Data in Brief 40 (2022) 107795

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

# Hornet 40: Network dataset of geographically placed honeypots

Veronica Valeros*, Sebastian Garcia

*Artificial Intelligence Center, Department of Computer Science, FEL, Czech Technical University in Prague, Czech Republic*

ARTICLE INFO

*Article history:*
Received 5 August 2021
Revised 3 January 2022
Accepted 4 January 2022
Available online 5 January 2022

*Keywords:*
Honeypots dataset
Cyber security
Network attacks
Geolocation
Distributed attacks

ABSTRACT

Deception technologies, and honeypots in particular, have been used for decades to understand how cyber attacks and attackers work. A myriad of factors impact the effectiveness of a honeypot. However, very few is known about the impact of the geographical location of honeypots on the amount and type of attacks. Hornet 40 is the first dataset designed to help understand how the geolocation of honeypots may impact the inflow of network attacks. The data consists of network flows in binary and text format, with up to 118 features, including 480 bytes of the content of each flow. They were created using the Argus flow collector. The passive honeypots are IP addresses connected to the Internet and do not have any honeypot software running, so attacks are not interactive. The data was collected from identically configured honeypot servers in eight locations: Amsterdam, Bangalore, Frankfurt, London, New York, San Francisco, Singapore, and Toronto. The dataset contains over 4.7 million network flows collected during forty days throughout April, May, and June 2021.

* Corresponding author.
*E-mail address:* valerver@fel.cvut.cz (V. Valeros).
*Social media:* (V. Valeros), (S. Garcia)

https://doi.org/10.1016/j.dib.2022.107795

## Specifications Table

| | |
|---|---|
| Subject | *Computer Science* |
| Specific subject area | Honeypots network cybersecurity attacks |
| Type of data | Binary<br>Table<br>Figure |
| How the data were acquired | The network flows were collected by running Argus network flow collectors [1] on the network interfaces of geographically distributed cloud servers. |
| Data format | Raw<br>Filtered |
| Parameters for data collection | Argus network traffic flow collectors run directly on the network interface of each honeypot and store all the received traffic into files. Administration connections were filtered out by deleting the traffic with the IP addresses used for administration. |
| Description of data collection | The network flow data was collected for forty days, from April 23rd, 2021 to June 1st, 2021, using the Argus network traffic flow collector previously installed in each honeypot. Argus captures all the packets on the network interfaces of the honeypots to generate the flows. During this time, approximately 4.7 million flows were captured in total. The network connections used to administer the honeypots were filtered out from the dataset. |
| Data source location | The data was collected from eight cloud servers used exclusively as honeypots from the Digital Ocean [2] cloud provider. The locations were Amsterdam, Bangalore, Frankfurt, London, New York, San Francisco, Singapore, and Toronto. |
| Data accessibility | The Hornet 40 dataset was uploaded to the Mendeley Data repository.<br>Data identification number: 10.17632/tcfzkbpw46<br>Direct URL to data: http://dx.doi.org/10.17632/tcfzkbpw46 |

## Value of the Data

- The Hornet 40 dataset contains a unique collection of network flows, considered attacks from geographically distributed honeypots. It is useful for statistical and behavioral analysis of identically configured honeypots located in different geographical locations and regions.
- The data can benefit security practitioners, machine learning researchers, and statisticians working on network analysis, cyber security, or threat intelligence. It may also benefit network providers and antivirus companies. Data can be used to track attackers, to understand their origin, to understand changes in attacks patterns.
- The large amount of network flows can be used to evaluate and develop better attack detection mechanisms.
- The geography factor can be further used to evaluate better location placement of servers to reduce the number of attacks received by production servers.

## 1. Data Description

The Hornet 40 [3] dataset contains forty days of raw flow data, captured from eight cloud Linux passive honeypot servers. In this paper 'raw flow data' refers to network flows generated by a network flow collector from raw network data. Raw network data are the packets in the network.

Each honeypot was located in a different city in the regions of North America, Asia, and Europe. The cities were chosen from the available locations by the cloud provider Digital Ocean

[2]. No honeypot software was running on the Linux servers, but each IP address received connections from the Internet.

