IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2020, accepted January 4, 2021, date of publication January 12, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051072

Cy-Through: Toward a Cybersecurity Simulation for Supporting Live, Virtual, and Constructive Interoperability

DONGHWAN LEE[1,2], (Graduate Student Member, IEEE), DONGHWA KIM[1], MYUNG KIL AHN[1], WONWOO JANG[2], (Member, IEEE), AND WONJUN LEE[2], (Fellow, IEEE)

[1]2nd R&D Institute - 3rd Directorate, Agency for Defense Development, Seoul 05771, South Korea
[2]School of Cybersecurity, Korea University, Seoul 02841, South Korea

Corresponding author: Wonjun Lee (wlee@korea.ac.kr)

This work was jointly supported by Agency of Defense Development, Republic of Korea and the National Research Foundation (NRF) of Korea grant funded by the Korea Government (Ministry of Science & Information and Communication Technology) under Grant 2019R1A2C2088812.

ABSTRACT Cybersecurity simulation is a useful, practical approach to provide insights to counter cyber threats for organizations with a large-scale, complex cyber environment. From the micro behavior of malware on a host to the macroscopic impact of a DDoS attack, various phenomena can be observed and analyzed with simulation scenarios. Many platforms for cybersecurity simulation have been developed to support simulation scenarios and models with different fidelity levels: live, virtual, and constructive. Many platforms for cybersecurity simulation have been developed to support simulation scenarios and models with varying fidelity levels: live, virtual, and constructive. Hence, the support of interoperability between models with different fidelities remained untrodden in the cybersecurity simulation literature. In this paper, we propose a novel cybersecurity simulation platform, Cy-Through, which enables full interoperability between models with different fidelity levels, live/virtual and constructive models. Through the development and demonstration of a prototype of the platform, we prove the possibility of a Live, Virtual, and Constructive (LVC)-interoperable cybersecurity simulation.

INDEX TERMS Cybersecurity simulation, simulation platform, virtualization platform, LVC interoperability, prototype demonstration.

I. INTRODUCTION

Simulation has been a powerful tool allowing cybersecurity researchers to explore different scenarios of cyber phenomenon to analyze and assess cyber threats efficiently. Due to the strengths of simulation, such as easy-to-construct scenarios, reproducibility, and traceability, simulation techniques are widely used in academia and the industry for evaluating the impact of cyber threats and the resiliency of existing/possible systems and network set-ups, and locating a weak link in various cyber environments. For example, it is fairly inefficient in time and effort to deploy hundreds of real hosts when finding out the most resilient network topology and configurations against a DDoS attack. Instead, we can attain the optimal topology and configurations by simply exploring various scenarios consisting of hundreds of host and network models.

Like other simulation models, cybersecurity simulation has three different fidelity levels of simulation models: live, virtual, and constructive. A live model is equivalent to a real machine. Real hosts, network equipment, and actual network protocols are used in a live simulation. A virtual model is a model where a part or a whole of a system/mechanism is virtualized or emulated. For instance, a virtual machine that runs on a virtualization platform such as VirtualBox [1], can be regarded as a virtual model. With advancements in virtualization technology, there is no significant difference between a live and a virtual model. Therefore, in cybersecurity literature, a live and a virtual model are often considered as models with the same fidelity. Lastly, a constructive model is a limited or representative model in which only the necessary logic of a system/mechanism is provided for the

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

purpose of a simulation. In a constructive simulation, as the interactions between constructive models are also abstracted as events, actual network packets may not necessarily be generated according to the purpose of a simulation. However, thanks to the low fidelity of the model, a scenario with a large scale network can be easily built and simulated compared to the other simulation models. A scenario built by constructive models may include up to thousands of hosts and a corresponding number of network segments, which can hardly be achieved in live and virtual model-based simulation.

Each simulation model, as described above, has a distinct purpose and usage, and a simulation model is carefully considered and determined in advance of simulation. As the demand for various simulation scenarios for cybersecurity applications increases, so does the need for a new integrated, hybrid simulation based upon the heterogeneous fidelity models. For example, the creation of artifacts and the changes on the system when compromising a host by a worm virus can be observed through live/virtual simulation. If we can extend the simulation to include constructive models, i.e., include a large number of simulated hosts and network segments, then it is also possible to attain the result of large scale compromises by worm viruses from one integrated simulation. The integration of simulation models enriches the diversity of simulation scenarios and improves the applicability of simulation not only for cybersecurity testing and evaluations but also for cybersecurity training.

There were a few researchers who tried to leverage the advantages of the LVC interoperable simulation [2]–[4]. However, most of the research projects simply interconnected heterogeneous models, making merely limited use of the other model. Specifically, in most cases, benign packets are interpreted and transferred seamlessly between the different models while threat packets are not allowed to roam between the models. Even in cases where threat packets are allowed to be transferred between the models, the other model only works as a relay node or a network segment that simply returns the received threat packets to its originating model. Since analyzing the impact of a cyber threat is one of the most important reasons to utilize cybersecurity simulation, the aforementioned limitations have kept us from using LVC interoperable simulation in a practical sense.

In this paper, we introduce a prototype of an LVC interoperable simulation where cyber threat packets are seamlessly exchanged between heterogeneous models and their impact of cyber threat packets is wholly analyzed. The rest of the paper is structured as follows. Section 2 has some of the related work of the study and the challenges for the LVC interoperable cyber threat simulation. Section 3 provides the detailed architecture and rationale of our prototype. Next, Section 4 proves the efficacy of our simulation with an LVC interoperable scenario. Lastly, Section 5 summarizes and concludes our study with discussions and future work.

II. RELATED WORK

In this section, we will introduce some related work to the LVC interoperable simulation and discuss challenges in enabling LVC interoperability on cybersecurity simulation.

A. DIS AND HLA

Distributed Interactive Simulation (DIS) [5] and High Level Architecture (HLA) [6] emerged in the military community due to the need for joint, distributed training and exercises supported by LVC interoperable simulation. DIS extended its predecessor, SIMNET [7], introducing dead reckoning to efficiently transmit the state of battlefield entities. HLA was developed by the merger of the DIS protocol with the Aggregate Level Simulation Protocol (ALSP) designed by MITRE. Both DIS and HLA were ratified later as the IEEE standards, IEEE 1278 [8]–[10] and IEEE 1516 [11]–[13], respectively. A security-aware CPS simulator, COSSIM [14], [15], can be taken as an up-to-date example of an LVC interoperable cybersecurity simulation built based on DIS-HLA technology.

Although DIS and HLA provide architectures and middlewares suitable for military training and exercises, they are barely adoptable for cybersecurity simulations. The middlewares of DIS and HLA are designed to support data exchanges between heterogeneous LVC simulation models and entities, however, in general simulation for military training and exercises, the levels of fidelity coherently correspond to the ones of abstraction, which are categorized as four levels: campaign, mission, engagement, and engineering. This means that most of events occur in each model, and the middleware only mediates the interactions between them. On the other hand, in the majority of cybersecurity simulation, levels of fidelity do not correspond to the ones of abstraction, which makes heavy weather of LVC interoperability - the different levels of fidelity are evenly used for cybersecurity simulation models, but the level of abstraction is limited to the engineering level in most cases. To achieve LVC interoperability in cybersecurity simulation, we will need to address frequent, atomic data exchange between the most complicated models while processing all the engineering-level events occurring in each model.

B. TENA

Test and Training Enabling Architecture (TENA) started as a research project to develop a simulation architecture which provides interoperability to the US Army test and training systems [16]. One of the differences between TENA and DIS/HLA is its intensive support for modularity and object-orientedness. TENA also has a middleware and a data exchange architecture, but its differentiated model-driven architecture enables the fast, simple deployment and management of simulation. To achieve this, TENA imposes strict requirements for participating models, unlike DIS/HLA, which gives us another feature known as computer-enforced

agreements. The computer-enforced agreements ensure that various LVC models detail the nature and form of the data to be exchanged between the models. The introduction of this scheme was based on consideration for the seamless incorporation of live and virtual models that have highly sophisticated communication sub-models. TENA has not been standardized publicly by any organization yet, however it is maintaining its position as *de facto* standard for LVC interoperable simulation in the US military community.

