Transpacific Testbed for Real-Time Experimentation

[1]Oluwaseyi Ajayi, [1]Huseyn Huseynov, [1]Tarek Saadawi, [2]Masato Tsuru, and [3]Kenichi Kourai

[1]*Department of Electrical Engineering, City University of New York, City College, NY, USA*
[2]*Department of Computer Science and Electronics, Kyushu Institute of Technology, Fukuoka, Japan.*
[3]*Department of Creative Informatics, Kyushu Institute of Technology, Fukuoka, Japan.*

Oajayi000@citymail.cuny.edu hhuseynov@ccny.cuny.edu saadawi@ccny.cuny.edu tsuru@cse.kyutech.ac.jp
kourai@ksl.ci.kyutech.ac.jp

***Abstract*—The transpacific testbed is a generic routing encapsulation (GRE) tunnel built between CUNY City College (CCNY), USA and Kyushu Institute of Technology (KYUTECH), Japan. The tunnel, built through internet2, originated from CCNY through the JGN network in Seattle and terminated at Kyutech in Japan. The testbed defines the future of the Internet by focusing on addressing research challenges associated with enabling trustworthy networks, supporting the Internet of Things (IoT), which encompasses everything connected to the Internet and cyber-physical systems (CPS) - a controlled mechanism monitored by computer-based algorithms. In this paper, we describe the setting up and testing of the testbed. Furthermore, we describe the real-time experiments conducted on the testbed and present the results. The experiments are classified into two: blockchain-based cooperative intrusion detection system (CoIDS) and Secure Virtual Machine introspection. In each of the experiments, we describe the method and present the results. Finally, we look into the ongoing works of extending the testbed to the COSM-IC global testbed.**

Keywords — B*lockchain, Cybersecurity, introspection, virtual machine, Intrusion detection System, IoT, Internet, GRE tunnel.*

I. INTRODUCTION

Recent developments in mobile networks, big data, and cloud computing led to the growth of connected devices and Internet of Things (IoT) use cases. Various domains in which IoT significantly facilitates our daily lives include smart cities, remote healthcare, autonomous cars, security systems, and many others. One of the critical factors that satisfy the quality of service for these technologies is 5G mobile networks that rely on software-defined networking (SDN) and network function virtualization (NFV) [1]. New use cases, protocols, and technologies add new attack surfaces to the existing ones. An increasing number of cloud-based services across multiple geographical zones makes enforcing security and availability of the network services to the end-users more challenging. Traditional attacks such as IP spoofing, Denial of Service (DoS), Distributed Denial of Service (DDoS), etc., along with increased ransomware, spyware, and insider attacks, are targeting public and private organizations running on cloud-based servers.

To address these security problems and conduct real-world experiments on next-generation wireless technologies, we developed a transpacific testbed between two labs located at the City College of New York and Kyushu Institute of Technology in Fukuoka, Japan. The project is part of the global development of an international networking and wireless testbed by federating US research testbeds including COSMOS, ORBIT, FABRIC, and PEERING with experimental facilities in Ireland, Greece, Brazil, and Japan. The federated international testbed was proposed to enable global experimental research on a wide range of optical, wireless, SDN and NFV, blockchain, interdomain routing, and edge computing experiments.

In this paper, we describe the testbed and present two ongoing real-time intrusion detection systems (IDS) experimentations: blockchain-based cooperative intrusion detection system (CoIDS) [2] and secure remote virtual machine introspection [3]. These real-time experiments have been successfully implemented and tested over the transpacific testbed. The CoIDS aims to securely exchange attack features among IDS nodes located at different sites, and in the process, detecting any malicious activities on the shared data. The architecture is novel because it securely shares attack information with nodes located in different sites in real-time with minimal dissemination latency. It also detects and prevents malicious activities on the stored data in real-time. The architecture encourages public nodes to join and leave the blockchain network in real-time without any security concern. The architecture achieves these novelties by leveraging blockchain's distributive and tamper-proof ability. It also makes use of the smart contract to verify and thwart outsider and insider malicious activities. The performance metrics show its effectiveness when disseminating stored information among IDS nodes. The second experiment implements a remotely secure virtual machine introspection (VMI) [3] using Artificial Immune System (AIS) algorithms. The application notifies system administrators about potential threats running in the VM in real-time via agentless operations. The monitored potential threats include keyloggers, rootkits, trojans, process hiding, and other intrusion artifacts. Experimental results of remote VMI on more than 50 different malicious codes demonstrated an average anomaly detection rate close to 97%. Based on these explanations, the contribution of the paper can be summarized as follows:

- To describe the setup and troubleshooting of a transpacific testbed between the United States and Japan.