The raw flow data was captured using the Argus network flow collector from April 23rd, 2021 to June 1st, 2021. The dataset contains a total of 4,758,657 bidirectional flows, generated by a total of 266,678 unique source IP addresses. A bidirectional network flow is an aggregation of features of all the packets in a network connection. It aggregates packets from both directions: from source IP to destination IP, and from destination IP to source IP.

The Argus flows have 118 features including 480 bytes of the content of the connection in both directions. Argus generates one binary flow file per day per honeypot. These binary files contain all the flows for that day. From this binary flow file, two more ASCII files were exported in CSV format, one with NetFlow v5 features, and other with up to 118 Argus features.

The Hornet 40 dataset consists of three main groups of files: one for the binary bidirectional Argus files (biargus), one for the ASCII files with NetFlow v5 features, and one for the ASCII files extended with up to 118 Argus features. Each of these groups is separated per honeypot, each separation per honeypot containing 40 files, one per day. The folders and file structure of the Hornet 40 dataset is as follows:

- hornet40-biargus
  - Honeypot-Cloud-DigitalOcean-Geo-1: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-2: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-3: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-4: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-5: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-6: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-7: Has 40 binary flow files
  - Honeypot-Cloud-DigitalOcean-Geo-8: Has 40 binary flow files
- hornet40-netflow-v5
  - Honeypot-Cloud-DigitalOcean-Geo-1: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-2: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-3: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-4: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-5: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-6: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-7: Has 40 ASCII files with NetFlow v5 features
  - Honeypot-Cloud-DigitalOcean-Geo-8: Has 40 ASCII files with NetFlow v5 features
- hornet40-netflow-extended
  - Honeypot-Cloud-DigitalOcean-Geo-1: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-2: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-3: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-4: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-5: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-6: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-7: Has 40 ASCII files with 118 flow features
  - Honeypot-Cloud-DigitalOcean-Geo-8: Has 40 ASCII files with 118 flow features

All source IP addresses (from now on Src IPs) communicating with the honeypots are considered attacking IPs, due to one of the definitions of honeypots: since a honeypot is not an authorized production service, nobody should connect to it, and therefore all connections are considered attacks [4]. The main difficulty of this definition is how to consider the scanning activities of companies and organizations mapping the Internet for probable *benign* purposes. In this paper we still consider these *benign* scans as attacks, since (i) they are unsolicited, or (ii) the data collected may be sold, or (iii) used by attackers later, or (iv) use in marketing campaigns and advertising. Moreover, it is not technically feasible to filter out some of these scans based on organizations names or IP ranges, especially since many scans may be done from shared cloud providers.

**Table 1**
Hornet 40 dataset overview per honeypot server.

| Honeypot Name | City | Amount Unique Src IPs | Amount Flows | Amount Bytes | Amount Packets |
|---|---|---|---|---|---|
| Honeypot-Cloud-DigitalOcean-Geo-1 | Amsterdam | 36,441 | 347,195 | 554,894,141 | 2,052,308 |
| Honeypot-Cloud-DigitalOcean-Geo-2 | Bangalore | 59,103 | 444,007 | 86,173,556 | 1,244,019 |
| Honeypot-Cloud-DigitalOcean-Geo-3 | Frankfurt | 83,254 | 1,399,437 | 215,631,133 | 2,023,323 |
| Honeypot-Cloud-DigitalOcean-Geo-4 | London | 60,273 | 1,169,506 | 146,574,789 | 2,565,162 |
| Honeypot-Cloud-DigitalOcean-Geo-5 | New York | 48,967 | 298,851 | 57,456,984 | 927,028 |
| Honeypot-Cloud-DigitalOcean-Geo-6 | San Francisco | 41,478 | 308,829 | 48,286,398 | 791,287 |
| Honeypot-Cloud-DigitalOcean-Geo-7 | Singapore | 71,891 | 352,572 | 63,369,397 | 961,555 |
| Honeypot-Cloud-DigitalOcean-Geo-8 | Toronto | 52,824 | 438,260 | 82,452,397 | 1,230,072 |

