Data Article

Cyber attack evaluation dataset for deep packet inspection and analysis

Shishir Kumar Shandilya\textsuperscript{a}, Chirag Ganguli\textsuperscript{a}, Ivan Izonin\textsuperscript{b,*}, Prof. Atulya Kumar Nagar\textsuperscript{c}

\textsuperscript{a}Vellore Institute of Technology, VIT Bhopal University, Bhopal, India
\textsuperscript{b}Lviv Polytechnic National University, Ukraine
\textsuperscript{c}Liverpool Hope University, United Kingdom

\textbf{A R T I C L E  I N F O}

\textbf{Article history:}
Received 25 October 2022
Revised 14 November 2022
Accepted 18 November 2022
Available online 24 November 2022

\textbf{Keywords:}
Cyber attacks
Evaluation dataset
Attack techniques
Defense mechanisms

\textbf{A B S T R A C T}

To determine the effectiveness of any defense mechanism, there is a need for comprehensive real-time network data that solely references various attack scenarios based on older software versions or unprotected ports, and so on. This presented dataset has entire network data at the time of several cyber attacks to enable experimentation on challenges based on implementing defense mechanisms on a larger scale. For collecting the data, we captured the network traffic of configured virtual machines using Wireshark and tcpdump. To analyze the impact of several cyber attack scenarios, this dataset presents a set of ten computers connected to Router1 on VLAN1 in a Docker Bridge network, that try and exploit each other. It includes browsing the web and downloading foreign packages including malicious ones. Also, services like File Transfer Protocol (FTP) and Secure Shell (SSH) were exploited using several attack mechanisms. The presented dataset shows the importance of updating and patching systems to protect themselves to a greater extent, by following attack tactics on older versions of packages as compared to the newer and updated ones. This dataset also includes an Apache Server hosted on a different subset of VLAN2 which is connected to the VLAN1 to demonstrate

\* Corresponding author.
\textit{E-mail address: ivan.v.izonin@lpnu.ua} (I. Izonin).

\textit{Social media:} @theshishir (S.K. Shandilya), @chirag_ganguli (C. Ganguli), @IvanIzonin (I. Izonin), @AtulyaNagar (Prof.A.K. Nagar)

\url{https://doi.org/10.1016/j.dib.2022.108771}
isolation and cross-VLAN communication. The services on this web server were also exploited by the previously stated ten computers. The attack types include Distributed Denial of Service, SQL Injection, Account Takeover, Service Exploitation (SSH, FTP), DNS and ARP Spoofing, Scanning and Firewall Searching and Indexing (using Nmap), Hammering the services to brute-force passwords and usernames, Malware attacks, Spoofing, and Man-in-the-Middle Attack. The attack scenarios also show various scanning mechanisms and the impact of Insider Threats on the entire network.

© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

| Table 1 | Specifications |
|---------|----------------|
| **Subject** | Computer Networks and Communication |
| **Specific subject area** | Simulation Dataset for analysis of various Attack and Defense scenarios |
| **Type of data** | Image |
| **Chart Graph Figure** | |
| **How the data were acquired** | All the data were generated in real-time using virtual machines running in a sandboxed (containerized) environment. Two VLANs namely VLAN1 and VLAN2 were created as part of the network-level organization. Thereafter, 10 virtual machines were attached to the VLAN1 network, and an Apache server was attached to the VLAN2 network to separate it so that it is unavailable locally to the attached machines. The attack and defense scenarios were developed considering the machines attached to the VLAN1 where they perform absolute permutations in attacking each other in their local subnet based on their vulnerabilities, and the web server on VLAN2 to exploit its capacity and gain Remote Code Execution to the server. |
| **Data format** | Raw CSV Analyzed CSV Filtered XLSX |
| **Real-time Capture pcap** | |
| **SQL format for dataset regeneration** | |
| **Description of Data Collection** | For the purpose of collecting the raw packet data, 11 virtual machines were created in a containerized environment where they were attached to a completely different network bridged to the host machine. Two separate bridge networks were created, one for connecting the 10 attack/victim machines and the other for maintaining the web server such that they are kept different from each other and cannot ping each other locally. Several attack scripts were executed simultaneously on the attack/victim machines to have different combinations of attack and defense scenarios for a period of 23 hours, 31 minutes, and 54 seconds. The collected data were then converted from Raw pcaps to a Database format using SQL such that they can be easily analyzed and altered. Also, the IP addresses of the known machines were flagged and annotated in the database to have a brief understanding of the already-known and foreign IPs. The machines were left connected to the Internet where they acted like normal day-to-day computers downloading online data and performing daily operations exploiting using specially crafted payloads. This provided a real-life operation scenario of attack and defense in regular machines that are connected to the Internet. |
| **Data Source Location** | • Institution: VIT Bhopal University  
• City/Town/Region: Bhopal, Madhya Pradesh  
• Country: India  
• Latitude and Longitude: 23.0774° N, 76.8513° E |
| **Data accessibility** | The dataset can be downloaded from  
The full version of the dataset is freely available at https://data.mendeley.com/datasets/3szjvt3w78 |

Table 1 provides a brief overview of the specifications of the dataset presented in this paper which includes the Subject area of the dataset, the type of the data, how the data was acquired,
the format of the data, a description of the data collection, source location and accessibility of the presented dataset.

