CyVIA keeps a record of detailed control information such as control type, assigned weight for each control, a recommended set of controls for different types of network devices, and the administrator-defined control set for a particular type of digital asset. Similarly, different types of adversaries (internal and external) can be defined and assigned weights based on their assumed capabilities. These information pieces are maintained under control master, and the various attributes of control master are as follows:

Control and Adversary Definition. Control definition document contains the master list of available security controls that can be used to secure digital assets. At present, we classify these controls into three main types. 1) Technical Controls

($T = \{T_1, T_2, \ldots, T_8\}$), where T_1=Strong Authentication, T_2=Antivirus/Patches/Updates, ..., T_8=Encryption. 2) Physical Controls ($P = \{P_1, P_2, \ldots, P_6\}$), where P_1=Video Surveillance, P_2=Locks, ..., P_6=Man-traps, and 3) Administrative Controls ($A = \{A_1, A_2, A_3, A_4\}$), where A_1=Security Policy, A_2=Security Training, A_3=Data classification, A_4=NDA Signing [15]. This document is used to specify the control set for each node on the network, representing administrator efforts for securing network nodes or digital assets. And the purpose of the adversary definition document is to define the types of adversaries that the organizational assets are exposed to. At the moment we have four types of adversarial actors: internal employees, and external adversaries with novice, intermediate, and expert expertise. Both of these documents can be expanded as per the organizational needs.

Control and Adversary Weights. Each of the defined controls is assigned a weight value and since the control application varies from asset to asset, we further introduce control application categories M (must have), G (good to have), O (optional) for different types of digital assets. Similarly, the level of protection provided by these controls will vary if the applied controls are exposed to adversarial entities. We assign two different types of weights, 1) NE (not exposed): when the controls are not exposed to the adversarial entities, and 2) E (exposed): when the adversaries are aware of what controls are applied to protect organizational assets. These weights are used to calculate the level of protection that can be expected by the applied controls.

Similarly, the threat posed by humans or adversarial entities is determined by the threat actor's level of access and skill set and it is critical to categorize individuals based on their competence and access location. An inside employee with a given level of access, for example, may pose a different risk than an external experienced attacker. Similar to controls, we categorize adversaries and assign weights based on their skill-set and location.

Master and User-Defined Policies. Master policy document contains the ideal or recommended control configurations for different types of devices on the network. The controls are categorized further into three more categories M, G, and O as explained earlier. Network devices are categorized into seven different types: 1) Servers: server computers providing services to other nodes on the network, 2) Workstations: client computers receiving services from servers, 3) Portables: portable devices such as laptops, tablets, etc., 4) Network: networking equipment such as routers, switches, access points, etc., 5) Network Security: firewall, IPS/IDS, etc.), 6) Storage: USB, Optical Disk, SAN, NAS, etc., 7) IoT: any device connecting to the network not classified in above categories.

For each type of device, the master policy holds a recommended M, G, O control that determines how secure the node is in terms of control security. For example, a server device must have the controls T1-T3, whereas T4 is good to have: "Server": ["T1:M", "T2:M", "T3:M", "T4:G", ...]. Each node profile specifies whether these recommended controls are applied or not. For example when T1-T3 are applied and T4 not applied: "ControlPolicy": ["T1:1", "T2:1", "T3:1", "T4:0", ...]. Similar to control mapping, adversarial threats are also mapped within node profiles for each node. If a particular control or threat is applied or applicable to a node, it will be represented by the value 1, otherwise by 0 stating that the control or threat is not applied or applicable. For example, a CCTV control and an external adversarial threat may not be applicable for a standalone scanner.

Ideally, each device under the same device category should have the same controls applied as per the defined control policy, however, it can change as per the network administrator's approval. CyVIA allows the administrators to have custom user-defined policies as per their needs. Another use case for this scenario is the third-party devices with limited access rights and policy options such as a DVR for CCTV recording. Administrators can further secure these devices by employing custom physical (locks) or administrative controls (policies).

3.6. Threat Modeling and Risk Analysis

This phase is mainly responsible for generating contextual analyses for the computing environment being analyzed.

Interdependency Between Nodes – Service Mapping. Dependencies between network nodes present a different set of challenges for a cyber defender. Because risk scores are usually centered on network/infrastructure, we add the dependency factor for nodes, which represents the number of service dependents for a node [15]. The higher the number of dependents, the more important the node is in the network. CyVIA is capable of generating the network map of the given infrastructure as well as service dependencies. The recorded information under each node's profiles is used to map the services that node K_i delivers to node K_j on the network. CyVIA's dependency map illustrates the service dependencies between network nodes and aids the administrator in identifying crucial network nodes. We keep track of services provided (service:port) and services received (IP:port) by every node on the network.

Severity of Nodes. How critical a node on the network is, can be determined by what risk the network node is introducing to the infrastructure. In our case, we consider the following factors while calculating risk scores:

(i) **Control-Based Risk**: This risk informs the administrator about what amount of protection should be expected from the applied security controls in light of adversarial threats.

(ii) **CVSS-Based or Vulnerability-Based Risk**: How vulnerable each node on the network and the overall infrastructure is seeing the discovered vulnerabilities.

By aggregating both scores, we can label the most critical nodes on the network that require urgent attention from the administrator to improve the general welfare of the network. Furthermore, the critical nodes can also be identified by analyzing the number of open ports vs actual dependents.

Potential Consequences and Mitigation. Once the vulnerabilities within the specific infrastructure have been identified, CyVIA can educate the administrator about the potential consequences of the discovered vulnerabilities as well as mitigation strategies that may be utilized to prevent such exploitation. For example, vulnerabilities under the category CWE-5, i.e. "J2EE Misconfiguration: Data Transmission Without Encryption" target the "Integrity" metric and are capable of modifying the application data. Using SSL or encryption for all access-controlled sites is a mitigation strategy that can be utilized to avoid such exploitation.