**Table 2**
Hornet 40 dataset comparison of number of flows by protocol and honeypot.

| Honeypot Name | TCP | UDP | ICMP | ARP | SCTP | UDT |
|---|---|---|---|---|---|---|
| Honeypot-Cloud-DigitalOcean-Geo-1 | 310,273 | 18,671 | 16,960 | 1,284 | 3 | 1 |
| Honeypot-Cloud-DigitalOcean-Geo-2 | 395,123 | 25,187 | 22,408 | 1,284 | 2 | 2 |
| Honeypot-Cloud-DigitalOcean-Geo-3 | 324,677 | 1,057,897 | 15,558 | 1,287 | 2 | 15 |
| Honeypot-Cloud-DigitalOcean-Geo-4 | 1,130,134 | 17,585 | 20,499 | 1,284 | 2 | 2 |
| Honeypot-Cloud-DigitalOcean-Geo-5 | 270,509 | 16,065 | 10,985 | 1,289 | 2 | 1 |
| Honeypot-Cloud-DigitalOcean-Geo-6 | 273,000 | 15,418 | 19,130 | 1,279 | 2 | 0 |
| Honeypot-Cloud-DigitalOcean-Geo-7 | 303,422 | 25,772 | 22,083 | 1,287 | 5 | 2 |
| Honeypot-Cloud-DigitalOcean-Geo-8 | 406,043 | 16,043 | 14,885 | 1,284 | 2 | 1 |

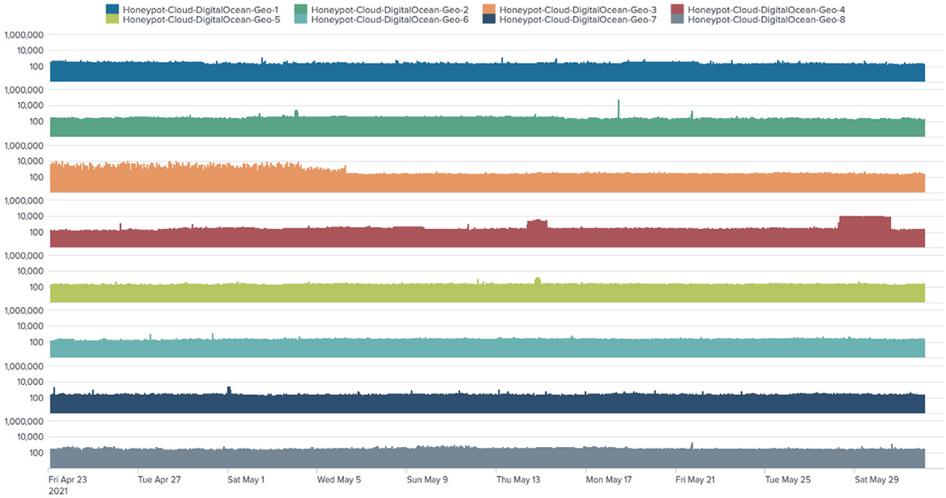


**Fig. 1.** Distribution of the number of flows per hour per scenario in logarithmic scale.

The Hornet 40 dataset contains a total of 4,758,657 network flows, containing 11,794,754 packets and 1,254,838,795 bytes, originating from 266,678 unique Src IPs. Table 1 provides a general overview of the Hornet 40 dataset per honeypot, including its location, the total unique source IPs, the total flows, the total bytes, and the total packets. Table 2 compares the number of connections per network protocol per honeypot.

To understand how the raw flow data is distributed in time, Fig. 1 shows the hourly traffic distribution per honeypot during the duration of the capture. The number of flows is displayed in logarithmic scale.

### *1.1. Argus binary network flows*

The raw flow data capture on the network interface of each honeypot was performed using the Argus network flow system [1] version 3.0.8.2. Argus stores raw flow data in a binary format that includes 118 features (e.g., total amount of bytes), and the first 480 bytes of the application content in each direction in the connection (to the honeypot, and from the honeypot). The number of features included in the binary Argus will change depending on the Argus version. These application content bytes are gathered by concatenating the content of the packets on each direction of each connection, until reaching 480 bytes. The full list of features is described in Table 3. The binary format is highly convenient due to its high compression and thus low final size. The binary files can be read and processed using Argus client tools to generate any desired output.