Despite the cyber-friendly features such as computer enforced agreements, TENA is not prepared to accommodate cybersecurity simulation. TENA per se is a competitive architecture with LVC interoperability, though there has been a limited number of contributions for simulation models to process cyber-specific events and interactions so far [2], [17]–[19]. In the long run, cybersecurity models can be developed and deployed for TENA, but it is premature to substantialize fully functional LVC interoperable cybersecurity simulation in the TENA architecture.

C. EMULYTICS

EMULYTICS is one of the earliest and most well-known LVC interoperable simulation platforms for cybersecurity testing and training [3]. The EMULYTICS project is led by Sandia National Lab and is still advancing as a comprehensive platform by including from moving target technology [20], [21] to CPS and embedded systems [4], [22]. The constructive models of EMULYTICS are built based on a COTS simulation tool, OPNET Modeler [23], currently known as Riverbed Modeler [24], and leverages a System-In-The-Loop (SITL) gateway [25], a bundled product with OPNET Modeler, to bring LVC interoperability to the platform. Under SITL gateway's packet-level interoperability, the EMULYTICS platform can smoothly accommodate the data exchange of fundamental cyber threats between live/virtual and constructive models. For example, in a simulation environment of EMULYTICS, the cyber threat packets generated by live/virtual models successfully flowed through constructive network models, therefore they could then impact the other live/virtual models. This alone can be regarded as a considerable achievement in the development of LVC interoperable cybersecurity simulation.

Nonetheless, the applicability of the EMULYTICS platform in LVC interoperable simulation is limited to particular cases. Given the feature of the SITL gateway, benign packets will be flawlessly exchanged between different LVC models. Threat packets, however, cannot deliver their effects to models with different fidelity, i.e., live/virtual threats do not affect constructive hosts and vice versa. A collective threat such as a DDoS attack where the individual packets can actually be benign, may be allowed in the platform exceptionally.

D. OTHER WORK

One of the most emerging applications of LVC interoperable cybersecurity simulation is the cybersecurity testbed for industrial control systems (ICS). Ever since TASSCS [26] adopted the SITL gateway to provide LVC interoperability between its simulated electric grid and anomaly detection testbed, similar work has continued to appear to date. Most of the work follows the conventional Hardware-In-the-Loop (HIL) architecture [27]–[30], where the hardware parts, such as sensors and actuators in industrial control systems, are often alternated by constructive simulation. In other words, the constructive simulation in the ICS cybersecurity testbeds is not an immediate target of cyber threats but merely a representative of the effect of cyber threats. For this reason, ICS cybersecurity testbeds do not support full LVC interoperability with the flawless threat exchange.

The essential prerequisite to achieving the full LVC interoperability for cybersecurity simulation is enabling the LVC interoperability of cyber threats, which remains unresolved due to the aforementioned difficulties. In this paper, we will try to make monumental progress in the effort to develop an LVC interoperable cybersecurity simulation by introducing additional modules to an EMULYTICS-like simulation platform.

III. CY-THROUGH PLATFORM

In this section, the detailed architecture and design of the Cy-through platform, and the rationale behind the architecture and design will be introduced.

A. OVERALL ARCHITECTURE

The architecture of the Cy-Through platform is essentially divided into two worlds: constructive and live/virtual ones. In the constructive world, we built a Discrete Event Simulation (DES) environment in which diverse cyber threat scenarios can be created, based on Riverbed Modeler. In a detailed view, a cyber threat scenario is composed of threat modules which are designed to initiate internal events and eventually trigger a designated threat effect when an event(s) meeting predefined conditions occurs. To ensure compatibility between internal simulation models, the events and conditions are made to contain parameters referred from the National Vulnerability Database (NVD) standard [31]. For example, suppose that a Linux host (cpe:/o:linux:linux_kernel:2.6.19:rc3) in a simulation was predefined to be compromised by a Use-After-Free vulnerability (CVE-2019-15292), then the host will be compromised when a threat with the UAF vulnerability is received. Other simulation models, such as firewalls and IPSs, were equipped with a similar mechanism to allow a wide range of cyber scenarios. In this method, we built a fundamental yet essential constructive cyber simulation platform that can accommodate large-scale cybersecurity models. We have developed 56 cyber threat modules based on 56 selected techniques among over 250 techniques provided in the MITRE ATT&CK model [32]. Table 1 shows the full list of cyber threat modules that we implemented.

Models in the live/virtual world, as in most cybersecurity simulations, can be regarded as real machines. There is little to no software or malware that recognize virtualized

TABLE 1. Selected MITRE ATT&CK techniques for cyber threat modules.

Tactics	Techniques	Tactics	Techniques	Tactics	Techniques	Tactics	Techniques
Impact 6*(Denial of Service)	TCP Flood	8*Collection	Audio Capture	5*Discovery	Network Service Scanning	Defense Evasion	Valid Accounts
	UDP Flood		Automated Collection		System Information Scanning	2* Privilege Escalation	Bypass User Account Control
	ICMP Flood		Clipboard Data		System Network Configuration		Valid Accounts
	HTTP Flood		Data Staged		System Network Connections	4*Persistence	Create Account
	TCP Fragmentation		Data from Local System		System Owner/User		Hidden Files and Directories
	UDP Fragmentation		Input Capture	2* Credential Access	Account Manipulation		Port Knocking
5* Command and Control	Commonly Used Port		Screen Capture		Input Capture		Valid Accounts
	Custom Command and Control		Video Capture	7* Defense Evasion	Bypass User Account Control	3*Execution	Command-Line Interface
	Port Knocking	2* Lateral Movement	Remote Desktop Protocol		Disabling Security Tools		Exploitation for Client Execution
	Remote Access Tools		Windows Remote Management		File Deletion		Windows Remote Management
	Uncommonly Used Port	4*Discovery	Account Discovery		Hidden Files and Directories	4* Initial Access	Drive-by Compromise
3*Exfiltration	Automated Exfiltration		Application Window Discovery		Modify Registry		Spearphishing Attachment
	Data Transfer Size Limits		Browser Bookmark Discovery		Port Knocking		Spearphishing Link
	Exfiltration Over Command and Control		File and Bookmark Discovery		Rundll32		Valid Accounts

or emulated environments, but excluding such exceptional cases, we can assume our live/virtual world is the same as real environments without loss of generality.

The features that set Cy-Through apart from the other simulation platforms are a Cy-Through gateway and a Cy-Through agent, which process data exchange between the constructive and the live/virtual world and emulates threat behaviors on an end-host, respectively. Fig. 1 depicts the overall architecture of our simulation platform including the Cy-Through gateway and the Cy-Through agent.

B. CY-THROUGH GATEWAY

The Cy-Though gateway plays a key role in our simulation platform, mediating threat-involved packets between the constructive and live/virtual world. The gateway is built upon the SITL gateway, complementing the limitations of the SITL gateway which mainly copes with the inter-world exchange of benign packets. One of the most critical limitations of the SITL gateway is that it only supports up to the transport layer, i.e., it does not deliver payloads of packets, nor process those when their destination is the constructive world unless there are explicit routines for those purposes. This is a neglectable problem in many cases when mostly dealing with benign packets, however, this problem becomes much more critical when it comes to threat packets. To overcome this challenge, we have taken a simple but clear approach with the Cy-Through gateway: filling the holes of the SITL gateway.

A major feature of the Cy-Through gateway used to mediate threat-involved packets is called *threat translation*. The threat translation has two modes of operation: live/virtual-to-constructive (LV-to-C) and constructive-to-live/virtual (C-to-LV) translation. In each mode, the real/live world packets are converted into the constructive world packets and vice versa, respectively.

