EXPLORING EXPLOITS TO ATTACK THE VULNERABILITY IN APT-GET

Dr. Bhawana Rudra
Faculty, Information Technology
National Institute of Technology, Karnataka
Surathkal, India
bhawanarudra@nitk.edu.in

Bavya Balakrishnan
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
bavyabalakrishnan.192it003@nitk.edu.in

Pritish Uday Naik
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
pritishnaik.192it010@nitk.edu.in

***Abstract*—We attempted to carry out Man in the middle attack using the exploit based on recently discovered vulnerability CVE-2019-3462 on APT/APT-GET. We implemented it with the help of 302 redirect HTTP transport method.**

***Index Terms*—MITM, Exploit, redirect, vulnerability, APT, APT-GET**

I. Introduction

Advanced Package Tool, or APT can be described as a free-software user interface to work with core libraries to handle various functionalities such as installation and removal of softwares on Debian and Linux distributions such as Ubuntu. It automates the process of retrieval, configuration and installation of software packages.

Max Justicz ,a security researcher discovered Remote Code Execution(RCE) vulnerability named as CVE-2019-3462 in Linux Apt / apt-get on 22 Jan 2019. Using http it is possible for an attacker to redirect the download link of an installation package to a malicious server. The attacker can exploit vulnerability via man-in-the-middle attack and thereby executing code with root privileges also causing denial of service.

Initially it was discovered in version 1.4.8 or earlier. Now in most of the apt versions available today this serious vulnerability has been fixed and patched up.

Debian and Ubuntu use plain http repositories by default. Switch to HTTPS as a default in debian and other distributions is difficult as it requires maintaining valid TLS certificates on the mirror network, plus it is costly and depends on caching in low-bandwith environments.

II. Literature review

A. How APT works

A main process which is interacting with user forks a child process. That child establishes connection with the online repository which contains data and makes the http get request to fetch data. Parent and helper use http for communication.

- Step 1: Parent process forks off /usr/lib/apt/methods/http
- Step 2: Helper responds with ”100 Capabilities” message once it is created. It then requests parent for the configuration details .
- Step 3: Parent process gives a reply to helper with the apt configuration details(”601 Configuration”)
- Step 4: Progress is then updated to parent process through messages such as ”102 Status” , ”200 URI Start” by child through regular communication with server.
- Step 5: The parent process sends a ”201 URI done response” once the packet is acquired.

B. HTTP Redirect Request and the Vulnerability

The http server responds with a 302 redirect message with the new URI location when the resource has been moved to a new URI location. Then it is the child process which constructs 103 redirect message with the new URI and sends it immediately to the parent process.

Unfortunately, the HTTP fetcher process decodes the HTTP location and blindly appends it to the 102 Redirect response. Thus,the vulnerability that we can exploit lies in the location

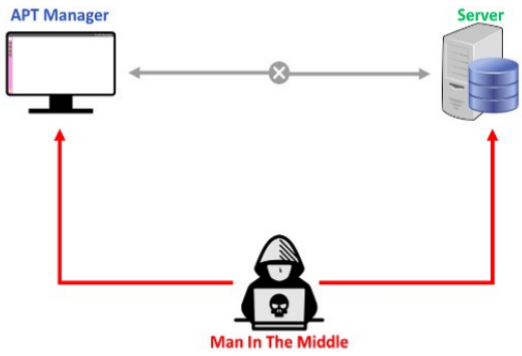

Fig. 1: Man in the middle attack

string. We can intercept the packets and modify the location string in the http redirect response after establishing a man in the middle attack between the victim and the gateway.

For example, attacker can append a ”200 URI done” message in the location string and trick parent process to think that package is surccessfully downloaded. Parent process doesn't do any recomputation on hash in URI done message. Whatever we specify in filename will be excuted with root privileges. That can be a malicious file given by attacker.

Control character allowed in location string is the threat here. Allowing the location string if it is free of such characters can help us fix this issue.

