CSE-CIC-IDS2018 on AWS

A collaborative project between the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC)

Anomaly detection has been the main focus of many researchers' due to its potential in detecting novel attacks. However, its adoption to real-world applications has been hampered due to system complexity as these systems require a substantial amount of testing, evaluation, and tuning prior to deployment. Running these systems over real labeled network traces with a comprehensive and extensive set of intrusions and abnormal behavior is the most idealistic methodology for testing and evaluation.

This itself is a significant challenge, since the availability of datasets is extremely rare, because from one side, many such datasets are internal and cannot be shared due to privacy issues, and on the other hand the others are heavily anonymized and do not reflect current trends, or they lack certain statistical characteristics, so a perfect dataset is yet to exist. Thus, researchers must resort to datasets that are often suboptimal. As network behaviours and patterns change and intrusions evolve, it has very much become necessary to move away from static and one-time datasets towards more dynamically generated datasets, which not only reflect the traffic compositions and intrusions of that time, but are also modifiable, extensible, and reproducible.

To overcome these shortcomings, a systematic approach has been devised to generate datasets to analyze, test, and evaluate intrusion detection systems, with a focus towards network-based anomaly detectors. The main objective of this project is to develop a systematic approach to generate diverse and comprehensive benchmark dataset for intrusion detection based on the creation of user profiles which contain abstract representations of events and behaviours seen on the network. The profiles will be combined to generate a diverse set of datasets each with a unique set of features, which covers a portion of the evaluation domain.

The final dataset includes seven different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. The attacking infrastructure includes 50 machines and the victim organization has 5 departments and includes 420 machines and 30 servers. The dataset includes the captures network traffic and system logs of each machine, along with 80 features extracted from the captured traffic using CICFlowMeter-V3 (//www.unb.ca/cic/research/applications.html#CICFlowMeter).

1. Introduction

In CSE-CIC-IDS2018 dataset, we use the notion of profiles to generate datasets in a systematic manner, which will contain detailed descriptions of intrusions and abstract distribution models for applications, protocols, or lower level network entities. These profiles can be used by agents or human operators to generate events on the network. Due to the abstract nature of the generated profiles, we can apply them to a diverse range of network protocols with different topologies. Profiles can be used together to generate a dataset for specific needs. We will build two distinct classes of profiles:

B-profiles: Encapsulate the entity behaviours of users using various machine learning and statistical analysis techniques (such as K-Means, Random Forest, SVM, and J48). The encapsulated features are distributions of packet sizes of a protocol, number of packets per flow, certain patterns in the payload, size of payload, and request time distribution of a protocol. The following protocols will be simulated in our testbed environment: HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP. Based on our initial observations majority of traffic is HTTP and HTTPS.

M-Profiles: Attempt to describe an attack scenario in an unambiguous manner. In the simplest case, humans can interpret these profiles and subsequently carry them out. Idealistically, autonomous agents along with compilers would be employed to interpret and execute these scenarios. For attacks we considered six different scenarios (Table 1):

Infiltration of the network from inside: Infiltration of the network from inside: In this scenario, we send a malicious file via an email to the victim and exploit an application vulnerability. After successful exploitation, a backdoor will be executed on the victim’s computer and then we use his computer to scan the internal network for other vulnerable boxes and exploit them if possible.

HTTP denial of service: HTTP denial of service: In this scenario, we utilize Slowloris and LOIC as our main tools, which have been proven to make Web servers completely inaccessible using a single attacking machine. Slowloris starts by making a full TCP connection to the remote server. The tool holds the connection open by sending valid, incomplete HTTP requests to the server at regular intervals to keep the sockets from closing. Since any Web server has a finite ability to serve connections, it will only be a matter of time before all sockets are used up and no other connection can be made. Also, HOIC is another famous application which can launch DoS attacks against websites.

Collection of web application attacks: Collection of web application attacks: In this scenario, we use Damn Vulnerable Web App (DVWA), which is developed to be an aid for security professionals to test their skills, as our victim web application. In the first step, we scan the website through a web application vulnerability scanner and then we conduct different types of web attacks on the vulnerable website, including SQL injection, command injection, and unrestricted file upload.