978-1-6654-4308-1/21/$31.00 ©2021 IEEE
DOI 10.1109/5GWF52925.2021.00095

- To describe how a blockchain-based cooperative intrusion detection system experiment is evaluated on the testbed and presents the corresponding real-time results.
- To describe how we set up a Secure Virtual Machine introspection on the transpacific testbed and explained the real-time results.

The remainder of this paper is organized as follows: Section II discusses the background and related works on the experimentation testbed, Section III describes the setup and troubleshooting of the transpacific testbed. Section IV presents the experimentations and the results. Finally, section V describes the conclusions and ongoing future works.

II. BACKGROUND AND RELATED WORKS

Experimentation testbeds

Several works have been done on setting up the testbed for conducting experiments in intrusion detection and mitigation systems. The authors in [1] provide a detailed description of their testbed for deploying IDS in virtualized networks. Their setup is based on Mininet [4] to virtualize the network and ONOS controller [5] to manage it. To connect the controller and simulated IoT devices, they deployed an Open vSwitch [6]. In [7], the authors described setting up a highly reconfigurable cloud testbed to support research and education in experimentally assessing the practical impact of attacks and mitigations on realistic cloud systems. Compared to the previous authors, their testbed, known as Hyperdrive, runs on a combination of bare-metal hardware and virtualized systems. The authors cover experiments with constructing an access-driven side-channel attack that enables a malicious VM to extract fine-grained information from a victim VM running on the same physical computer. In [8], the authors illustrated examples of a low-cost testbed based on off-the-shelf hardware and open-source software to investigate security and privacy issues of many IoT devices, including HDMI sticks, IP cameras, activity trackers, smartwatches, and drones. Their research, supported by many actual experiments, indicates that many IoT devices have serious vulnerabilities. The Transpacific Testbed described in this paper is also based on a combination of bare-metal hardware and virtualized environment that provides a comprehensive platform for conducting various security experimentations.

III. TESTBED DESCRIPTION

The Transpacific testbed is a GRE tunnel through the internet2 that originates from the Center for Information Networking and Telecommunication (CINT) lab, The City College of New York(CCNY), USA, through the JGN network in Seattle and terminates at the Kyushu Institute of Technology(KYUTECH), Fukuoka, Japan. The testbed comprises two data-plane virtual local area networks (VLANs) (VLAN 1091 and 1092) and one control plane VLAN (VLAN 1941) for experiments, as shown in Fig. 1. Each VLAN contains 254 private IP address spaces. In this section, we give the full description of each side of the testbed's setup.

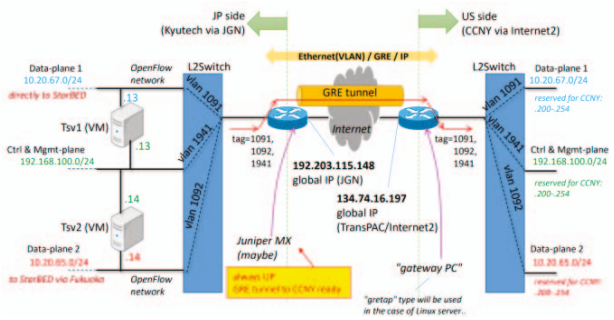

Fig. 1. The custom private blockchain

1. *CCNY Side:* The entrance to the CCNY side of the testbed has a dedicated PC that acts as a gateway to the network. The GRE tunnel is terminated at this gateway PC. The gateway PC has two network interfaces; public and private interfaces. The public interface opens to the GRE tunnel, which contains a routable public IP address, while the private interface features a private IP address. The private interface of the gateway PC is connected to a tagged port of a 4-ports layer2 switch. Each untagged port of the layer2 switch is configured to three separate VLANs mentioned above, as shown in Fig. 2. Each VLAN port is connected to a multiport layer2 switch using cat6 UTP internet cable for experiments. The nodes used for experiments in this VLAN are connected to these multiport switches.

