OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Generation of Distributed Denial of Service Network Data with Python and Scapy

Stefan Görtz
Ostbayerische Technische Hochschule Regensburg
27.06.2023

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

Abstract

- We aimed to create DDoS attack data suitable for machine learning (labeled datasets)
- DDoS Attacks are among the most common network attacks
  They are difficult to detect and defend
- There are not very many suitable DDoS data sets, so we generate our own data
- We implemented a Python program with the library Scapy to perform DoS Attacks on an Internet-of-Things (IoT) Test setup with simulated distribution by multithreading
- Advantages of this approach: Create DDoS data individually tailored to our needs

DDoS attacks in 2022 and Q1 2023

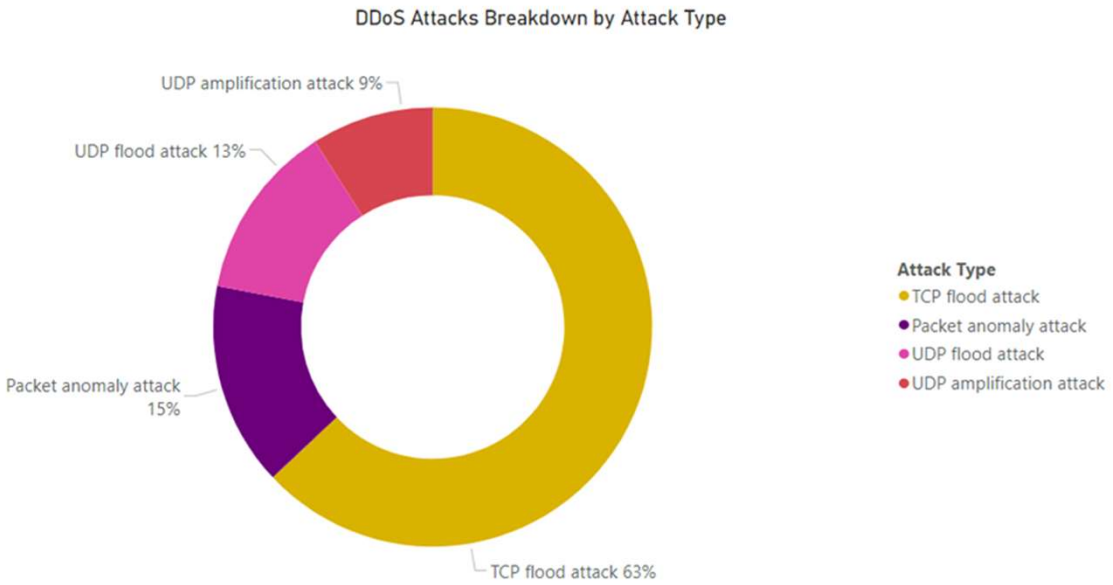
Most common DDoS Attacks in 2022:

- TCP: e.g. Synflood
- UDP Flooding

Q1 2023:

- 22% of the attacks were Synflood attacks [1]
- 21% were Udpflooding attacks [1]

[1] https://blog.cloudflare.com/ddos-threat-report-2023-q1



http://www.microsoft.com/en-us/security/blog/2023/02/21/2022-in-review-ddos-attack-trends-and-insights/

DoS and DDoS

- Denial of Service (DoS) attacks aim to make a service, reachable over the Internet inaccessible
- They aim to overload a network protocol (e.g. TCP) or exhaust the victims hardware resources (CPU cycles, bandwidth, memory)
- DDoS attacks use DoS techniques with the help of many hosts
- DDoS attacks are are financially or politically motivated
- DDoS attacks cause major economic damage

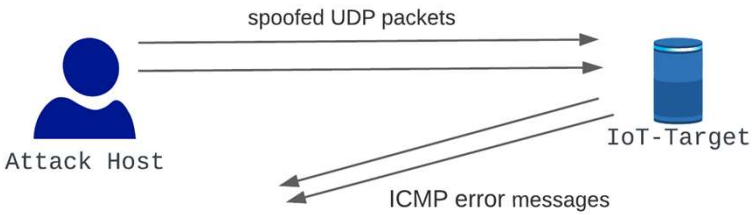
UDP Flooding

UDP

- UDP: minimal protocol for connectionless data transfer, with no guarantee of completeness and correctness
- Applications create an UDP header and transfer their data over the Internet Protocol (IP)