Taking a closer look at the LV-to-C translation mode, the contained threat translation can be defined as the abstraction of a threat packet. Once a threat signature is detected in the packet, the Cy-Through gateway alternates the original payload with a prepared one, which includes the following data:

- Identifier that indicates whether the packet is benign or a threat;
- Size of the original payload;
- Tactic to which the threat belongs;
- Technique to which the threat belongs;
- Other information can be referred for simulation running

The specific structure of a preprocessed payload is depicted in Fig. 2, in which the numbers in parentheses indicate the size of each data field. The preprocessing of benign packets can be conducted in a similar way if necessary. For example, firewall control packets from an administrator clearly make an impact on the simulation environment even though it is not a threat packet. To enable signature detection

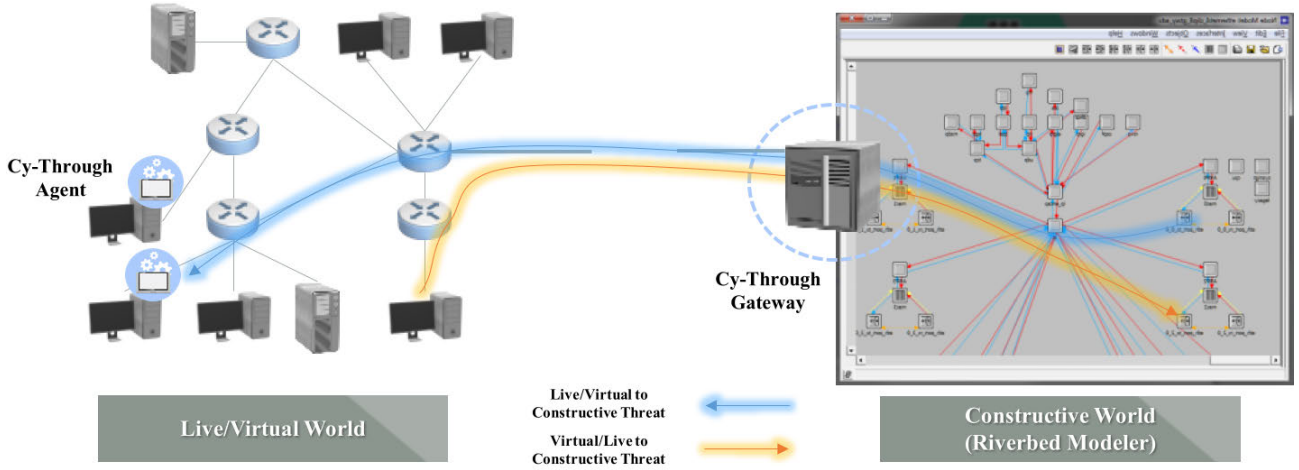


FIGURE 1. Overall architecture of Cy-Through platform.

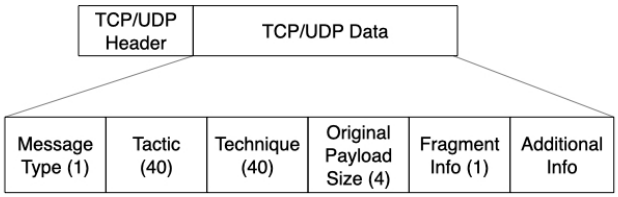


FIGURE 2. Structure of preprocessed threat packet.

in this step, we have utilized Snort [33], an open-source intrusion prevention system for the deep packet inspection. The overall procedure of the preprocessing process is in Algorithm 1.

When the preprocessed packet arrives at the SITL gateway, the corresponding packet event is created through the *packet stream interrupt* feature provided by the SITL gateway. The data of the threat provided in the preprocessing step is included in the packet event so the target node can determine what kind of threat event should be generated.

On the other hand, a completely reversed process is performed in the C-to-LV translation. It is required to literally "reconstruct" cyber threats with abstracted data from the constructive world. To deal with this requirement, we used real payloads extracted from real threat packets, saved as pcap (packet capture) files. The Cy-Thorough gateway has its own node in Riverbed Modeler and captures all the packets passing through the gateway. Once a packet destined to the live/virtual world has arrived at the gateway, its process model fetches the relevant threat payload from the database with a search condition provided from the sender node and assembles a threat packet using the fetched threat payload. The assembled packet, finally, is sent to the target host by the Cy-Through gateway.

Fig. 3 shows the two operation modes of the threat translation, in which database tables with real and simulated threat information help the gateway to identify cyber threats and convert them into ones for the target worlds. One thing to consider when designing complete, seamless LVC

Algorithm 1 Preprocessing Process for LV-to-C Packet Translation

// $(MSG, TCQ, TTC, PSZ, FRG, DAT)$ is a tuple in which each element corresponds to each field in Fig. 3
// $(\cdot)_i$ is a tuple of the database Δ, associated to the signature i
Function `subPayload(P,N)`**:**
  substitutes the payload of packet P with the new payload N and creates a new packet P'
  return P'
end
Function `sendPacket(P)`**:**
  sends the packet P to the Cy-Through gateway
  return
end
while *simulation continues to run* **do**
  if *a packet P is entered to the gateway* **then**
    $P' \leftarrow P$
    for *each signature* $i \in \Delta$ **do**
      if $i \subset P$ **then**
        $N \leftarrow (MSG, TCQ, TTC, PSZ, FRG, DAT)_i$
        $P' \leftarrow$ `subPayload(P',N)`
      end
    end
    `sendPacket(P')`
  end
end

interoperability is to make sure all the necessary packets are translated. That is, benign packets also need to be translated in cases where their effect on the target host is unignorable.

C. CY-THROUGH AGENT

The Cy-Through agent contributes to our platform in two major ways: the diversification of cyber threats and the

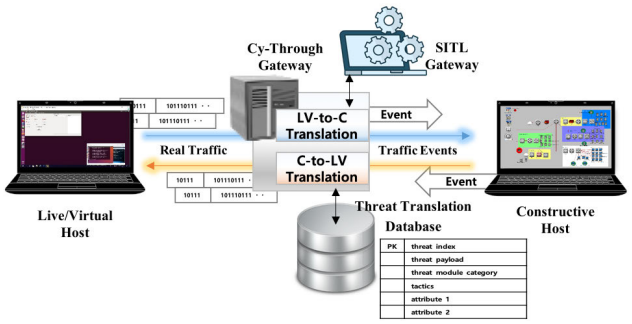


FIGURE 3. Cy-Through gateway and modes of operation.

simplification of the C-to-LV threat translation. It is worth noting that both are related to the C-to-LV threat translation which has more challenges than the opposite. It is very difficult to capture a whole variety of cyber threat packets as pcap files to cope with diverse threats for the C-to-LV translation. Therefore, we introduced a new type of threat packet, called *emulated threat* which can embed a specially designed payload. The agent is designed to download the payload, which is formatted similar to the one in the preprocessed threat packet and includes an attack script. Once the script is triggered by the agent, it automatically performs predetermined malicious behaviors. By this method, we could abridge the preparation of exploit packets which is one of the most challenging tasks, and freely create any malicious behaviors following exploitation.

For instance, let us assume a case where there is a server protected with single packet authorization and an attacker who forges an authorization packet using a vulnerability of the authorization protocol and sends it to the server, which corresponds to the port knocking technique of the defense evasion tactic in Table 1. This case can be simulated in the constructive world without difficulty whereas, in the live/virtual world, the case can be reproduced only if the forged attack packet and a firewall with the vulnerability are prepared. This is where an emulated threat comes in. Although an emulated threat is not real, it can invoke the effect that is expected from the real attack. We can emulate the aforementioned case simply by opening the port using an emulated threat instead of looking for the matching attack packet and vulnerable application/service. Fig. 5 illustrates the exemplified case. The upper part of the figure describes a situation in which the real threat is delivered to the live/virtual world and vice versa, using the real port knocking packet. Once the attack packet is received by the targeted port, the corresponding service opens the port (= 8080) and authorizes the attacker to use it. However, at the lower side of the figure, we can expect the same effect on the targeted port with the Cy-Through agent and the emulated threat even without the real attack packet. In this manner, the coverage of our simulation platform on cyber threats can be expanded infinitely, regardless of whether they are existing or potential ones.

In many cases in reality, a cyber attack can be resolved into a Course of Action (CoA) [34]. Similarly, an advanced scenario of a cybersecurity simulation should be able to deal with the CoAs of cyber threats. In the context of the LVC interoperability, it is suggested that the LVC gateway needs to process a series of threat-related packets in the correct order based upon the history of the previous ones. A simple approach for this can be the use of a state machine, which we adopted for our gateway. However, the more points requiring control means the more chances of failure. Even if the Cy-Through gateway handled threat packets flawlessly, the threat could fail in some unexpected situations in between exploitation and following behaviors on the target host. The Cy-Through agent helps our platform tackle the above problems, allowing smoother interoperation between the different simulation worlds. The detailed control flows of the LV-to-C and C-to-LV threat translation represented with UML sequence diagrams can be found in the appendices.

IV. PROTOTYPE DEMONSTRATION

In this section, we provide the prototype configuration of the Cy-Through platform and explain the experimental results of the demonstration according to a detailed cyber attack scenario.

