# Cybersecurity (Security+) and P4 Programmable Switches

# Lab 8: Detecting and Mitigating the DNS Amplification Attack

Ali AlSabeh, Jorge Crichigno
University of South Carolina
http://ce.sc.edu/cyberinfra

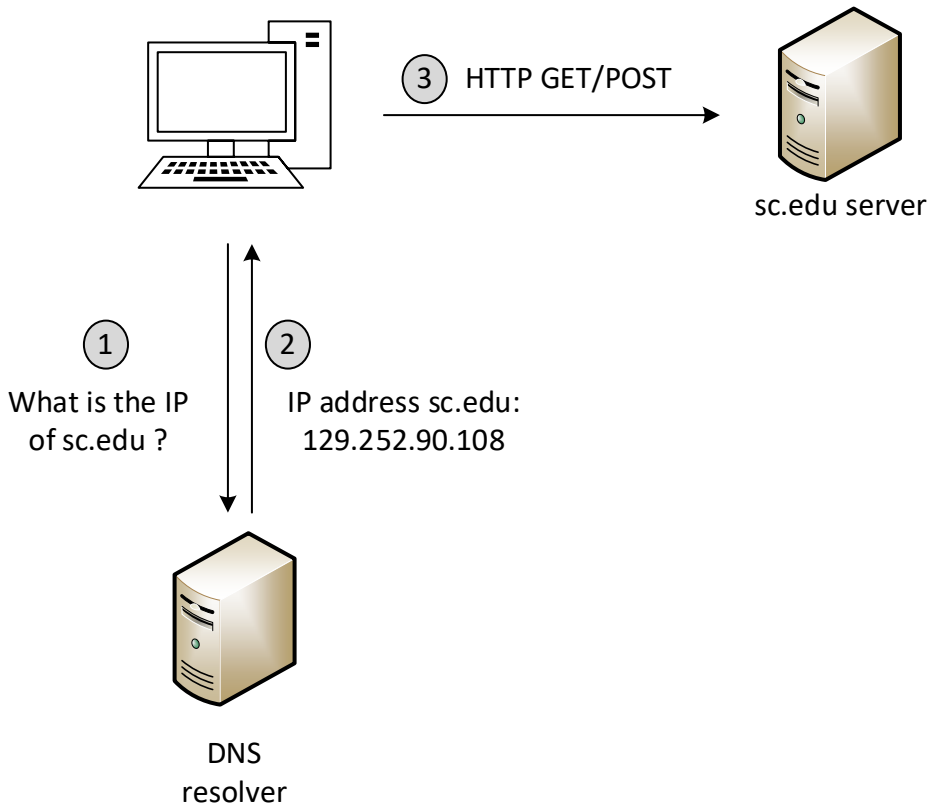
Western Academy Support and Training Center (WASTC)
University of South Carolina (USC)
Energy Sciences Network (ESnet)

