April 30, 2025

GHSL-2025-012_GHSL-2025-022: Multiple vulnerabilities in Retrieval-based-Voice-Conversion-WebUI - CVE-2025-43842_CVE-2025-43852

Sylwia Budzynska

Coordinated Disclosure Timeline

- 2025-01-20: Created an issue in the repository asking for a vulnerability contact.
- 2025-01-25: Received an answer from one of the contributors, that the repository is not fully active.
- 2025-02-26: Tried reaching out again, asking to enable the private vulnerability reporting feature on GitHub.
- 2025-04-23: GitHub Security Lab assigned CVEs due to our 90 day disclosure policy.

Summary

Retrieval-based-Voice-Conversion-WebUI is vulnerable to commandline injection

Project

Retrieval-based-Voice-Conversion-WebUI

Tested Version

2.2.231006

Details

Issue 1: command injection in infer-web.py `preprocess_dataset` (`GHSL-2025-012`)

The variables `exp_dir1`, `np7`, `trainset_dir4` and `sr2` take user input and pass it to the `preprocess_dataset` function, which concatenates them into a command that is run on the server. This can lead to arbitrary command execution.

```
def preprocess_dataset(trainset_dir, exp_dir, sr, n_p):
    sr = sr_dict[sr]
    os.makedirs("%s/logs/%s" % (now_dir, exp_dir), exist_ok=True)
    f = open("%s/logs/%s/preprocess.log" % (now_dir, exp_dir), "w")
    f.close()
    cmd = '"%s" infer/modules/train/preprocess.py "%s" %s %s "%s/logs/%s" %s %.1f' % (
        config.python_cmd,
        trainset_dir,
        sr,
        n_p,
        now_dir,
        exp_dir,
        config.noparallel,
        config.preprocess_per,
    )
```

Impact

This issue may lead to arbitrary command execution.

CWEs

- CWE-77: Improper Neutralization of Special Elements used in a Command (‘Command Injection’)

Issue 2: command injection in infer-web.py `extract_f0_feature` (`GHSL-2025-013`)

The variables `exp_dir1`, `np7` and `f0method8` take user input and pass it to the `extract_f0_feature` function, which concatenates them into a command that is run on the server. This can lead to arbitrary command execution. Note that the commands are run several times in the `extract_f0_feature` function, and all of them would have to be fixed to prevent commandline injection:

- https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/blob/7ef19867780cf703841ebafb565a4e47d1ea86ff/infer-web.py#L276-L278
- https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/blob/7ef19867780cf703841ebafb565a4e47d1ea86ff/infer-web.py#L307-L309
- https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/blob/7ef19867780cf703841ebafb565a4e47d1ea86ff/infer-web.py#L330-L332
- https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI/blob/7ef19867780cf703841ebafb565a4e47d1ea86ff/infer-web.py#L373-L375

```
def extract_f0_feature(gpus, n_p, f0method, if_f0, exp_dir, version19, gpus_rmvpe):
  gpus = gpus.split("-")
  os.makedirs("%s/logs/%s" % (now_dir, exp_dir), exist_ok=True)
  f = open("%s/logs/%s/extract_f0_feature.log" % (now_dir, exp_dir), "w")
  f.close()
  if if_f0:
      if f0method != "rmvpe_gpu":
          cmd = (
              '"%s" infer/modules/train/extract/extract_f0_print.py "%s/logs/%s" %s %s'
              % (
                  config.python_cmd,
                  now_dir,
                  exp_dir,
                  n_p,
                  f0method,
              )
          )
          logger.info("Execute: " + cmd)
          p = Popen(
              cmd, shell=True, cwd=now_dir
          )  # , stdin=PIPE, stdout=PIPE,stderr=PIPE
```

Impact

This issue may lead to arbitrary command execution.