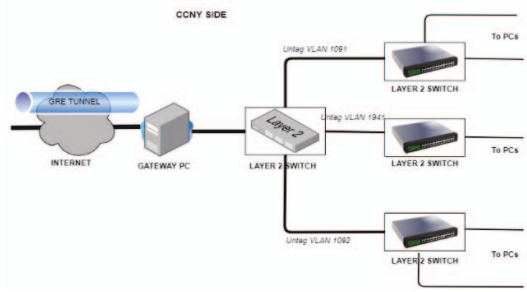

Fig. 2. CCNY side of the Testbed

2. *KYUTECH Side.* Unlike the CCNY side, the GRE tunnel at the KYUTECH side is terminated with a router. The router acts as the network's gateway, and it is connected to a tagged port of the layer2 switch. Like the CCNY side, the untagged switch port is configured to the same three VLANs. Each VLAN port is connected to a multiport layer2 switch, accommodating multiple node connections for experiments. Based on this configuration, VLANs 1091, 1092, and 1941 are replicated on both sides of the testbed. An experiment implementation conducted on VLAN 1091 can have part of the nodes set up in the USA while other nodes set up at the Japan side of the testbed, as described in section IV.

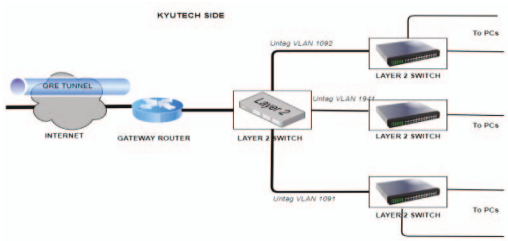

Fig. 3. KYUTECH side of the Testbed

3. *Testbed Testing and Troubleshooting*: After setting up the GRE tunnel between the two sites, we performed connectivity tests and measured the throughput. To measure overall network performance and determine possible throughput issues on the tunnel, we deployed the *perfSONAR* toolkit (Performance Service-Oriented Network monitoring Architecture) [9]. The perfSONAR allows creating a grouping of tests, more commonly known as a "mesh." A shared configuration file is written in the JSON format made with a graphical tool and published for other hosts to download. Each host then tests the different mesh members based on the configuration [10]. Using perfSONAR's *scheduler* service, we made several automated tests to measure:
   - Round trip time and related statistics between hosts
   - TCP/UDP throughput in both directions (uses built-in *iperf3* utility)
   - One way latency between hosts (uses *owping*)

Table 1 shows average TCP and UDP throughputs of incoming and outgoing traffic within the GRE tunnel after conducting more than 50 continuous tests.

Table 1. Average throughput between two labs after conducting more than fifty tests using perfSONAR.

Protocol	Source	Destination	Throughput (Mbits/sec)
TCP	CCNY lab	Kyutech lab	80
TCP	Kyutech lab	CCNY lab	75
UDP	CCNY lab	Kyutech lab	78
UDP	Kyutech lab	CCNY lab	72

During continuous automated latency tests, no packet loss was reported. Packet statistics and jitter measurements for every 1000 packets sent and received during one-way latency tests are shown in Table 2.

Table 2. One-way Latency Statistics (CCNY-Kyutech)

Data	Value (ms)
Delay Median	78.96
Delay Minimum	78.85
Delay Maximum	80.40
Delay Mean	78.97
Max Clock Error	0.13
Jitter (P95-P50)	0.06
Jitter (P75-P25)	0.04

IV. EXPERIMENTATIONS

We describe two real-time experiments that were conducted on the transpacific testbed. For each of these experiments, we describe the objective, setup, and result.

1. *Blockchain-based Cooperative Intrusion detection system (CoIDS)*

Objective: This work proposed a blockchain-based architecture that securely exchanges cyberattack signatures and features among participating nodes in real-time. This architecture allows companies to form a consortium and build a shared repository of up-to-date cyber attack features/signatures in real-time, allowing a non-participating company to download the attack information without any security concerns. This architecture uses blockchain technology's tamper-proof ability, distributive nature, and data immutability to build a resilient solution to common cyberattacks on stored data. The architecture is novel because it detects and prevents malicious activities on the stored data in real-time. Also, the architecture facilitates a real-time attack signature/feature exchange. It is robust to public companies joining and leaving the blockchain network in real-time without any security concern. [2], [11], and[12] show some of the major publications from this work. Figure 4 shows the pictorial representation of Blockchain-based CoIDS.