3.7. Assumptions and Limitations

Assumptions. We have considered the following assumptions for CyVIA: 1) we assume that various CVE features, such as CVSS scores, CWE IDs, Severity values, etc., stored in the NVD are correctly assigned. 2) Because NVD is fed by MITRE data, and CWE is managed by MITRE, we take the final CWE features from MITRE. 3) The final list of possible vulnerabilities is matched with MITRE's CVE search engine. 4) We use a Raspberry Pi as a device on the perception layer that represents IoT devices and communicates with different sensors for data collection. 5) Due to limited resources, we are unable to deploy CyVIA on a live large network, however, we have conducted several trials of CyVIA on various network clusters containing different versions of Microsoft Windows and Linux, and we are confident that it can be deployed on any large network.

Limitations. CyVIA at this point is limited to: 1) Local agent that can capture information from nodes running Windows 7 onward, having power-shell script execution enabled. And for Linux, we have tested agents on Ubuntu, Kali, Debian, and Fedora. 2) Services offered by nodes are captured through the remote scan, however, the nodes utilizing these services are identified by the administrator.

Integration Overview. A cyber defender present within the target network is capable of interacting with all components of CyVIA whereas limited interaction with different components is available from outside the network using the API.

4. Comparative Study and Evaluation of CyVIA

We evaluate CyVIA on a large VM setup having different clusters of nodes, representing different parts of the network. Nodes are mapped and evaluated during this process. Table 1 lists the subset cluster being evaluated in this Section, its nodes, their IP addresses, and the installed OS. All nodes have a default set of applications installed and a few custom applications such as MySQL, SQL Server, etc. to create dependencies between nodes. The node cluster includes nodes from each layer as seen in Figure 2. We selected three state-of-the-art vulnerability scanning tools, Nessus Essentials by Tenable, InsightVM by Rapid7, and Greenbone Security Manager (GSM) by Greenbone, and scanned the network using these tools. We also scanned the network using CyVIA.

Table 1. Network Node List

Node	IP	OS
Win7	50.50.50.4	Windows 7 ENT
Win81	50.50.50.5	Windows 81 ENT
Win10	50.50.50.6	Windows 10 ENT
Windows11	50.50.50.7	Windows 11 Pro
Server2012	50.50.50.8	Server 2012 R2
Server2016	50.50.50.9	Server 2016 Datacenter
Centos	50.50.50.23	Centos 8.3.1
Debian	50.50.50.24	Debian 10
Fedora	50.50.50.25	Fedora 33
OpenSUSE	50.50.50.26	OpenSUSE 15.2 1
Raspbian	50.50.50.27	Raspbian
Ubuntu16	50.50.50.28	Ubuntu 16 LTS
Ubuntu18	50.50.50.29	Ubuntu 18 LTS
Ubuntu20	50.50.50.30	Ubuntu 20 LTS

In the following subsections, we initially discuss the findings by CyVIA and then for each tool followed by a comparison between the four. Please note that we only provided the node IPs and OS credentials to each tool for scanning and kept everything else as default. Each tool was installed on a fresh virtual machine with no other application installed or running, and assigned 8GB of RAM and 2 threads of Intel i7 processor.

4.1. Analysis by CyVIA

CyVIA is capable of generating contextual information based on the network nodes, applied security controls

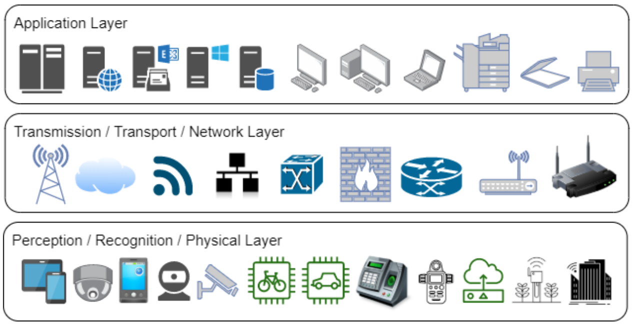
Application Layer
Transmission / Transport / Network Layer
Perception / Recognition / Physical Layer

Figure 2. Layers

and policies on these nodes, and the found vulnerabilities within the installed OS and applications on these nodes. Therefore, the execution process is slightly different as compared with other tools. In the following subsections, we discuss the major components, their execution, and responsibilities.

Node Profiling. CyVIA is capable of detecting network nodes using the scheduler module. Once a node is detected, CyVIA tries to obtain node information remotely using a profiling agent. Based on the information captured in this process, further analyses are generated, therefore, it is critical to verify and update each node profile to have the most accurate results. The scheduler module has two sub-components, a client-side scheduler, and a server-side scheduler, responsible for evaluating the changes in node profiles. These schedulers work closely with the profiling agents. A server-side profiling agent captures node profiles remotely, and a client-side profiling agent runs on each client.

(i) **Server Side Scheduler**: CyVIA keeps track of changes by closely monitoring the recorded node profiles and any new observed changes on the network. For example, any newly discovered node(s), process(es), application(s), or vulnerabilities are highlighted in this process. The server-side scheduler relies more on the recorded information and the remote profiling agent. The following output sample shows the server-side scheduler execution where a network id is required to start monitoring the specified network. The recorded information is displayed for each node and in case of any change, it is highlighted for consideration. The server-side scheduler schedules tasks to run after every few minutes to keep track of changes.

```
Please provide network id: 50.50.50.0
Server scheduler started at 21:21:19
21:21:19 Fetching existing data...
Win10: [Processes: 55 , Users: 3 , Apps: 5 ,
Open Ports: 19 , Vulnerabilities: 227]
... more ...
Ubuntu20: [Processes: 187 , Users: 12 , Apps:
9 , Open Ports: 2 , Vulnerabilities: 430]
** Starting network scanner at 21:22:19
Found 10 alive hosts. Newly discovered
node(s) 2
** New host(s): ['50.50.50.90',
'50.50.50.99']
21:26:20 Looking for changes in node
processes, applications, and ports...
** 4 New process(es) found **
Win10: ['SystemSettingsBroker.exe',
'sppsvc.exe', 'SppExtComObj.Exe',
'ApplicationFrameHost.exe']
** 1 New application(s) found **
Win10: ['Free Cam 8']
No new open ports found.
No new vulnerabilities found.
... more ...
```

We can see that 2 new nodes on the network are found, and 4 new processes with 1 new application on the Win10 node are detected and prompted in the above sample.

