IISC/CTF: replme

Review of Enowars 8

Jacob Bachmann

SecT
TU Berlin

January 1, 1980

About service: replme

About service: replme

- Clone of replit.com

About service: replme

- Clone of replit.com
- Provides ”DEVENVs” in browser

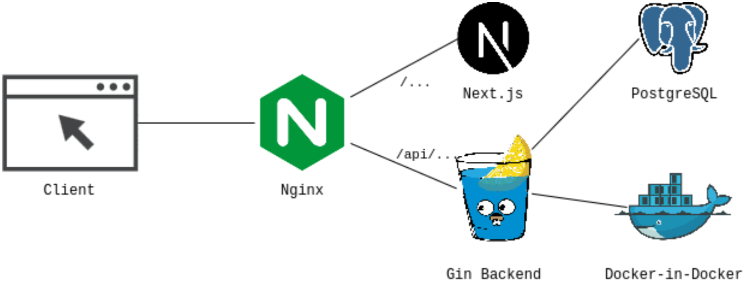
About service: replme

- Clone of replit.com
- Provides ”DEVENVs” in browser
- Provides ”REPLs” in browser

DEMO

Architecture

Architecture



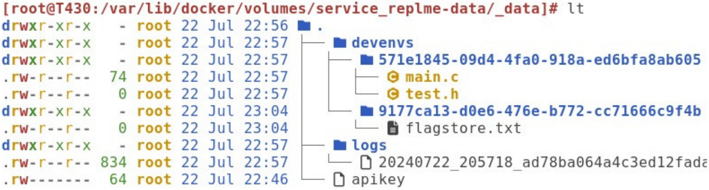
Vuln 1: Path traversal

Vuln 1: Path traversal

- Flagstore is file in devenv

Vuln 1: Path traversal

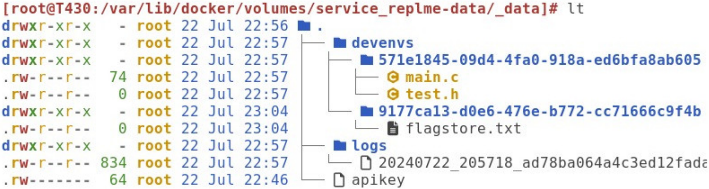
- Flagstore is file in devenv
- Devenv files are stored in FS (docker volume)



```
[root@T430:/var/lib/docker/volumes/service_replme-data/_data]# lt
drwxr-xr-x    - root 22 Jul 22:56 .
drwxr-xr-x    - root 22 Jul 22:57 ├── devenvs
drwxr-xr-x    - root 22 Jul 22:57 │   ├── 571e1845-09d4-4fa0-918a-ed6bfa8ab605
.rw-r--r--   74 root 22 Jul 22:57 │   │   ├── main.c
.rw-r--r--    0 root 22 Jul 22:57 │   │   └── test.h
drwxr-xr-x    - root 22 Jul 23:04 │   └── 9177ca13-d0e6-476e-b772-cc71666c9f4b
.rw-r--r--    0 root 22 Jul 23:04 │       └── flagstore.txt
drwxr-xr-x    - root 22 Jul 22:57 ├── logs
.rw-r--r--  834 root 22 Jul 22:57 │   └── 20240722_205718_ad78ba064a4c3ed12fada
.rw-------   64 root 22 Jul 22:46 └── apikey
```

Vuln 1: Path traversal

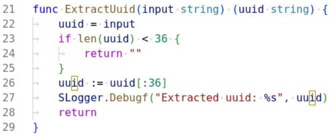
- Flagstore is file in devenv
- Devenv files are stored in FS (docker volume)



```
[root@T430:/var/lib/docker/volumes/service_replme-data/_data]# lt
drwxr-xr-x    - root 22 Jul 22:56 .
drwxr-xr-x    - root 22 Jul 22:57 ├── devenvs
drwxr-xr-x    - root 22 Jul 22:57 │  ├── 571e1845-09d4-4fa0-918a-ed6bfa8ab605
.rw-r--r--   74 root 22 Jul 22:57 │  │  ├── main.c
.rw-r--r--    0 root 22 Jul 22:57 │  │  └── test.h
drwxr-xr-x    - root 22 Jul 23:04 │  └── 9177ca13-d0e6-476e-b772-cc71666c9f4b
.rw-r--r--    0 root 22 Jul 23:04 │     └── flagstore.txt
drwxr-xr-x    - root 22 Jul 22:57 ├── logs
.rw-r--r--  834 root 22 Jul 22:57 │  └── 20240722_205718_ad78ba064a4c3ed12fada
.rw-------   64 root 22 Jul 22:46 └── apikey
```

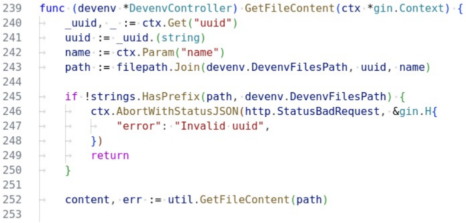
- /api/devenv/{571..}/files/flagstore.txt
  ?uuid={571..}%2F..%2F{917..}

Vuln 1: Path traversal



```go
func ExtractUuid(input string) (uuid string) {
	uuid = input
	if len(uuid) < 36 {
		return ""
	}
	uuid := uuid[:36]
	SLogger.Debugf("Extracted uuid: %s", uuid)
	return
}
```