Brute force attacks: Brute force attacks: Brute force attacks are very common against networks as they tend to break into accounts with weak username and password combinations. The final scenario has been designed with the goal of acquiring an SSH and MySQL account by running a dictionary brute force attack against the main server.

Last updated attacks: Last updated attacks: There are some attacks based on some famous vulnerabilities that can be conducted during a specific amount of time (these are extraordinary vulnerabilities which sometimes affects millions of servers or victims, and normally it takes months to patch all vulnerable computers around the world), one of the most famous ones in recent years is Heartbleed.

Table 1: List of executed attacks and duration

Attack	Tools	Duration	Attacker	Victim
Bruteforce attack	FTP – Patator SSH – Patator	One day	Kali linux	Ubuntu 16.4 (Web Server)
DoS attack	Hulk, GoldenEye, Slowloris, Slowhttptest	One day	Kali linux	Ubuntu 16.4 (Apache)

DoS attack	Heartleech	One day	Kali linux	Ubuntu 12.04 (Open SSL)
Web attack	• Damn Vulnerable Web App (DVWA) • In-house selenium framework (XSS and Brute-force)	Two days	Kali linux	Ubuntu 16.4 (Web Server)
Infiltration attack	• First level: Dropbox download in a windows machine • Second Level: Nmap and portscan	Two days	Kali linux	Windows Vista and Macintosh
Botnet attack	• Ares (developed by Python): remote shell, file upload/download, capturing • screenshots and key logging	One day	Kali linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit)
DDoS+PortScan	Low Orbit Ion Canon (LOIC) for UDP, TCP, or HTTP requests	Two days	Kali linux	Windows Vista, 7, 8.1, 10 (32-bit) and 10 (64-bit)

It is important to note that a profile needs an infrastructure to be used effectively. Our testbed will consist of some interconnected Windows and Linux based workstations. For Windows machines, we will use different service packs (because each pack has a diverse set of known vulnerabilities) and for Linux machines we will use Metasploit-able distribution, which is developed for being attacked by the new penetration testers.

2. Infrastructure and implementation

2.1 B-Profile

To produce benign background traffic, B-Profile is designed to extract the abstract behaviour of a group of human users. It tries to encapsulate network events produced by users with machine learning and statistical analysis techniques. The encapsulated features are distributions of packet sizes of a protocol, number of packets per flow, certain patterns in the payload, size of payload, and request time distribution of protocols. Once B-Profiles are derived from users, an agent (CIC-BenignGenerator (https://www.riverpublishers.com/journal_read_html_article.php?j=JSN/2017/1/009)) or a human operator can use them to generate realistic benign events on the network. Organizations and researchers can use this approach to easily generate realistic datasets; therefore, there is no need to anonymize datasets.

2.2 M-Profile

We have implemented seven attack scenarios. For each attack, we define a scenario based on the implemented network topology and execute the attack from one or more machines outside the target network. Figure 1 shows the implemented network which is a common LAN network topology on the AWS computing platform. To have a diversity of machines similar to real-world networks, we have installed 5 subnets, namely R&D department (Dep1), Management Department (Dep2), Technician department (Dep3), Secretary and operation department (Dep4), IT department (Dep5), and server rooms. For all departments except the IT department we have installed sets of different MS Windows OSs (Windows 8.1 and Windows 10) and all computers in the IT department are Ubuntu. For the server room, we implemented, different MS Windows servers such as 2012 and 2016. The rest of this section presents the seven attacks scenarios and tools.