The Hornet 40 dataset contains one file for the Argus flows, called `hornet40-biargus.tar.gz` that consists of eight folders, one per honeypot, each containing 40 Argus binary flow files with all the 118 features, and the 480 bytes in both directions.

### *1.2. ASCII flows using NetFlow v5 features*

The Argus binary flow files were processed to generate flow files compatible with the NetFlow v5 standard [5]. This format is widely used, allowing to easily use and analyze data in CSV format. The NetFlow file is a format transformation from the original Argus file.

The Hornet 40 dataset contains one file, called `hornet40-netflow-v5.tar.gz`, that includes all the flows in the eight scenarios, and for each flow, it has the following features: StartTime, Dur, Proto, SrcAddr, Sport, Dir, DstAddr, Dport, State, sTos, dTos, TotPkts, TotBytes, SrcBytes, and SrcPkts. These features are described in Table 3.

### *1.3. ASCII flows using Argus features*

The Argus binary flow files were processed to generate files compatible with the NetFlow standard but with an extended number of features. These ASCII extended file are a format transformation from the original Argus file.

The Hornet 40 dataset contains one file for the extended flows, called `hornet40-netflow-extended.tar.gz`, that includes all the flows in the eight scenarios, and for each flow, it has all the 118 features described in Table 3.

### *1.4. Network features*

The complete list of the 118 features in the biargus and flow extended files of the Hornet 40 dataset is shown in Table 3.

## 2. Experimental design, materials and methods

The dataset was collected from cloud server instances. The cloud server provider chosen for this dataset was Digital Ocean [2]. All cloud servers, have the same technical specifications:

- Operating System: Ubuntu 20.04LTS
- Capacity: 1GB / 1 Intel CPU
- Storage: 25 GB NVMe SSDs
- Transfer capacity: 1000 GB transfer

**Table 3**
Full list of network features per flow as captured by Argus network collector tool. The Ra field is the name of the field according to the Argus Ra tool.

| Attribute | Ra Field | Attribute Description |
|---|---|---|
| StartTime | stime | record start time |
| Dur | dur | record total duration |
| Proto | proto | transaction protocol |
| SrcAddr | saddr | source IP address |
| Sport | sport | source port number |
| Dir | dir | direction of transaction |
| DstAddr | daddr | destination IP address |
| Dport | dport | destination port number |
| State | state | transaction state |
| sTos | stosv | source TOS (type of service) byte value |
| dTos | dtos | destination TOS (type of service) byte value |
| TotPkts | pkts | total transaction packet count |
| TotBytes | bytes | total transaction bytes |
| SrcBytes | sbytes | source to destination transaction bytes |
| SrcPkts | spkts | source to destination packet count |
| SrcId | srcid | argus source identifier |
| LastTime | ltime | record last time |
| Trans | trans | aggregation record count |
| Flgs | flgs | flow state flags seen in transaction |
| Seq | seq | argus sequence number |
| StdDev | stddev | standard deviation of aggregated duration times |
| SrcMac | smac | source MAC addr |
| DstMac | dmac | destination MAC addr |
| sDSb | sdsb | source diff serve byte (Differentiated Services) value |
| dDSb | ddsb | destination diff serve byte (Differentiated Services) value |
| sCo | sco | source IP address country code |
| dCo | dco | destination IP address country code |
| sTtl | sttl | source to destination TTL value |
| dTttl | dttl | destination to source TTL value |
| sHops | shops | estimate of number of IP hops from src to this point |
| dHops | dhops | estimate of number of IP hopes from dst to this point |
| sIpId | sipid | source IP identifier |
| dIpId | dipid | destination IP identifier |
| sMpls | smpls | source MPLS identifier |
| dMpls | dmpls | destination MPLS identifier |
| DstBytes | dbytes | destination to source transaction bytes |
| TotAppByte | appbytes | total application bytes |
| SAppBytes | sappbytes | source to destination application bytes |
| DAppBytes | dappbytes | destination to source application bytes |
| Load | load | bits per second |
| SrcLoad | sload | source bits per second |
| Dstload | dload | destination bits per second |
| Loss | loss | pkts retransmitted or dropped |
| SrcLoss | sloss | source pkts retransmitted or dropped |
| DstLoss | dloss | destination pkts retransmitted or dropped |
| pLoss | ploss | percent pkts retransmitted or dropped |
| Rate | rate | pkts per second |
| SrcRate | srate | source pkts per second |
| DstRate | drate | destination pkts per second |
| SIntPkt | sintpkt | source interpacket arrival time (mSec) |
| SIntPktAct | sintpktact | source active interpacket arrival time (mSec) |
| SIntPktIdl | sintpktidl | source idle interpacket arrival time (mSec) |
| DIntPkt | dintpkt | destination interpacket arrival time (mSec) |
| DIntPktAct | dintpktact | destination active interpacket arrival time (mSec) |
| DIntPktIdl | dintpktidl | destination idle interpacket arrival time (mSec) |
| SrcJitter | sjit | source jitter (mSec) |
| SrcJitAct | sjitact | source active jitter (mSec) |
| DstJitter | djit | destination jitter (mSec) |
| DstJitAct | djitact | destination active jitter (mSec) |