A. PROTOTYPE CONFIGURATION & DEMONSTRATION SCENARIO

We have built a network environment on Riverbed Modeler 17.5A PL6 for the demonstration of LVC interoperable simulation. As presented in Fig. 4, the network environment is composed of multiple partitioned network segments to accommodate an Advanced Persistent Threat (APT) scenario. An e-mail server and web servers is/are deployed in the DB zone and the server farm zone, respectively, that are abused to attack the final targets, victim hosts in the user zone. Each zone and the live/virtual world are operated on 100 Mbps full-duplex Ethernet, and a 1 Gbps full-duplex Ethernet link is used to connect the zones. An attacker host is only a live/virtual node outside the constructive network environment. An attacker host is actually a virtual machine on VMware Workstation Pro 15 [35], on which we installed Ubuntu 18.04.4 LTS and the Cy-Through agent. In addition, a customized version of Snort based on Snort 2.9.15.1 was installed on the virtual machine as a part of the Cy-through gateway. By monitoring the traffic between the attacker and the victims, we can verify if the LVC interoperability of the Cy-Through platform is realistically achieved.

The feature of the Cy-Through agent was expanded to send and receive real packets including real threat packets, as well as emulated threats. Note that these are handled by different applications in the real environment. However, we introduced the functions into the Cy-Through agent to build an integrated simulation environment, enabling a smooth simulation workflow.

The demonstration procedure is thoroughly designed as a typical APT scenario so that we can run through most

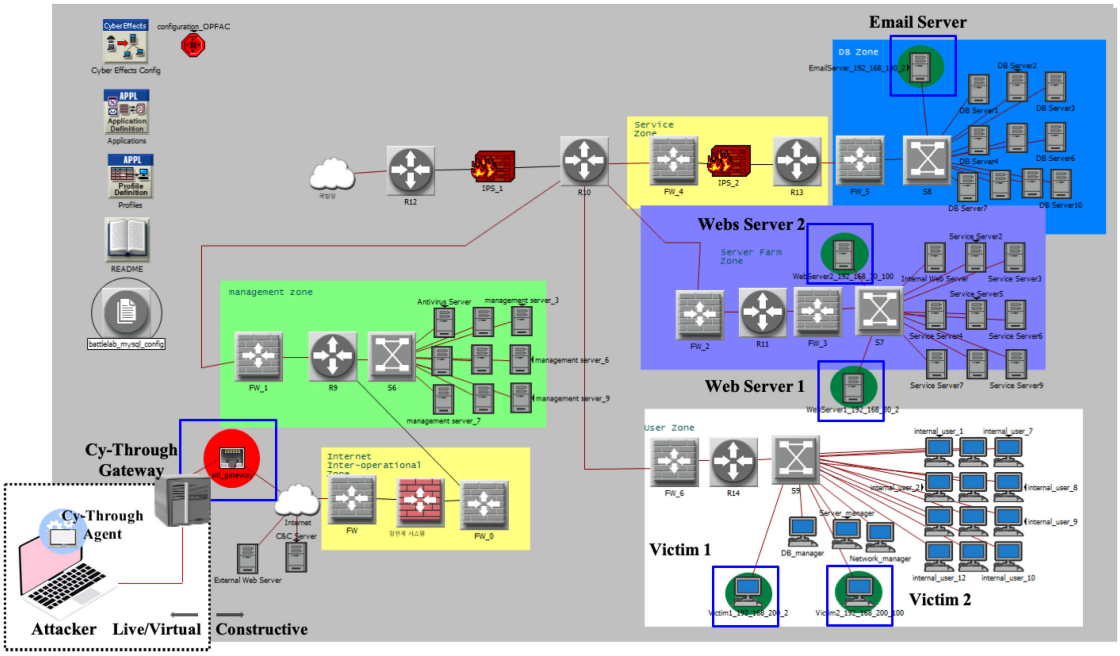


FIGURE 4. Network environment for prototype demonstration of Cy-Through platform.

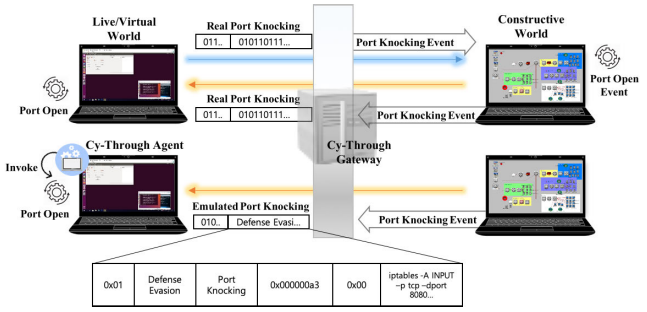


FIGURE 5. Example of real and emulated port knocking attack.

of the platform features. The demonstration procedure is divided into two phases. Phase 1 demonstrates most of the simulation scenario including the LV-to-C translation. Phase 2 was specifically designed to present the C-to-LV translation, where the roles of the attacker and the victim are reversed. The following is a detailed expatiation of the demonstration procedure provided in Fig. 6:

Demonstration Procedure - Phase 1

Step. 1 Web server 1 is compromised by Attacker, setting up a spear-phishing site on the web server.

Step. 2-3 Attacker sends Victim 1 a spear-phishing email to lure the victim to Web Server 1, the compromised one.

Step. 4-9 After Victim 1 reaches Web Server 1, his credentials for Web Server 2 is acquired by Attacker.

Step. 10 Attacker sets up another trap on Web Server 2, with a drive-by compromise technique using the valid account acquired from the previous attack.

Step. 11-12 Victim 2 is compromised by the drive-by compromise attack.

Step. 13-15 Web server 2 used as a C&C server delivers Attacker's command to infiltrate data from Victim 2. The victim sends back the targeted data to Attacker via FTP.

Demonstration Procedure - Phase 2

Step. 16-20 Switching their roles between Attacker and Victim 2, Victim 2 delivers Automated Exfiltration as an emulated threat to Attacker (Step. 16). Then, Victim 2 sends an e-mail and subsequently a threat packet with Win32_Miner to Attacker. This procedure is intended to represent a scenario where Attacker is victimized by a phishing email and downloads a malicious file from Web Server 1 (Step. 17-20). In this case, Win32_Miner is not an emulated threat, but a binary threat file which is actually used in cyber attack cases.

To make the above traffic flow without packet-specific counterparts at the live/virtual host, we developed an integrated processing agent to deal with every packet received through the Cy-Through gateway. In addition, the agent is designed to generate different types of traffic delivered across the gateway. Without loss of generality, UDP was chosen as a transport layer protocol for traffic generation, instead of TCP with which operations are more complicated because of its session management. We intentionally differentiated destination ports into 11223 and 11224, which is for benign packets and threat packets, respectively.

Note that although all the packet types in the procedure can be categorized into threat packets in a broad sense, only the packet whose payload is directly associated with the ATT&CK techniques (hence the simulation/emulation of the threat effect is explicitly required), was considered as a threat packet. In other words, the packets were categorized as benign, except for the packets that required the engagement of