III. Methodology

A. ARP Poisoning

There are various tools to carry out Man in the middle attacks. One method is ARP poisoning. We send a ARP response to client saying that IP address of host belongs to our MAC address.Similarly, we send a ARP response to host saying that IP address of client belongs to our MAC address. That is how ARP poisoning is done between machines. ARP poisoning can be carried out with Ettercap or scapy.

First we have to enable Packet forwarding. We should not forget to enable IP forwarding when you want the system to transfer IP packets from one network to another ie to act as a router between networks.

Commands to do the settings are given below

- To get the current state of IP forwarding:
  cat /proc/sys/net/ipv4/ip_forward
- To enable IP forwarding on a Linux system:
  sudo sysctl -wnet.ipv4.ip_forward=1

1) Ettercap: It is a free open sourse network security software used in applications such as security auditing and computer network protocol analysis. We can use Unified sniffing to sniff the packets on a particular interface that we have chosen. Host scan will list all the hosts in the network with their IP addresses and corresponding MAC address. Among them we choose the targets and lauch ARP poisoning over them. Then ettercap will send arp-responses to launch the attack.

Once the mitm is established we can capture and modify the packet with the help of script in ettercap.Since we found the Ettercap is unstable in listing hosts in the network we started to use Scapy which is comparatively more stable and programmable.

2) Scapy: **Scapy** is a tool written in Python for packet manipulation in computer networks. It can perform the functionalities such as forging or decoding packets, sending or capturing them on the wire and match requests and replies. Tasks like scanning, tracerouting, probing, unit tests, attacks, and network discovery can also be handled using Scapy.

Initially we should enable the packet forwarding in the machines. Run the following comments to get the status.

- To find the connected interface **netstat -i**
- To list the ip addresses of gateway and other machines in the network. **arp -a**

Among the the clients we chose one as target. We sent ARP requests using scapy to get the mac address of victim and gateway. Now, until the victim machine and gateway are poisoned malicious ARP responses are sent to them. Once the mitm is established using tcpdump we can see all the packets transmitting from victim to gateway **tcpdump -i eth0 -n port 80 and host victimip**

B. Packet Interception and Manipulation

We can sniff the forwarded packets once the mitm is established. We make use of **Netfilterqueue** for modifying sniffed packets. Default firewalls present in Linux distributions are the Iptables. They contain Tables with chains that contains various rules. Those rules are applicable for packet filtering and Network address translation. There are 5 chains in total. We focus on Forward chain.

- INPUT CHAIN: The rules mentioned in this are applicable to the traffic/packets coming towards the server.
- OUTPUT CHAIN: The rules mentioned in this are applicable on outgoing traffic/packets from our server.
- FORWARD CHAIN: Rules related to forwarding an IP packet can be added here.

```
// From methods/basehttp.cc
NextURI = DeQuoteString(Req.Location);
...
Redirect(NextURI);

// From apt-pkg/acquire-method.cc
void pkgAcqMethod::Redirect(const string &NewURI)
{
   std::cout << "103 Redirect\nURI: " << Queue->Uri << "\n"
             << "New-URI: " << NewURI << "\n"
             << "\n" << std::flush;
   Dequeue();
}
```

Fig. 2: HTTP fetcher process decoding HTTP Location

```
HTTP/1.1 302 Found
Location:/payload%0A%0A201%20URI%20Done%0AURI%3A%20http%3Anew-uri%0A
Filename%3A%20xxxxx%0ASize%3A%yyyyy%0ALast-Modified%3A%20xxxxx%0A
MD5-Hash%3A%20yyyyy%0A
```

```
103 Redirect
URI: http://deb.debian.org/debian/<old-uri>
New-URI: http://deb.debian.org/payload


201 URI Done
URI: new-uri
Filename: xxxxx
Size: yyyyy
Last-Modified: xxxxx
MD5-Hash: yyyyy
```

Fig. 3: Sample packet content for attack

- PRE-ROUTING CHAIN: It has the rules to define actions that has to be executed before a routing decision is made by the kernel.
- POST-ROUTING CHAIN: It has the rules to define actions that need to be taken after a routing decision is made by the kernel.