June 22nd, 2023

# Lab 8: Detecting and Mitigating the DNS Amplification Attack

# Domain Name System (DNS)

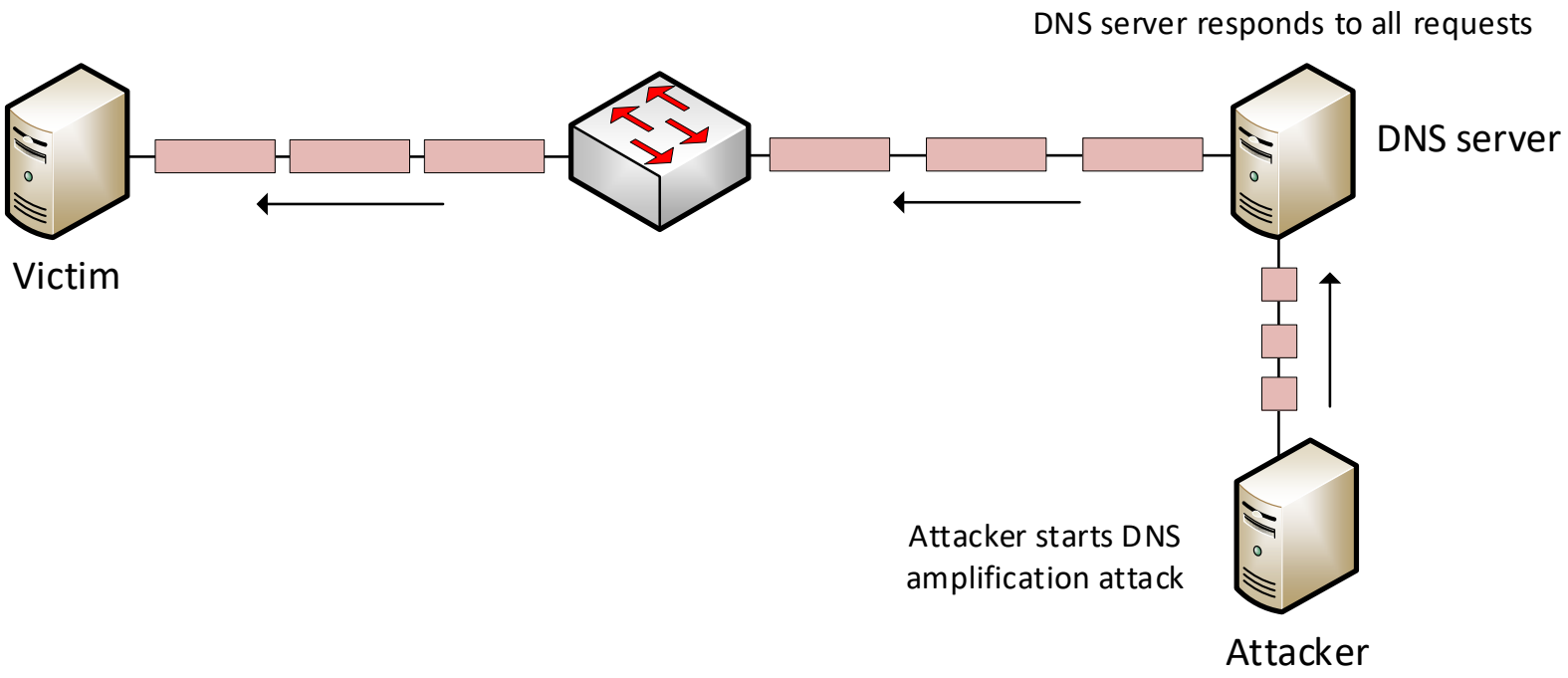
- The Domain Name System (DNS) is an essential component of the internet
- It is responsible for translating human-readable domain names into IP addresses that can be understood by devices connected to the internet[1]