(ii) **Remote Profiling Agent**: CyVIA initially detects network nodes remotely and tries to obtain individual node information using a remote profiling agent as shown previously in the output sample. During this process, not necessarily all nodes are discovered depending on the security settings on each node. The undiscovered node(s) information is further captured with the help of the local profiling agent discussed next. This process took ≈ 10 minutes in our case of 14 nodes network. The information captured is stored and the sample output is as follows:

```
Please provide router IP / Network ID:
50.50.50.1
Scanning network please wait...
Found host: 50.50.50.1
Found host: 50.50.50.5
... more ...
Total alive hosts: 9
Scanning hosts, please wait...
Collecting information for the IP 50.50.50.1
Host: 50.50.50.1, State: up
OS Vendor: Linux, OS: Linux, OS Ver: 2.6.X,
OS Type: general purpose, Accuracy percent:
100.
Running protocol(s) : tcp
port : 22 state : open
port : 80 state : open
port : 443 state : open
... more ...
```

(iii) **Client Side Scheduler**: Client Scheduler is responsible for monitoring any new process, application, or vulnerability on the client-side.

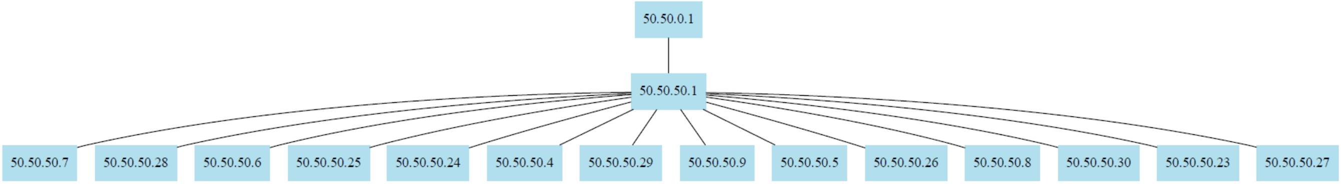

Figure 3. CyVIA Network Map

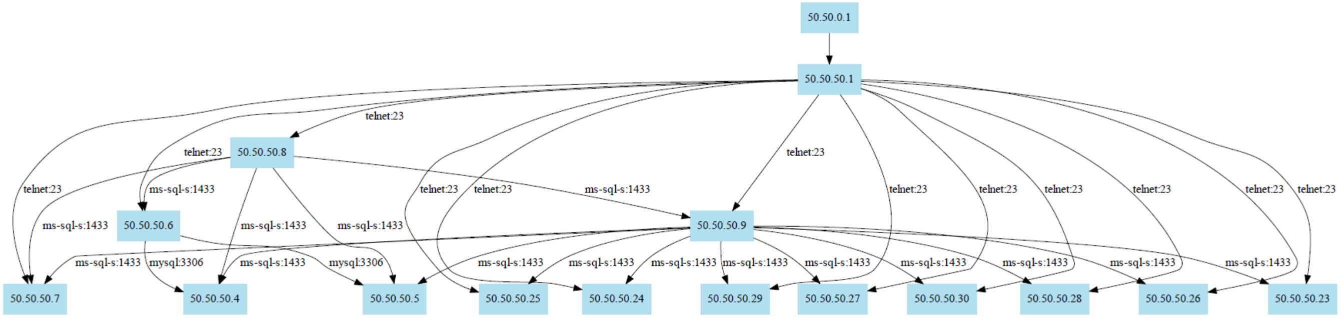

Figure 4. CyVIA Dependency Map

The discovered items are reported to the server scheduler for further action. The client-side scheduler also schedules tasks to run after every few minutes to keep track of changes. The sample output is shown below.

```
Client scheduler started at 20:21:03
** Starting process scanner at 20:22:03
HostName: Win10 HostIP: 192.168.0.199
Running Processes: 55 Previously recorded: 55
Finding new processes if any...
## New application ApplicationFrameHost.exe
found (not recorded previously) with 55
processes:
."ApplicationFrameHost.exe": [
... "pid": 3796,
... "exe": "C:/Windows/System32/
ApplicationFrameHost.exe",
... "username": "WIN10/IEUser",
... "num_threads": 1,
... "cpu_percent": 0.0,
... "memory_percent": 0.5453594831106481,
... "cpu_times": [
...... 0.125,
...... 0.21875,
...... 0.0,
...... 0.0
... ]
. ]
... more ...
** New PID 2100 under MsMpEng.exe
** New PID 4996 under NisSrv.exe
** New PID 1444 under OneDrive.exe
Found 1 new application(s) and 3 new
process(es) among the 55 currently running.
Recording newly discovered process... Done.
```

(iv) **Local Profiling Agent**: With the help of the local administrator, a local agent can be deployed and executed on each node on the network that captures the remaining pieces of information required to complete the node profiles. This process takes ≈ 1 minute on each node and ≈ 14 minutes for the entire network. The administrator can verify the captured information and fine-tune node profiles as discussed in Sections 3.3 and 3.4.

Interdependency Between Nodes – Service Mapping. Complete node profiles allow CyVIA to generate network and dependency diagrams as seen in Figure. 3 and Figure. 4. This allows the administrator to understand the network hierarchy and service load on each node. We can see that node 50.50.50.1 and 50.50.50.9 has a higher number of service dependants as compared with other nodes on the network. This process takes a few seconds to generate the analysis.

