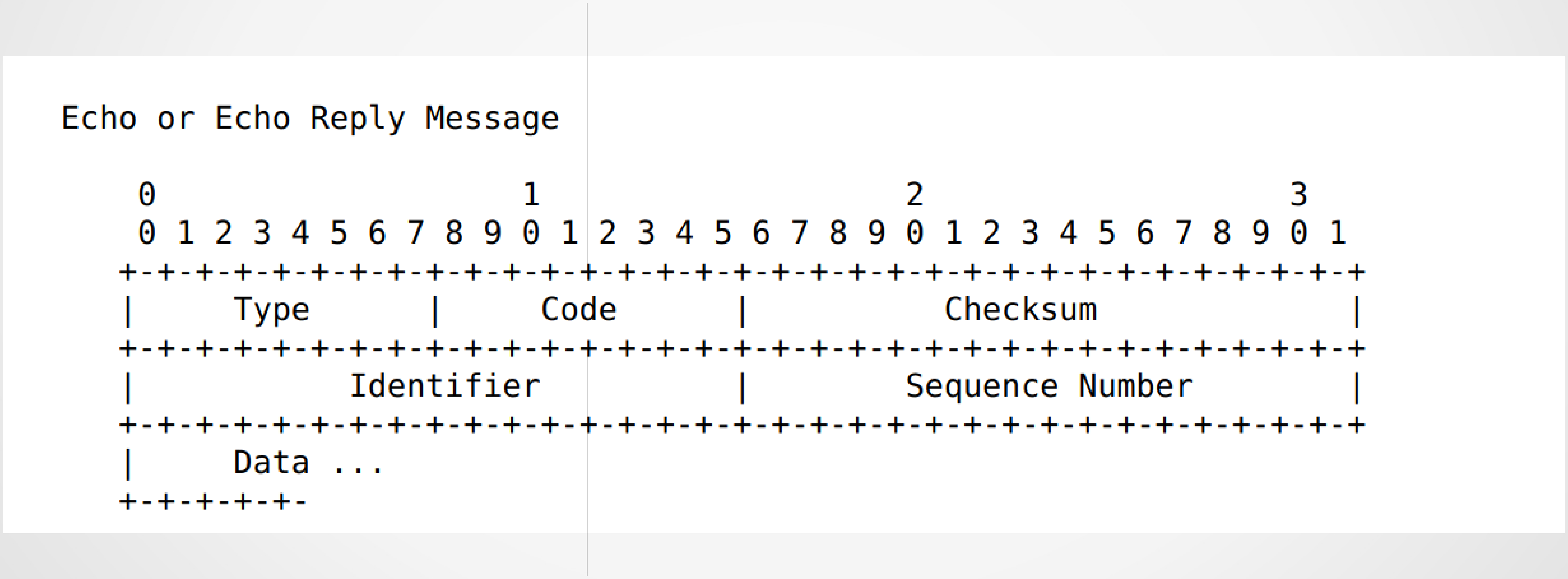
Kernel Mania

Theme: Embedded System for Medical Device

Backdoor in ICMP

RFC says this should remain 0...