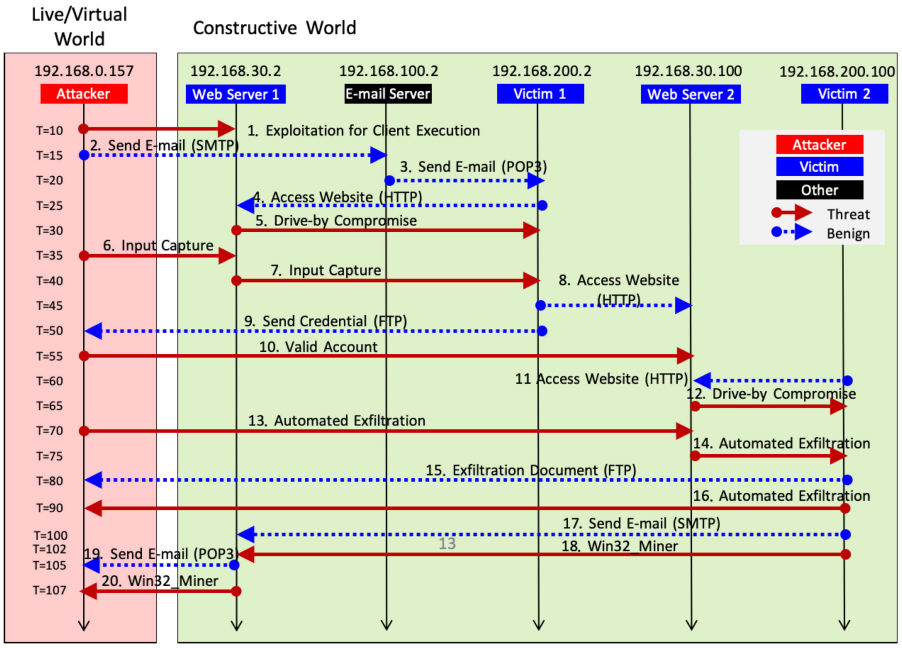
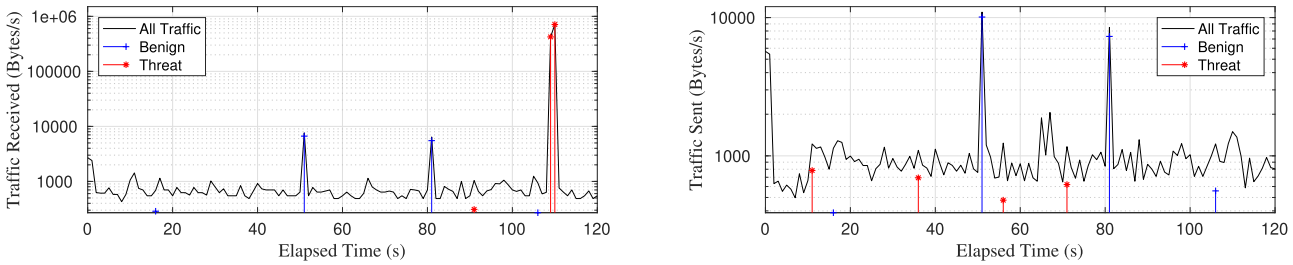


FIGURE 6. Procedure of Cy-Through demonstration.



(a) Traffic received by Attacker

(b) Traffic sent by Attacker

FIGURE 7. Measured traffic flow received/sent by Attacker.

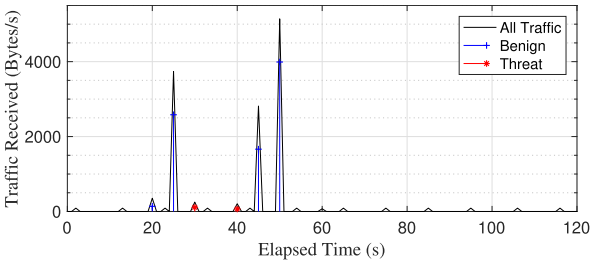
a threat-related process of the Cy-Through platform. It should also be noted that the direction of each flow does not represent a single packet necessarily. Each flow may consist of multiple packets, which can include ones in the opposite direction as well.

We added background traffic between the live/virtual world and the constructive world to prove the real-time capability of our simulation platform in a real-life environment. The emulated database access traffic, added up to 1 MByte/s on average, from both directions, was given between the Attacker host and the DB servers in the DB zone.

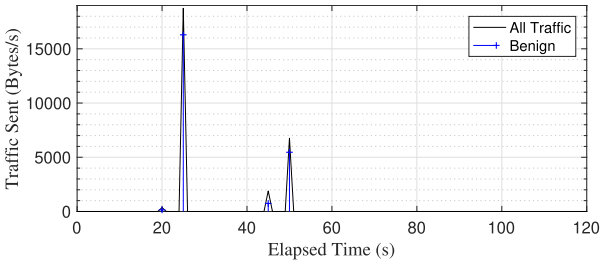
B. EXPERIMENTAL RESULTS

First, the received and sent traffic at the major nodes was examined to make sure that the overall simulation and the generation of the traffic flows were carried out as planned in the demonstration procedure. Fig. 7 and Fig. 8-9 present the byte rate of received/sent traffic by the attacker node and the victim nodes, respectively. It can be observed that the traffic pattern is following the demonstration procedure throughout the simulation. Comparably higher peaks appear around $T = 50$ (at Attacker and Victim 1) and $T = 80$ (at Attacker and Victim 2), which are related to the infiltration traffic via FTP. On the other hand, only Victim 1 has peaks around $T = 25$, which is related to the website access traffic via HTTP between Web Server 1 and Victim 1. The highest peak is found around $T = 107$, which arose from the transmission of Win32_Miner [36], the binary file with a size of 1.4 MByte.

More importantly than the byte rate, we needed to identify the content of the traffic flow to verify if the threat translation was performed properly between the live/virtual and the constructive worlds. The packet captures with Wireshark in Fig. 10 confirm how the Cy-Through gateway translated different types of packets. In Fig. 10a-10b, we can look into the LV-to-C packets before and after preprocessing. The packet in Fig. 10a is the first threat packet (Exploitation for Client Execution at $T = 10$) sent from Attacker. Any readable information is not found here as the packet is delivering binary data. This packet is not supposed to be received from the constructive world as our gateway captures and preprocess it before its delivery. The inside of the prepro-

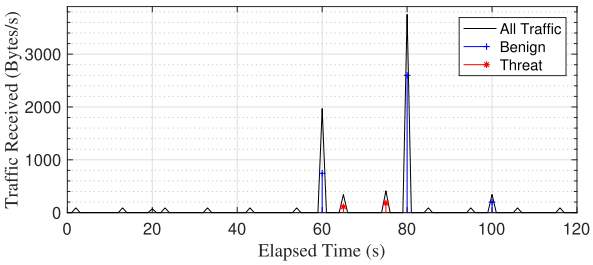


(a) Traffic received by Victim 1

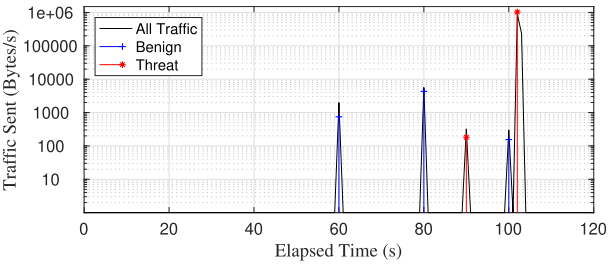


(b) Traffic sent by Victim 1

FIGURE 8. Measured traffic flow received/sent by Victim 1.



(a) Traffic received by Victim 2



(b) Traffic sent by Victim 2

FIGURE 9. Measured traffic flow received/sent by Victim 2.

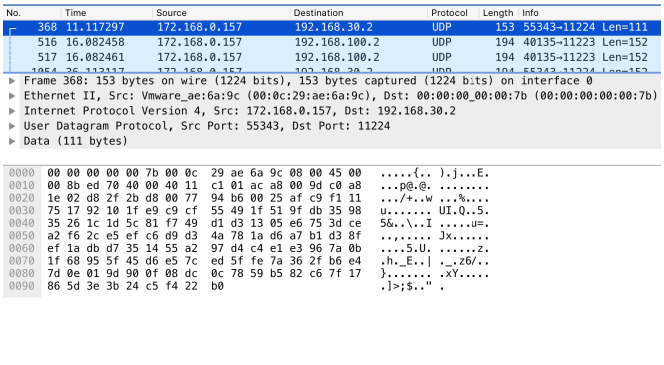
cessed packet is seen in Fig. 10b, where the structure of the preprocessed threat packet described in Fig. 2 is identified clearly through the highlighted part. The first 0×01 part denotes that it is a threat packet while 0×00 means a benign packet. The size of the original payload ($= 111$ Byte) can also be confirmed from the '6f' part at offset $0 \times 007b$ in the packet byte pane.

Looking into the C-to-LV packets depicted in Fig. 10c-10d, the two different types of threats, a real threat and an emulated threat, are clearly distinguished from each other. Fig. 10c shows the packet dump from the delivery of an actual malware (Win32_Miner at $T = 107$). We can confirm the name of the malware as well as the signatures of a PE format including the magic number ($= 0 \times 4d5a$). On the other hand, Fig. 10d presents the packet bytes of an emulated threat (Automated Exfiltration at $T = 90$) delivered to the Cy-Through agent at the Attacker host. As mentioned above, the structure of the data is the same as the one in a preprocessed packet, however, the additional info field is filled out with an attack script so the agent can emulate cyber attack behavior.