Severity of Nodes. Nodes can be flagged critical in several ways, as discussed above, a node with the highest number of dependents is also critical for the network. Two major risk categories used by CyVIA to flag critical nodes are as follows:

(i) **Control-Based Risk**: During this process, CyVIA at first ensures that the control documents exist and respective weights are assigned to each of the categories. After this, each node is analyzed in terms of control security based on the control and adversarial policy applied to each node. The following output sample shows an analysis of three different cases during this process.

```
Host: Win11, IP: 50.50.50.7
```

Table 2. CyVIA Node and Infrastructure-based Control Risk

IP	Node	M_Av	M_Ap	G_Av	G_Ap	O_Av	O_Ap	Deps.	NR	CR
50.50.50.1	Router1	12	12	2	2	2	2	13	0.00	0.00
50.50.50.9	Win2016	12	12	3	3	2	2	12	0.00	0.00
50.50.50.8	Win2012	12	5	3	2	2	1	6	0.51	0.11
50.50.50.6	Win10	7	3	2	2	7	3	3	0.43	0.10
50.50.50.27	Raspbian	7	0	2	0	7	0	1	1.00	0.22
50.50.50.5	Win8	7	4	2	0	7	6	1	0.54	0.12
50.50.50.24	Debian10	7	2	2	2	7	4	1	0.51	0.11
50.50.50.4	Win7	7	4	2	1	7	5	1	0.44	0.10
50.50.50.29	Ubuntu18	7	6	2	0	7	5	1	0.37	0.08
50.50.50.25	Fedora33	7	7	2	0	7	4	1	0.29	0.06
50.50.50.23	Centos	7	6	2	1	7	7	1	0.21	0.04
50.50.50.30	Ubuntu20	7	5	2	2	7	6	1	0.20	0.04
50.50.50.7	Win11	7	7	2	2	7	7	1	0.00	0.00
50.50.50.28	Ubuntu16	7	7	2	2	7	7	1	0.00	0.00
50.50.50.26	openSUSE	7	7	2	2	7	7	1	0.00	0.00

```
Must have controls: 7 applied, out of 7
Good to have controls: 2 applied, out of 2
Optional controls: 7 applied, out of 7
All recommended controls applied
Host: Win10, IP: 50.50.50.6
Must have controls: 3 applied, out of 7
Good to have controls: 2 applied, out of 2
Optional controls: 3 applied, out of 7
Recommended controls not applied: ['T2:M ->
T2:O', 'T6:M -> T6:O', 'A2:M -> A2:O', 'A4:M
-> A4:O']
Matched controls: ['T1:M', 'T3:G', 'T4:O',
'T5:O', 'T7:G', 'T8:O', 'P1:O', 'P2:O',
'P3:O', 'P4:O', 'P5:N', 'P6:N', 'A1:M',
'A3:M']
Host: Raspbian, IP: 50.50.50.27
Must have controls: 0 applied, out of 7
Good to have controls: 0 applied, out of 2
Optional controls: 0 applied, out of 7
Recommended controls not applied: ['T1:M ->
T1:O', 'T2:M -> T2:O', 'T3:G -> T3:O', 'T6:M
-> T6:O', 'T7:G -> T7:O', 'A1:M -> A1:O',
'A2:M -> A2:O', 'A3:M -> A3:O', 'A4:M ->
A4:O'] Matched controls: ['T4:O', 'T5:O',
'T8:O', 'P1:O', 'P2:O', 'P3:O', 'P4:O',
'P5:N', 'P6:N']
... more nodes ...
```

In the above example, workstation 50.50.50.7 has all controls applied, workstation 50.50.50.6 is missing 4 must have controls and 4 optional controls, and workstation 50.50.50.27 has no controls applied. Must have controls are highlighted whereas the optional controls are ignored because they are optional. Table 2 lists the network nodes, applied controls, number of dependents, associated node-based, and infrastructure-based control (CR). We can see that node 50.50.50.27 (Raspbian) has no security control applied (M_Ap, G_Ap, O_Ap) and it is at a high risk of 100% (NR), followed by workstation 50.50.50.5 (Win8) at 54%. We can also see that nodes 50.50.50.1 and 50.50.50.9 have the highest number of dependents (Deps.). Please note that nodes with risk 0 do not mean they are 100% secure. This process also takes a few seconds to execute.

(ii) **Vulnerability-Based Risk**: CyVIA flags nodes based on the number of vulnerabilities found in each. There may be a case where on one hand a node has a higher number of reported vulnerabilities most medium or low severities. And on the other hand, a node with a high number of high severity vulnerabilities. CyVIA is not only capable of highlighting both cases, but also the applications with the highest numbers of reported vulnerabilities and their classifications. Table 3 provides a summary of node-based vulnerabilities (Total), the number of applications (Apps.), open ports (O.P.), control-risk (CR), vulnerability-risk (VR), and the aggregated risk (TR). We can see that node 50.50.50.27 (Raspbian) has the highest number of vulnerabilities (30%), highest control risk (22%), and the highest risk portion within the infrastructure (26%). This process takes ≈ 1 minute, and depending on the number of applications installed on a node it can take up to 4 minutes. For our network, it took ≈ 20 minutes to complete the analysis.

Table 3. CyVIA Infrastructure-based Risk Summary

IP	Node	O.P.	Apps.	High	Med.	Low	Total	CR	VR	TR
50.50.50.27	Raspbian	2	1,824	4,070	6,326	909	11,305	0.22	0.30	0.26
50.50.50.8	Win2012	39	43	4,780	2,957	888	8,625	0.11	0.23	0.17
50.50.50.9	Win2016	17	42	6,347	3,657	588	10,592	0.00	0.28	0.14
50.50.50.24	Debian10	1	1,618	965	2,043	423	3,431	0.11	0.09	0.10
50.50.50.5	Win8	15	25	572	570	253	1,395	0.12	0.04	0.08
50.50.50.6	Win10	19	30	329	679	92	1,100	0.10	0.03	0.06
50.50.50.4	Win7	10	23	113	56	6	175	0.10	0.00	0.05
50.50.50.29	Ubuntu18	4	16	106	182	28	316	0.08	0.01	0.04
50.50.50.25	Fedora33	2	1,740	3	9	4	16	0.06	0.00	0.03
50.50.50.30	Ubuntu20	2	12	146	239	47	432	0.04	0.01	0.02
50.50.50.23	Centos	2	1,403	6	3	0	9	0.05	0.00	0.02
50.50.50.7	Win11	17	14	2	7	9	18	0.00	0.00	0.00
50.50.50.28	Ubuntu16	2	15	86	162	21	269	0.00	0.01	0.00
50.50.50.1	Router1	1	1	8	10	9	27	0.00	0.00	0.00
50.50.50.26	openSUSE	5	2,320	25	23	3	51	0.00	0.00	0.00