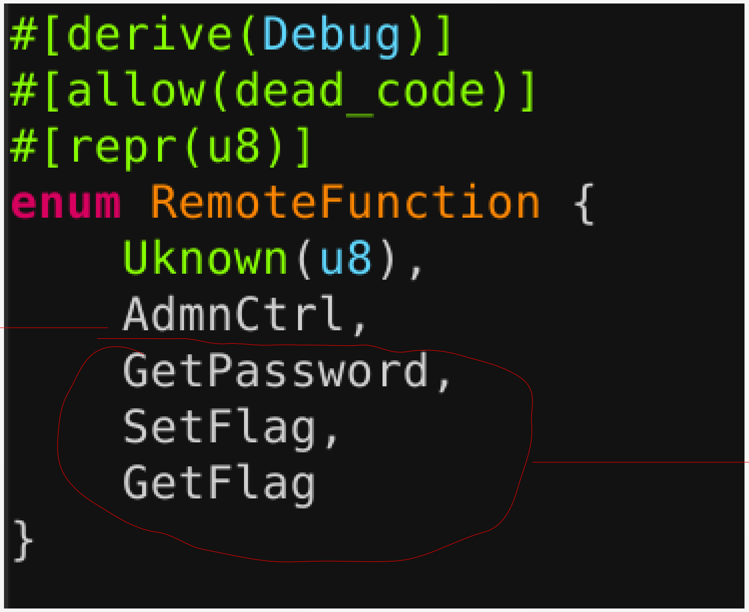
```
Echo or Echo Reply Message

    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |     Code      |          Checksum             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Identifier          |        Sequence Number        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Data ...
   +-+-+-+-+-
```

I’m using it for a 1 byte checksum.

If it matches, the kernel decrypts the data section with XOR

Remote function

Depending on the first byte it calls the apropriate function



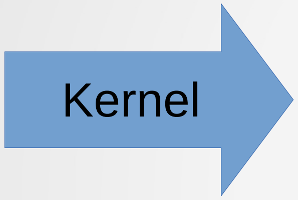
```
#[derive(Debug)]
#[allow(dead_code)]
#[repr(u8)]
enum RemoteFunction {
    Uknown(u8),
    AdmnCtrl,
    GetPassword,
    SetFlag,
    GetFlag
}
```

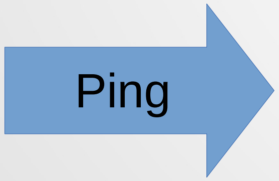
Exploitable function
with hardcoded password