Value of the Data

Communication systems form an important factor in the daily life of users. Computer Networks are used for various data processing, learning processes, widespread collaboration, and data review [1]. The Internet in today’s world is full of attacks and account takeovers where unauthorized adversaries try to gain access to user data and exploit their access.

- To analyze the impact of several attack scenarios, this dataset presents a set of 10 computers connected to Router1 on VLAN1 in a Docker Bridge network, that try and exploit each other. It includes browsing the web and downloading foreign packages including malicious ones. Also, services like FTP (File Transfer Protocol) and SSH (Secure Shell) were exploited using several attack mechanisms.
- The presented dataset shows the importance of updating and patching systems to protect themselves to a greater extent, by following attack tactics on older versions of packages as compared to the newer and updated ones.
- This data can further elaborate on how regular users connected to the Internet can get exploited by using several unsigned packages from the Internet and outdated services and protocols.

In order to comprehend and analyze the effect of any possible cyber-attack on a critical asset of an organization, real-time network activity data is needed by the researchers. The presented dataset contains the recorded network data of sandboxed machines which demonstrates several attack vectors in a controlled network environment. To isolate an actual attack on a large network, it is important to inspect and analyze the effect of the attack on a smaller scale and determine the risk mitigation steps. The presented dataset is intended to provide standard network data to the community to facilitate comparative research in this domain.

In Table 2, existing datasets containing network captures and network enumeration live sets were studied to follow the types of network topologies and patterns of the system used, includ-

| Paper | Domain | Contributions |
|-------|--------|---------------|
| Uses and Challenges for Network Datasets [2] | General Network | Analysis of research based concerns on several datasets including current and suggested practices |
| Anomaly, event, and fraud detection in large network datasets [3] | Large Networks | Comprehensive overview of several anomalies, events, and fraud detection. Analysis of data mining and machine algorithm algorithms |
| Local Learning for Mining Outlier Subgraphs from Network Datasets [4] UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) [5] | Mining Networks | Graphical representation of determining outliers on several synthetic and real datasets |
| Computer network database of attack and defense [6] APT datasets and attack modeling for automated detection methods: A review [7] A Survey of Intrusion Detection Systems Leveraging Host Data [8] A detailed analysis of the KDD CUP 99 data set [9] | Network Databases Automated Detection Review Host-based Intrusion Network Intrusion Detection | Computer network Databases attacks and mitigation steps analysis and experimentation Description of different stages of attacks on cyber-physical systems and large networks based on an attack model Access to different types of host-based data including intrusions, and research activities Statistical analysis of the KDDCUP’99 Dataset. Proposed NSL-KDD dataset that avoids performance and poor evaluation concerns using the KDDCUP’99 dataset |
ing the type of traffic transfer and the significance of different types of attack mechanisms when or if imposed on such networks.

The presented dataset provides an overall distribution of a separate network divided into 2 different subnets and is specifically used to determine the attack and defense effects of several network and host-based attacks in an isolated versus open environment.

1. Data Description

The dataset presented in this article includes raw data which was prepared using live machines built into a virtual sandboxed environment, which consists of 10 machines that are connected to a Router on VLAN1, and an Apache Server hosted on a different subset on VLAN2 which is connected to the VLAN1 to demonstrate isolation and cross VLAN communication. The services on this web server were also exploited by the previously stated 10 computers.

The Exiting Files in this dataset include:

- L1 Cap 10PC 1S.pcapng: This is the Raw pcap file captured using Wireshark for a specific time period on 11 machines (10 Machines & 1 Apache Server). Total Packets: 3,962,784.
- L1 Cap 10PC 1S dissec.xlsx: This file is the segregated first 1,048,576 rows of the Raw pcap file for small compute analysis. Note: 1,048,576 rows is the maximum row limit for Microsoft Excel (Version Home and Business 2021) as of 5th September 2022.
- L1 Cap 10PC 1S dissec.csv: This file is the segregated first 1,048,576 rows of the Raw pcap file for small compute analysis.
- L1 Cap 10PC 1S dissec complete.csv: This is the complete Raw exported pcap file in .csv format. Total Packets: 3,962,784.
- L1 Cap 202209051617.csv: This is the labeled dataset which 2 columns added to the Raw Dataset – Source Known and Destination Known which flags known IP addresses in the Source and Destination fields to belong to the 10+1 machine range. Here, ‘1’ represents True, and ‘0’ represents False.
- L1 Cap 202209051620.sql: This is the labeled dataset compiled in .sql format so that this can be easily imported into an SQL Database and analyzed based on the user requirement. This makes the analysis of large datasets easier and more convenient.

2. Experimental Design, Materials, and Methods

Fig. 1 provides a network architecture diagram of the implemented dataset along with the IP addresses of the nodes included in the Virtual LANs and the Host Network. This diagram also provides the Host Server on IP address ‘172.16.49.135’, on which the Wireshark tool was executed to capture the traffic flowing through the VLANS 1 and 2. The packets are directed to flow from each VLAN through the mentioned server to reach the Host Network router located on IP address ‘172.16.49.1’.