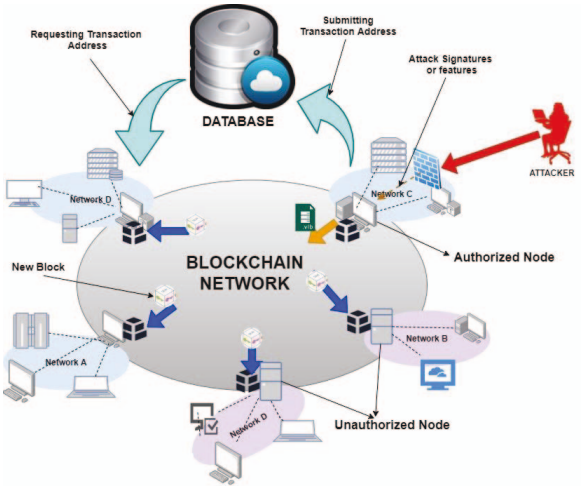

Fig. 4. Blockchain-Based CoIDS

The architecture, built on the Ethereum blockchain platform, describes how the signature or features of detected attacks are retrieved, turned into transactions, and submitted to the blockchain network for verification. It further describes the verification and validation steps of each transaction to detect malicious activities. A successfully verified transaction is attached to the blockchain, which is populated in the public

ledger of other participating nodes. The participating nodes could download the stored information after obtaining the transaction information from the database. [2], [11], and [12]. [17] describes the performance and security analysis using network dissemination latency, scalability, and ability to detect typical insider and outsider threats on stored data.

The architecture was tested in the laboratory. During the testing process, we performed the security analysis by conducting some typical outsider and insider attacks to show how the architecture performs in real-time. The results showed that the architecture could detect and prevent malicious activities on shared information. Also, we evaluated the performance using the architecture's dissemination latency and scalability as the network grows. The dissemination latency defines the time it takes for the attack feature/ signature to be successfully distributed to all participating nodes after an attack has been detected. At the same time, scalability defines the effect of expanding the network (i.e., increasing the number of participants) on the architecture's dissemination latency. The laboratory results showed that the latency is very low (typically less than 2 seconds).

Despite the promising results, it is challenging to ascertain the architecture's behavior when deployed in a real-life scenario. The reason is that the architecture's testing platforms have light network traffic, which does not depict a real-life situation. We deploy the architecture to the transpacific testbed to evaluate the performance for a near real-life situation. Here, we set up part of the blockchain network on the CCNY side while the other domain is set up at KYUTECH, as shown in Fig. 5. The two fragmented blockchain networks are set up over VLAN 1091 and connected between the two sites, and we measured both the security and performance metrics. Testing the architecture across these inter-domain networks enables us to assess its scalability, throughput, and transaction dissemination latency over a wide geographical area with diverse network traffic.

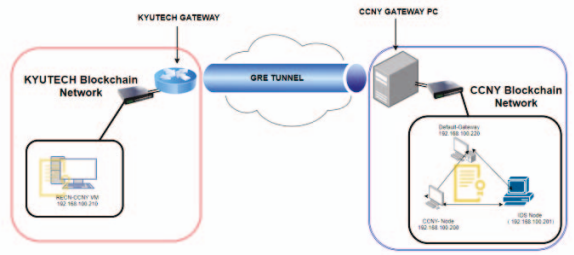

Fig. 5. The Blockchain-based CoIDS on the testbed

Results: The preliminary result of the architecture's performance is shown in Fig. 6 and 7. The preliminary experiment was implemented with three blockchain nodes, as shown in Fig. 5. For a detailed description of the setup and implementation, visit [2] and [12]. We measured each node's dissemination latency and average dissemination latency. Fig. 6 shows the dissemination latency of blockchain nodes for twenty transactions. As explained in [11], dissemination latency is from the time a transaction is submitted to the blockchain to the time each node retrieves the information from its ledger. This time comprises verification time, commit and mining time, new block broadcasting time, and time it is taken to retrieve it from their ledger. Fig. 7 shows the average dissemination time of each node. We observe that the architecture's average response time is low (less than 2 seconds), even for the node located in Japan. For further work, we will repeat the experiment but with more blockchain nodes on each side of the testbed and compares the result to what was obtained initially.