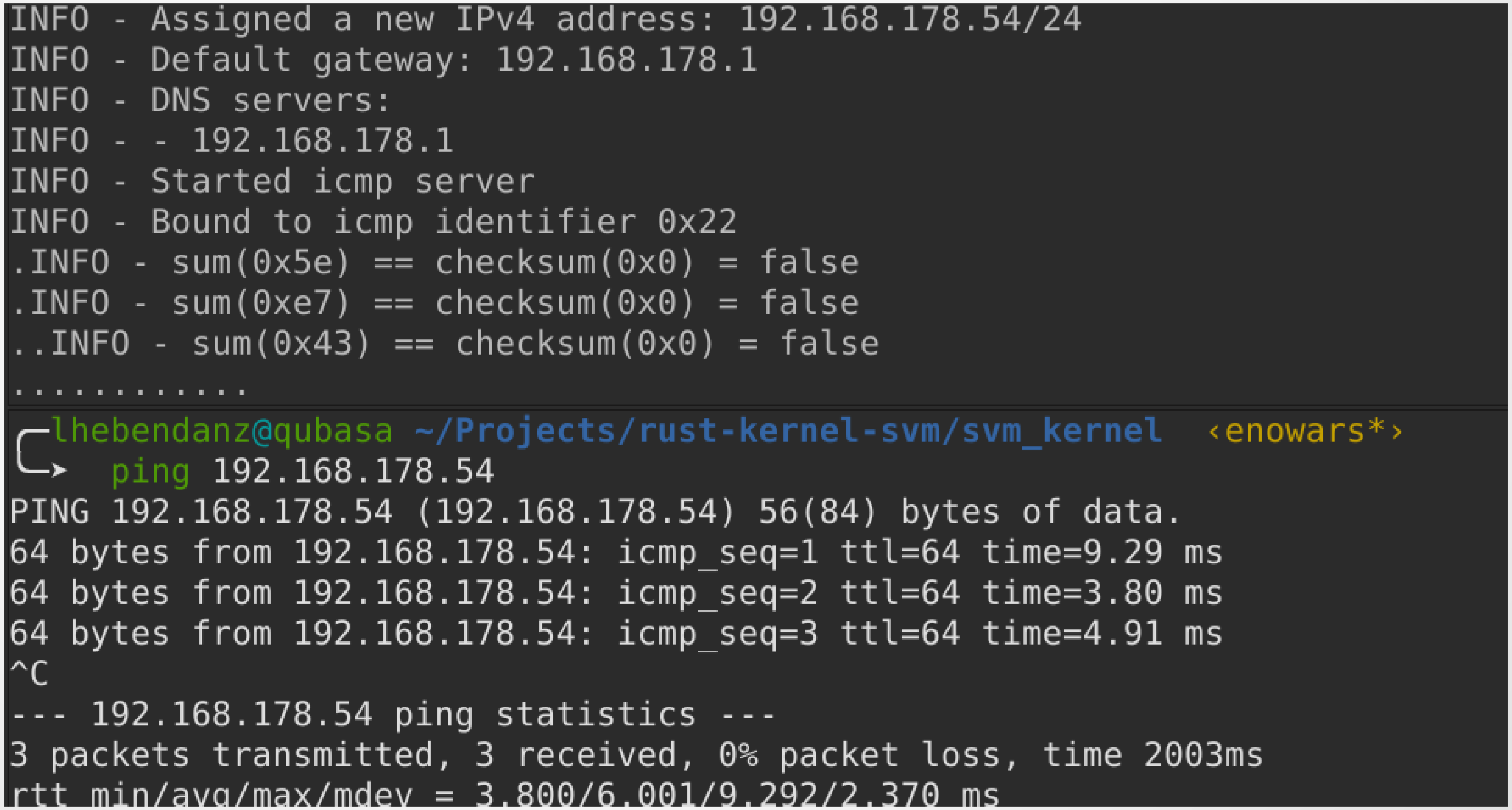
Functions for checker

TODO: Needs pub / priv crypto verification

Kernel responds normally on ICMP



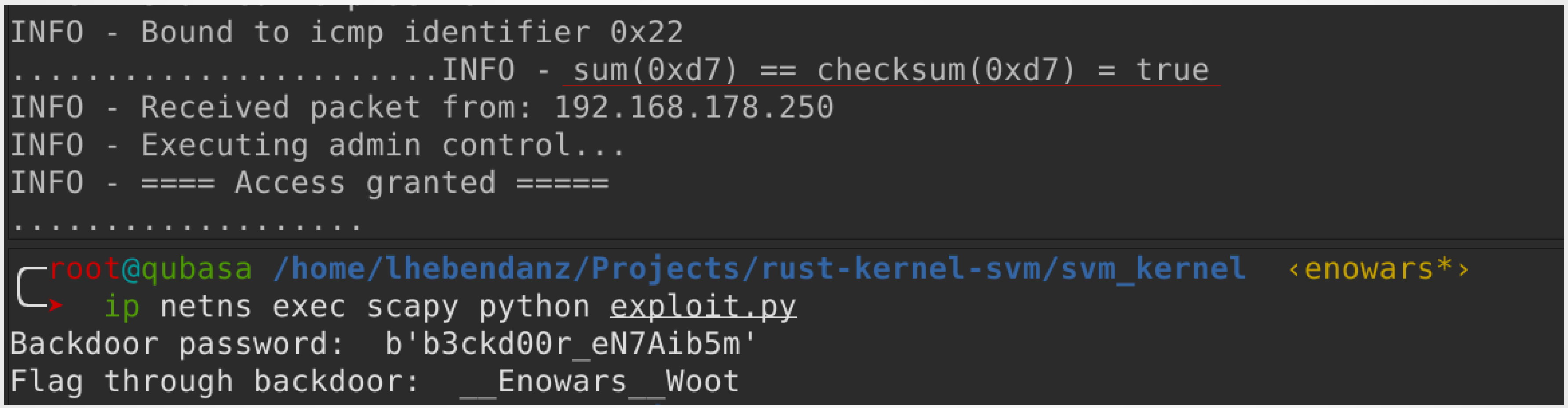




```
INFO - Assigned a new IPv4 address: 192.168.178.54/24
INFO - Default gateway: 192.168.178.1
INFO - DNS servers:
INFO - - 192.168.178.1
INFO - Started icmp server
INFO - Bound to icmp identifier 0x22
.INFO - sum(0x5e) == checksum(0x0) = false
.INFO - sum(0xe7) == checksum(0x0) = false
..INFO - sum(0x43) == checksum(0x0) = false
...........
```

```
┌─lhebendanz@qubasa ~/Projects/rust-kernel-svm/svm_kernel  ‹enowars*›
└─➤  ping 192.168.178.54
PING 192.168.178.54 (192.168.178.54) 56(84) bytes of data.
64 bytes from 192.168.178.54: icmp_seq=1 ttl=64 time=9.29 ms
64 bytes from 192.168.178.54: icmp_seq=2 ttl=64 time=3.80 ms
64 bytes from 192.168.178.54: icmp_seq=3 ttl=64 time=4.91 ms
^C
--- 192.168.178.54 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.800/6.001/9.292/2.370 ms
```