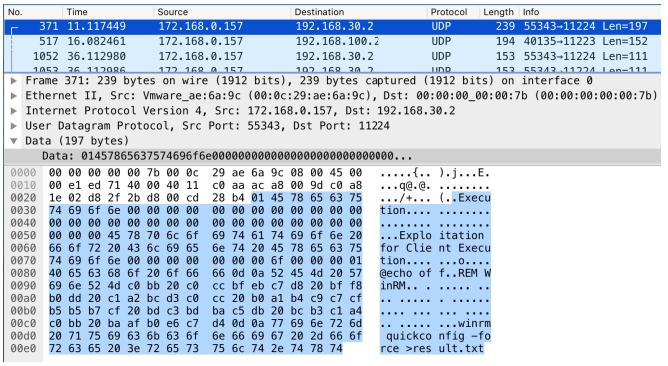
As a result of the extensive demonstration, we were able to ensure that the LVC interoperable simulation was carried out with precision per the demonstration procedure. Particularly, no significant delay was observed in the process of packet exchange although additional background traffic was given between the live/virtual world and the constructive world. The different types of packets between the two worlds were translated by the Cy-Through gateway as we intended, and the Cy-Through agent processed and generated the delivered packets in a timely and accurate manner. This encouraging result shows promise for a fully functional LVC interoperable simulation platform that provides a seamless, flawless exchange of cyber threats between simulation models with different fidelity. Nevertheless, our prototype also shows some limitations to improve upon, which will be further discussed in the next section of the study.

V. DISCUSSIONS

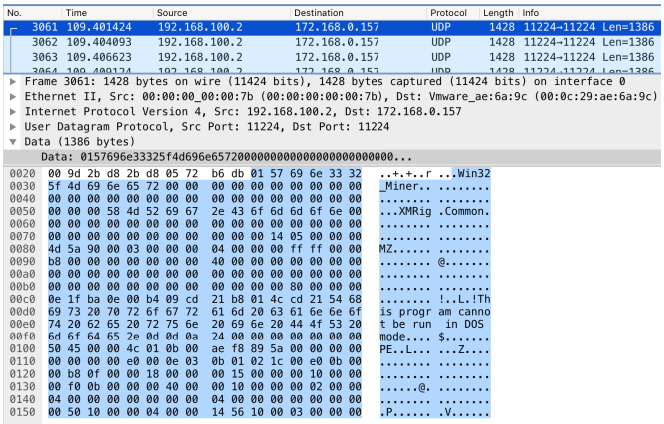
The LVC interoperability of the Cy-Through platform with the support of the threat translation and emulated threat features could be confirmed in the prototype demonstration. However, no direct performance comparison with other simulation schemes has been provided since very limited work on the LVC interoperation of cyber threats has been done. For this reason, we conducted a comparative study in functional aspects of the simulation schemes. Table 2 provides the comparison results between a number of simulation schemes/platforms. As listed in the table, a significant amount of work has been proposed as ICS/SCADA testbeds since TASSCS [26] adopted SITL as an LVC gateway and RTDS for its constructive environment. All the other ICS/SCADA schemes in the table also followed a similar architecture to support the LVC interoperability, however, no work that considered the LVC interoperation of cyber threats in earnest until recently. In the other words, existing ICS/SCADA cyber security testbeds assume that cyber attacks occur only in the live/virtual world, which can be a plausible assumption when the constructive world is simply composed of sensors and actuators. Nevertheless, considering the recent changes in the pattern of cyber attacks, where the threats to embedded systems and hardware are on the rise, there will be a need to cover cyber threats existent in the



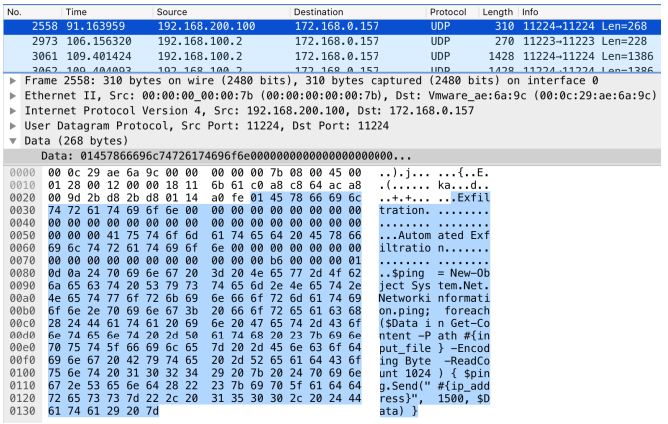
(a) LV-to-C packet before preprocessing



(b) LV-to-C packet after preprocessing



(c) C-to-LV traffic with a real threat (Win32_Miner)



(d) C-to-LV traffic with an emulated threat (Automated Exfiltration)

FIGURE 10. **Captured traffic between the live/virtual world and the constructive world.**

ICS/SCADA constructive environment. Although StealthNet is a great example that demonstrated the simulation of cyber threats in the constructive environment, supporting a wide range of hardware-targeted cyber attacks such as radio jamming and channel scanning, the cyber threats cannot roam freely between the different worlds due to the limitation of the simulation scheme.

Emulytics can be said to be one of the most advanced work in the LVC interoperation of cyber threats. However, as mentioned above, the scheme also has a limitation in the seamless exchange of cyber threats. For instance, in [21], the experiments in the cyber defense and performance aspects of moving target defense framework were carried out each in the live/virtual and constructive environments separately. This is due to the limited LVC interoperability of the simulation framework, which allows only partial support of cyber threat translation such as the pass-through of cyber threats.

Given all the analysis, it is obvious that the Cy-through platform delivers unparalleled interoperabiltiy in a cyber security simulation, compared to the other simulation schemes. Despite the achievement with our prototype, we still have challenges to overcome before a fully functional interoperable LVC simulation can be achieved. The major challenges that can be posed for a fully interoperable LVC platform include the following: 1) *Scarcity of real threats*: In contrast with the LV-to-C translation, the real-world threat packets are necessary to convert threats in the C-to-LV translation. Collecting almost every cyber threat mfrom the real-world is a pious hope at a laboratory level. We believe the emulated threat feature of the Cy-Through agent will be helpful to address this issue, however, some problems remain. For example, emulated threats cannot be used when testing a network-based IPS, as an emulated threat packet does not include signatures found in the real-world threats. An additional feature like signature injection might be needed for such cases. 2) *Lack of TCP support*: The Cy-Through platform can process the exchange of TCP packets at the packet-level, but we had to choose UDP as a default protocol for the packet exchange because of the uncertainty in session management. However, considering that TCP is also a dominant protocol in cyber threats, the support for TCP will be essential for future simulation platforms. There are a few more limitations to our simulation platform. For instance, for a similar reason with 2), support for TLS (Transport Layer Security) such as SSL (Secure Socket Layer) is hardly expected in our platform. All these limitations will be tackled in our future iterations.

TABLE 2. Comparison between LVC interoperable cyber simulation schemes/platforms.

Work	Application	NVD	Cyber Attack Framework	LVC Threat Translation	Underlying Technology/Tools	Emulated Threat
Cy-Through	Cyber Trainer/ Testbed	YES	ATT&CK	N/A	SITL	YES
StealthNet [2] [19]	Military Trainer/ Testbed	YES	N/A	N/A	TENA	N/A
Emulytics [3] [4] [20] [21] [22]	Cyber Trainer/ Testbed	N/A	Kill Chain [37]	N/A	SITL	N/A
COSSIM [14] [15]	ICS/SCADA Testbed	N/A	N/A	N/A	HLA	N/A
CSDS [17]	Cyber Trainer/ Testbed	N/A	N/A	N/A	TENA	N/A
NCR [18]	Military Trainer/ Testbed	N/A	Kill Chain	N/A	TENA	N/A
TASSCS [26]	ICS/SCADA Testbed	N/A	N/A	N/A	SITL	N/A
RTCPS [27]	ICS/SCADA Testbed	N/A	N/A	N/A	SITL	N/A
NISTIR 8089 [28]	ICS/SCADA Testbed	N/A	N/A	N/A	Simulink [38]	N/A
RTTE [29]	ICS/SCADA Testbed	N/A	N/A	N/A	Simulink	N/A
Softgrid [30]	ICS/SCADA Testbed	N/A	N/A	N/A	PowerWorld [39]	N/A

