ToN_IoT Datasets for Network Traffic: New Generations of Heterogeneous Data sources in IoT Networks for Cyber Applications-based Artificial Intelligence

Nour Moustafa

Executive Summary

Existing security solutions, including intrusion detection, threat intelligence and hunting, privacy preservation and digital forensics, have a great interest in the cyber security domain. Those solutions have been basically developed using knowledge-based models that often can not trigger new families. With the boom of Artificial Intelligence (AI), especially deep learning, such security solutions have been plugged-in with AI models to discover, trace, mitigate, or respond to incidents of new security events. However, the models are not reliable in the industry due to the variety and complexity of new hacking categories, non-availability of heterogeneous data sources to train and validate AI models.

To address these open gaps, we designed a new architectural testbed at the IoT lab of UNSW Canberra to create new heterogeneous datasets that would be used to effectively evaluate the fidelity of new security solutions-based AI. The testbed was deployed in three tiers, edge, fog and cloud. The edge tier includes IoT and network devices, the fog layer involves virtual machines and gateways, and the cloud tier comprises cloud services such as data analytics and visualization linked to the other tiers. These layers were elastically managed using the technologies of software-Defined Network (SDN) and Network-Function Virtualization (NFV) using the VMware NSX and vCloud NFV platform.

While deploying the testbed, normal and attack scenarios were executed to gather labeled datasets. The datasets are named 'ToN_IoT' as they include data sources collected from **T**elemetry datasets of **IoT** services, **O**perating systems datasets of Windows and Linux, as well as datasets of **N**etwork traffic. The datasets have new properties: 1) a realistic testbed of design, 2) a variety of normal and attack events, 3) heterogeneous data sources, 4) a ground truth table of security events. The datasets can be publicly accessed from [1].

Proposed Testbed

The proposed testbed architecture of the ToN_IoT datasets is represented in Figure 1. The testbed was designed based on interacting network and IoT systems with the three layers of edge, fog and cloud to mimic the realistic implementation of recent real-world IoT networks. The dynamism of the three layers, including physical and simulated systems, is flexibly managed by the technologies of SDN and NVF. The NSX-VMware data center platform [2] is used to provide an SDN solution of the proposed testbed of the TON_IoT datasets. This technology permits the creation of overlay networks with the same capabilities of physical networks.

VMware NSX was deployed with VMware vSphere hypervisor NFV to allow the creation and management of various virtual machines that concurrently operate to offer the IoT/IIoT and network services. In VMware NSX, the vCloud NFV platform was employed to provide a modular design with abstractions that enable multi-domain, hybrid physical, and VM deployments [3]. The NSX vCloud NFV platform enables to design of a dynamic testbed IoT/IIoT network of the ToN_IoT with creating and controlling several VMs for hacking and normal operations, allowing the communications between the edge, fog and cloud layers.