CWEs

- CWE-77: Improper Neutralization of Special Elements used in a Command (‘Command Injection’)

Issue 3: command injection in infer-web.py `click_train` (`GHSL-2025-014`)

The variables `exp_dir1`, among others, take user input and pass it to the `click_train` function, which concatenates them into a command hereand here that is run on the server. This can lead to arbitrary command execution.

```
if gpus16:
    cmd = (
        '"%s" infer/modules/train/train.py -e "%s" -sr %s -f0 %s -bs %s -g %s -te %s -se %s %s %s -l %s -c %s -sw %s -v %s'
        % (
            config.python_cmd,
            exp_dir1,
            sr2,
            1 if if_f0_3 else 0,
            batch_size12,
            gpus16,
            total_epoch11,
            save_epoch10,
            "-pg %s" % pretrained_G14 if pretrained_G14 != "" else "",
            "-pd %s" % pretrained_D15 if pretrained_D15 != "" else "",
            1 if if_save_latest13 == i18n("是") else 0,
            1 if if_cache_gpu17 == i18n("是") else 0,
            1 if if_save_every_weights18 == i18n("是") else 0,
            version19,
        )
    )
else:
```

```
        cmd = (
            '"%s" infer/modules/train/train.py -e "%s" -sr %s -f0 %s -bs %s -te %s -se %s %s %s -l %s -c %s -sw %s -v %s'
            % (
                config.python_cmd,
                exp_dir1,
                sr2,
                1 if if_f0_3 else 0,
                batch_size12,
                total_epoch11,
                save_epoch10,
                "-pg %s" % pretrained_G14 if pretrained_G14 != "" else "",
                "-pd %s" % pretrained_D15 if pretrained_D15 != "" else "",
                1 if if_save_latest13 == i18n("是") else 0,
                1 if if_cache_gpu17 == i18n("是") else 0,
                1 if if_save_every_weights18 == i18n("是") else 0,
                version19,
            )
        )
    logger.info("Execute: " + cmd)
    p = Popen(cmd, shell=True, cwd=now_dir)
```

Impact

This issue may lead to arbitrary command execution.

CWEs

- CWE-77: Improper Neutralization of Special Elements used in a Command (‘Command Injection’)

Issue 4: code injection in infer-web.py `change_info_` (`GHSL-2025-015`)

The `ckpt_path2` variable takes user input (e.g. a path to a model) and passes it to `change_info_` function, which opens and reads the file on the given path (except it changes the final on the path to `train.log`), and passes the contents of the file to `eval`, which can lead to remote code execution.

```
def change_info_(ckpt_path):
    if not os.path.exists(ckpt_path.replace(os.path.basename(ckpt_path), "train.log")):
        return {"__type__": "update"}, {"__type__": "update"}, {"__type__": "update"}
    try:
        with open(
            ckpt_path.replace(os.path.basename(ckpt_path), "train.log"), "r"
        ) as f:
            info = eval(f.read().strip("\n").split("\n")[0].split("\t")[-1])
```

Impact

This issue may lead to `remote code execution`.

CWEs

- CWE-94: Improper Control of Generation of Code (‘Code Injection’)

Issue 5: unsafe deserialization in process_ckpt.py `show_info` function (`GHSL-2025-016`)

The `ckpt_path1` variable takes user input (e.g. a path to a model) and passes it to the `show_info` function in `process_ckpt.py`, which uses it to load the model on that path with `torch.load`, which can lead to unsafe deserialization.

```
def show_info(path):
    try:
        a = torch.load(path, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

Issue 6: unsafe deserialization in process_ckpt.py `extract_small_model` function (`GHSL-2025-017`)

The `ckpt_path2` variable takes user input (e.g. a path to a model) and passes it to the `extract_small_model` function in `process_ckpt.py`, which uses it to load the model on that path with `torch.load`, which can lead to unsafe deserialization.

```
def extract_small_model(path, name, sr, if_f0, info, version):
    try:
        ckpt = torch.load(path, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

Issue 7: unsafe deserialization in process_ckpt.py `change_info` function (`GHSL-2025-018`)

The `ckpt_path0` variable takes user input (e.g. a path to a model) and passes it to the `change_info` function in `process_ckpt.py`, which uses it to load the model on that path with `torch.load`, which can lead to unsafe deserialization.

```
def change_info(path, info, name):
    try:
        ckpt = torch.load(path, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

Issue 8: unsafe deserialization in process_ckpt.py `merge` function (`GHSL-2025-019`)

The `ckpt_a` and `cpkt_b` variables take user input here and here (e.g. a path to a model) and pass it to the `merge` function in `process_ckpt.py`, which uses them to load the models on those paths with `torch.load`, which can lead to unsafe deserialization.