UDP-Flooding

Send large number of UDP packets to exhaust victims ressources



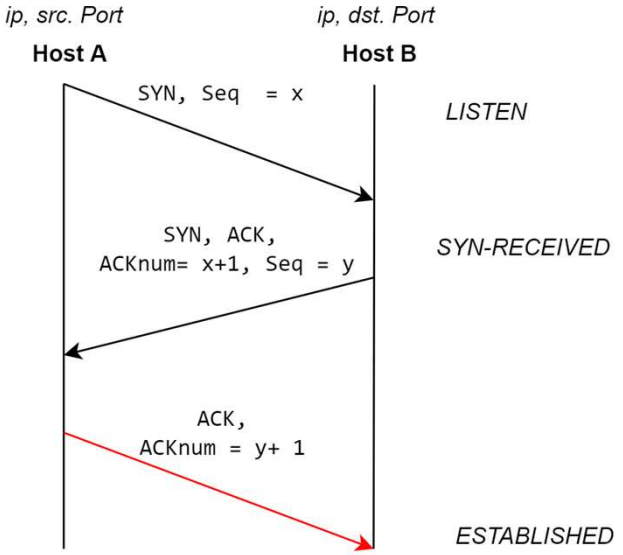
TCP: Synflooding

TCP:

- reliable, connection-oriented protocol for data transmission, error checking, flow control, retransmission
- 3-way-handshake: Establish connection between client and server

Synflooding:

- Exhaust victims backlog with SYN packets to disrupt its ability to establish legitimate connections



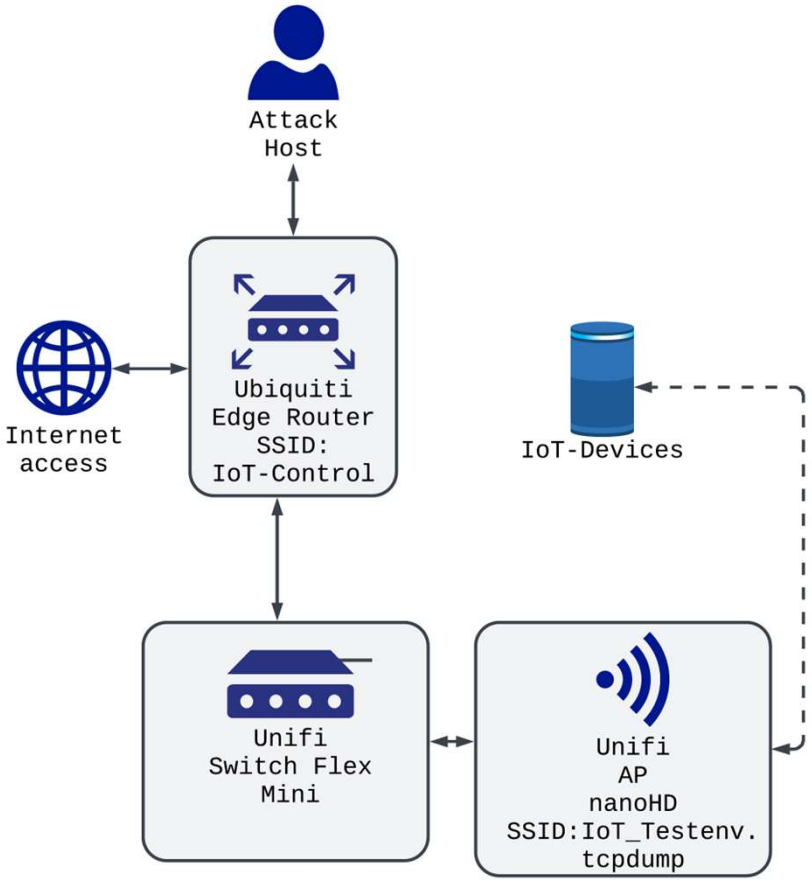
Testsetup: IoT targets

IoT Devices:

- Nedis Cam Grau
- Tapo Cam C 100
- Antela Speed Cam
- Nedis Cam klein
- Amazon Echo dot
- Amazon Echo 2
- Tapo Wifi Socket
- Brennenstuhl Smoke Detector