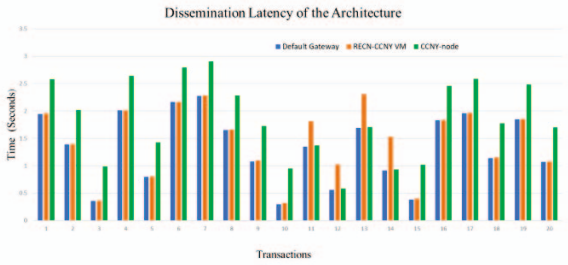

Fig. 6. The dissemination latency for blockchain nodes

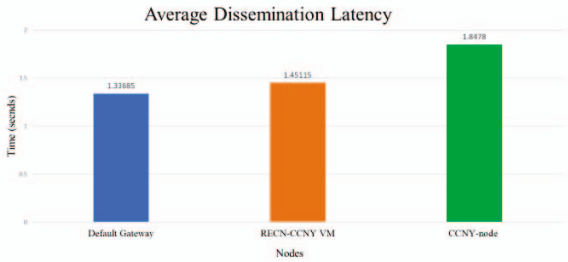

Fig. 7. The average dissemination latency of blockchain nodes

2. VM Introspection

Objective: The proposed work describes an Artificial Immune System (AIS) based Intrusion Detection System (IDS) that aims to introspect multiple Linux-based Virtual Machines (VMs) to detect malicious applications (such as keyloggers, spyware, rootkits, trojans, worms, etc.) while running from the outside of the infected VM. Developed software can reside on a client machine or operate directly within the host OS to continuously monitor multiple virtual machines. Once the suspicious process is detected based on anomalous behavior, the application in real-time notifies the system administrator and provides detailed information about the threat. Virtual Machine Introspection (VMI) addresses several security issues outside the guest OS without relying on functionality rendered unreliable by advanced malware. The proposed application uses a lightweight VMI module *KVMonitor* [13,14], developed in C to access VM's memory file located in the host OS and get all the necessary information about processes running in the VM. A controller is a cross-platform application developed in Python that utilizes an Artificial Immune System algorithm, known as Negative Selection Algorithm, to collect the data and detect potential anomalies by tracking the events (*interrupts, memory writes, network activities*, and so on) [15].

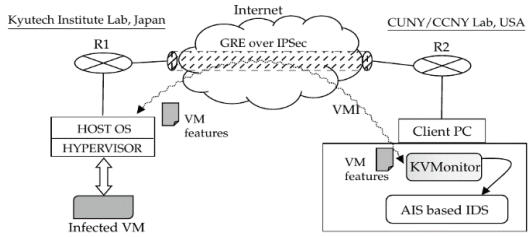

Fig. 8. VM introspection on the testbed

The overall architecture of the proposed application, as depicted below, provides security of cloud servers through automated virtual machine monitoring, intelligent analysis of network flows, and system-level process hidings, followed by mitigation actions being taken per the decision of intrusion detection component.

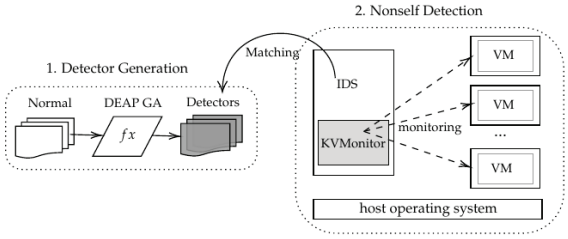

Fig. 9. Building blocks of the IDS

The end-to-end intrusion detection and mitigation process rely on two main steps: *Detector Generation* and *Nonself Detection*. As a part of the first phase, using a *Genetic Algorithm* within the DEAP framework [16] and feeding it in advance with a set of normal features, the application generates a sufficient number of detectors. During this process, squared (Euclidean) distance is implemented as a fitness function to measure the distance between self and randomly generated nonself features. In the second phase, IDS constantly introspects multiple virtual machines and returns raw features. The application converts these values into the appropriate binary forms and begins the matching process. If the IDS finds a match for any incoming set of features among the detectors, it immediately notifies the administrator about a potential anomaly.

Results: Developed application was initially tested on a local PC, with a pre-installed VM running within the same OS. After demonstrating high detection accuracy (F1 score 96.87%) and experimenting with over 50 different malicious applications, we deployed it to a remote host at the Kyutech. The IDS was successfully tested on remote VM through the established GRE tunnel. The maximum available bandwidth between the switch and hosts in the network is set to 100 Mbps. The packet sending rate was 1000 packets per second, and the payload of the packets was 1000 bytes. In this experiment, all features for 20,000 flow entries were collected in 416.4 milliseconds, whereas it took 280 milliseconds for retrieving the eight best features for the same number of flows. Remote VMI delivers a new way of securing cloud-based servers through the centralized controller. An IDS cannot be subverted by any malware running in the VM that resides outside of the guest machine. This also provides a significantly low-performance impact on VM and the host machine.