```
def merge(path1, path2, alpha1, sr, f0, info, name, version):
    try:

        def extract(ckpt):
            a = ckpt["model"]
            opt = OrderedDict()
            opt["weight"] = {}
            for key in a.keys():
                if "enc_q" in key:
                    continue
                opt["weight"][key] = a[key]
            return opt

        ckpt1 = torch.load(path1, map_location="cpu")
        ckpt2 = torch.load(path2, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

Issue 9: unsafe deserialization in export.py (`GHSL-2025-020`)

The `ckpt_dir` variable takes user input (e.g. a path to a model) and passes it to the `change_info` function in `export.py`, which uses it to load the model on that path with `torch.load`, which can lead to unsafe deserialization.

```
def export_onnx(ModelPath, ExportedPath):
    cpt = torch.load(ModelPath, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

Issue 10: unsafe deserialization in vr.py `AudioPre` (`GHSL-2025-021`)

The `model_choose` variable takes user input (e.g. a path to a model) and passes it to the `uvr` function in `vr.py`. In `uvr` , a new instance of `AudioPre` class is created with the `model_path` attribute containing the aformentioned user input (here called locally `model_name`). Note that in this step there is added the `.pth` extension to the path.

In the `AudioPre` class, the user input, here called `model_path`, is used to load the model on that path with `torch.load`, which can lead to unsafe deserialization.

```
class AudioPre:
    def __init__(self, agg, model_path, device, is_half, tta=False):
        self.model_path = model_path
        self.device = device
        self.data = {
            # Processing Options
            "postprocess": False,
            "tta": tta,
            # Constants
            "window_size": 512,
            "agg": agg,
            "high_end_process": "mirroring",
        }
        mp = ModelParameters("infer/lib/uvr5_pack/lib_v5/modelparams/4band_v2.json")
        model = Nets.CascadedASPPNet(mp.param["bins"] * 2)
        cpk = torch.load(model_path, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

Issue 11: unsafe deserialization in vr.py `AudioPreDeEcho` (`GHSL-2025-022`)

The `model_choose` variable takes user input (e.g. a path to a model) and passes it to the `uvr` function in `vr.py`. In `uvr` , if `model_name` contains the string `"DeEcho"`, a new instance of `AudioPreDeEcho` class is created with the `model_path` attribute containing the aforementioned user input (here called locally `model_name`). Note that in this step there is added the `.pth` extension to the path.

In the `AudioPreDeEcho` class, the user input, here called `model_path`, is used to load the model on that path with `torch.load`, which can lead to unsafe deserialization.

```
class AudioPreDeEcho:
    def __init__(self, agg, model_path, device, is_half, tta=False):
        self.model_path = model_path
        self.device = device
        self.data = {
            # Processing Options
            "postprocess": False,
            "tta": tta,
            # Constants
            "window_size": 512,
            "agg": agg,
            "high_end_process": "mirroring",
        }
        mp = ModelParameters("infer/lib/uvr5_pack/lib_v5/modelparams/4band_v3.json")
        nout = 64 if "DeReverb" in model_path else 48
        model = CascadedNet(mp.param["bins"] * 2, nout)
        cpk = torch.load(model_path, map_location="cpu")
```

Impact

The issue can lead to remote code execution.

CWEs

- CWE-502: Deserialization of Untrusted Data

CVE

- GHSL-2025-012 - CVE-2025-43842
- GHSL-2025-013 - CVE-2025-43843

- GHSL-2025-014 - CVE-2025-43844
- GHSL-2025-015 - CVE-2025-43845
- GHSL-2025-016 - CVE-2025-43846
- GHSL-2025-017 - CVE-2025-43847
- GHSL-2025-018 - CVE-2025-43848
- GHSL-2025-019 - CVE-2025-43849
- GHSL-2025-020 - CVE-2025-43850
- GHSL-2025-021 - CVE-2025-43851
- GHSL-2025-022 - CVE-2025-43852

Credit

These issues were discovered and reported by GHSL team member @sylwia-budzynska (Sylwia Budzynska).

Contact

You can contact the GHSL team at `securitylab@github.com`, please include a reference to `GHSL-2025-012`, `GHSL-2025-013`, `GHSL-2025-014`, `GHSL-2025-015`, `GHSL-2025-016`, `GHSL-2025-017`, `GHSL-2025-018`, `GHSL-2025-019`, `GHSL-2025-020`, `GHSL-2025-021`, or `GHSL-2025-022` in any communication regarding these issues.

-