We specify a rule in the Forward chain to add the packets into netfilterqueue. These are packets routed/forwarded through our system but not meant for local delivery.

iptablesr = ”iptables -A FORWARD -j NFQUEUE”

Using Python script we access this queue and thus the packets in it. Our goal is to modify the TCP payload . For that we convert this packet to scapy packets through scripts. We then access the different layers such as IP,TCP and Raw in the packet and modify the content. Using python script we accept those modified packets in Netfilterqueue.

With the methods mentioned above we can only modify http packets and the thing is apt uses http by default.

C. Modifying the packet to launch attack

Redirect response from the server is causing the Vulnerability in apt. We capture all the http responses with 200 OK and modify them to 302 redirect response. Inorder to identify TCP packets with payload 200 OK we match it with our reference packet. If we get a match replace the payload of TCP package malicious content as given in fig:3. We get the correct hash values from /var/lib/apt/lists. To get the correct format of 200 URI Done message by the child process we can run apt with command “strace -f”. In the output file we will get the format.

We keep a malicious packet in location/URI mentioned in manipulated packet. We make netfilterqueue accept this packet.

Later when parent process send a request to this malicious URL, attack occures and code is excuted (packet is downloaded) with root access. Even if this failes attempt can cause denial of service on victim machine.

IV. Results

A. Attacking apt version 1.6.11

Since the vulnerability CVE-2019-3462 is fixed in this version it is no more susceptible to the this exploit. Even though, the attack carried out on this version gave a security warning ”SECURITY: URL redirect target contains control characters, rejecting.”. This actually confirmed that the packet is indeed manipulated and we were able to launch Denial of service attack. The manipulated packet payload is given in fig:4 and the security warning with abort in apt version 1.6.11 in fig:5.