The files mentioned in the above points can be used for analysis either in the form of segregated data or in the form of a complete set of real-world traffic defining different attack and defense scenarios. The presented subfolder of files can be opened by the following means:

- File 1 (L1 Cap 10PC 1S.pcapng):
  - Install Wireshark from https://www.wireshark.org/#download
  - Double Click on the file to open it using Wireshark for packet-level deep inspection and analysis
Fig. 1. Position of Wireshark in the network
• File 2 and 3 (L1 Cap 10PC 1S dissecc.xlsx, L1 Cap 10PC 1S dissecc.csv):
  – Open any Text Editor to view these files – Notepad (Windows), TextEdit (Linux),
    TextMate (Mac)

  – These files are limited to 1,048,576 rows which is the standard Excel row limit so these
    can also be opened and viewed using Excel.

  * Right Click on the File and Open With: Microsoft Excel

• File 4 and 5 (L1 Cap 10PC 1S dissecc complete.csv, L1 Cap 202209051617.csv):
  – Open any Text Editor to view these files – Notepad (Windows), TextEdit (Linux),
    TextMate (Mac)
• File 6 (L1 Cap 202209051620.sql):
  - To import the MySQL file install MySQL from https://www.mysql.com/downloads/
  - Open a terminal prompt and type `mysql -u root -p`. If prompted for a password, enter the password used during the installation of MySQL or keep blank if None was set. Then execute the below commands in MySQL prompt
    * mysql> create database pcdataset;
    * mysql> use pcdataset;
    * mysql> CREATE TABLE L1 Cap('No.' int, 'Time' int, Source varchar(50), Destination varchar(50), Protocol varchar(50), 'Length' int, 'Time to Live' int, Info varchar(1024), Source Known int, Destination Known int);
    * mysql> exit
  - Now, Open a terminal prompt and type `mysql -u root -p pcdataset < file-path/L1 Cap 202209051620.sql`. The dataset will be imported in the SQL Database "pcdataset".
  - To view the dataset, use:
    * prompt> mysql -u root -p
    * mysql> use pcdataset;
    * mysql> show tables;
    * mysql> select * from L1 Cap;
  - Several other queries can now to executed to perform analysis on the data
  - Alternatively, this data can be viewed using GUI Database Managers like DBeaver (https://dbeaver.io/download)
Open the DBeaver app and connect to local host to view the tables and databases

* Click on Tables and Double on the Table Name to view the table

| Station | Protocol | Length | Time to Live | Info | Source Known | Destination Known |
|---------|----------|--------|--------------|------|--------------|-------------------|
| 1       | DNS      | 76     | 64           | Standard query 0x5002 A ports:ubuntu.com | 1               | 0                |
| 2       | DNS      | 108    | 127          | Standard query response 0x5002 A ports:ubuntu | 0               | 1                |
| 3       | TCP      | 74     | 64           | SYN Seq=0 Win=64240 Len=0 MS | 1               | 0                |
| 4       | TCP      | 58     | 127          | 80 > 51742 (SYN, ACK) Seq=0 Ack=1 Win=64240 | 0               | 1                |
| 5       | TCP      | 54     | 64           | GET /ubuntu-ports/Pool/main/libc/libcap2/libcap2 | 1               | 0                |
| 6       | TCP      | 207    | 127          | 80 > 51742 (ACK) Seq=1 Ack=64 Win=64240 L | 0               | 1                |
| 7       | HTTP     | 14334  | 127          | 80 > 51742 (PSH, ACK) Seq=1 Ack=154 Win=64 | 0               | 1                |
| 8       | TCP      | 94     | 64           | 51742 > 80 (ACK) Seq=1 Ack=154 Ack=12821 Win=955 | 1               | 0                |
| 9       | TCP      | 2910   | 127          | 80 > 51742 (PSH, ACK) Seq=14281 Ack=164 Wi | 0               | 1                |

* Also, SQL queries can be executed using the build-in SQL Query Editor

The Data Input Output per second was measured and plotted to result into a potential view of traffic capture statistics for a stipulated time of 23 h, 31 min, and 54 s.

Fig. 2 represents the Packet to Time ratio of the packets that are presented in the dataset which also provides a clear representation of the Distributed Denial of Service Attacks (DDoS) and Service Enumeration path that were performed on a local packet capturing tool in order to prevent the virtual environment from crashing due to heavy packet flow.
Fig. 2. DDoS attack and service enumeration chart.

Fig. 3 provides a baseline timeline of the various attack and activities performed to generate the data presented in the proposed dataset. The timeline starts from $t_0$ and ends at $t_{10}$, which is labeled in Fig. 2 to indicate the referred time stamp of script execution. This figure represents the data collected through Wire- shark for the activities or attacks performed in correlation with time. The detailed summary of the activities or attacks performed along with their outcomes and the tools used to perform them with correlation with time is attached in Table 3.
Table 3
Summary of activities / attacks performed.