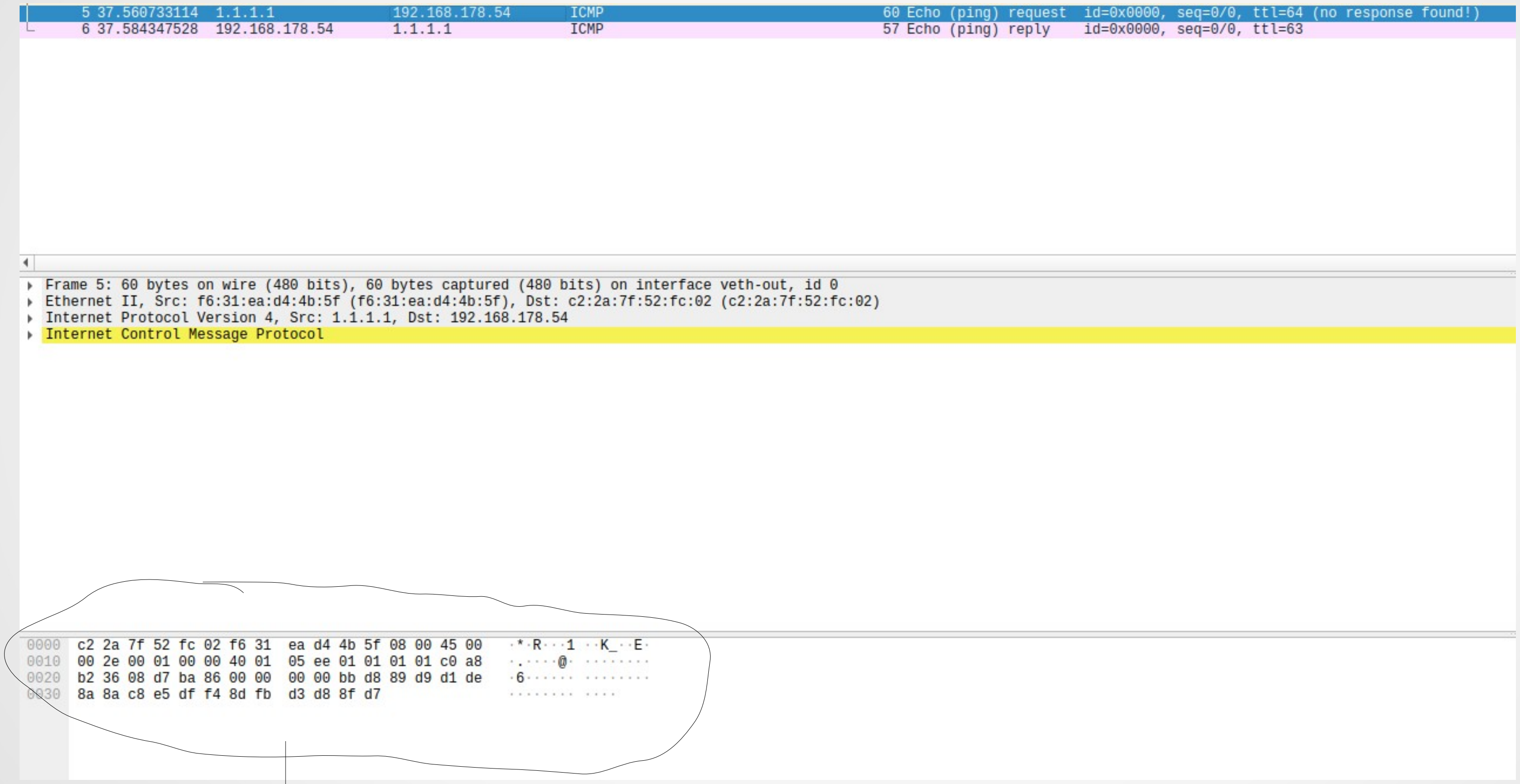
But differently if checksum correct



```
INFO - Bound to icmp identifier 0x22
.......................INFO - sum(0xd7) == checksum(0xd7) = true
INFO - Received packet from: 192.168.178.250
INFO - Executing admin control...
INFO - ==== Access granted =====
...................
```

```
┌root@qubasa /home/lhebendanz/Projects/rust-kernel-svm/svm_kernel  ‹enowars*›
└➤  ip netns exec scapy python exploit.py
Backdoor password:  b'b3ckd00r_eN7Aib5m'
Flag through backdoor:  __Enowars__Woot
```