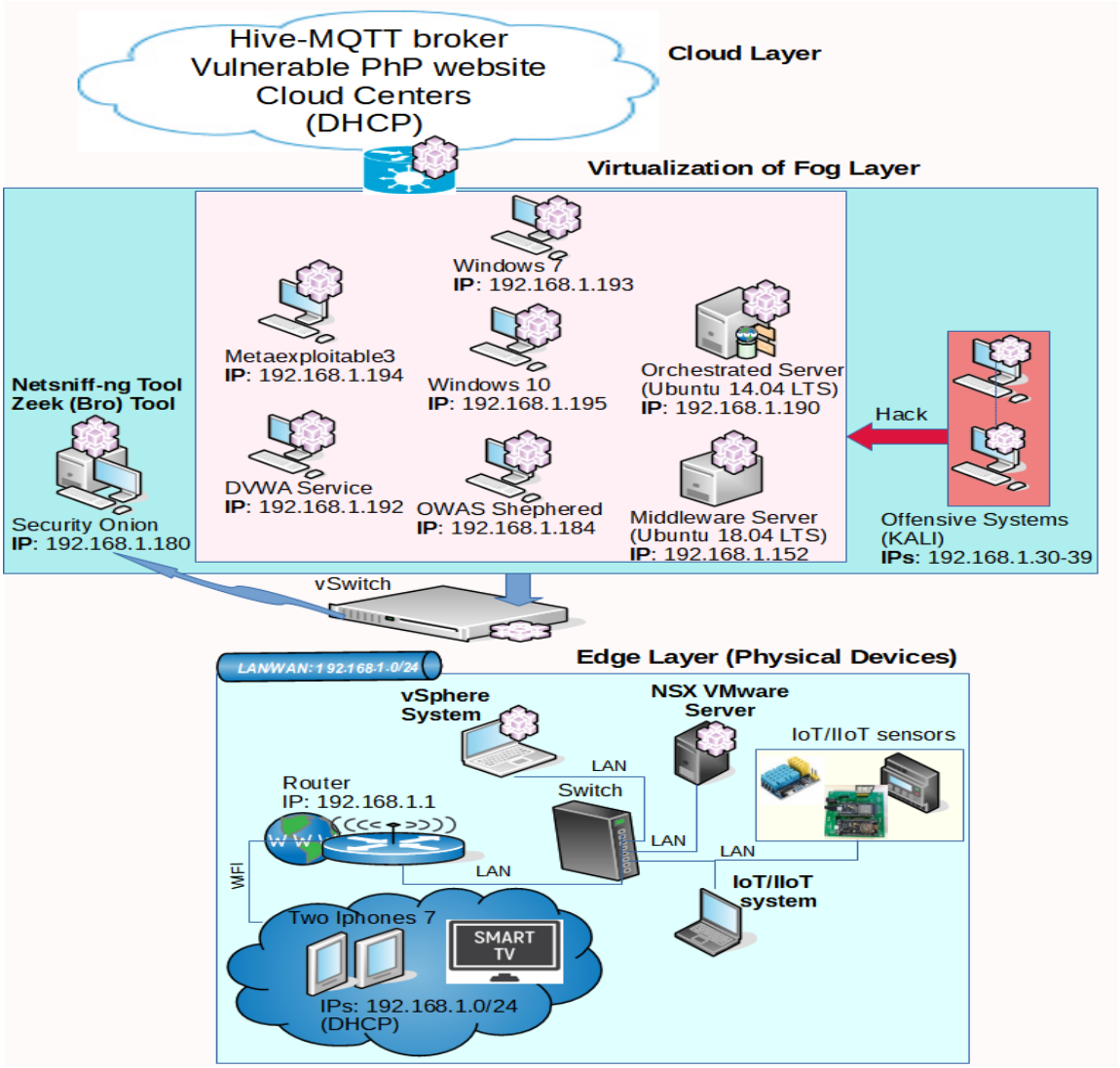

Figure 1: Configured Testbed of TON_IoT datasets for collecting network data

The components of the testbed are explained for the three layers as follows:

1. **Edge layer** – involves the physical devices and their operating systems utilized as the infrastructure of configuring the virtualization technology and cloud services at the layers of fog and cloud, respectively. It includes multiple IoT/IIoT devices, such as Modbus and light bulb sensors, smartphones and smart TVs, as well as host systems, such as workstations and servers, used to connect IoT/IIoT devices, hypervisors and physical gateways (i.e., routers and switches) to the Internet. The hypervisor technology of NSX-VMware was installed on a host server at the edge layer to manage the Virtual Machines (VMs) created at the fog layer.

2. **Fog layer**- includes the virtualization technology that controls the VMs and their services using the NSX-VMware and vCloud platform to offers the framework of executing SDN and NFV in the proposed testbed. The NSX vCloud NFV platform enables the design of a dynamic testbed

IoT/IIoT network of the ToN_IoT with creating and controlling several VMs for hacking and normal operations, allowing the communications between the edge, fog and cloud layers via vSwitches and gateways. This layer includes the nodes of virtual machines configured to generate the datasets, as explained in the following:

- **Orchestrated server**- is one of the main virtualized servers configured in the testbed using the Ubuntu 14.04 LTS with the IP address 192.168.1.190. This server offered many orchestrated services, such as FTP, Kerberos, HTTPS, and DNS to simulate real production networks and generate more simulated network traffic using the Ostinato Traffic Generator [4] that transmits traffic to other VMs in the testbed.