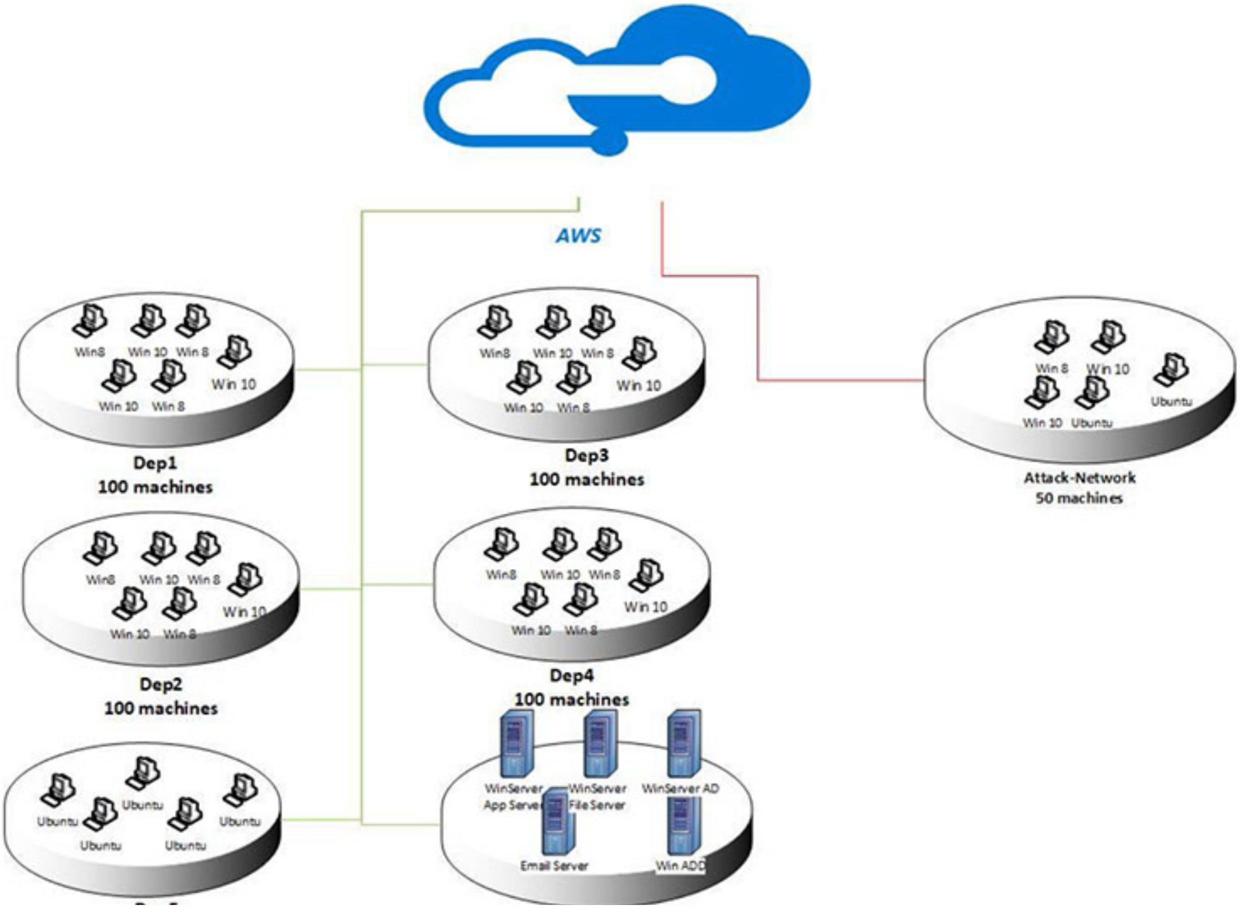

Figure 1: Network Topology

2.2.1. Brute-force attack

There are many tools for conducting brute-force attacks and password cracking such as Hydra, Medusa, Ncrack, Metasploit modules, and Nmap NSE scripts. Also, there are some tools such as hashcat and hashpump for password hash cracking. But one of the most comprehensive multi-threaded tools is Patator, which is written in Python and seems to be more reliable and flexible than others. It can also save every response in a separate log file for later review. In this dataset we use two modules, FTP and SSH on the Kali Linux machine as the attacker machine and an Ubuntu 14.0 system as the victim machine. For a list of passwords, we use a large dictionary that contains 90 million words.