[1] Amazon, “What is DNS?” [Online]. Available: https://tinyurl.com/ynb9esn6

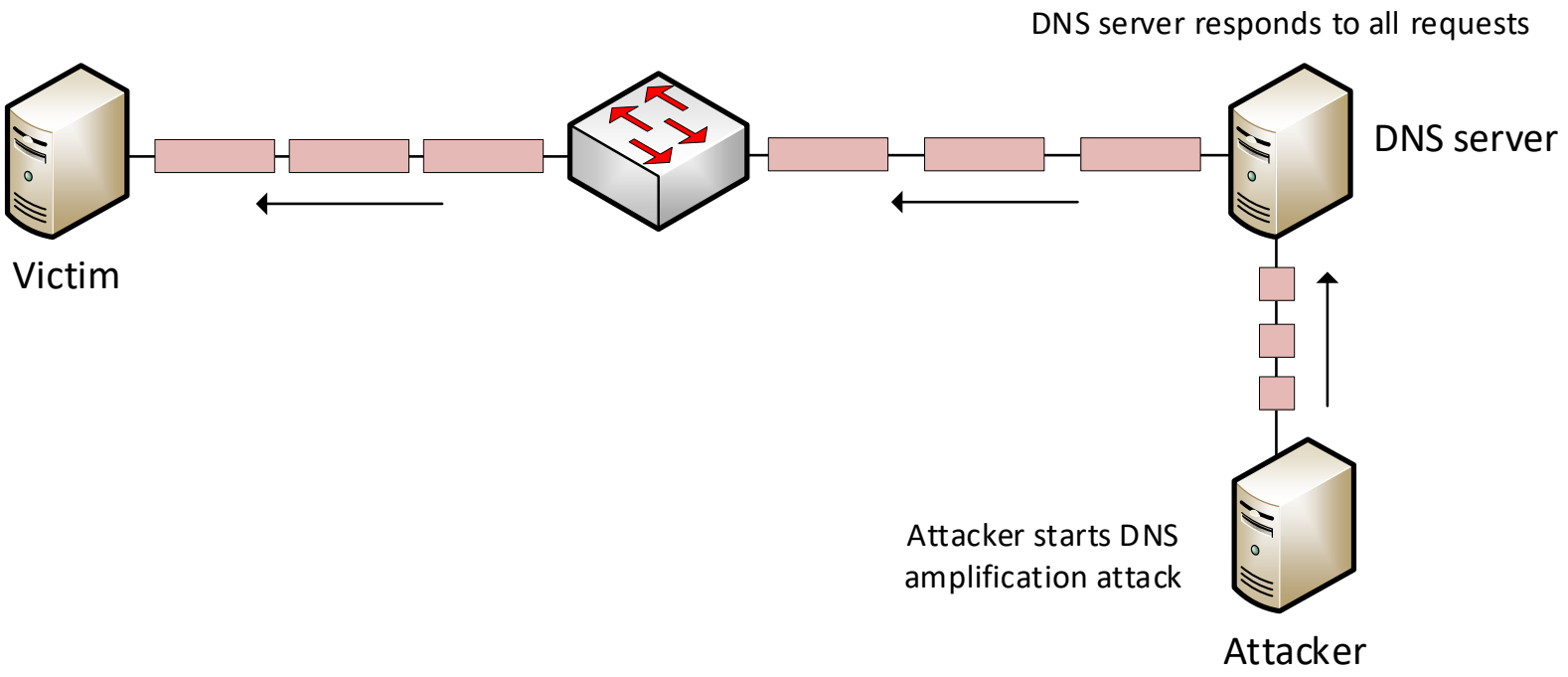
# DNS Amplification

- Due its frequent use on the Internet, the DNS is susceptible to many attacks
- DNS amplification is an attack that exploits the infrastructure of the DNS to create a Distributed Denial of Service (DDoS) on a target victim
  - Thus, rendering the victim unresponsive due to the large amount of traffic received