- **Middleware server**- is the IoT/IIoT virtualized server deployed in the testbed using the Ubuntu 18.04 with the IP address 192.168.1.152. This server included the scripts that run IoT/IIoT services through public and local MQTT gateways utilized in the testbed and linked with the cloud layer to subscribe and publish the telemetry data of IoT/IIoT sensors.

- **Client Systems-** includes a Windows 7 VM (IP address: 192.168.1.193), Windows 10 VM(192.168.1.195), DVWA web service (IP address: 192.168.1.192), OWASP security Sphered VM (192.168.1.184), Metaspoitable3 (IP address: 192.168.194). The two windows were used as the remote web interface of the node-red IP (192.168.1.152) and their network traffic and audit traces were logged. The DVWA (Damn Vulnerable Web App) [5] was utilized to make security vulnerabilities through web applications hacked using the virtualized offensive systems. The OWASP security Sphered VM [6] is an open-source platform that has many security vulnerabilities against mobile and web applications exploited using the offensive systems. In addition, the Metasploitable3 VM [7] was deployed in the testbed to increase vulnerable fog nodes and hack them using various attacking techniques by the offensive systems.

- **Offensive systems-** include the kali Linx VMs and scripts of hacking scenarios that exploit vulnerable systems in the testbed network. Ten static IP addresses (i.e.,192.168.1.30-39) were employed in the testbed to launch attacking scenarios and breach vulnerable systems either IoT/IIoT services (client and public MQTT brokers and node-red IP), operating systems (i.e., Windows 7 and 10, and Ubuntu 14.04 LTS and 18.04 LTS), and network systems (i.e., IP addresses and open protocols of the VMs).

- **Data Logger System** – is to log network traffic of the ToN_IoT datasets, the Security Onion VM [8] (IP address: 192.168.1.180) was used to log network data from all the active systems in the testbed using a virtual mirror switch that forwards the entire network traffic to this VM without dropping any traffic. As shown in Figure 1, the netsniff-ng tool [9] was used to capture the entire network packets from the entire systems in pcap formats without packet drops. The Zeek Network Security Monitor tool (previously named Bro) [10] was used to generate data features from the pcap files, discussed below.

- **Cloud layer**- contains the cloud services configured online in the testbed, as shown in Figure 1. The fog and edge services connected with the public HIVE MQTT dashboard [11], public PHP vulnerable website [12], cloud virtualization, and cloud data analytics services (e.g., Microsoft

Azure or AWS). The public HIV MQTT dashboard enabled us to publish and subscribe to the telemetry data of IoT/IIoT services via the configuration of the node-red tool. The public PHP vulnerable website used to launch injection hacking events against websites. The other cloud services were configured either in Microsoft Azure or AWS to transmit sensory data to the cloud and visualize their patterns.

Hacking Scenarios

Hacking scenarios were utilized to launch nine attack categories against vulnerable elements of IoT/IIoT applications, operating systems, network systems. The scripts and some links of the attacking categories are published in [13]. The nine attack families utilized in the datasets are explained as follows:

1. **Scanning attack**- We used the Nessus [14] and Nmap [15] tools from the offensive systems with IP addresses 192.168.1.20-38 against the target subnet 192.168.1.0/24 and all other public vulnerable systems such as the Public MQTT broker and vulnerable PHP website. For example, *nmap 192.168.1.40-254*, and the scans of the Nessus tool for the same range of IP addresses.
2. **Denial of Service (DoS) attack-** We utilized DoS attack scenarios on the offensive systems with IP addresses 192.168.1.{30,31,39} to hack vulnerable elements in the IoT testbed network. We created Python scripts using the Scapy package to launch the DoS attacks [16].
3. **Distributed Denial of Service (DDoS) attack-** We used DDoS attacks in the offensive systems wit IP addresses 192.168.1.{30,31,34,35,36,37,38} to breach several weaknesses in the IoT testbed network. We developed Python scripts using the Scapy package to launch the DoS attacks [16]. Further, automated bash scripts were developed to launch DDoS against vulnerable nodes of the testbed using the ufonet toolkit [17].
4. **Ransomware attack-** We utilised the Kali Linux with IP addresses 192.168.1.{33, 37} to execute this malware against windows operating systems and their webpages of monitoring IoT services included in the testbed network. This attack executed using the Metasploit framework that hacks the SMB vulnerability of the systems, named eternalblue [18].
5. **Backdoor attack -** We used the offensive systems with IP addresses 192.168.1.{33,37} to keep the hacking persistence using the Metasploit framework by executing a bash script of the command “run persistence -h” [19].
6. **Injection attack-** We used various injection scenarios from the offensive systems with IP addresses 192.168.1.{30, 31, 33, 35} to inject data inputs against web applications of DVWA and Security Shepherd VMs and webpages of IoT services through other VMs, including SQL injection, client-side injection, broken authentication and data management, and unintended data leakage.
7. **Cross-site Scripting (XSS) attack-** We employed the offensive systems with IP addresses 192.168.1.{32,35,36,39} to illegally inject web applications of DVWA and Security Shepherd VMs and webpages of IoT services through other VMs. In these systems, we created malicious bash scripts of python codes to hack the web applications of the testbed network using the Cross-Site Scripter toolkit (named XSSer) [20].
8. ***Password attack-*** We used the offensive systems with IP addresses 192.168.1.{30, 31, 32, 35, 38}. In these systems, the hydra [21] and cewl [22] toolkits were configured using automated bash scripts to concurrently launch password hacking scenarios against vulnerable nodes in the testbed.

9. **Man-In-The-Middle (MITM) attack-** We utilized the offensive systems with IP addresses 192.168.1.{31,34} to launch various MITM scenarios in the testbed network. In the systems, we employed the Ettercap tool [23] to execute ARP spoofing, ICMP redirection, port stealing and DHCP spoofing.

Acknowledgment

We thank the Australian Research Data Center (ARDC) *RG192500* and UNSW Canberra *PS51776* for funding this work.

References

1. ToN_IoT datasets, https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/, January 2020.
2. VMware SDN, https://lenovopress.com/lp0661.pdf, January 2020.
3. VMWARE VCLOUD NFV, https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nfv/vmware-vcloud-nfv-vcloud-director-edition-datasheet.pdf, January 2020.
4. Ostinato Traffic Generator, https://ostinato.org/, January 2020.
5. DVWA web service, http://www.dvwa.co.uk/
6. Owasp Security Shepherd https://owasp.org/www-project-security-shepherd/
7. Metasploitable3, https://github.com/rapid7/metasploitable3
8. Security onion, https://securityonion.net/
9. The netsniff-ng tool, http://netsniff-ng.org/
10. The Zeek/Bro tool, https://www.zeek.org/
11. Public Hive MQTT broker, https://www.hivemq.com/public-mqtt-broker/
12. Public PhP vulnerable website, http://testphp.vulnweb.com/
13. ToN_IoT hacking scenarios, https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i?path=%2FSecurityEvents_GroundTruth_datasets%20%20%2FHacking_scenarios%20
14. Nessus tool, https://www.tenable.com/products/nessus
15. Nmap tool, https://nmap.org/
16. DoS and DDoS attacks by Scapy, https://www.tutorialspoint.com/python_penetration_testing/python_penetration_testing_dos_and_ddos_attack.htm
17. Ufonet toolkit, https://ufonet.03c8.net/
18. Metasploit for eternal blue, https://null-byte.wonderhowto.com/how-to/exploit-eternalblue-windows-server-with-metasploit-0195413/
19. The backdoor attack, https://www.hacking-tutorial.com/hacking-tutorial/5-steps-to-set-up-backdoor-after-successfully-compromising-target-using-backtrack-5/#sthash.QgjjbNYM.ULO04i1b.dpbs
20. Cross-site-script toolkit, https://xsser.03c8.net/
21. Hydra tool, https://tools.kali.org/password-attacks/hydra
22. Cwel tool, https://tools.kali.org/password-attacks/cewl
23. Man-in-the-middle attack using Ettercap, https://www.1337pwn.com/how-to-perform-a-man-in-the-middle-attack-using-ettercap-in-kali-linux/