Table 4. CyVIA Infrastructure-based Top 10 Most Vulnerable Products

S#	Product	CVEs	CWEs
1	Microsoft MPI ...	6,377	97
2	jackd 5+nmu1	3,070	87
3	chromium 90.0.4430...	1,468	59
4	Windows 8.1 Enterprise	1,107	62
5	Windows Server 2012 R2	949	42
6	ssh 1:7.9p1-10	748	73
7	SQL Server 2017	640	37
8	zip 3.0-11+b1	584	54
9	SQL Server 2017	516	14
10	SQL Server 2017	516	14

Additional Analysis. CyVIA produces various analyses that play a significant role in securing the cyber infrastructure. Table 4 provides information about the top 10 most vulnerable products with the highest number of vulnerabilities and their associated weakness types found by CyVIA. Table 5 on the other hand provides information on which product has the highest observed mean, max, and mode scores. Although Microsoft MPI has the highest number of reported vulnerabilities (6,377), however, simple-scan has the highest vulnerability scores, meaning it is more vulnerable as compared with Microsoft MPI. Furthermore, Table 6 spotlights the top 10 weakness types, their percentage and count. For example, 12.20% vulnerabilities fall under SQL injection type and 11.15% are related to buffer overflow.

Figure 5 provides information on the open ports found on each node versus the actual number of dependents. For example, node Win2012 has 39 ports open whereas the actual number of dependents is only 6. This raises a red flag for the administrator. Figure 6 illustrates an overview of control and vulnerability risk. Node Raspbian has the highest control and vulnerability risk as compared with all other nodes, whereas nodes Win11, Router1, and OpenSUSE15 have

Table 5. CyVIA Infrastructure-based Top 10 Mean, Max, and Mode Scores

Product	Mean	Max	Mode
simple-scan 3.30.1.1-1+b1	10.0	6.40	10.0
gpicview 0.2.5-2+b1	10.0	6.39	10.0
tcl8.6 8.6.9+dfsg-2	10.0	10.00	10.0
lp-solve 5.5.0.15-4+b1	10.0	6.40	10.0
SolarWinds Collector	10.0	10.00	10.0
mscompress 0.4-3+b1	10.0	6.40	10.0
eog 3.28.4-2+b1	10.0	6.38	10.0
enchant 1.6.0-11.1+b1	10.0	6.42	10.0
user-setup 1.81	10.0	7.03	10.0
whiptail 0.52.20-8	10.0	10.00	10.0

Table 6. CyVIA Infrastructure-based Top 10 Weakness Types

Description	%	Count
Other	14.73	5,563
SQL Injection	12.20	4,607
Buffer Overflow	11.15	4,211
Insufficient Information	8.97	3,388
Improper Input Validation	6.17	2,330
Cross-site Scripting	5.55	2,095
Unauthorized Access	5.45	2,057
Access Controls	4.70	1,774
Resource Management Errors	3.18	1,200
Code Injection	3.11	1,176

very low risks. Figure 7 provides the percentage of vulnerability severities and access vector. Among the found vulnerabilities, 46.5% are high severity and 83.5% can be exploited through network access. Table 7 provides further statistics related to the found vulnerability severities. We can observe a low standard deviation for the high severity vulnerabilities meaning most high severity vulnerabilities are closer to the mean value i.e. 8.29, which can also be noticed by the percentile values. Figure 8 highlights the top 10 CVEs found among the current network nodes. Similarly CyVIA is capable of highlighting common CVEs across different products or the vulnerabilities that are present in multiple products. This is very helpful for generating relational analysis.

```
CVE-2010-1444: [’vlc 3.0...’, ’zip 3.0..’]
CVE-2018-6559: [’Ubuntu16...’, ’Ubuntu18...’,
’Ubuntu20...’]
CVE-2015-0095: [’Microsoft MPI...’, ’Windows
8.1...’, ’Server2012...’]
CVE-2017-9383: [’curl 7.47...’, ’curl 7.64...’,
’wget 1.20...’, ’curl 7.58...’, ’curl 7.68...’]
```

Table 7. CyVIA Infrastructure-Based Vulnerability Severity Analysis

Sv.	Count	Mean	Std.	Min	50%	75%	Max
H	17558	8.29	1.0	7.1	7.6	9.3	10.0
M	16923	5.30	1.0	4.0	5.0	6.5	6.9
L	3281	2.58	0.7	1.0	2.1	3.5	3.8

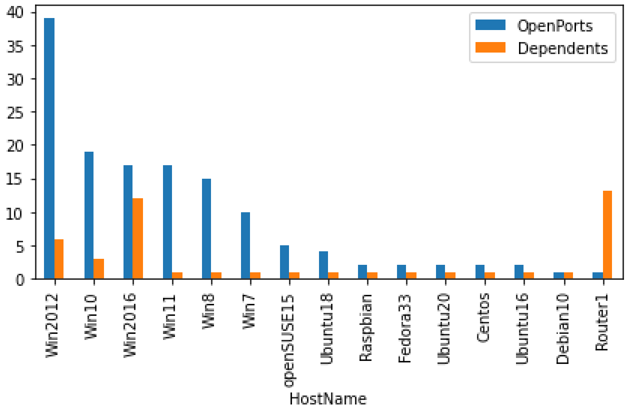

Figure 5. CyVIA Infrastructure-based Open Ports vs Dependents

Furthermore, CyVIA provides detailed information about each network node. For example, CVE-2019-12068 is the most common vulnerability among the 11,305 found vulnerabilities on the high-risk node (Raspbian). This vulnerability is basically a software bug (an infinite loop) that can lead to a successful denial-of-service attack. 36% of these vulnerabilities are high severity, 53.5% can be exploited via the network, and the majority of vulnerabilities belong to the class "Other," followed by "Cross-site Scripting". On the given network cluster, there are 37,761 vulnerabilities found in total and for 156 vulnerabilities, no information is found within CyVIA dataset. These are newly discovered vulnerabilities for which relational information within the CyVIA dataset was not present at the time of scan. The server-side scheduler is responsible to update vulnerability information and is currently set to update once a week.