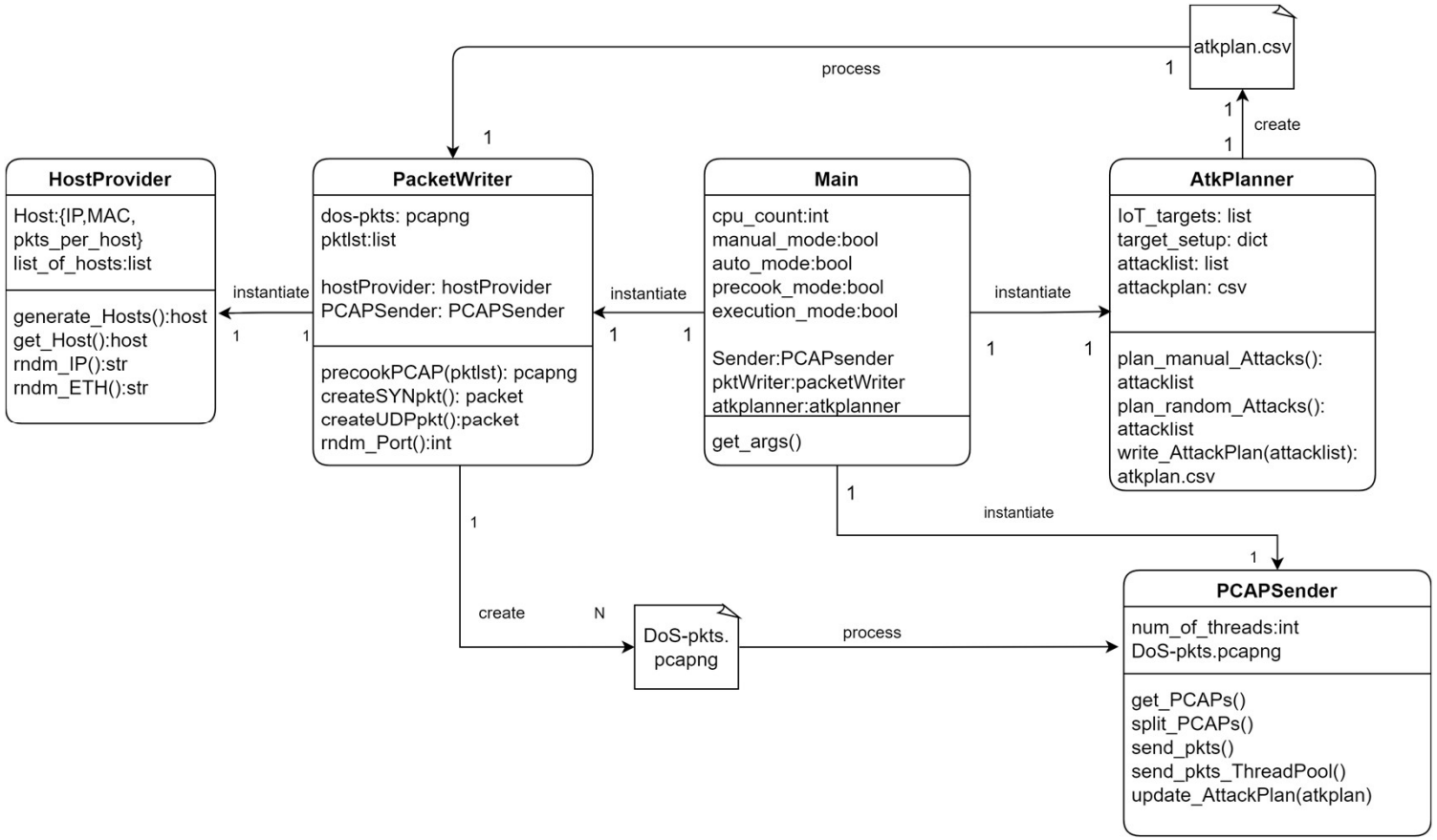
OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