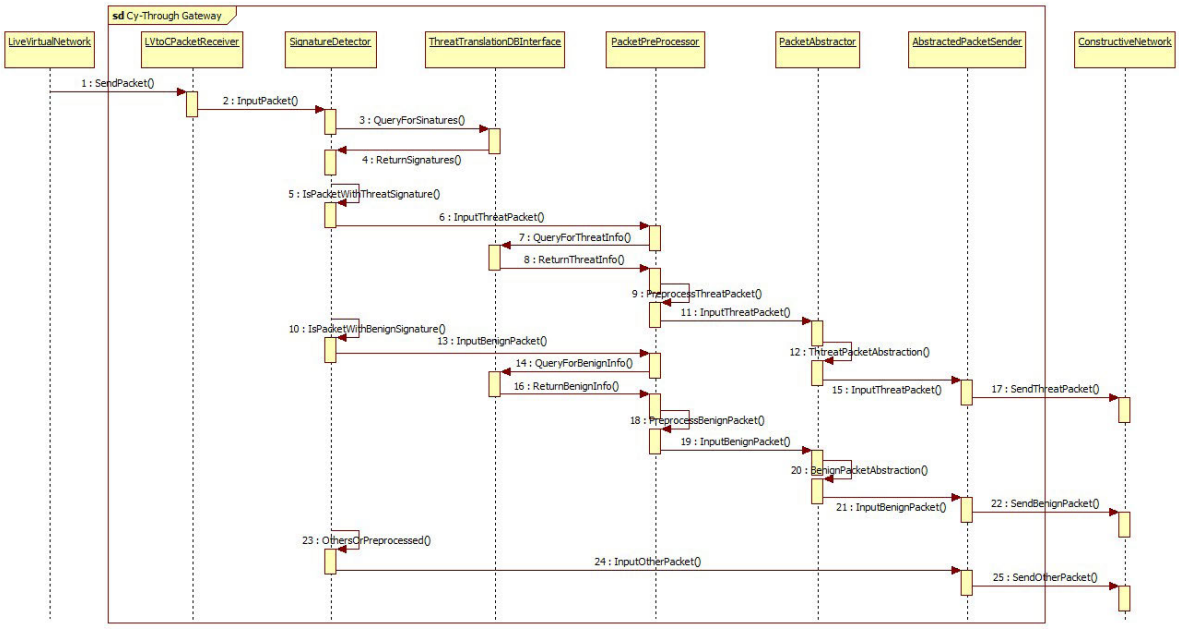


FIGURE 11. Sequence diagram of LV-to-C translation.

VI. CONCLUSION

In this paper, we proposed a novel cybersecurity simulation platform, Cy-Through. Our platform is designed to provide interoperability between simulation models with different fidelity, live/virtual and constructive models. To enable this, the Cy-Through gateway and Cy-Through agent were

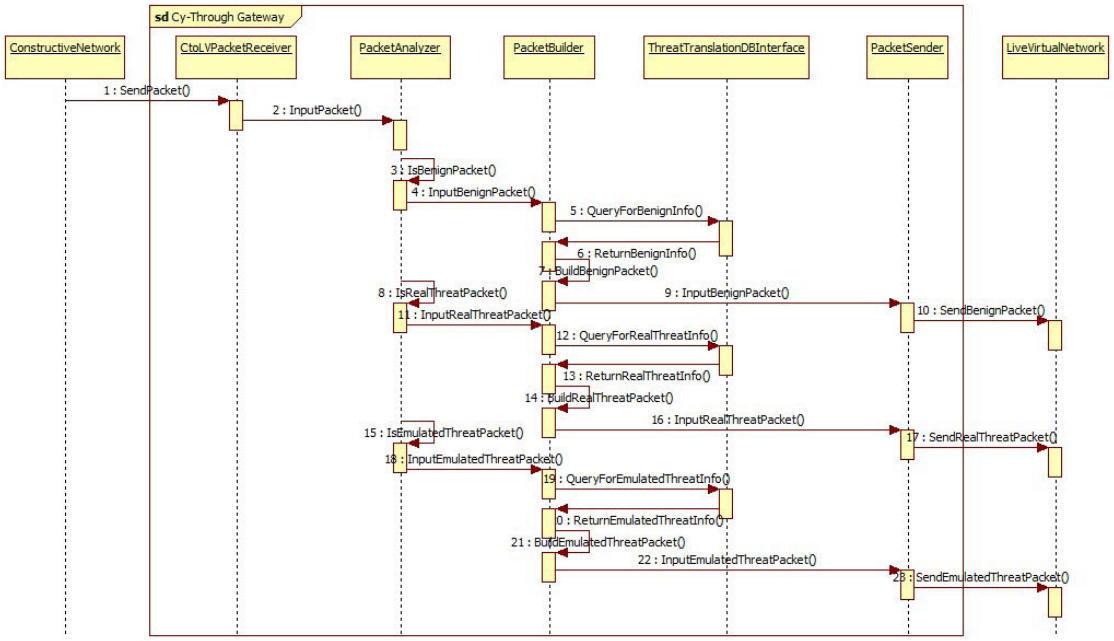


FIGURE 12. Sequence diagram of C-to-LV translation.

introduced. The Cy-Through gateway carries out the exchange of cyber threats with a packet-specific conversion process called threat translation. The Cy-Through agent processes delivered packets at end-point hosts to support smoother threat delivery and various types of cyber threats including emulated threats, which can enrich cyber threats for diverse attack scenarios. Despite the imperfections aforementioned in the previous section, we expect that our platform will pave the road towards cybersecurity simulation with full LVC interoperability by greatly enriching cyber attack scenarios in cybersecurity simulation that demand the LVC interoperable exchange of cyber threats. Our platform is expected to be widely adopted for a range of applications, including cybersecurity simulation for decision making and analysis as well as cybersecurity training and exercise systems. Additional synergies are expected to arise through an integration between our platform and a widely used model-driven architecture such as TENA.

APPENDIX A
SEQUENCE DIAGRAM OF LV-TO-C TRANSLATION

See Fig. 11.

APPENDIX B
SEQUENCE DIAGRAM OF C-TO-LV TRANSLATION

See Fig. 12.

REFERENCES

[1] J. Watson, "Virtualbox: Bits and bytes masquerading as machines," *Linux J.*, vol. 2008, no. 166, p. 1, 2008.
[2] M. Varshney, K. Pickett, and R. Bagrodia, "A live-virtual-constructive (LVC) framework for cyber operations test, evaluation and training," in *Proc. - MILCOM Mil. Commun. Conf.*, Nov. 2011, pp. 1387–1392.
[3] V. Urias, B. Van Leeuwen, and B. Richardson, "Supervisory command and data acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed," in *Proc. MILCOM IEEE Mil. Commun. Conf.*, Oct. 2012, pp. 1–8.
[4] V. Urias, B. Van Leeuwen, B. Wright, and W. Stout, "Emulytics at Sandia national laboratories," in *Proc. 21st Int. Congr. Modelling Simul. (MODSIM World)*, 2015, pp. 1–10.
[5] R. C. Hofer and M. L. Loper, "DIS today [distributed interactive simulation]," *Proc. IEEE*, vol. 83, no. 8, pp. 1124–1137, Aug. 1995.
[6] J. S. Dahmann, "High level architecture for simulation," in *Proc. 1st Int. Workshop Distrib. Interact. Simulation Real Time Appl.*, 1997, pp. 9–14.
[7] D. C. Miller and J. A. Thorpe, "SIMNET: The advent of simulator networking," *Proc. IEEE*, vol. 83, no. 8, pp. 1114–1123, Aug. 1995.
[8] *IEEE Standard for Information Technology—Protocols for Distributed Interactive Simulation Applications—Entity Information and Interaction*, Standard 1278-1993, 1993, pp. 1–64.
[9] *IEEE Standard for Distributed Interactive Simulation—Application Protocols*, Standard 1278.1-2012 (Revision of IEEE Std 1278.1-1995), 2012, pp. 1–747.
[10] *IEEE Standard for Distributed Interactive Simulation (DIS)—Communication Services and Profiles*, Standard 1278.2-2015 Revision IEEE Std 1278.2-1995, 2015, pp. 1–42.
[11] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules*, Standard 1516-2010 (Revision IEEE Std 1516-2000), 2010, pp. 1–38.
[12] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Federate Interface Specification*, Standard 1516.1-2010 (Revision IEEE Std 1516.1-2000), 2010, pp. 1–378.
[13] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Object Model Template (OMT) Specification*, Standard 1516.2-2010 (Revision IEEE Std 1516.2-2000), 2010, pp. 1–110.
[14] A. Brokalakis, N. Tampouratzis, A. Nikitakis, S. Andrianakis, I. Papaefstathiou, and A. Dollas, "An open-source extendable, highly-accurate and security aware CPS simulator," in *Proc. 13th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Jun. 2017, pp. 81–88.
[15] N. Tampouratzis, I. Papaefstathiou, A. Nikitakis, A. Brokalakis, S. Andrianakis, A. Dollas, M. Marcon, and E. Plebani, "A novel, highly integrated simulator for parallel and distributed systems," *ACM Trans. Archit. Code Optim.*, vol. 17, no. 1, pp. 1–28, Mar. 2020.
[16] J. R. Noseworthy, "The test and training enabling architecture (TENA) supporting the decentralized development of distributed applications and LVC simulations," in *Proc. 12th IEEE/ACM Int. Symp. Distrib. Simul. Real-Time Appl.*, Oct. 2008, pp. 259–268.