(*continued on next page*)

**Table 3** (*continued*)

| Attribute | Ra Field | Attribute Description |
|---|---|---|
| srcUdata | suser | source user data buffer |
| dstUdata | duser | destination user data buffer |
| SrcWin | swin | source TCP window advertisement |
| DstWin | dwin | destination TCP window advertisement |
| sVlan | svlan | source VLAN identifier |
| dVlan | dvlan | destination VLAN identifier |
| sVid | svid | source VLAN identifier |
| dVid | dvid | destination VLAN identifier |
| sVpri | svpri | source VLAN priority |
| dVpri | dvpri | destination VLAN priority |
| SRange | srng | start time for the filter timerange |
| SrcTCPBase | stcpb | source TCP base sequence number |
| DstTCPBase | dtcpb | destination TCP base sequence number |
| TcpRtt | tcprtt | TCP connection setup round-trip time ('synack' plus 'ackdat') |
| SynAck | synack | TCP connection setup time, time between SYN and SYN_ACK packets |
| AckDat | ackdat | TCP connection setup time, time between SYN_ACK and ACK packets |
| SrcStartTime | sstime | source start time |
| SrcLastTime | sltime | source last time |
| SrcDur | sdur | source duration |
| DstStartTime | dstime | destination start time |
| DstLastTime | dltime | destination last time |
| DstDur | ddur | destination duration |
| DstPkts | dpkts | destination to source packet count |
| pSrcLoss | sploss | percent source pkts retransmitted or dropped |
| pDstLoss | dploss | percent destination pkts retransmitted or dropped |
| sEnc | senc | source encoding |
| dEnc | denc | destination encoding |
| SIntPktMax | sintpktmax | maximum source interpacket arrival time |
| SIntPktMin | sintpktmin | minimum source interpacket arrival time |
| DIntPktMax | dintpktmax | maximum destination interpacket arrival time |
| DIntPktMin | dintpktmin | minimum destination interpacket arrival time |
| SIPActMax | sintpktactmax | source longest active interpacket arrival time |
| SIPActMin | sintpktactmin | source shortest active interpacket arrival time |
| DIPActMax | dintpktactmax | destination longest active interpacket arrival time |
| DIPActMin | dintpktactmin | destination shortest active interpacket arrival time |
| SIPIdlMax | sintpktidlmax | source longest inactive interpacket arrival time |
| SIPIdlMin | sintpktidlmin | source shortest inactive interpacket arrival time |
| DIPIdlMax | dintpktidlmax | destination longest inactive interpacket arrival time |
| DIPIdlMin | dintpktidlmin | destination shortest inactive interpacket arrival time |
| SrcJitIdl | sjitidl | source inactive jitter time |
| DstJitIdl | djitidl | destination inactive jitter time |
| dsPkts | dspkts | delta source packets |
| ddPkts | ddpkts | delta destination packets |
| dsBytes | dsbytes | delta source bytes |
| ddBytes | ddbytes | delta destination bytes |
| pdsPkt | pdspkts | percent delta source packets |
| pddPkt | pddpkts | percent delta destination packets |
| pdsByte | pdsbytes | percent delta source bytes |
| pddByte | pddbytes | percent delta destination bytes |
| (null) | tcpext | TCP extensions |
| JDelay | jdelay | join delay |
| LDelay | ldelay | leave delay |
| Bins | bins | |
| Bin | bin | |
| Inode | inode | ICMP intermediate node |
| sMaxPktSz | smaxsz | maximum packet size for traffic transmitted by the source |
| sMinPktSz | sminsz | minimum packet size for traffic transmitted by the source |
| dMaxPktSz | dmaxsz | maximum packet size for traffic transmitted by the destination |
| dMinPktSz | dminsz | minimum packet size for traffic transmitted by the destination |