2.2.2 Heartbleed attack

One of the most famous tools to exploit Heartbleed is Heartleech. It can scan for systems vulnerable to the bug, and can then be used to exploit them and exfiltrate data. Some important features:

- Conclusive/inconclusive verdicts as to whether the target is vulnerable
- Bulk/fast download of heartbleed data into a large file for offline processing using many threads
- Automatic retrieval of private keys with no additional steps
- Some limited IDS evasion
- STARTTLS support
- IPv6 support
- Tor/Socks5n proxy support
- Extensive connection diagnostic information

To exploit the vulnerability, we compiled OpenSSL version 1.0.1f, which is a vulnerable version. Then we use Heartleech to retrieve the memory of the server.

2.2.3 Botnet

In this dataset we use Zeus, which is a Trojan horse malware package that runs on versions of

Microsoft Windows. While it can be used to carry out many malicious and criminal tasks, it is often used to steal banking information by man-in-the-browser keystroke logging and form grabbing. It is also used to install the Crypto-Locker ransomware. Zeus is spread mainly through drive-by downloads and phishing schemes. Also, as a complement we use Ares botnet which is an open source botnet and has the following capabilities:

- remote cmd.exe shell
- persistence
- file upload/download
- screenshot
- key logging

In this scenario, we infect machines with two different botnets (Zeus and Ares), also every 400 seconds we request screenshots from the zombies.

2.2.4 Denial-of-Service

Slowloris is a type of denial of service attack tool invented by Robert Hansen which allows a single machine to take down another machine's web server with minimal bandwidth and side effects on unrelated services and ports. In this scenario, we use a Slowloris Perl-based tool to take down the web server.

2.2.5 Distributed Denial-of-Service

The High Orbit Ion Cannon, often abbreviated to HOIC, is an open source network stress testing and denial-of-service attack application written in BASIC designed to attack as many as 256 URLs at the same time. It has been designed to replace the Low Orbit Ion Cannon which was developed by Praetox Technologies. In this scenario, we use free HOIC tool to conduct DDoS attack by using 4 different computers.

2.2.6 Web Attacks

In this work, we use Damn Vulnerable Web App (DVWA) to conduct our attacks. DVWA is a PHP/MySQL web application that is vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment. To automate the attacks in XSS and Brute-force section we developed an automation code with Selenium framework.

2.2.7 Infiltration of the network from inside

In this scenario, a vulnerable application (such as Adobe Acrobat Reader 9) should be exploited. First the victim receives a malicious document through the email. Then, after successful exploitation using Metasploit framework, a backdoor will be executed on the victim's computer. Now we can conduct different attacks on the victim's network include IP sweep, full port scan and service enumerations using Nmap.

3. Capturing data and final dataset

Based on all selected attacks and defined scenarios in previous section, we implemented the infrastructure and execute the attack scenarios. Table 2 shows, the list of attacks, related attackers and victims IP(s), Date, start and finish time of attack(s).

Table 2: List of daily attacks, Machine IPs, Start and finish time of attack(s)

Attacker	Victim	Attack Name	Date	Attack Start Time	Attack Finish Time
172.31.70.4 (Valid IP:18.221.219.4)	172.31.69.25 (Valid IP:18.217.21.148)	FTP-BruteForce	Wed-14-02-2018	10:32	12:09
172.31.70.6 (Valid IP:13.58.98.64)	18.217.21.148- 172.31.69.25	SSH-Bruteforce	Wed-14-02-2018	14:01	15:31
172.31.70.46 (Valid IP:18.219.211.138)	18.217.21.148- 172.31.69.25	DoS-GoldenEye	Thurs-15-02-2018	9:26	10:09
172.31.70.8 (Vazlid IP:18.217.165.70)	18.217.21.148- 172.31.69.25	DoS-Slowloris	Thurs-15-02-2018	10:59	11:40