| Time   | Activity                                      | Outcome                                                                 | Tools                                      |
|--------|-----------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------|
| t0 - t1| Scanning                                      | Network Scanning within VLAN1                                           | nmap, custom ping utility script           |
| t1 - t2| Services Setup                                | Setup of ftp, ssh, apache                                               | vsftpd, openssh, apache                    |
| t2 - t3| Services Utility Testing                      | Internal file sharing and communication on VLAN1 and VLAN2              | ftp, ssh, apache                           |
| t3 - t4| Distributed Denial of Service Attack          | Includes system unavailability - On VLAN2 web server from VLAN1         | hping3 flooding with random source         |
| t4 - t5| Service Restoration                           | Restored after DDoS and Services Utility Testing Includes higher wait times and response from VLAN2 web server | -                                          |
| t5 - t6| Distributed Denial of Service Attack          | Testing within VLAN2 on Services ssh and ftp communication - Includes services unavailability | hping3 flooding with random source         |
| t6 - t7| Brute Force                                   | On ftp and ssh services to login to gain access within VLAN2            | hydra using pre-determined wordlists of usernames and passwords |
| t7 - t8| Service Enumeration and Exploitation (Brute Force) | Includes running scripts for Privilege Escalation                     | Hydra using customized wordlists of usernames and passwords created using cewl, Customized privilege escalation scripts |
| t8 - t9| Web Based Enumeration and Exploitation        | On test website hosted on VLAN1 web server from VLAN2                  | nikto, sqlmap (SQL Injection), XSS (Manual Testing), Command Injection and Directory Traversal (through URL manual testing) |
| t9 - t10| ARP Spoofing                                  | Between VLAN1 and VLAN2                                                 | dnsniff tools to listen to network traffic (arpspoof) |
The Domain Name System is a naming convention to identify machines that are reachable through the Internet or over a network. The records link IP to Domain names and resolve requests for names to IP Addresses of the reachable machines and vice versa. The Domain Name System Resolution graph for the proposed dataset describes the packet count for each resolution factor along with their burst rate.

| Topic / Item          | Count | Average | Min Val | Max Val | Rate (ms) | Percent | Burst Rate | Burst Start |
|-----------------------|-------|---------|---------|---------|-----------|---------|------------|-------------|
| Total Packets         | 149   |         |         |         | 0.0000    | 100%    | 0.0700     | 76426.403   |
| rcode                 | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| No such name          | 10    |         |         |         | 0.0000    | 6.71%   | 0.0300     | 76426.417   |
| No error              | 139   |         |         |         | 0.0000    | 93.29%  | 0.0600     | 2177.567    |
| opcodes               | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| Standard query        | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| Query/Response        | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| Response              | 73    |         |         |         | 0.0000    | 48.99%  | 0.0300     | 2177.575    |
| Query                 | 76    |         |         |         | 0.0000    | 51.01%  | 0.0400     | 76426.403   |
| Query Type            | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| SRV (Server Selection)| 12    |         |         |         | 0.0000    | 8.05%   | 0.0200     | 817241      |
| PTR (domain name PoinTer) | 14   |         |         |         | 0.0000    | 9.40%   | 0.0200     | 76582.821   |
| AAAA (IPv6 Address)   | 4     |         |         |         | 0.0000    | 3.66%   | 0.0400     | 76426.403   |
| A (Host Address)      | 119   |         |         |         | 0.0000    | 79.87%  | 0.0600     | 2177.567    |
| Class                 | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| IN                    | 149   |         |         |         | 0.0000    | 100.00% | 0.0700     | 76426.403   |
| Service Stats         | 0     |         |         |         | 0.0000    | 100%    | -          | -           |
| request-response time (msec) | 73 | 427.70 | 0.701064 | 15018.991211 | 0.0000 | 0.0300 | 2177.575 |
| no. of unsolicited responses | 0 |         |         |         | 0.0000 | -          | -           |
| no. of retransmissions | 0 |         |         |         | 0.0000 | -          | -           |
| Response Stats        | 0     |         |         |         | 0.0000    | 100%    | -          | -           |
| no. of questions      | 146   | 1.00    | 1       | 1       | 0.0000    | 0.0600  | 2177.575   |
| no. of authorities    | 146   | 0.11    | 0       | 1       | 0.0000    | 0.0600  | 2177.575   |
| no. of answers        | 146   | 1.56    | 0       | 2       | 0.0000    | 0.0600  | 2177.575   |
| no. of additional     | 146   | 0.00    | 0       | 0       | 0.0000    | 0.0600  | 2177.575   |
| Query Stats           | 0     |         |         |         | 0.0000    | 100%    | -          | -           |
| Qname Len             | 76    | 17.51   | 8       | 27      | 0.0000    | 0.0400  | 76426.403  |
| Label Stats           | 0     |         |         |         | 0.0000    | -       | -          | -           |
| 4th Level or more     | 13    | 0.0000  | 0.0100  | 817241  |
| 3rd Level             | 57    | 0.0000  | 0.0300  | 2177.567 |
| 2nd Level             | 4     | 0.0000  | 0.0200  | 76426.426 |
| 1st Level             | 2     | 0.0000  | 0.0200  | 76426.403 |
| Payload size          | 149   | 51.17   | 26      | 106     | 0.0000    | 100%    | 0.0700     | 76426.403   |