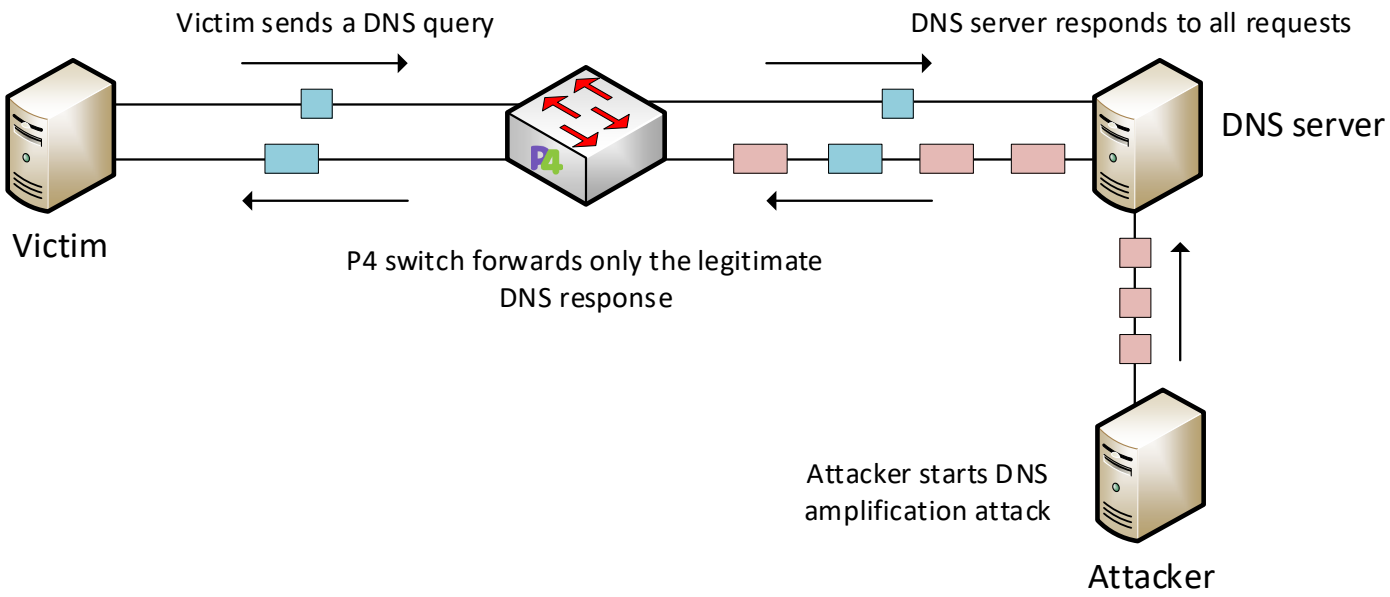
# DNS Amplification

- In a DNS amplification attach, the attacker sends DNS requests with spoofed source IP address of the target victim
- The DNS server responds to all the requests and send them to the target victim
- Typically, a valid response is much larger than the request



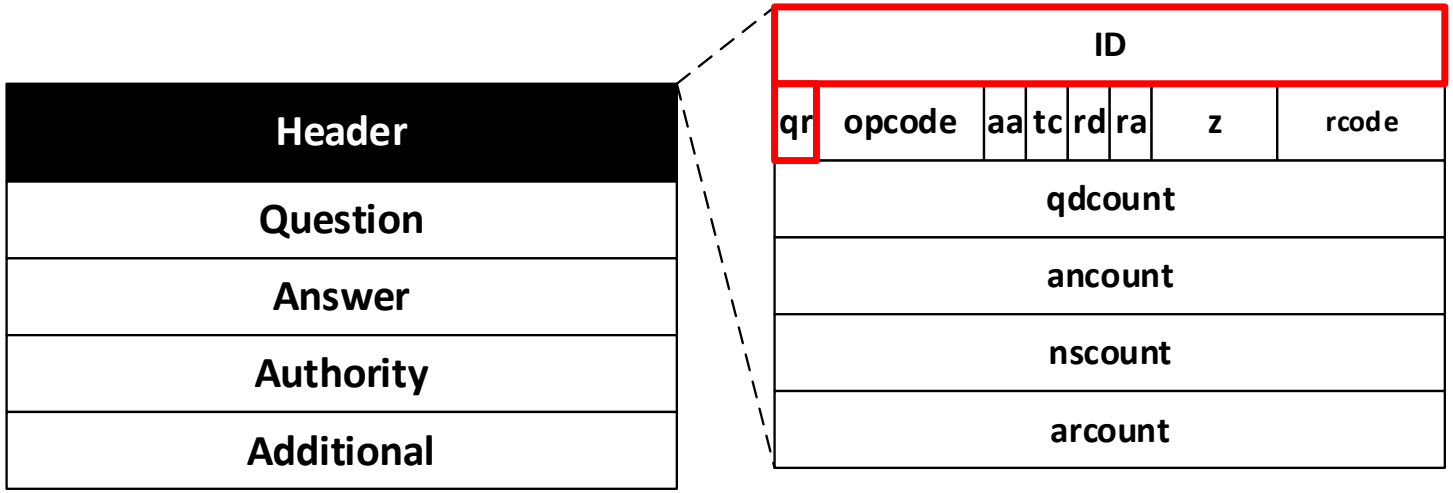
# Attack Scenario

- The attacker performs a DNS amplification attack using the source IP address of the victim
- The DNS server responds to the DNS requests and sends them to the victim
- The P4 switch drops all the DNS replies that do not associate with DNS requests issued by the victim
- The P4 switch forwards legitimate DNS replies requested by the victim