Attacker	Victim	Attack Name	Date	Attack Start Time	Attack Finish Time
172.31.70.23 (Valid IP: 13.59.126.31)	18.217.21.148- 172.31.69.25	DoS-SlowHTTPTest	Fri-16-02-2018	10:12	11:08
172.31.70.16 (Valid IP:18.219.193.20)	18.217.21.148- 172.31.69.25	DoS-Hulk	Fri-16-02-2018	13:45	14:19
18.218.115.60 18.219.9.1 18.219.32.43 18.218.55.126 52.14.136.135 18.219.5.43 18.216.200.189 18.218.229.235 18.218.11.51 18.216.24.42	18.217.21.148- 172.31.69.25	DDoS attacks-LOIC-HTTP	Tues-20-02-2018	10:12	11:17
18.218.115.60 18.219.9.1 18.219.32.43 18.218.55.126 52.14.136.135 18.219.5.43 18.216.200.189 18.218.229.235 18.218.11.51 18.216.24.42	18.217.21.148- 172.31.69.25	DDoS-LOIC-UDP	Tues-20-02-2018	13:13	13:32
18.218.115.60 18.219.9.1 18.219.32.43 18.218.55.126 52.14.136.135 18.219.5.43 18.216.200.189 18.218.229.235 18.218.11.51 18.216.24.42	18.218.83.150-172.31.69.28	DDOS-LOIC-UDP	Wed-21-02-2018	10:09	10:43

Attacker	Victim	Attack Name	Date	Attack Start Time	Attack Finish Time
18.218.115.60 18.219.9.1 18.219.32.43 18.218.55.126 52.14.136.135 18.219.5.43 18.216.200.189 18.218.229.235 18.218.11.51 18.216.24.42	18.218.83.150-172.31.69.28	DDOS-HOIC	Wed-21-02-2018	14:05	15:05
18.218.115.60	18.218.83.150-172.31.69.28	Brute Force - Web	Thurs-22-02-2018	10:17	11:24
18.218.115.60	18.218.83.150-172.31.69.28	Brute Force - XSS	Thurs-22-02-2018	13:50	14:29
18.218.115.60	18.218.83.150-172.31.69.28	SQL Injection	Thurs-22-02-2018	16:15	16:29
18.218.115.60	18.218.83.150-172.31.69.28	Brute Force - Web	Fri-23-02-2018	10:03	11:03
18.218.115.60	18.218.83.150-172.31.69.28	Brute Force - XSS	Fri-23-02-2018	13:00	14:10
18.218.115.60	18.218.83.150-172.31.69.28	SQL Injection	Fri-23-02-2018	15:05	15:18
13.58.225.34	18.221.148.137-172.31.69.24	Infiltration	Wed-28-02-2018	10:50	12:05
13.58.225.34	18.221.148.137-172.31.69.24	Infiltration	Wed-28-02-2018	13:42	14:40
13.58.225.34	18.216.254.154-172.31.69.13	Infiltration	Thursday-01-03-2018	9:57	10:55
13.58.225.34	18.216.254.154-172.31.69.13	Infiltration	Thursday-01-03-2018	14:00	15:37
13.58.225.34	18.216.254.154-172.31.69.13	Infiltration	Thursday-01-03-2018	14:00	15:37

Attacker	Victim	Attack Name	Date	Attack Start Time	Attack Finish Time
18.219.211.138	18.217.218.111-172.31.69.23 18.222.10.237-172.31.69.17 18.222.86.193-172.31.69.14 18.222.62.221-172.31.69.12 13.59.9.106-172.31.69.10 18.222.102.2-172.31.69.8 18.219.212.0-172.31.69.6 18.216.105.13-172.31.69.26 18.219.163.126-172.31.69.29 18.216.164.12-172.31.69.30	Bot	Friday-02-03-2018	10:11	11:34
18.219.211.138	18.217.218.111-172.31.69.23 18.222.10.237-172.31.69.17 18.222.86.193-172.31.69.14 18.222.62.221-172.31.69.12 13.59.9.106-172.31.69.10 18.222.102.2-172.31.69.8 18.219.212.0-172.31.69.6 18.216.105.13-172.31.69.26 18.219.163.126-172.31.69.29 18.216.164.12-172.31.69.30	Bot	Friday-02-03-2018	14:24	15:55