The attack types include Distributed Denial of Service, SQL Injection, Account Takeover, Service Exploitation (SSH, FTP), DNS and ARP Spoofing, Scanning and Firewall Searching and Indexing (using Nmap), Hammering the services to brute-force passwords and usernames, Malware attack, Spoofing, and Man-in-the-Middle Attack. The attack scenarios also show various scanning mechanisms and the impact of Insider Threats on the entire network (Table 4).
Table 4

Machine Information

| Machine | Description |
|---------|-------------|
| 172.18.0.1 | Router1 |
| 172.18.0.2 | dev |
| 172.18.0.3 | devone |
| 172.18.0.4 | devtwo |
| 172.18.0.5 | devthree |
| 172.18.0.6 | devfour |
| 172.18.0.7 | devfive |
| 172.18.0.8 | devsix |
| 172.18.0.9 | devseven |
| 172.18.0.10 | deveight |
| 172.18.0.11 | devnine |
| 172.17.0.1 | Router2 |
| 172.17.0.2 | server |

The dataset presented was for the following time span:

| Time | Value |
|------|-------|
| First packet: | 2022-08-30 13:54:31 |
| Last packet: | 2022-08-31 13:26:26 |
| Elapsed: | 23:31:54 |

And the packet capture statistics can be viewed as:

| Measurement | Captured |
|-------------|----------|
| Packets     | 3962784  |
| Time span, s| 84714.903 |
| Average pps | 46.8     |
| Average packet size, B | 217 |
| Bytes       | 858082296 |
| Average bytes/s | 10 k |
| Average bits/s | 81 k |

The proposed dataset provides a set of attack and defense approaches to a network topology consisting of two VLANs connected to each other. The VLANs are:

- 172.17.0.1 /16
- 172.18.0.1 /16

172.17.0.1/16 consists of an Apache Server and is connected to a Router. 172.18.0.1/16 consists of 10 machines interconnected to each other and are connected to a different Router. The assigned machines' IPs and names are as follows:

Table 3 displays the specified machines that were categorized into both Attack and Victim machines having all probabilities of attack scenarios along with defense mechanisms implemented into each of them. The environment used for maintaining these machines was isolated into 2 different bridged networks which were managed by a traffic capture device and were connected independently to 2 different Routers to route traffic specifically to those subnets. The subnet routers are further interconnected to form a larger set of machines that are talking to each other and sharing resources and hosting web applications.
The packet captured by the capturing device was used not only to capture the traffic between the machines connected but also the traffic from the external Internet which includes downloading repositories, updating the machines, pulling old versions of software, and various external exploitations caused by connecting from unknown sources.

| No. | Source         | Destination         | Protocol | Time to Live | Info               | Source_Known | Destination_Known |
|-----|----------------|---------------------|----------|--------------|--------------------|--------------|--------------------|
| 0   | 172.18.0.2     | 172.18.0.2          | DNS      | 70           | 64 Standard query  | 1            | 0                  |
| 2   | 172.18.0.2     | 172.18.0.2          | DNS      | 108          | 127 Standard query | 0            | 1                  |
| 3   | 172.18.0.2     | 185.125.190.36      | TCP      | 74           | 64 51742 > 80 [BYN] | 1            | 0                  |
| 4   | 185.125.190.36 | 172.18.0.2          | TCP      | 58           | 127 80 > 51742 [BYN] | 0            | 1                  |
| 5   | 172.18.0.2     | 185.125.190.36      | TCP      | 64           | 64 51742 > 80 [ACK] | 1            | 0                  |
| 6   | 172.18.0.2     | 185.125.190.36      | HTTP     | 217          | 64 GET [javantu-permai] | 1            | 0                  |
| 7   | 185.125.190.36 | 172.18.0.2          | TCP      | 54           | 127 80 > 51742 [ACK] | 0            | 1                  |
| 8   | 185.125.190.36 | 172.18.0.2          | TCP      | 44,534       | 127 80 > 51742 [PSH] | 0            | 1                  |
| 9   | 172.18.0.2     | 185.125.190.36      | TCP      | 54           | 64 51742 > 80 [ACK] | 1            | 0                  |
| 10  | 185.125.190.36 | 172.18.0.2          | TCP      | 2,910        | 127 80 > 51742 [PSH] | 0            | 1                  |
| 11  | 172.18.0.2     | 185.125.190.36      | TCP      | 54           | 64 51742 > 80 [ACK] | 1            | 0                  |
| 12  | 185.125.190.36 | 172.18.0.2          | TCP      | 2,910        | 127 80 > 51742 [PSH] | 0            | 1                  |
| 13  | 172.18.0.2     | 185.125.190.36      | TCP      | 54           | 64 51742 > 80 [ACK] | 1            | 0                  |
| 14  | 185.125.190.36 | 172.18.0.2          | TCP      | 5,766        | 127 80 > 51742 [PSH] | 0            | 1                  |
| 15  | 172.18.0.2     | 185.125.190.36      | TCP      | 54           | 64 51742 > 80 [ACK] | 0            | 1                  |
| 16  | 185.125.190.36 | 172.18.0.2          | HTTP     | 143          | 127 HTTP/1.1 200 OK | 0            | 1                  |
| 17  | 172.18.0.2     | 185.125.190.36      | TCP      | 54           | 64 51742 > 80 [ACK] | 1            | 0                  |
| 18  | 172.18.0.2     | 185.125.190.36      | HTTP     | 355          | 64 GET [javantu-permai] | 1            | 0                  |