The *honeypots* were created within the same hour of the same day. Once created, the servers were configured simultaneously using the open-source tools `parallel-ssh` and `parallel-scp`. Each honeypot has its own public IP address assigned, which has not changed during the data capture.

The steps to create and configure the honeypots were as follows:

- Create one *honeypot* per available region in Digital Ocean with the technical specifications listed above.
- Store the list of public IPs from the *honeypots* in a text file named `hosts`.
- Update the software repository of all honeypots simultaneously using parallel-ssh: `pssh -h hosts -l root -o output/` “apt update”, where `hosts` contains the list of IP addresses of the honeypots, `-l root` specifies the SSH user name, `-o output/` the folder where to store the results, and “`apt update`” is the command that will be executed in parallel in the honeypots.
- Install the Argus network collector tool: `pssh -h hosts -l root -o output/` “`apt install -yq argus-client argus-server`”. The Argus version 3.0.8.2 was used.
- Upload a new SSH configuration of all honeypots. The updated configuration moves the service to a non-standard port (See Subsection *SSH Configuration*) to avoid attacks on a real service: `pscp -h hosts -l root sshd_config /etc/ssh/sshd_config`
- Restart the SSH server to load the new SSH configuration: `pssh -h hosts -l root -o output/` “`/etc/init.d/ssh restart`”
- Update the `hosts` file to specify the new SSH port to use, e.g.: *0.0.0.0:902*
- Upload a common Argus configuration (See Subsection *Argus Configuration*) to each honeypot: `pscp -h hosts -l root argus.conf /etc/argus.conf`
- Create a folder to store the raw flow data: `pssh -h hosts -l root -o output/` “mkdir /root/dataset”
- Start the Argus network collector for all honeypots on the network interface `eth0`: `pssh -h hosts -l root -o output/` “`argus -F /etc/argus.conf -i eth0`”
- Start the Argus service *rasplit* to store the network data received by Argus: `pssh -h hosts -l root -o output/` “`rasplit -S 127.0.0.1:900 -M time 1h -w /root/dataset/%Y/%m/%d/do-sensor.%H.%M.%S.biargus`”, where `-S` indicates the Argus server and port where to retrieve data from, -M for splitting the collection by time every one hour, and -w indicates where to store the data and how to name the files.

The Argus and *rasplit* tools produce raw flow data files from each honeypot. The binary raw flow data can be read using the *ra* tool provided in the Argus-clients suite. The command to read the files is:

```
ra -F /etc/ra.conf -n -Z b -r 2021-04-23_honeypot-cloud-digitalocean-
geo-1.biargus -
```

where -F specifies the configuration file to use (see Subsection *Ra Configuration*), -n avoids resolving port numbers to service names, -Z modifies the status field to show TCP flag values, -r specifies which file to read from, and - sends the output to the standard output in the terminal.