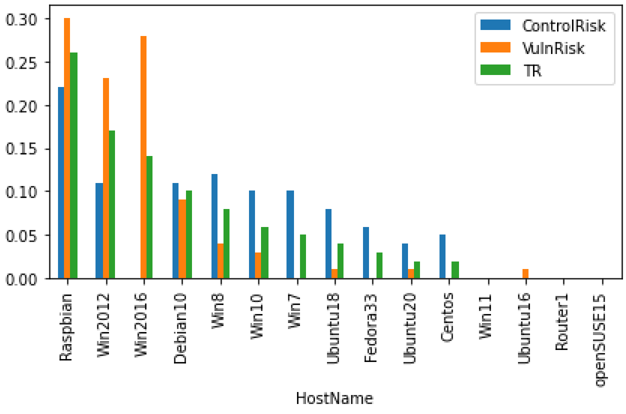

Figure 6. CyVIA Infrastructure-based Control and Vulnerability Risks

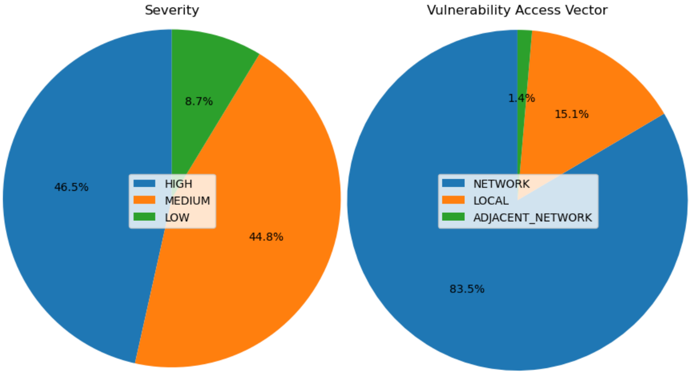

Figure 7. CyVIA Infrastructure-based Severity and Vulnerability Access Vector

Potential Consequences and Mitigation. After identifying the found weaknesses in the infrastructure, CyVIA is capable of educating the cyber defender about the common consequences caused by the found weaknesses and at the same time how to mitigate them. The sample output below provides the information about CWE-200 i.e. unauthorized access.

```
CWE-200 - Exposure of Sensitive Information to an
Unauthorized Actor...
CWE-200 - Common Consequences:
. Confidentiality:
.. IMPACT:
```

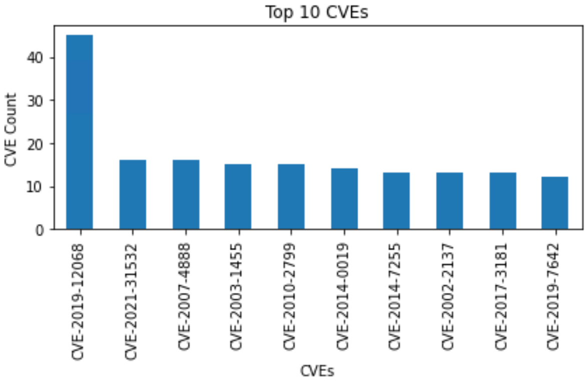

Figure 8. CyVIA Infrastructure-based Top 10 CVEs

```
.. - Read Application Data
CWE-200 - Potential Mitigations:
. Architecture and Design:
.. DESCRIPTION:
.. - Compartmentalize the system to have safe
areas where trust boundaries can be ... more ...
.. STRATEGY:
.. - Separation of Privilege.
```

4.2. Analysis by other Tools

The trial version of Nessus Essentials allows scanning of up to 16 nodes on the network. Nessus results showed asset classification based on vulnerability severity as seen in Table 8. On the other hand, Nessus also provides remediation information for the found vulnerabilities. As per the results, node 50.50.50.8 i.e. a Windows Server 2012 R2 has the highest number of found vulnerabilities followed by 50.50.50.26 (OpenSuse 15.2.1). Time taken by Nessus to scan the network was ≈ 33 minutes.

Table 8. Nessus Results

Node	Critical	High	Med	Low	T.
Server2012	**25**	**238**	**83**	8	**354**
OpenSUSE	29	106	63	5	**203**
Debian10	36	85	18	1	**140**
Server2016	16	51	23	0	**90**
Fedora33	5	49	22	1	**77**
Ubuntu16	7	13	6	0	**26**
Win81	0	2	8	1	**11**
Ubuntu18	3	5	3	0	**11**
Centos831	1	5	3	1	**10**
Raspbian	1	6	1	0	**8**
Win7	1	1	1	0	**3**
Win10	0	0	2	0	**2**
Win11	0	0	2	0	**2**
Ubuntu20	0	1	1	0	**2**

InsightVM by Rapid7 allows the creation of sites and asset assignments to each site making asset management much easier. InsightVM keeps track of asset risk over time, providing a classification of assets by OS (Windows, Linux, etc.), exploitability (by adversary skill e.g. novice, intermediate, expert, etc.), vulnerabilities, exploits, malware, and risk scores. InsightVM also keeps track of software packages and services. The results generated are shown in Table 9. It was observed that node 50.50.50.9 (Server2016) has the highest number of vulnerabilities, however, node 50.50.50.8 (Server2012) has the highest risk score value compared with node 50.50.50.9. Time taken by InsightVM to scan the network was ≈ 10 minutes.