Network traffic is hidden



```
    5 37.560733114  1.1.1.1          192.168.178.54   ICMP                    60 Echo (ping) request  id=0x0000, seq=0/0, ttl=64 (no response found!)
    6 37.584347528  192.168.178.54   1.1.1.1          ICMP                    57 Echo (ping) reply    id=0x0000, seq=0/0, ttl=63
```

```
▸ Frame 5: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface veth-out, id 0
▸ Ethernet II, Src: f6:31:ea:d4:4b:5f (f6:31:ea:d4:4b:5f), Dst: c2:2a:7f:52:fc:02 (c2:2a:7f:52:fc:02)
▸ Internet Protocol Version 4, Src: 1.1.1.1, Dst: 192.168.178.54
▸ Internet Control Message Protocol
```

```
0000  c2 2a 7f 52 fc 02 f6 31  ea d4 4b 5f 08 00 45 00   ·*·R···1 ··K_··E·
0010  00 2e 00 01 00 00 40 01  05 ee 01 01 01 01 c0 a8   ·.····@· ········
0020  b2 36 08 d7 ba 86 00 00  00 00 bb d8 89 d9 d1 de   ·6······ ········
0030  8a 8a c8 e5 df f4 8d fb  d3 d8 8f d7               ········ ····
```

You do not see cleartext password because of XOR

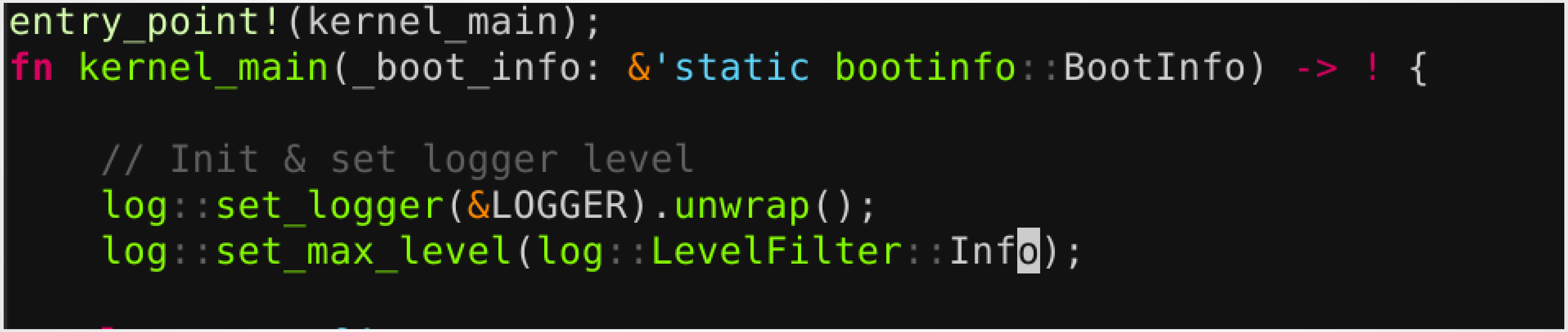
⟶ I will add somekind of “randomness” to foil copy and paste of network traffic