The graphical representation of the Database provides a brief overview of the dataset with the following properties of the captured network data:

- No.: Packet Serial Number
- Time: Time for packet capture starting from 0
- Source: Source IP Address for Packet Transfer
- Destination: Destination IP Address for Packet Transfer
- Protocol: Rules for transmission of information across a communication channel

| Topic / Item | Count | Average | Min Val | Max Val | Rate (ms) | Percent | Burst Rate | Burst Start |
|--------------|-------|---------|---------|---------|-----------|---------|------------|-------------|
| IP Protocol Types | | | | | | | | |
| UDP          | 159   | 0.0468  | 100%    | 0.0000  | 0.0000    | 0.0000  | 0.2000     | 228.555     |
| TCP          | 3961640 | 0.0468  | 99.99%  | 0.0000  | 0.0000    | 0.0000  | 0.2000     | 228.555     |
| NONE         | 196   | 0.0000  | 0.00%   | 0.0000  | 0.0000    | 0.0000  | 0.0000     | 0.0000      |

- Length: Length of each packet
- Time of Time: Count of hops for a packet to exist in a network before it is removed from the routing device
- Info: Packet Information
- Source Known: Flag to check if the Source IP Address falls under the category of known machine IPs
- Destination Known: Flag to check if the Destination IP Address falls under the category of known machine IPs
To determine the Source Known and Destination Known flag, the set of Known IPs are provided in a separate table to form a link between the existing dataset and the IP sets based on joining.

| # | IP Address       |
|---|------------------|
| 1 | 172.18.0.1       |
| 2 | 172.18.0.2       |
| 3 | 172.18.0.3       |
| 4 | 172.18.0.4       |
| 5 | 172.18.0.5       |
| 6 | 172.18.0.6       |
| 7 | 172.18.0.7       |
| 8 | 172.18.0.8       |
| 9 | 172.18.0.9       |
| 10| 172.18.0.10      |
| 11| 172.18.0.11      |
| 12| 172.18.0.12      |
| 13| 172.17.0.1       |
| 14| 172.17.0.2       |

2.1. Monitored Potential Outcome of Several Activities on a Host or Network

2.1.1. Unpatched Hosts

Ten machines were set up to use an old version of their Operating System with old repositories which made them potentially more vulnerable. The older version was not having certain firewall mechanisms and system hardening techniques that made an attacker machine discover potentially vulnerable services on the machine and also opened a way of pivoting to the other machines present in the same subnet network. This clearly indicated that an unpatched machine not only opens a gate to it, being breached by an attacker machine but also becomes a reason behind large-scale attack scenarios where multiple hosts connected to the same network reach a state of getting exploited.

The systems were then updated to experiment if the latest patch causes any effect on the hosts under the same attack conditions. The said machines were brought back to a stable safe snapshot and rebuilt to the latest build version available. On execution of the same attack scanning and pivoting script, the targeted victim machine detected the traffic flow and blocked the ping probes, which when bypassed by modifying the scripts, prevented it from execution by allowing only certain traffic to enter the network while blocking repeated packet hits thus blocking brute force approaches to a host getting exploited. The host was then made vulnerable manually in order to get access to the host in an exceptional case to test out the pivoting mechanism. Unlike the expected network getting exploited, the pivoting was immediately blocked by
the firewalls that hardened the systems by determining the type of packets hammered toward them.

This also provided a mechanism to determine and infer that whether one particular machine is vulnerable to a certain service and gets exploited, the other machines if patched, can prevent themselves from getting attacked.

2.1.2. Network Scanning

In order to record the impact of several network mapping tools on the network, the proposed dataset contains a set of machines performing scanning of ports, services, and versions of the services using Network Mappers and their impact or network logs were recorded and managed by an external Network Capture Device. The machines were configured to run different services and tools and were maintained on different bridged connections which were NATed to the host network for connecting to the External Internet. The Network Mapper tools were executed on all machines to scan each other and the traffic to monitor a united probability of the ports and services running on each machine.

The Mapper was configured under the following set of configurations:

- Execute Default Network Scanning Scripts.
- Determine the Services running on the hosts.
- Determine the Version of the services scanned.
- Identify all ports.
- Enumerate port 80 for the Apache web server hosted on different subversions.

All 10 host systems were then hardened to manage the scanning mechanism which added a secured mechanism to prevent pinging of the machines which might cause blocking of ping probes and show that the host is down for an attacker scanning the ports and services on the host. But this can be easily bypassed using scripts on the Network Mapper that considers each host to be up and starts scanning them. However, when the firewall was activated on the machines a Filtered flag was added to the services and the version enumeration failed.

2.1.3. Service Exploitation

Various host-based services were exploited, for example, FTP (File Transfer Protocol) and SSH (Secure Shell).

File Transfer Protocol. A specific service called "vsftpd" was installed and configured on the systems. The service was then initialized on 5 host machines and the other 4 were used as an authenticated client on a simultaneous connection based on a different subnet and 1 attacker machine to brute force login credentials, to the following defaults:

- Default home folder.
- Allow Anonymous login.
- Default owner permissions - Read, Write, Execute.