Table 9. InsightVM Results

Node	Exploits	Malw.	Vulns.	Risk
Server2016	**134**	**0**	**1,946**	**670,047**
Server2012	342	10	**1,798**	700,461
OpenSUSE	11	0	**751**	159,590
Debian10	9	0	**595**	162,486
Win7	16	0	**567**	170,516
Win10	4	0	**142**	30,080
Raspbian	0	0	**63**	11,682
Ubuntu16	0	0	**61**	17,308
Centos831	1	0	**56**	11,411
Win81	3	0	**37**	16,718
Ubuntu18	0	0	**17**	8,508
Win11	0	0	**4**	845
Ubuntu20	0	0	**4**	1,495
Fedora33	0	0	**3**	742

Table 10. GSM Results

Node	Sev.	Score	H.	M.	L.	T.
OpenSUSE	High	9.3	39	110	10	**159**
Debian10	High	10	48	80	7	**135**
Fedora33	High	7.2	9	13	5	**27**
Win10	High	7.7	2	14	0	**16**
Win81	High	7.8	3	7	1	**11**
Server2012	High	10	1	9	1	**11**
Win7	High	7.8	3	4	1	**8**
Raspbian	High	7.5	1	3	2	**6**
Ubuntu16	Med.	4.9	0	2	2	**4**
Ubuntu18	High	10	1	2	1	**4**
Win11	Med.	5	0	1	1	**2**
Server2016	Med.	5	0	1	1	**2**
Centos831	Med.	4.3	0	1	1	**2**
Ubuntu 20	Low	2.6	0	0	2	**2**

Open Vulnerability Assessment System (OpenVAS) has become a part of Greenbone Vulnerability Manager (GVM) which is still available to the community. GSM, on the other hand, is the professional edition and is only available under multiple licensing options similar to InsightVM and Nessus. GSM classifies the nodes by severity of nodes and OS severity based on the found vulnerabilities. GSM also generates network topology based on the found network nodes and keeps track of open ports and installed packages. Results generated by GSM are shown in Table 10. We can see that node 50.50.50.26 (OpenSUSE 15.2 1) has the highest number of vulnerabilities found, followed by 50.50.50.24 (Debian 10). Time taken by GSM to scan the network was ≈ 48 minutes.

4.3. Comparison of CyVIA with Other Tools

Each tool has some strengths that make the tool better than the other, for example, Greenbone tools are open-sourced and still available to the community whereas Tenable and Rapid7 products are not. Tenable provides customize-able reports options whereas Greenbone products do not offer such rich reporting options. Rapid7 on the other hand provides a very informative interface and customize-able reports as well. Among the three tools, Greenbone is very stable and ran without any issues, whereas Rapid7 took the minimum time for scanning the network and generating analysis. One main difference between these tools and CyVIA is that all three generate on-demand analysis whereas CyVIA provides dynamic risk assessment and keeps the administrator informed at all times for any changes in node configurations or risk. Table 11 lists the vulnerability counts by all four tools, however, CyVIA provides further details of contextual cyber risk assessment that is very useful for the administrator.

Table 11. Tool Comparison in Terms of Detected Vulnerabilities

Node	CyVIA	O.VAS	Nessus	Nexpose
Win7	175	8	3	567
Win81	1,395	11	11	37
Win10	1,100	16	2	142
Windows11	18	2	2	4
Server2012	8,625	11	354	1,798
Server2016	10,592	2	90	1,946
Centos831	9	2	10	56
Debian10	3,431	135	140	595
Fedora33	16	27	77	3
OpenSUSE	51	159	203	751
Raspbian	11,305	6	8	63
Ubuntu16	269	4	26	61
Ubuntu18	316	4	11	17
Ubuntu20	432	2	2	4

The number of observed vulnerabilities is higher in CyVIA because CyVIA considers the vulnerabilities in the OS and each of the user-installed applications. CyVIA provides deeper insights into the overall infrastructure-based risk as well as node-based risk highlighting various critical areas, whereas the other tools simply focus on individual nodes.

5. Conclusion and Future Work

Heterogeneity in cyberspace has introduced a wide spectrum of weaknesses and uncertainties for cyber defenders to defend against. In such a scenario, keeping the organizational infrastructure safe is a major challenge. We propose a threat intelligence system CyVIA, that provides contextual cyber situational awareness to a cyber defender. CyVIA considers various key elements that play a significant role in evaluating organizational cybersecurity. We evaluate CyVIA on a network cluster and compare the results with the state-of-the-art. Our results indicate that CyVIA provides an extensive amount of analyses indicating infrastructure-based loopholes as compared with other tools. In the future, we plan to 1) deploy CyVIA on a large network for evaluation, 2) integrate the AI engine of CyVIA for evaluating and predicting risks and provide recommendations to a cyber defender on where to focus, and 3) allow the cyber defender to add additional risk layers to the framework to expose high-risk nodes based on custom criteria. We also plan to introduce CyVIA API for coordinated vulnerability disclosure and CyVIA as a service that can be accessed from anywhere.

Acknowledgement. This research was supported partially by U.S. Department of Energy's Office of Fossil Energy (FE) Award # DE-FE0031744.

References

[1] Database, N.V. (2021 (Online; accessed Aug 1, 2021)) *NVD Yearly Report*. https://bit.ly/2H5sOGI.

[2] Redscan (2021 (Online; accessed Aug 1, 2021)) *Redscan Report 2021*. https://bit.ly/3gtr0a1.

[3] Umezawa, K., Mishina, Y. and Takaragi, K. (2019) Threat analyses using vulnerability databases—possibility of utilizing past analysis results—. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)* (IEEE): 1–6.

[4] Ghani, H., Luna, J., Khelil, A., Alkadri, N. and Suri, N. (2013) Predictive vulnerability scoring in the context of insufficient information availability. In *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)* (IEEE): 1–8.

[5] Syed, R. (2020) Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system. *Information & Management* **57**(6): 103334.

[6] Kritikos, K., Magoutis, K., Papoutsakis, M. and Ioannidis, S. (2019) A survey on vulnerability assessment tools and databases for cloud-based web applications. *Array* **3**: 100011.

[7] Chalvatzis, I., Karras, D.A. and Papademetriou, R.C. (2019) Evaluation of security vulnerability scanners for small and medium enterprises business networks resilience towards risk assessment. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (IEEE): 52–58.