### *2.1. SSH configuration for administration of the honeypots*

The honeypots are configured to be remotely administered using the SSH protocol. The port 902/TCP is used for the SSH server. The SSH server configuration file used, called `sshd_config`, is shown below:

```
AcceptEnv LANG LC_*
ChallengeResponseAuthentication no
Include /etc/ssh/sshd_config.d/*.conf
PasswordAuthentication no
```

```
PermitRootLogin yes
Port 902
PrintMotd no
Subsystem sftp/usr/lib/openssh/sftp-server
UsePAM yes
X11Forwarding yes
```

### 2.2. Argus configuration

The Argus servers in all honeypots used the same configuration for the collection of network data. The file used was argus.conf, and its content is shown below:

```
ARGUS_FLOW_TYPE="Bidirectional"
ARGUS_FLOW_KEY="CLASSIC_5_TUPLE"
ARGUS_ACCESS_PORT=900
ARGUS_INTERFACE=eth0
ARGUS_FLOW_STATUS_INTERVAL=3600
ARGUS_MAR_STATUS_INTERVAL=60
ARGUS_GENERATE_RESPONSE_TIME_DATA = yes
ARGUS_GENERATE_PACKET_SIZE=yes
ARGUS_GENERATE_JITTER_DATA=yes
ARGUS_GENERATE_MAC_DATA=yes
ARGUS_GENERATE_APPBYTE_METRIC=yes
ARGUS_GENERATE_TCP_PERF_METRIC=yes
ARGUS_GENERATE_BIDIRECTIONAL_TIMESTAMPS = yes
ARGUS_CAPTURE_DATA_LEN=480
ARGUS_BIND_IP="::1,127.0.0.1"
```

### 2.3. Ra configuration

The following Ra configuration can be used to read the binary flow files and export all the attributes of Argus:

```
RA_PRINT_LABELS=0
RA_FIELD_DELIMITER=','
RA_USEC_PRECISION=6
RA_PRINT_NAMES=0
RA_TIME_FORMAT="%Y/%m/%d %T.%f"
RA_FIELD_SPECIFIER= srcid seq stime ltime dur sstime sltime sdur
dstime dltime ddur srng drng trans flgs avgdur stddev mindur maxdur
saddr dir daddr proto sport dport sco dco stos dtos sdsb ddsb sttl
dttl shops dhops sipid dipid pkts spkts dpkts bytes sbytes dbytes
appbytes sappbytes dappbytes load sload dload rate srate drate loss
sloss dloss ploss sploss dploss senc denc smac dmac smpls dmpls svlan
dvlan svid dvid svpri dvpri sintpkt dintpkt sintpktact dintpktact
sintpktidl dintpktidl sintpktmax sintpktmin dintpktmax dintpktmin
sintpktactmax sintpktactmin dintpktactmax dintpktactmin sintpktidlmax
sintpktidlmin dintpktidlmax dintpktidlmin jit sjit djit jitact
sjitact djitact jitidl sjitidl djitidl state deldur delstime delltime
dspkts ddpkts dsbytes ddbytes pdspkts pddpkts pdsbytes pddbytes
suser:1500 duser:1500 tcpext swin dwin jdelay ldelay bins binnum
stcpb dtcpb tcprtt synack ackdat inode smaxsz sminsz dmaxsz dminsz
```

## Ethics Statements

The work did not involve any human subject or animal experiments. The Hornet 40 dataset files were analyzed by more than 50 Antivirus systems in VirusTotal and as of January 2021 it does not trigger any detection.

## 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.

## CRediT Author Statement

**Veronica Valeros:** Conceptualization, Methodology, Data curation, Writing – original draft; **Sebastian Garcia:** Validation, Writing – review & editing.

## Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Supplementary Material

Supplementary material associated with this article can be found online at Mendeley Data at doi:10.17632/tcfzkbpw46, including the data for the Tables 1–3, and Fig. 1.

## References

[1] C. Bullard, "OpenArgus". [Online]. Available: https://openargus.org/. Accessed 5 2021.
[2] Digital Ocean, "DigitalOcean – the developer cloud", 2011 Digital Ocean. [Online]. Available: https://www.digitalocean.com/. Accessed 5 2021.
[3] V. Valeros, "Hornet 40: network dataset of geographically placed honeypots," 2021. [Online]. Available: https://data.mendeley.com/datasets/tcfzkbpw46/3.
[4] N. Provos, A virtual honeypot framework, in: Proceedings of the USENIX Security Symposium, 2004.
[5] "Netflow:: Version 5," 2009. [Online]. Available: https://netflow.caligare.com/netflow_v5.htm. Accessed 5 2021.