The system was further scanned using Network Mapper and also tried to be connected to using an FTP Client. Since Anonymous Login was allowed, the login attempt brute force passed on the first attempt and the root folder was accessible to the outside world as the owner of the home folder was the root or a super administrator of the system. Most users use their system as super administrators to avoid permission concerns that were inferred to be a potential breach by an adversary.

The FTP Configuration file was then reverted to the previous snapshot and was re-initialized updating the following setting:

- Default folder: Different Users specific to FTP file sharing.
- Anonymous Login: Disabled.
- Default owner permissions: Read Only.
This updated configuration prevented access to the root folder using anonymous login. The presented dataset shows a way to hammer the network and get access to weak FTP credentials. This leads to file system access to a machine, however, allows an adversary to have only read permissions and prevents script execution or data manipulation.

Secure Shell. As for the case with FTP, SSH is widely used for easy remote access to a host machine. In this experiment, two ways of login were analyzed on each of the 9 host machines excluding the dev machine that was used as the client machine:

- Password-based Login
- Key & Passphrase based login

For Case 1: The password was hammered using password guessing tools that were used to brute force commonly used passwords and a weak password was easily cracked and SSH access was granted to the machine.

For Case 2: Public-private key pairs were generated using ssh-keygen for safe password-free login. Whoso-ever holds the valid key can get access to the system. A passphrase refers to a secret text that can be used to safeguard the encryption key. This passphrase is initialized during the key generation phase and prevents access by brute force. This when enabled, presents no way to login through a password-based mechanism. However, protecting the keys form a major step in this case, as unauthorized access to the key can allow access to the system by an adversary.

2.1.4. Web-Based Exploitation

An Apache server was initialized on a different server-end subnet mask however connected to allow external traffic. The machine was successfully scanned to have Port 80 and 443 open Password-based service. The traffic on the web page was captured using the Packet Capture and Management DeWvere. Web-based attacks like Standard Query Language Injection and Cross Site Scripting attacks were implemented however since the website logs generate and manage user input, the packet flow shows no specific analysis based on user input. Even so when the website was hosted on a vulnerable host, the Directory Traversal Attack passed and the /var/www folder was accessible over the web which opened a way of easy access to system directory access and command injection through User Input on the hosted website.

The directory traversal was performed from a host browser using a port forwarding mechanism to transfer the Internal IP through the router so that it is accessible to the host machine under a controlled environment. Note:

- SQL Injection refers to a code injection mechanism that can be used to get access to the underlying Database used to store user data posted using the POST method from the website. The exploitation of a database can cause sensitive user data leakage and outcomes in a major breach.
- Cross Site Scripting refers to a mechanism of injecting malicious scripts into a website allowing user access and therefore crashing the website functionalities and providing access to the machine hosting the website, which could further provide unauthorized access to website data and sensitive user information.
- Command Injection refers to a process for executing malicious host-based commands to grant access to the website hosting machine.
- Directory Traversal refers to the file data reading based on the path of the file on the server. This can be achieved by manipulating the website URL to something similar to "http://172.17.0.2/..commands/www/.." where "../" refers to moving to the higher directory.
The web-based packet capture statistics as presented in the proposed dataset can be displayed as:

| Topic / Item                | Count | Average | Min Val | Max Val | Rate (ms) | Percent | Burst Rate | Burst Start |
|-----------------------------|-------|---------|---------|---------|-----------|---------|------------|-------------|
| Total HTTP Packets          | 1151  | 0.0000  | 100%    |         | 0.1500    | 83628.672|
| Other HTTP Packets          | 77    | 0.0000  | 6.69%   | 0.1200  | 2180.960  |         |            |             |
| HTTP Response Packets       | 526   | 0.0000  | 45.70%  | 0.1000  | 83628.672 |
| - ??: broken                | 0     | 0.0000  | 0.00%   | -       | -         |         |            |             |
| - 5xx: Server Error         | 0     | 0.0000  | 0.00%   | -       | -         |         |            |             |
| - 4xx: Client Error         | 0     | 0.0000  | 0.00%   | -       | -         |         |            |             |
| - 3xx: Redirection          | 0     | 0.0000  | 0.00%   | -       | -         |         |            |             |
| - 2xx: Success              | 526   | 0.0000  | 100.00% | 0.1000  | 83628.672 |
| - 200 OK                    | 526   | 0.0000  | 100.00% | 0.1000  | 83628.672 |
| - 1xx: Informational        | 0     | 0.0000  | 0.00%   | -       | -         |         |            |             |
| HTTP Request Packets        | 548   | 0.0000  | 47.61%  | 0.1100  | 2148.238  |
| GET                         | 548   | 0.0000  | 100.00% | 0.1100  | 2148.238  |

2.1.5. Distributed Denial of Service

The most important factor in a service hosting platform is to maintain the Availability of its resource such that the authorized personnel always have access to the requested data without any interruption. A Denial-of-Service attack requires an attacker to flood the web service with arbitrary traffic such that the server gets too busy to reply to actual requests. The said attack when performed by several machines at the same time can cause a Distributed Denial-of-Service or disruption to the Availability constraint to the website data.