# DNS Header Fields

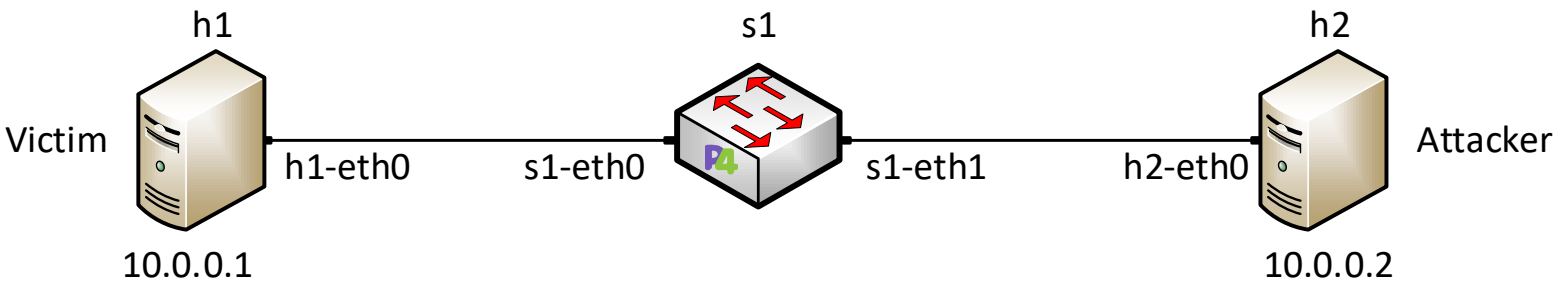
- UDP packets with source or destination port 53 are DNS packets
- The transaction ID is generated by the client upon sending a DNS request
- The DNS *qr* flag indicates that the DNS packet is request (*qr*=0) or response (*qr*=1)



# Lab Topology

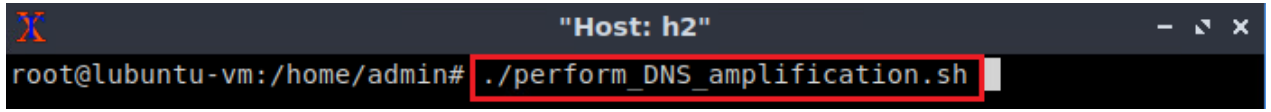
The topology consists of:

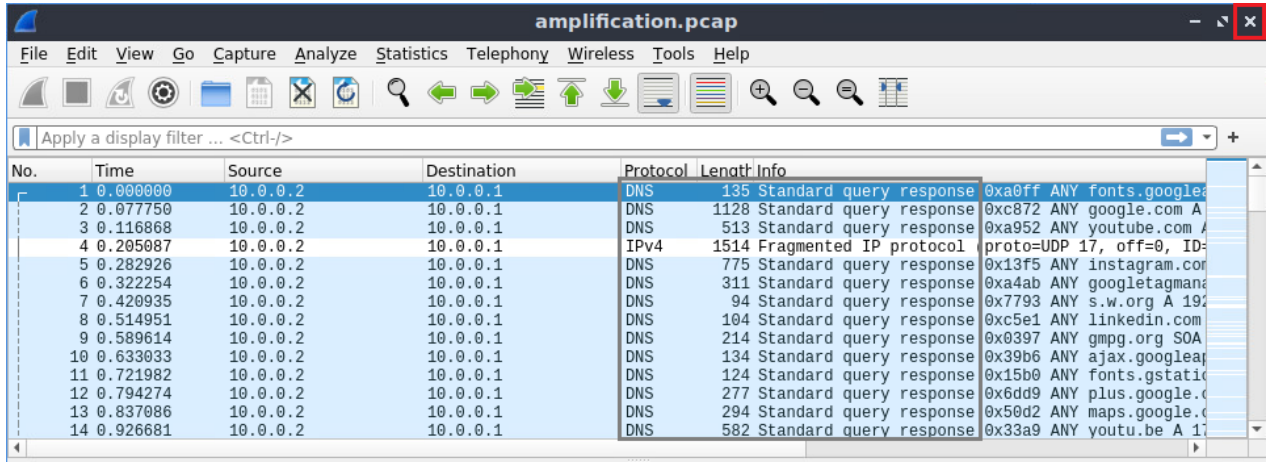
- Host h1 representing the victim of the DNS amplification attack
- Host h2 representing the attacker and the DNS resolver
- Programmable switch s1 that forwards traffic and protects against DNS amplification attack