How to defend?

- Extract bootloader.elf from bootimage-svm_kernel.iso
- Extract kernel.elf from bootloader.elf section called .kernel to get debug symbols for kernel
- Find hardcoded password and change it
- Repackage everything and reexecute

Quality of life considerations

- I choose the rtl8139 network driver because it works with qemu emulation without kvm → You can copy the iso and execute it on every OS with qemu



```
entry_point!(kernel_main);
fn kernel_main(_boot_info: &'static bootinfo::BootInfo) -> ! {

    // Init & set logger level
    log::set_logger(&LOGGER).unwrap();
    log::set_max_level(log::LevelFilter::Info);
```

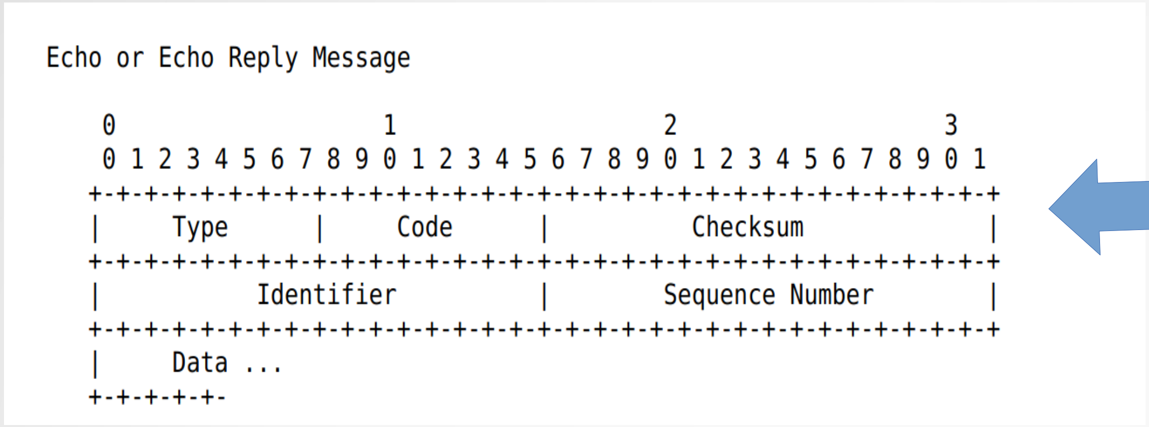
If the player needs more information, he can patch the log level from Info to Debug.
I placed debug prints at important sections of the code

Problems I did encounter….