A Distributed Denial of Service attack was performed on the hosted web server by all the 10 host machines under a different subnet which was disconnected from the server subnet to prevent concerns based on the same subnet flooding, using the attack machines under the following parameter until the web server crashed and was no longer accessible unless the service was restarted:

- Flood the server with arbitrary data requests
- Send continuous SYN traffic
- Make the flooding quit or reduce logging on the web server end
- Set data size to 120
- Request numeric output
- Target web server port

The attack scenario when launched caused all the 10 attacker machines (dev, devone, devtwo, devthree, devfour, devfive, devsix, devseven, deveight, and devnine) transfer data each of size 120 simultaneously for the server, to halt after some time as the firewall blocked the traffic from the suspected machines and the flooding stopped. This was bypassed when a random source flag was specified on the attacker machines (also known as botnets) and the web server could no longer determine the attacker machines, and the flooding continued after which the server stopped responding and the services were denied access by legitimate users.

2.1.6. ARP Spoofing

ARP Spoofing attack refers to the Man-in-the-Middle attack which allows an adversary to sit in between a communication channel between two machines and listen to the communication.

The ARP Spoofing attack was initialized on the standard Ethernet protocol of the dev and devone, dev and devtwo, dev and devthree, dev and devfour, dev and devfive, dev and devsix, dev and devseven, dev and deveight & dev and devnine machines and the target IP was set to the victim machine and access point, the machine is connected to which informs the target access point that the attacker is the targeted client. Next, the targeted user is informed that the target access point is the attacker machine, such that the machines keep exchanging data without suspecting the attacker is listening to the communication between them. This was then prevented.
by using encryption and decryption algorithms on the sender and receiver side which allowed the attacker to intercept data but the data was not decrypted and therefore the communication channel remained secure. All 10 machines on the same subnet were set to arpspoof on the dev routing protocol on Router 1 such that all machine traffic goes through dev and the dev machine can act as the Man in the Middle and perform packet and data sniffing.

2.2. Deployment

To generate the dataset in its complete form in the local system, navigate to the 'L1 Cap 202209051620.sql' and import it into MySQL on the local system. Please note that the table with the defined properties needs to be created before the import process. Inside the database, the L1 Cap file can be found which consists of the PCAP file database for the analysis of the packets that were captured to display both attack and defense mechanisms. The Info tab includes the Information about each packet which can be filtered from the generated database in MySQL and analyzed. For the packets to be analyzed graphically, the 'L1 Cap 10PC 1S.pcapng' can be directly imported into Wireshark. This can properly segregate the packets and can provide a graphical representation for analysis.

Ethics Statements

The authors confirm that the provided data set and presented work strictly meet the ethics requirements for publication in Data in Brief as mentioned in https://www.elsevier.com/authors/journal-authors/policies-and-ethics

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Data Availability

Cyber Attack Evaluation Dataset for Deep Packet Inspection and Analysis (Original data) (Mendeley Data).

CRediT Author Statement

Shishir Kumar Shandilya: Conceptualization, Methodology; Chirag Ganguli: Data curation, Writing – original draft, Visualization, Investigation; Ivan Izonin: Methodology, Formal analysis; Prof. Atulya Kumar Nagar: Conceptualization.

Acknowledgments

This research was partially funded by Department of Artificial Intelligence of Lviv Polytechnic National University, Ukraine.
References

[1] L. Dhanabal, S.P. Shantharajah, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, International Journal of Advanced Research in Computer and Communication Engineering 4 (6) (2015).

[2] J. Heidemann, C. Papdopoulos, Uses and challenges for network datasets, in: Proceedings of the Cybersecurity Applications Technology Conference for Homeland Security, 2009, pp. 73–82, doi:10.1109/CATCH.2009.29.

[3] L. Akoglu, C. Faloutsos, Anomaly, event, and fraud detection in large network datasets, in: Proceedings of the 6th ACM International Conference on Web Search and Data Mining, WSDM ’13, New York, NY, USA, Association for Computing Machinery, 2013, pp. 773–774, doi:10.1145/2433396.2433496.

[4] M. Gupta, A. Mallya, S. Roy, J. Cho, J. Han, Local Learning for Mining Outlier Subgraphs from Network Datasets, Proceedings of the 2014 SIAM International Conference on Data Mining (2014).

[5] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: Proceedings of the Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6, doi:10.1109/MilCIS.2015.7348942.

[6] Z. Wang, G. Lin, Computer network database of attack and defense, in: Proceedings of the International Conference on Consumer Electronics, Communications and Networks (CECNet), 2011, pp. 3986–3989, doi:10.1109/CECNET.2011.5768367.

[7] B. Stojanović, K. Hofer-Schmitz, U. Kleb, Apt datasets and attack modeling for automated detection methods: a review, Comput. Secur. 92 (2020) 101734 https://www.sciencedirect.com/science/article/pii/S0167404820300213, doi:10.1016/j.cose.2020.101734.

[8] R.A. Bridges, T.R. Glass-Vanderlan, M.D. Iannacone, M.S. Vincent, Q.G. Chen, A survey of intrusion detection systems leveraging host data, ACM Comput. Surv. 52 (6) (nov 2019), doi:10.1145/3344382.

[9] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD cup 99 data set, in: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6, doi:10.1109/CISDA.2009.5356528.