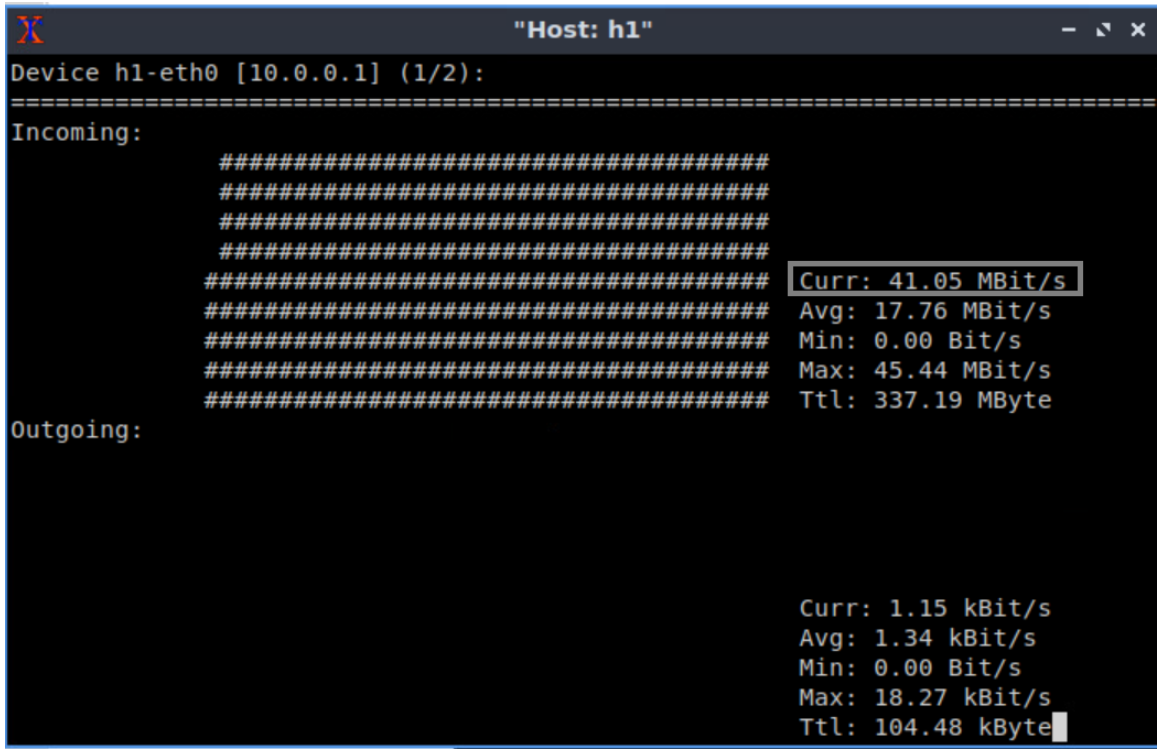
# DNS Amplification without Mitigation

Performing DNS amplification




Inspecting resource usage at the victim



# DNS Amplification Mitigation with P4

Defining DNS header

headers.p4 - lab8 - Visual Studio Code

```
header udp_t {
    bit<16> srcPort;
    bit<16> dstPort;
    bit<16> len;
    bit<16> checksum;
}

/* Define the DNS header below*/
header dns_t {
    bit<16> transaction_id;
    bit<1> qr_flag;
    bit<7> padding;
}

struct metadata {
    /*empty*/
}
```