Testsetup: Network Diagram



OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

Implemented Python program



OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

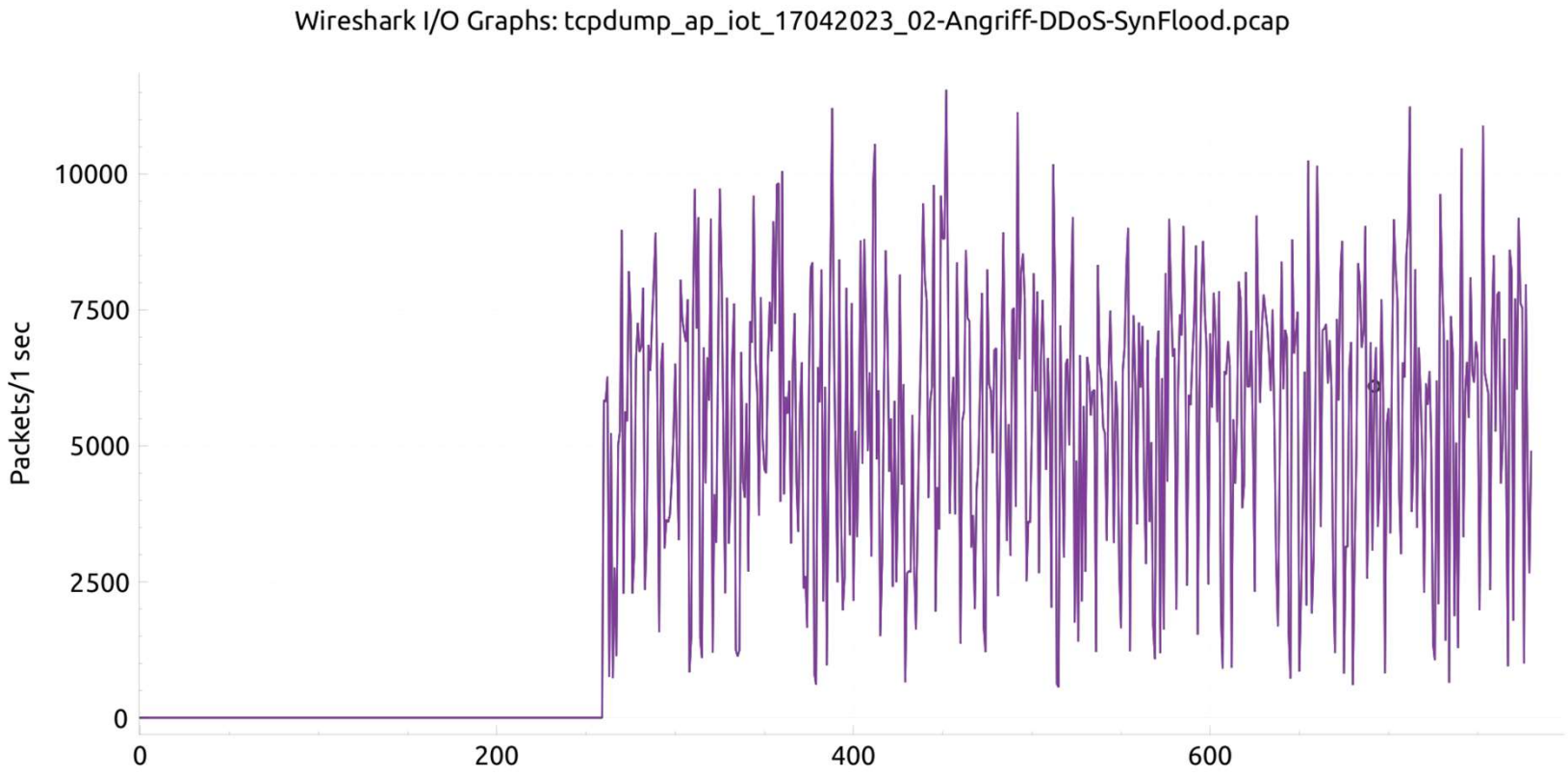
SYN Packet

```
▸ Frame 32904: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
▸ Ethernet II, Src: 32:41:82:86:98:9e (32:41:82:86:98:9e), Dst: AmazonTe_52:62:eb (00:71:47:52:62:eb)
▸ Internet Protocol Version 4, Src: 99.69.179.169, Dst: 192.168.1.4
▾ Transmission Control Protocol, Src Port: 23629, Dst Port: 0, Seq: 0, Len: 70
    Source Port: 23629
    Destination Port: 0
    [Stream index: 112]
    [Conversation completeness: Incomplete (9)]
    [TCP Segment Len: 70]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 0
    [Next Sequence Number: 71    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x002 (SYN)
    Window: 0
    [Calculated window size: 0]
    Checksum: 0x8ace [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▸ [Timestamps]
  ▸ [SEQ/ACK analysis]
    TCP payload (70 bytes)
    Retransmitted TCP segment data (70 bytes)
```

```
0000  00 71 47 52 62 eb 32 41  82 86 98 9e 08 00 45 00   ·qGRb·2A ·····E·
0010  00 6e 00 01 00 00 40 06  a1 ee 63 45 b3 a9 c0 a8   ·n····@· ··cE····
0020  01 04 5c 4d 00 00 00 00  00 00 00 00 00 00 50 02   ··\M···· ·····P·
0030  00 00 8a ce 00 00 53 59  4e 46 4c 4f 4f 44 58 58   ·····SY NFLOODXX
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0070  58 58 58 58 58 58 58 58  58 58 58 58               XXXXXXXX XXXX
```