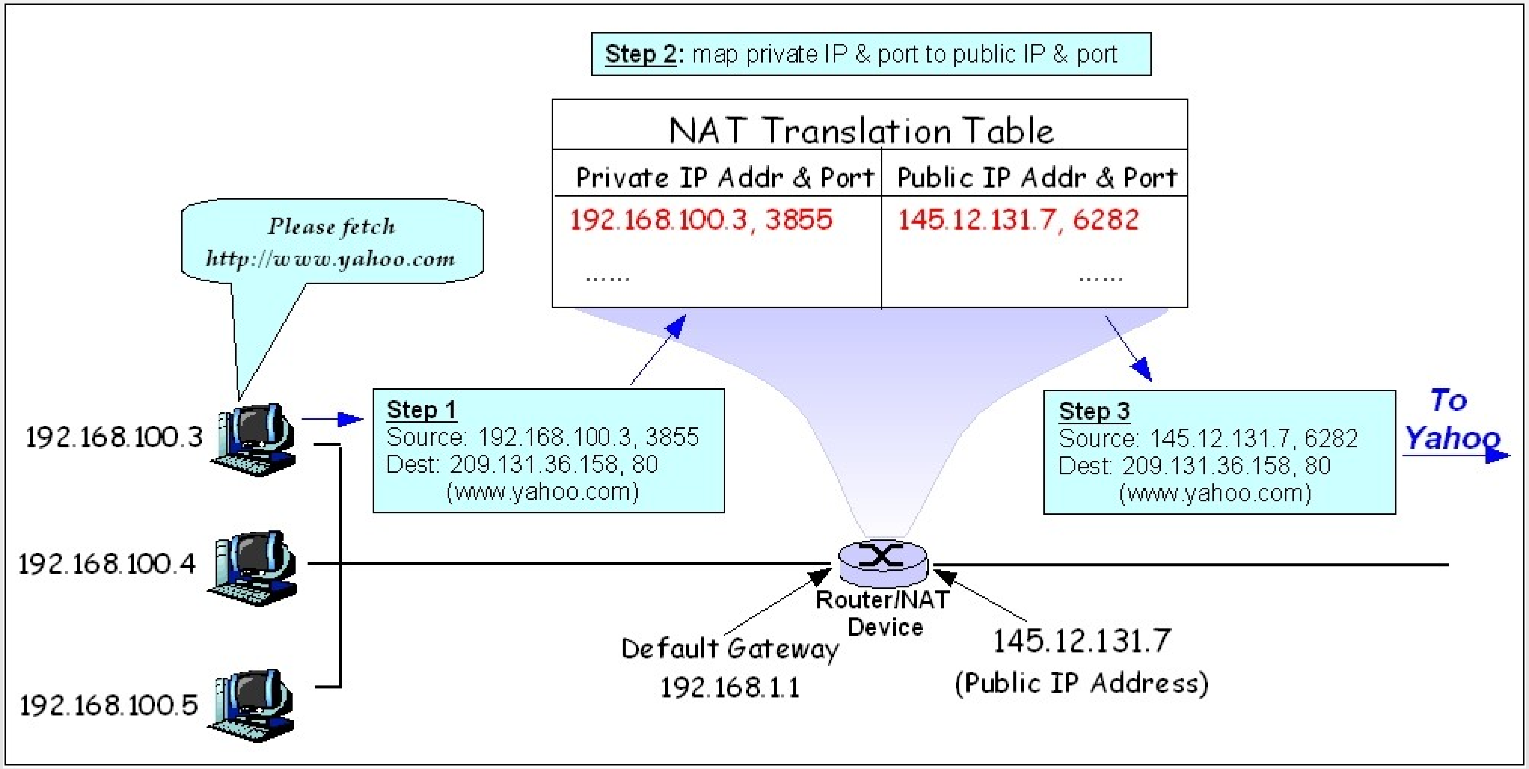
- My network driver crashed… times and times again.

  → Incredibly badly documented and example code is broken. Fixed now
- Creating a NATed environment to check if my protocol works through NAT.
- ...My protocol did not work in NAT…
- Now it does ;)

Reason it did not work in NAT



```
Echo or Echo Reply Message

    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |     Code      |          Checksum             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Identifier          |        Sequence Number        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Data ...
   +-+-+-+-+-
```



Current Problems

- Kernel has own IP how does the checker get it?
- How to display the Kernel IP to every other team?
- My checker needs raw socket priviliges because of scapy, how do I do this in Docker with python?