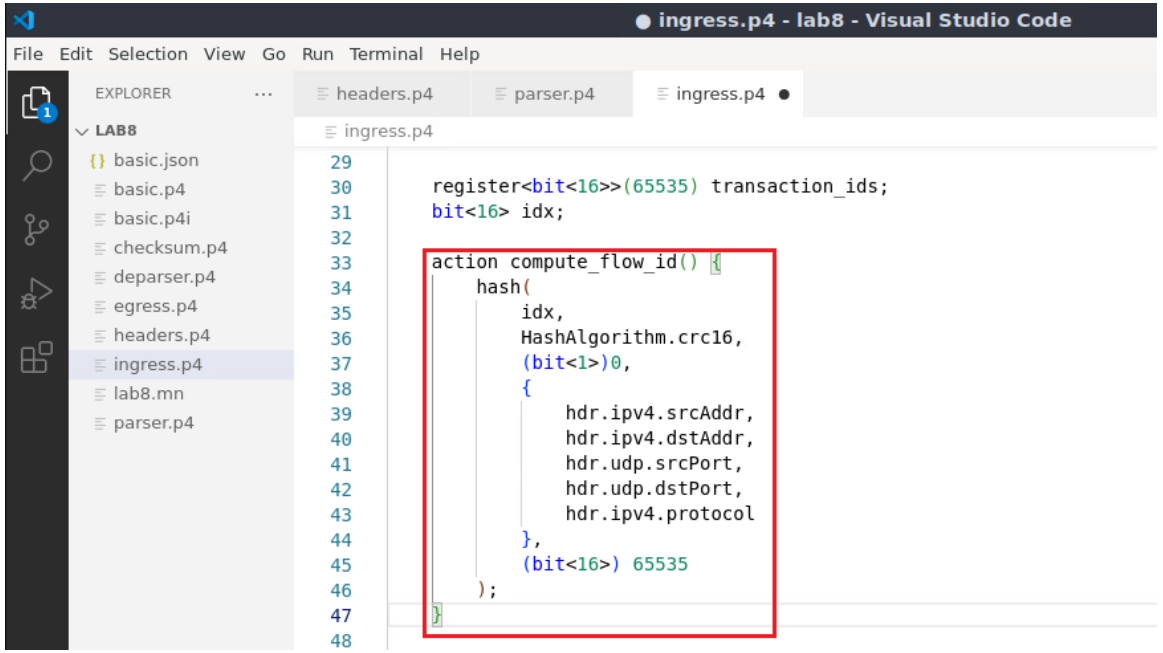
Transitioning and extracting DNS header

parser.p4 - lab8 - Visual Studio Code

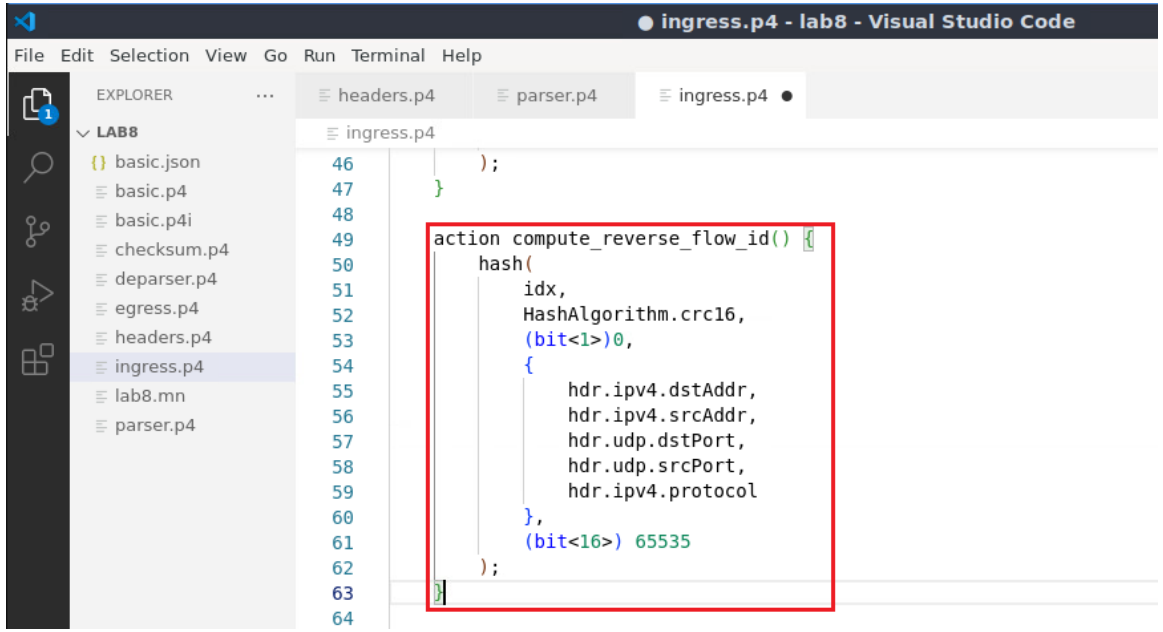
```
    state parse_ipv4 {
        packet.extract(hdr.ipv4);
        transition select(hdr.ipv4.protocol) {
            TYPE_UDP: parse_udp;
            default: accept;
        }
    }
    state parse_udp {
        packet.extract(hdr.udp);
        transition select(hdr.udp.srcPort, hdr.udp.dstPort) {
            (TYPE_DNS,_): parse_dns;
            (_,TYPE_DNS): parse_dns;
            (_,_): accept;
        }
    }
    state parse_dns {
        packet.extract(hdr.dns);
        transition accept;
    }
```

# DNS Amplification Mitigation with P4

Defining a function that computes a unique flow ID based on the 5-tuple (used for DNS requests)