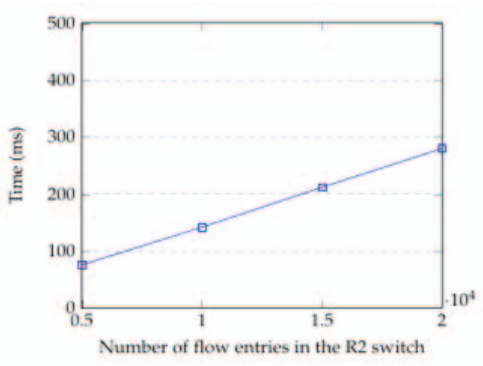

Fig. 10 Features retrieval and processing time

IDS should retrieve attack features quickly for efficient attack detection and prevention. Feature retrieval time is significant, as attacks need prompt detection before causing severe damage. Additionally, the feature retrieval process should not consume many client resources; otherwise, network performance would be adversely affected. We observe that the feature retrieval time increases linearly with the number of flow entries in the switch. However, our IDS performs feature processing on the fly and does not wait to finish every flow entry in the switch before taking action.

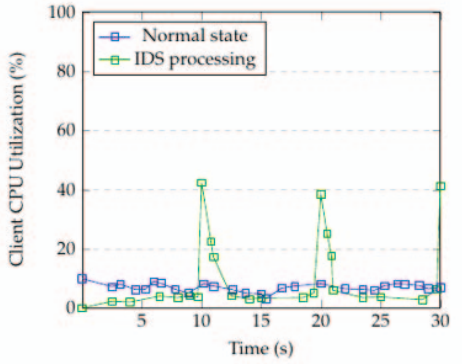

Fig. 11 Client CPU utilization for normal state Vs. IDS activity

The figure above shows the CPU utilization on the host machine during the normal state and while KVMonitor accesses the VM memory file. Remote offloading was 8.5% slower than local due to network delay and additional booting time using VNC and SSH. During intervals of VM introspection, the CPU of the host machine correlated around 2.8 - 3.4%, which is expected because *KVMonitor* is only accessing the QEMU-KVM memory file stored in the host in *qcow2* format. First, KVMonitor obtains the value of the CR3 register in a virtual CPU of a VM by communicating with QEMU-KVM. For this purpose, we added a new CR3 command to obtain the value to QEMU-KVM because QEMU-KVM provides only the command for dumping the values of all registers in[14].

Conducted experiments on more than 50 different malware show the average F1 score (detection rate) of the nonself detection 96.86%. We divided experiments into two parts:

first, we exposed remote VM separately to each of the listed malicious applications and measured performance and detection time. Second, we exposed the remote VM to all four listed malware simultaneously and then launched our IDS. In both cases, anomalies were detected with almost similar rates, and IDS successfully responded on time.

V. CONCLUSION AND ONGOING WORK

This paper described the setup and experimentation of a transpacific testbed between the USA and Japan. The testbed, a GRE tunnel through internet2, was built to define the future of the Internet by focusing on addressing research challenges associated with enabling trustworthy networks, supporting the Internet of Things (IoT), which encompasses everything connected to the Internet and cyber-physical systems (CPS). We described the tools used to evaluate the overall performance and troubleshoot. Furthermore, we described the two experiments that were conducted on the testbed and presented the results. The results show that blockchain-based CoIDS has a low dissemination latency. The proposed artificial immune system-based remote virtual machine introspection is efficient for achieving real-time, highly accurate detection and mitigation of attacks in various cloud-based servers.

Ongoing work: As part of the ongoing work, the transpacific testbed is being extended to an NSF-funded COSMOS Interconnecting Continent global testbed (COSM-IC). The COSMOS Interconnecting Continents (COSM-IC) is a National Science Foundation (NSF) sponsored global testbed spanning Asia, Europe, South, and North America continents. The testbed proposes to develop capabilities that will enhance unique multi-technology and software-defined wireless, optical, and edge cloud network testbed infrastructure to perform a groundbreaking international collaborative experiment. The goal will be achieved by leveraging the COSMOS [RSZ+20, COS20] and ORBIT [RSO05, BCL14] testbeds' interfaces with the PEERING [SAC+19] and FABRIC [FAB20] testbeds and adding connections to leading testbeds worldwide, including CPQD (Brazil) [CPQD20], Kyutech U./StarBED (Japan) [Sta20], OneLab/NITOS (EU/Greece) [One20, NIT20], and CONNECT and Pervasive Nation at Trinity College Dublin (EU/Ireland) [CON20]. After completion, the proposed experiments will be conducted on the global testbed Fig. 12 shows the extension of the transpacific testbed to COSM-IC.