Figure: service/backend/util/encoding.go



```go
func (devenv *DevenvController) GetFileContent(ctx *gin.Context) {
	_uuid, _ := ctx.Get("uuid")
	uuid := _uuid.(string)
	name := ctx.Param("name")
	path := filepath.Join(devenv.DevenvFilesPath, uuid, name)

	if !strings.HasPrefix(path, devenv.DevenvFilesPath) {
		ctx.AbortWithStatusJSON(http.StatusBadRequest, &gin.H{
			"error": "Invalid uuid",
		})
		return
	}

	content, err := util.GetFileContent(path)

```

Figure: service/backend/controller/devenv.go

Vuln 2: 2nd preimage

Vuln 2: 2nd preimage

- Flagstore is file in FS of REPL

Vuln 2: 2nd preimage

- Flagstore is file in FS of REPL
- Identifier of REPLs is CRC(username)

Vuln 2: 2nd preimage

- Flagstore is file in FS of REPL
- Identifier of REPLs is CRC(username)
- CRC is no cryptographically secure hash func

$$h(a) = a \texttt{ \% } p$$

Vuln 2: 2nd preimage

- Flagstore is file in FS of REPL
- Identifier of REPLs is CRC(username)
- CRC is no cryptographically secure hash func

$$h(a) = a \ \texttt{\%} \ p$$

- Calculate deltas, such that:
  CRC(username) == CRC(username+delta)

Vuln 2: 2nd preimage

- Flagstore is file in FS of REPL
- Identifier of REPLs is CRC(username)
- CRC is no cryptographically secure hash func

$$h(a) = a \;\texttt{\%}\; p$$

- Calculate deltas, such that:
  CRC(username) == CRC(username+delta)

$$\begin{aligned} h(a \oplus \Delta) &= (a \oplus b \cdot p) \;\texttt{\%}\; p \\ &= a \;\texttt{\%}\; p \end{aligned}$$

Vuln 3: RCE (Bonus)

Vuln 3: RCE (Bonus)

- Server on REPLs exposes register endpoint

Vuln 3: RCE (Bonus)

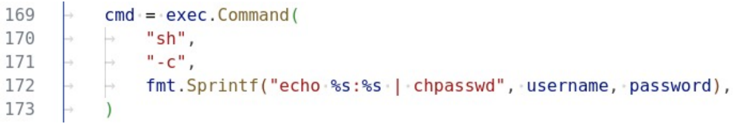
- Server on REPLs exposes register endpoint
- Endpoint secured by apikey
  http://{ip}:{port}/api/{apikey}/auth/register

Vuln 3: RCE (Bonus)

- Server on REPLs exposes register endpoint
- Endpoint secured by apikey
  http://{ip}:{port}/api/{apikey}/auth/register
- Password is not sanitized

Vuln 3: RCE (Bonus)

- Server on REPLs exposes register endpoint
- Endpoint secured by apikey
  http://{ip}:{port}/api/{apikey}/auth/register
- Password is not sanitized



```go
	cmd = exec.Command(
		"sh",
		"-c",
		fmt.Sprintf("echo %s:%s | chpasswd", username, password),
	)
```

Figure: service/image/service/user.go

DEMO

What worked

What worked

- Service stable

What worked

- Service stable
- SLA was suprisingly good

What worked

- Service stable
- SLA was suprisingly good
- People had fun

What did'nt work

What did’nt work

- Unintended vuln

What did’nt work

- Unintended vuln
- Performance issues due to strict timeout

What did’nt work

- Unintended vuln
- Performance issues due to strict timeout
- CORS ❤

What did’nt work

- Unintended vuln
- Performance issues due to strict timeout
- CORS ❤
- proxy.prod.bambi.ovh blacklisted

What did’nt work

- Unintended vuln
- Performance issues due to strict timeout
- CORS ❤
- proxy.prod.bambi.ovh blacklisted
- CRC unexploited

Feedback

Feedback

- ”the return bug was truly evil btw”

Feedback

- ”the return bug was truly evil btw”
- ”i wanted to do that, but i missed the crypto knowledge”

Feedback

- ”the return bug was truly evil btw”
- ”i wanted to do that, but i missed the crypto knowledge”
- ”exploitation wasn’t simple even with this unintended bug though, so it was a fun task”

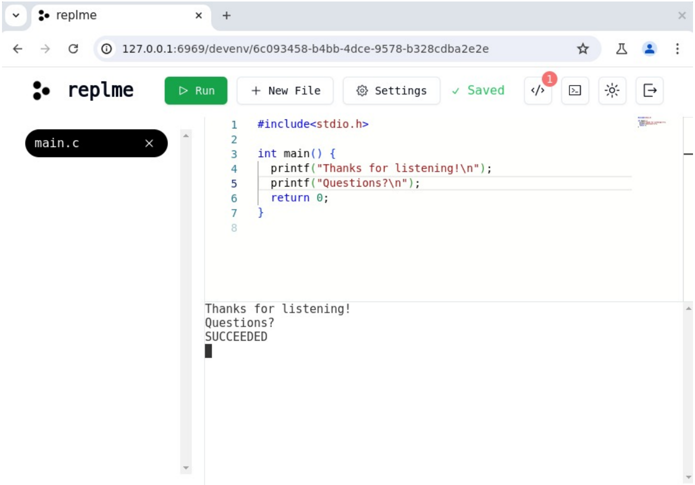
Lessons learned

Lessons learned

- Stay calm and take the time to think

Lessons learned

- Stay calm and take the time to think
- Do not get lost in details



```c
#include<stdio.h>

int main() {
  printf("Thanks for listening!\n");
  printf("Questions?\n");
  return 0;
}
```

```
Thanks for listening!
Questions?
SUCCEEDED
```