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

SYN flooding packets / second



OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

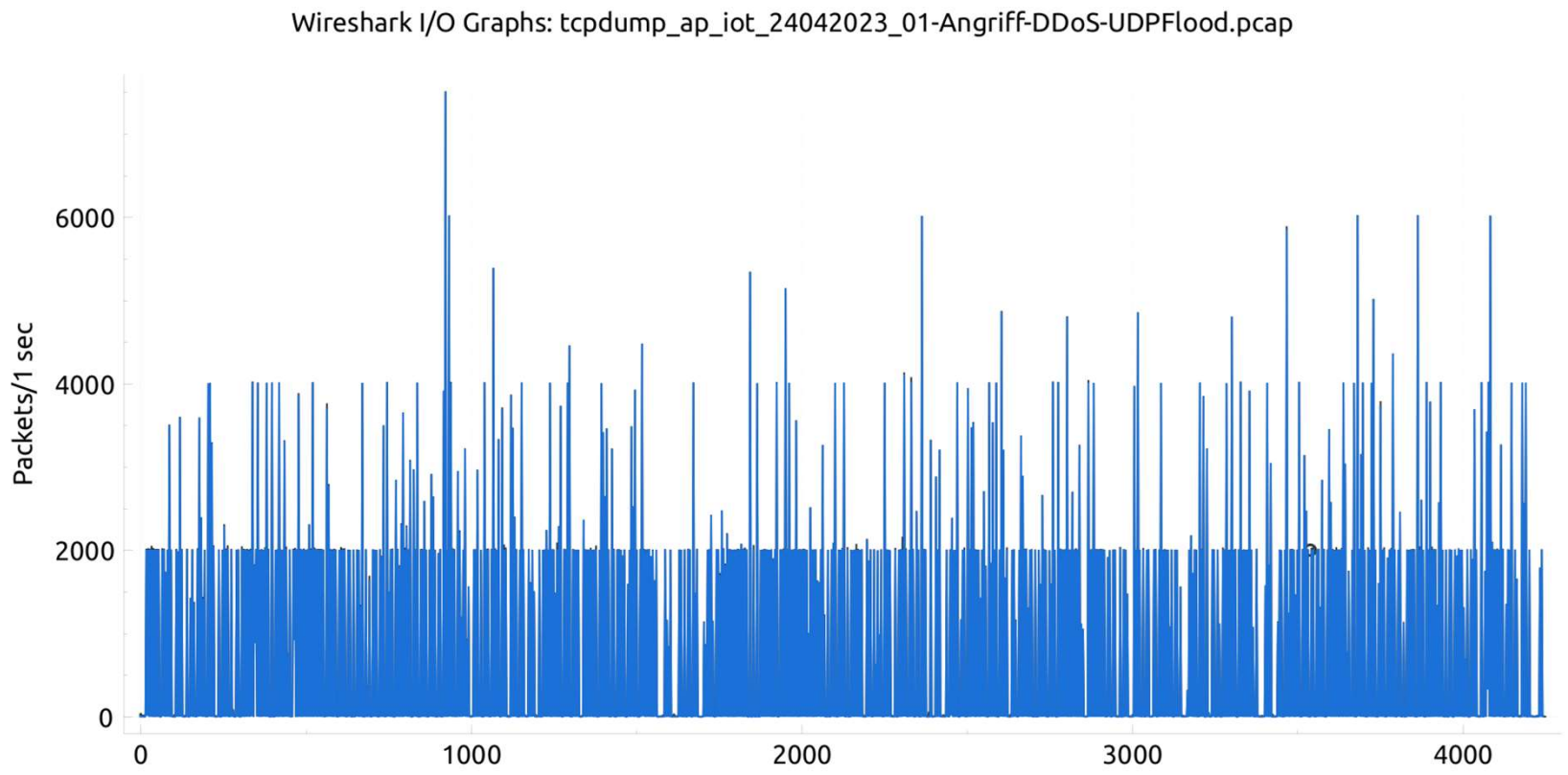
UDP packet

```
▸ Frame 963299: 1015 bytes on wire (8120 bits), 1015 bytes captured (8120 bits) on interface enp1s0, id 0
▸ Ethernet II, Src: 28:85:bf:2c:80:92 (28:85:bf:2c:80:92), Dst: Shenzhen_0b:ab:be (20:32:33:0b:ab:be)
▸ Internet Protocol Version 4, Src: 4.164.215.248, Dst: 192.168.1.14
▾ User Datagram Protocol, Src Port: 61686, Dst Port: 41113
    Source Port: 61686
    Destination Port: 41113
    Length: 981
    Checksum: 0xd8eb [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1141]
  ▸ [Timestamps]
    UDP payload (973 bytes)
▾ Data (973 bytes)
    Data: 554450464c4f4f44585858585858585858585858585858585858585858585858585858…
    [Length: 973]
```

```
0000  20 32 33 0b ab be 28 85  bf 2c 80 92 08 00 45 00   23···(· ·,····E·
0010  03 e9 00 01 00 00 40 11  d8 b0 04 a4 d7 f8 c0 a8   ······@· ········
0020  01 0e f0 f6 a0 99 03 d5  d8 eb 55 44 50 46 4c 4f   ········ ··UDPFLO
0030  4f 44 58 58 58 58 58 58  58 58 58 58 58 58 58 58   ODXXXXXX XXXXXXXX
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00b0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00c0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00d0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
```