3. Feature extraction

CICFlowMeter (//www.unb.ca/cic/research/applications.html#CICFlowMeter) is a network traffic flow generator which has been written in Java and offers more flexibility in terms of choosing the features you want to calculate, adding new ones, and having a better control of the duration of the flow timeout. It generates Bidirectional Flows (Biflow), where the first packet determines the forward (source to destination) and backward (destination to source) directions, hence the 83 statistical features such as Duration, Number of packets, Number of bytes, Length of packets, etc. are also calculated separately in the forward and reverse direction.

The output of the application is in CSV file format with six columns labeled for each flow, namely FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort, and Protocol with more than 80 network traffic features. Normally the TCP flows are terminated upon connection teardown (by FIN packet) while UDP flows are terminated by a flow timeout. The flow timeout value can be assigned arbitrarily by the individual scheme, e.g. 600 seconds for both TCP and UDP. The CICFlowMeter-V3 (//www.unb.ca/cic/research/applications.html) can extract more than 80 features which are listed in the table below:

Table 3: List of extracted traffic features by CICFlowMeter-V3 (//www.unb.ca/cic/research/applications.html#CICFlowMeter)

Feature Name	Description
fl_dur	Flow duration
tot_fw_pk	Total packets in the forward direction
tot_bw_pk	Total packets in the backward direction
tot_l_fw_pkt	Total size of packet in forward direction
fw_pkt_l_max	Maximum size of packet in forward direction
fw_pkt_l_min	Minimum size of packet in forward direction
fw_pkt_l_avg	Average size of packet in forward direction
fw_pkt_l_std	Standard deviation size of packet in forward direction
Bw_pkt_l_max	Maximum size of packet in backward direction
Bw_pkt_l_min	Minimum size of packet in backward direction
Bw_pkt_l_avg	Mean size of packet in backward direction
Bw_pkt_l_std	Standard deviation size of packet in backward direction
fl_byt_s	flow byte rate that is number of packets transferred per second
fl_pkt_s	flow packets rate that is number of packets transferred per second
fl_iat_avg	Average time between two flows
fl_iat_std	Standard deviation time two flows
fl_iat_max	Maximum time between two flows
fl_iat_min	Minimum time between two flows
fw_iat_tot	Total time between two packets sent in the forward direction
fw_iat_avg	Mean time between two packets sent in the forward direction

fw_iat_std	Standard deviation time between two packets sent in the forward direction
fw_iat_max	Maximum time between two packets sent in the forward direction
fw_iat_min	Minimum time between two packets sent in the forward direction
bw_iat_tot	Total time between two packets sent in the backward direction
bw_iat_avg	Mean time between two packets sent in the backward direction
bw_iat_std	Standard deviation time between two packets sent in the backward direction
bw_iat_max	Maximum time between two packets sent in the backward direction
bw_iat_min	Minimum time between two packets sent in the backward direction
fw_psh_flag	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
bw_psh_flag	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
fw_urg_flag	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
bw_urg_flag	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
fw_hdr_len	Total bytes used for headers in the forward direction
bw_hdr_len	Total bytes used for headers in the forward direction
fw_pkt_s	Number of forward packets per second
bw_pkt_s	Number of backward packets per second
pkt_len_min	Minimum length of a flow
pkt_len_max	Maximum length of a flow
pkt_len_avg	Mean length of a flow
pkt_len_std	Standard deviation length of a flow
pkt_len_va	Minimum inter-arrival time of packet
fin_cnt	Number of packets with FIN
syn_cnt	Number of packets with SYN
rst_cnt	Number of packets with RST