Defining a function that computes a unique flow ID based on the 5-tuple (used for DNS replies)



# DNS Amplification Mitigation with P4

Using the flow ID to save the transaction ID of DNS request in a P4 register

ingress.p4 - lab8 - Visual Studio Code

```
54                  {
55                      hdr.ipv4.dstAddr,
56                      hdr.ipv4.srcAddr,
57                      hdr.udp.dstPort,
58                      hdr.udp.srcPort,
59                      hdr.ipv4.protocol
60                  },
61                  (bit<16>) 65535
62              );
63          }
64
65          apply {
66              if (hdr.dns.isValid()) {
67                  if (hdr.dns.qr_flag == 0) {
68                      compute_flow_id();
69                      transaction_ids.write((bit<32>) idx, hdr.dns.transaction_id);
70                      forwarding.apply();
71                  }
72              }
73          }
74      }
```

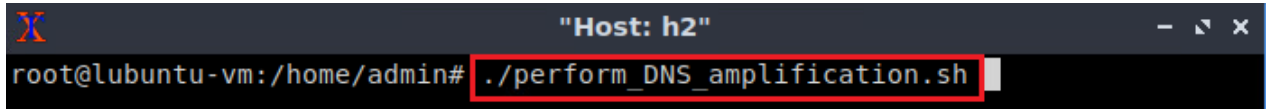
Using the flow ID to retrieve the transaction ID of DNS request saved in a P4 register and comparing it with the transaction ID of the received DNS reply

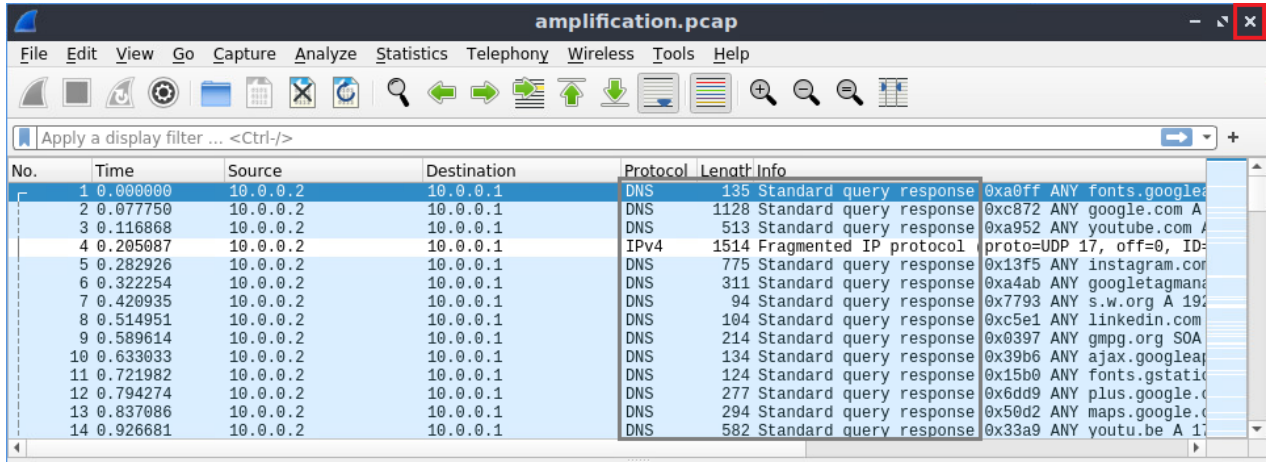
ingress.p4 - lab8 - Visual Studio Code

```
64
65          apply {
66              if (hdr.dns.isValid()) {
67                  if (hdr.dns.qr_flag == 0) {
68                      compute_flow_id();
69                      transaction_ids.write((bit<32>) idx, hdr.dns.transaction_id);
70                      forwarding.apply();
71                  }
72                  else if (hdr.dns.qr_flag == 1) {
73                      bit<16> transaction_id;
74                      compute_reverse_flow_id();
75                      transaction_ids.read(transaction_id, (bit<32>) idx);
76                      if (transaction_id == hdr.dns.transaction_id) {
77                          forwarding.apply();
78                      }
79                      else {
80                          drop();
81                      }
82                  }
83              }
84          }
85      }
```

# DNS Amplification with Mitigation

Performing DNS amplification




Inspecting resource usage at the victim