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

UDP Flooding packets / second



Results: Effects on IoT devices

TABLE I
DDoS ATTACK EFFECTS ON IoT DEVICES

DDoS effects on IoT Devices		
IoT device	**Attack type**	**Effect**
Amazon Echo	synflood	success
Amazon Echo	udpflood	no success
Nedis Cam Gray	synflood	success
Nedis Cam Gray	udpflood	success
Tapo Cam C100	synflood	success
Tapo Cam C100	udpflood	success
Antela Speed Cam	synflood	success
Antela Speed Cam	udpflood	no success
Nedis cam white	synflood	success
Nedis cam white	udpflood	no success
SV3C camera	synflood	success
SV3C camera	udpflood	no success

- Synflooding was always successful
- Udpflooding not always: some devices ignore incoming udp traffic

Results

- Conduct UDP/SYN-Flooding attacks on IoT Devices
- Caputre the traffice with tcpdump (pcap)
- Data processing: Convert packets of pcaps to dicts, check for a intrusion flag in the payload, then expand dicts and label them: (intrusion = 1, attack_type = „udp“ or attack_type = „syn“)
- Insert the labeled datasets into a sql database for further use

Discussion

Advantages

- Create data suitable for machine learning according to users needs
- Program is expandable by other protocols (e.g. ICMP flooding)
- Our method ensures there are no malformed packets
- Cost efficient method

Limitations

- Simulated distribution is limited by the number of threads on attack host
- Data transfer rate is also lower than in a ressource strong DDoS attack

=> **Augmentation** of packet timestamps to create larger, simultaneous attacks

Outlook

- Validation of training data: Re-create available DDoS datasets
- Test our data with an available IDS
- Expand the program by further attack types
- Conduct a larger field test: Use multiple attack hosts (e.g. raspberry pis) in combination with multithreading to create larger attacks