[17] A. Oltramari, C. Lebiere, L. Vizenor, W. Zhu, and R. Dipert, "Towards a cognitive system for decision support in cyber operations," in *Semantic Technology for Intelligence, Defense, and Security (STDIS)*, vol. 1097. Aachen, Germany: CEUR, 2013, pp. 94–100.

[18] G. Hudgins. *Successful Distributed and Cyber Testing With Tena and Jmetc*. Accessed: Jun. 14, 2020. [Online]. Available: http://www.dtic.mil/ndia/2017/test/BestPracticesHudgins.pdf

[19] G. Torres, K. Smith, J. Buscemi, S. Doshi, H. Duong, D. Xu, and H. K. Pickett, "Distributed StealthNet (D-SN): Creating a live, virtual, constructive (LVC) environment for simulating cyber-attacks for test and evaluation (T&E)," in *Proc. MILCOM IEEE Mil. Commun. Conf.*, Oct. 2015, pp. 1284–1291.

[20] B. Van Leeuwen, W. Stout, and V. Urias, "MTD assessment framework with cyber attack modeling," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2016, pp. 1–8.

[21] W. M. S. Stout, B. Van Leeuwen, V. E. Urias, J. Tuminaro, S. Schrock, and N. Dossaji, "Leveraging a LiveNirtual/constructive testbed for the evaluation of moving target defenses," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2018, pp. 1–5.

[22] B. Van Leeuwen, J. Eldridge, and V. Urias, "Cyber analysis emulation platform for wireless communication network protocols," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2017, pp. 1–6.

[23] X. Chang, "Network simulations with OPNET," in *Proc. WSC Winter Simulation Conf. Simulation Bridge Future*, Dec. 1999, pp. 307–314.

[24] I. Riverbed Technology. *Riverbed Modeler 18.8*. Accessed: Jun. 14, 2020. [Online]. Available: https://www.riverbed.com/products/steelcentral/steelcentral-riverbed-mo%deler.html

[25] Riverbed Technology. *System-in-the-Loop | Riverbed OPNET Modeler*. Accessed: Jun. 14, 2020. [Online]. Available: https://opnetmodeler.wordpress.com/system-in-the-loop/

[26] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of SCADA control systems (TASSCS)," in *Proc. ISGT*, Jan. 2011, pp. 1–7.

[27] B. Chen, K. L. Butler-Purry, A. Goulart, and D. Kundur, "Implementing a real-time cyber-physical system test bed in RTDS and OPNET," in *Proc. North Amer. Power Symp. (NAPS)*, Sep. 2014, pp. 1–6.

[28] R. Candell, T. Zimmerman, and K. Stouffer, "An industrial control system cybersecurity performance testbed," *NIST Interagency/Internal Report (NISTIR)*, vol. 8089, pp. 1–56, Dec. 2015.

[29] G. Koutsandria, R. Gentz, M. Jamei, A. Scaglione, S. Peisert, and C. McParland, "A real-time testbed environment for cyber-physical security on the power grid," in *Proc. 1st ACM Workshop Cyber-Phys. Syst.-Secur. PrivaCy CPS-SPC*, 2015, pp. 67–78.

[30] P. Gunathilaka, D. Mashima, and B. Chen, "SoftGrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions," in *Proc. 2nd ACM Workshop Cyber-Phys. Syst. Secur. Privacy - CPS-SPC*, 2016, pp. 113–124.

[31] S. Zhang, X. Ou, and D. Caragea, "Predicting cyber risks through national vulnerability database," *Inf. Secur. J., A Global Perspective*, vol. 24, nos. 4–6, pp. 194–206, Dec. 2015.

[32] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," Mitre, Colshire, VA, USA, Tech. Rep. MP180360, Jul. 2018.

[33] M. Roesch, "Snort: Lightweight intrusion detection for networks," *Lisa*, vol. 99, no. 1, pp. 229–238, Jun. 1999.

[34] M. S. Boddy, J. Gohde, T. Haigh, and S. A. Harp, "Course of action generation for cyber security using classical planning," in *Proc. Int. Conf. Automated Planning Scheduling (ICAPS)*, Monterey, CA, USA: AAAI Press, Jun. 2005, pp. 12–21.

[35] *vmware: Workstation Pro*, vmware, Inc, Palo Alto, CA, USA, Aug. 2020.

[36] Microsoft, Inc. *Trojan:Win32/Miner*. Accessed: Aug. 14, 2020. [Online]. Available: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-descr%iption?Name=Trojan:Win32/Miner

[37] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues Inf. Warfare Secur. Res.*, vol. 1, no. 1, p. 80, 2011.

[38] T. MathWorks. *Simulink: Simulation and Model-Based Design*. Accessed: Oct. 14, 2020. [Online]. Available: https://www.mathworks.com/products/simulink.html

[39] PowerWorld Corporation. *PowerWorld Simulator*. Accessed: Oct. 14, 2020. [Online]. Available: https://www.powerworld.com/products/simulator/overview

DONGHWAN LEE (Graduate Student Member, IEEE) received the B.E. degree in industrial engineering and the M.S. degree in computer science and engineering from Korea University, Seoul, South Korea, in 2006 and 2008, respectively, where he is currently pursuing the Ph.D. degree in cybersecurity. He is also a Senior Researcher with the 2nd R&D Institute, Agency for Defense Development, Seoul. His research interests include wireless communications, parallel and distributed computing, wireless security, and virtualization technologies for cybersecurity.

DONGHWA KIM received the B.S. and M.S. degrees from the School of Electrical Engineering, Korea University, Seoul, South Korea, in 2004 and 2007, respectively. He is currently a Senior Researcher with the 2nd R&D Institute, Agency for Defense Development, Seoul. His research interests include cybersecurity training systems and red team automation.

MYUNG KIL AHN received the B.S. degree in information and communication engineering from Chungnam National University, Daejeon, South Korea, in 1997, and the M.S. degree in computer engineering from Sogang University, Seoul, South Korea, in 2003. She is currently pursuing the Ph.D. degree in electrical and electronics engineering with Chung-Ang University. She is also a Principal Researcher with the 2nd R&D Institute, Agency for Defense Development, Seoul. Her research interests include computer security and cyberwarfare modeling and simulation.

WONWOO JANG (Member, IEEE) received the B.S. degree in computer from the College of Computing, Hanyang University ERICA campus, Ansan, Gyeonggi-do, South Korea, in 2018, and the M.E. degree in cybersecurity from the School of Cyber Security, Korea University, Seoul, South Korea, in 2020. He is currently pursuing the Ph.D. degree in cybersecurity with Korea University. His research interests include security and privacy in mobile computing, and RF-powered computing and networking, next-generation transport protocols, and deep learning.

WONJUN LEE (Fellow, IEEE) received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, South Korea, in 1989 and 1991, respectively, the M.S. degree in computer science from the University of Maryland, College Park, MD, USA, in 1996, and the Ph.D. degree in computer science and engineering from the University of Minnesota, Minneapolis, MN, USA, in 1999. He joined the Faculty of Korea University, Seoul, in 2002, where he is currently a Professor with the School of Cybersecurity. He has authored or coauthored over 220 papers in refereed international journals and conferences. His research interests include communication and network protocols, optimization techniques in wireless communication and networking, security and privacy in mobile computing, and RF-powered computing and networking. He has served on the TPC and/or an Organizing Committee Member of the IEEE INFOCOM from 2008 to 2021, a PC Vice Chair of the IEEE ICDCS 2019, and the ACM MobiHoc from 2008 to 2009, and over 130 international conferences.