[8] Mburano, B. and Si, W. (2018) Evaluation of web vulnerability scanners based on owasp benchmark.

In *2018 26th International Conference on Systems Engineering (ICSEng)* (IEEE): 1–6.

[9] Weerawardhana, S., Mukherjee, S., Ray, I. and Howe, A. (2014) Automated extraction of vulnerability information for home computer security. In *International Symposium on Foundations and Practice of Security* (Springer): 356–366.

[10] Mu, D., Cuevas, A., Yang, L., Hu, H., Xing, X., Mao, B. and Wang, G. (2018) Understanding the reproducibility of crowd-reported security vulnerabilities. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*: 919–936.

[11] Roldán-Molina, G., Almache-Cueva, M., Silva-Rabadão, C., Yevseyeva, I. and Basto-Fernandes, V. (2017) A comparison of cybersecurity risk analysis tools. *Procedia computer science* **121**: 568–575.

[12] Alguliyev, R., Imamverdiyev, Y. and Sukhostat, L. (2018) Cyber-physical systems and their security issues. *Computers in Industry* **100**: 212–223.

[13] Ashibani, Y. and Mahmoud, Q.H. (2017) Cyber physical systems security: Analysis, challenges and solutions. *Computers & Security* **68**: 81–97.

[14] Humayed, A., Lin, J., Li, F. and Luo, B. (2017) Cyber-physical systems security—a survey. *IEEE Internet of Things Journal* **4**(6): 1802–1831.

[15] Malik, A.A. and Tosh, D.K. (2020) Quantitative risk modeling and analysis for large-scale cyber-physical systems. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)* (IEEE): 1–6.

[16] Webroot (2021 (Online; accessed Aug 1, 2021)) *Social Engineering, examples & prevention*. https://bit.ly/3tIJirG.

[17] Zhang, Y., Li, Y., Deng, W., Huang, K. and Yang, C. (2021) Complex networks identification using bayesian model with independent laplace prior. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **31**(1): 013107.

[18] Zhang, Y., Yang, C., Huang, K., Jusup, M., Wang, Z. and Li, X. (2020) Reconstructing heterogeneous networks via compressive sensing and clustering. *IEEE Transactions on Emerging Topics in Computational Intelligence* .

[19] Musman, S. and Turner, A. (2018) A game theoretic approach to cyber security risk management. *The Journal of Defense Modeling and Simulation* **15**(2): 127–146.

[20] Institute, P. (2021 (Online; accessed Aug 1, 2021)) *Ponemon Library*. https://www.ponemon.org/research/ponemon-library/.

[21] Institute, P. (2021 (Online; accessed Aug 1, 2021)) *Ponemon Study*. https://bwnews.pr/3BOFOrR.

[22] Ventures, C. (2021 (Online; accessed Aug 1, 2021)) *Cybersecurity Ventures' 2019 Cybersecurity Market Report*. https://bit.ly/3f1r3Iy.

[23] El, M., McMahon, E., Samtani, S., Patton, M. and Chen, H. (2017) Benchmarking vulnerability scanners: An experiment on scada devices and scientific instruments. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)* (IEEE): 83–88.

[24] Benz, M. and Chatterjee, D. (2020) Calculated risk? a cybersecurity evaluation tool for smes. *Business Horizons* **63**(4): 531–540.

[25] Sabillon, R., Serra-Ruiz, J., Cavaller, V. and Cano, J. (2017) A comprehensive cybersecurity audit model to improve cybersecurity assurance: The cybersecurity audit model (csam). In *2017 International Conference on Information Systems and Computer Science (INCISCOS)* (IEEE): 253–259.

[26] Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., Sastry, S. *et al.* (2009) Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security* (Citeseer), **5**.

[27] Liu, C.H. and Zhang, Y. (2015) *Cyber physical systems: architectures, protocols and applications*, **22** (CRC Press).

[28] Kure, H.I., Islam, S. and Razzaque, M.A. (2018) An integrated cyber security risk management approach for a cyber-physical system. *Applied Sciences* **8**(6): 898.

[29] Sauerwein, C., Sillaber, C., Mussmann, A. and Breu, R. (2017) Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives .

[30] Mavroeidis, V. and Bromander, S. (2017) Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In *2017 European Intelligence and Security Informatics Conference (EISIC)* (IEEE): 91–98.

[31] Wagner, T.D., Mahbub, K., Palomar, E. and Abdallah, A.E. (2019) Cyber threat intelligence sharing: Survey and research directions. *Computers & Security* **87**: 101589.

[32] Tosh, D.K., Molloy, M., Sengupta, S., Kamhoua, C.A. and Kwiat, K.A. (2015) Cyber-investment and cyber-information exchange decision modeling. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems* (IEEE): 1219–1224.

[33] Haustein, M., Sighart, H., Titze, D. and Schoo, P. (2013) Collaboratively exchanging warning messages between peers while under attack. In *2013 International Conference on Availability, Reliability and Security* (IEEE): 726–731.

[34] Fisk, G., Ardi, C., Pickett, N., Heidemann, J., Fisk, M. and Papadopoulos, C. (2015) Privacy principles for sharing cyber security data. In *2015 IEEE Security and Privacy Workshops* (IEEE): 193–197.

[35] Sander, T. and Hailpern, J. (2015) Ux aspects of threat information sharing platforms: An examination & lessons learned using personas. In *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security*: 51–59.

[36] MITRE (2021 (Online; accessed Aug 1, 2021)) *Common Attack Pattern Enumeration and Classification*. https://capec.mitre.org/.

[37] Strom, B.E., Applebaum, A., Miller, D.P., Nickles, K.C., Pennington, A.G. and Thomas, C.B. (2018 (Online; accessed Aug 1, 2021)) *MITRE ATT&CK™ : DESIGN AND PHILOSOPHY*. https://bit.ly/2SZtNy7.

[38] Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G. and Thomas, C.B. (2018) Mitre att&ck: Design and philosophy. *Technical report* .