pst_cnt	Number of packets with PUSH
ack_cnt	Number of packets with ACK
urg_cnt	Number of packets with URG
cwe_cnt	Number of packets with CWE
ece_cnt	Number of packets with ECE
down_up_ratio	Download and upload ratio
pkt_size_avg	Average size of packet
fw_seg_avg	Average size observed in the forward direction
bw_seg_avg	Average size observed in the backward direction
fw_byt_blk_avg	Average number of bytes bulk rate in the forward direction
fw_pkt_blk_avg	Average number of packets bulk rate in the forward direction
fw_blk_rate_avg	Average number of bulk rate in the forward direction
bw_byt_blk_avg	Average number of bytes bulk rate in the backward direction
bw_pkt_blk_avg	Average number of packets bulk rate in the backward direction
bw_blk_rate_avg	Average number of bulk rate in the backward direction
subfl_fw_pk	The average number of packets in a sub flow in the forward direction
subfl_fw_byt	The average number of bytes in a sub flow in the forward direction
subfl_bw_pkt	The average number of packets in a sub flow in the backward direction
subfl_bw_byt	The average number of bytes in a sub flow in the backward direction
fw_win_byt	Number of bytes sent in initial window in the forward direction
bw_win_byt	# of bytes sent in initial window in the backward direction
Fw_act_pkt	# of packets with at least 1 byte of TCP data payload in the forward direction
fw_seg_min	Minimum segment size observed in the forward direction
atv_avg	Mean time a flow was active before becoming idle
atv_std	Standard deviation time a flow was active before becoming idle
atv_max	Maximum time a flow was active before becoming idle

atv_min	Minimum time a flow was active before becoming idle
idl_avg	Mean time a flow was idle before becoming active
idl_std	Standard deviation time a flow was idle before becoming active
idl_max	Maximum time a flow was idle before becoming active
idl_min	Minimum time a flow was idle before becoming active

After extracting the features and creating the CSV file, now we need to label the data. Here we used our attack scenarios schedule and the IPs and ports of the source and destination along with the protocol name to label the data per flow.

4. Using the dataset

The dataset has been organized per day. For each day, we recorded the raw data including the network traffic (Pcaps) and event logs (windows and Ubuntu event Logs) per machine. In features extraction process from the raw data, we used the CICFlowMeter-V3 and extracted more than 80 traffic features and saved them as a CSV file per machine.

If you want to use the AI techniques to analyze, you can download our generated data (CSV) files and analyze the network traffic.

If you want to use a new feature extractor, you can use the raw captured files (PCAP and Logs) to extract your features. And then, you can use the data mining techniques for analyzing the generated data.

5. License

You may redistribute, republish, and mirror the CSE-CIC-IDS2018 dataset in any form. However, any use or redistribution of the data must include a citation to the CSE-CIC-IDS2018 dataset and a link to this page in AWS (https://registry.opendata.aws/cse-cic-ids2018).

Research paper outlining the details of analyzing the similar IDS/IPS dataset and related principles

- Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization (https://

www.semanticscholar.org/paper/Toward-Generating-a-New-Intrusion-Detection-Dataset-Sharafaldin-Lashkari/a27089efabc5f4abd5ddf2be2a409bff41f31199)", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018

To download this dataset

- Install the AWS CLI (https://aws.amazon.com/cli/), available on Mac, Windows and Linux
- Run: *aws s3 sync --no-sign-request --region <your-region> "s3://cse-cic-ids2018/" dest-dir* (Where your-region is your region from the AWS regions list (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html) and dest-dir is the name of the desired destination folder in your machine)

If you are interest in the CSE-CIC-IDS2018 on AWS, you may also be interested in the Large-Scale Intrusion Detection Dataset (BCCC-CSE-CIC-IDS2018) (https://www.yorku.ca/research/bccc/ucs-technical/cybersecurity-datasets-cds/) made available by our colleagues at the Behaviour-Centric Cybersecurity Center (https://www.yorku.ca/research/bccc/), York University.

Online & Continuing Ed ›

Leadership ›

Connect with UNB

Contact UNB (//www.unb.ca/contact/)

(https://www.facebook.com/uofnb) (https://twitter.com/UNB)

(https://ca.linkedin.com/school/university-of-new-brunswick/)

Privacy › Web feedback ›