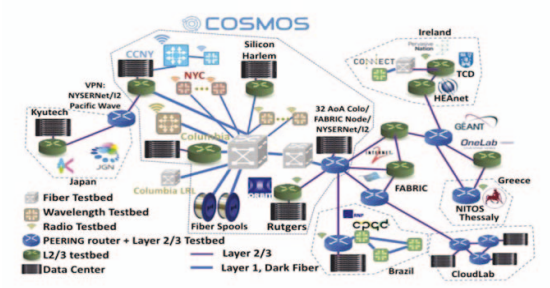

Fig. 12. COSM-IC global Testbed

ACKNOWLEDGEMENT

This work is supported in part by NSF JUNO2 (*Japan-US Network Opportunity 2*) (Award No. 1818884) and NSF IRNC (Award No. 2029295).

REFERENCES

[1] Sarica, A.K.; Angin, P. Explainable Security in SDN-Based IoT Networks. Sensors 2020, 20, 7326. https://doi.org/10.3390/s20247326

[2] O. Ajayi and T. Saadawi, "Blockchain-Based Architecture for Secured Cyber-Attack Features Exchange," 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York, NY, USA, 2020, pp. 100-107, doi: 10.1109/CSCloud-EdgeCom49738.2020.00025.

[3] K. Kourai and K. Juda, "Secure Offloading of Legacy IDSes Using Remote VM Introspection in Semi-trusted Clouds", 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 2016.

[4] Lantz, B.; Heller, B.; McKeown, N. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, CA, USA, 20–21 October 2010; ACM: New York, NY, USA, 2010; pp. 19:1–19:6.

[5] ONOS. Available online: accessed on 27 August 2021.

[6] Open vSwitch. Available online: accessed on 27 August 2021.

[7] A. Sanatinia et al., "Hyperdrive: A flexible cloud testbed for research and education," 2017 IEEE International Symposium on Technologies for Homeland Security (HST), 2017, pp. 1-4, doi: 10.1109/THS.2017.7943500.

[8] A. Tekeoglu and A. Ş. Tosun, "A Testbed for Security and Privacy Analysis of IoT Devices," 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2016, pp. 343-348, doi: 10.1109/MASS.2016.051.

[9] J. Zurawski, S. Balasubramanian, A. Brown, E. Kissel, A. Lake, M. Swany, B. Tierney, and M. Zekauskas, "perfSONAR: On-board diagnostics for big data." In IEEE International Conference on Big Data. 2013.

[10] perfSONAR. Performance Service-Oriented Network monitoring Architecture. accessed on 8.25.2021.

[11] O. Ajayi, M. Cherian and T. Saadawi, "Secured Cyber-Attack Signatures Distribution using Blockchain Technology." 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), New York, NY, USA, 2019, pp. 482-488.

[12] O. Ajayi, M. Abouali and T. Saadawi, "Secure Architecture for Inter-Healthcare Electronic Health Records Exchange," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 2020, pp. 1-6, doi: 10.1109/IEMTRONICS51293.2020.9216336.

[13] K. Kourai and K. Juda, "Secure Offloading of Legacy IDSes Using Remote VM Introspection in Semi-trusted Clouds", 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 2016.

[14] K. Kourai and K. Nakamura, "Efficient VM Introspection in KVM and Performance Comparison with Xen", Proceedings of the 2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing, 2014.

[15] H. Huseynov, K. Kourai, T. Saadawi, and O. Igbe, "Virtual Machine Introspection for Anomaly-Based Keylogger Detection" 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR).

[16] DEAP. Distributed Evolutionary Algorithms in Python. accessed on 01 August 2021.

[17] O. Ajayi and T. Saadawi, "Detecting Insider attacks in BlockchainNetworks" accepted for oral presentation at the 2021 International Symposium on Networks, Computers and Communications: Trust, Security, and Privacy (ISNCC-TSP), October 31 - November 2, 2021. Dubai UAE.