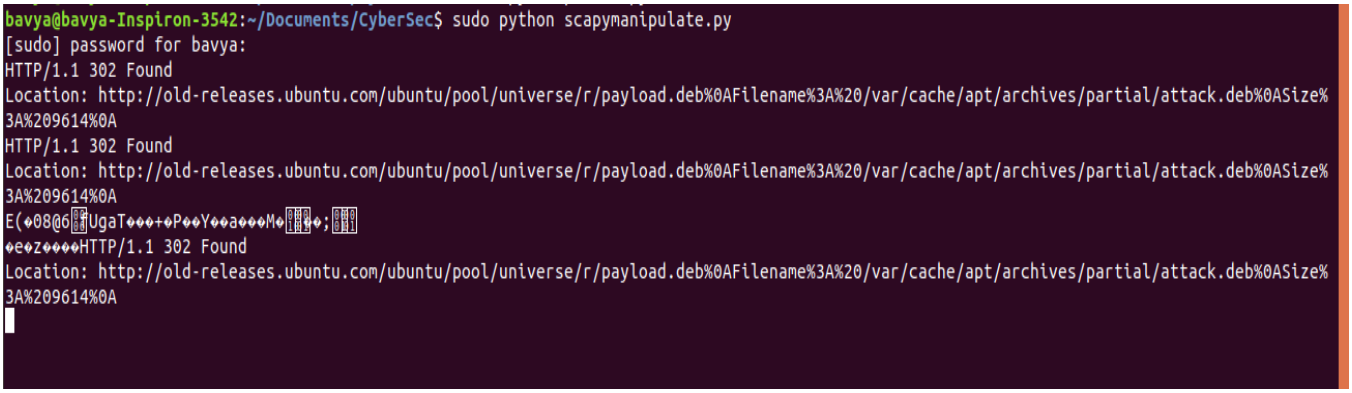

Fig. 4: Modified packet payload

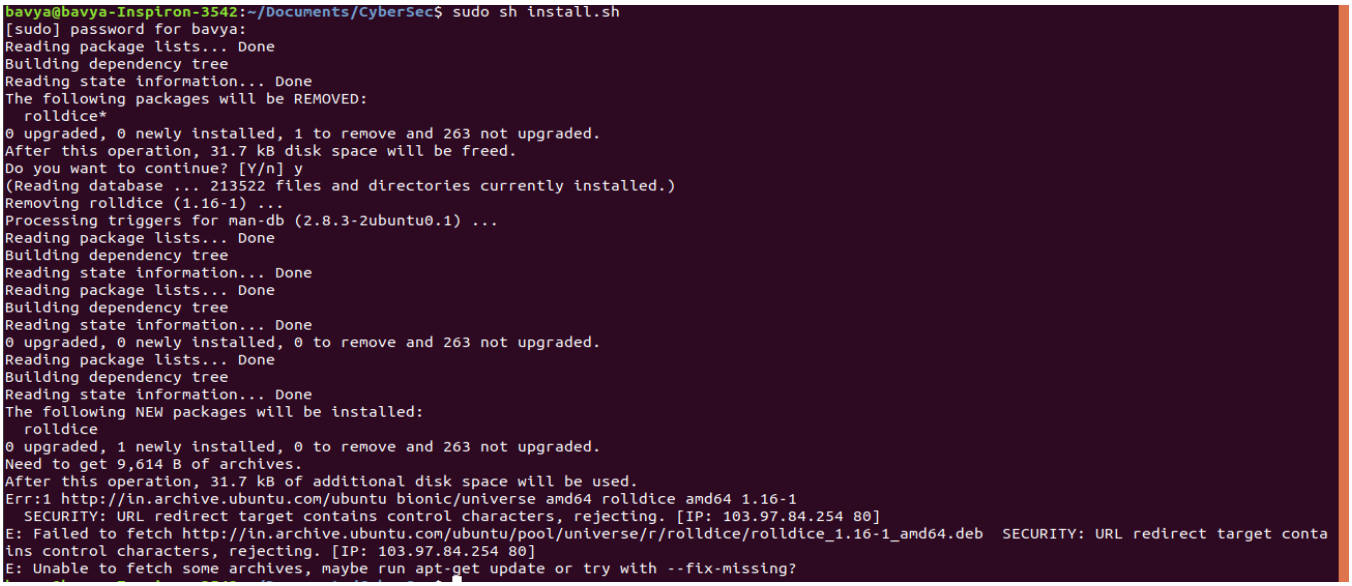

Fig. 5: Denial of Service attack on apt version 1.6.11

B. Attack on apt 1.4 - The vulnerable version

1) Setting up the vulnerable environment: We tried to pull the git repository of apt version 1.4.2 which doesn't have the security update for CVE-2019-3462. But we were not successful in compiling that version of apt. Since the exploit was published in 2019 we looked for an Ubuntu iso file, whose support has reached end of life so that it will contain the vulnerable version of apt. We used the version 17.04 for Ubuntu which was last modified in 2018 for carrying out the attack. This ubuntu version has the vulnerable version of apt (1.4) and that was confirmed when it did not produce any security warning as in version 1.6.11.

2) Attack on apt 1.4: We captured the packets matching "200 OK" and modified them to be HTTP redirect. This triggered a "404 Not found" fig:6.error message as the process tried to fetch the package from new URI mentioned in 103 redirect message.

V. Remedy

Ubuntu and Debian released patches for this vulnerability. It advises users to upgrade to the latest APT version for the system protection and by using below commands users can disable HTTP redirects during the upgrade.

sudo apt update –o Acquire::http::AllowRedirect=false
sudo apt upgrade –o Acquire::http::AllowRedirect=false

VI. Conclusion

We are trying to implement a known vulnerability(CVE-2019-3462) in apt/apt-get. After successfully launching the man in the middle attack we do packet manipulation . The attack resulted in Denial of service in apt version 1.6.11 where this serious vulnerability has been fixed and patched up. We also tested the same on vulnerable version and we could successfully modify the packets to redirect the process to new URI.

Switch to https can actually avoid this kind of attacks but the apt's only intention is to protect the integrity of

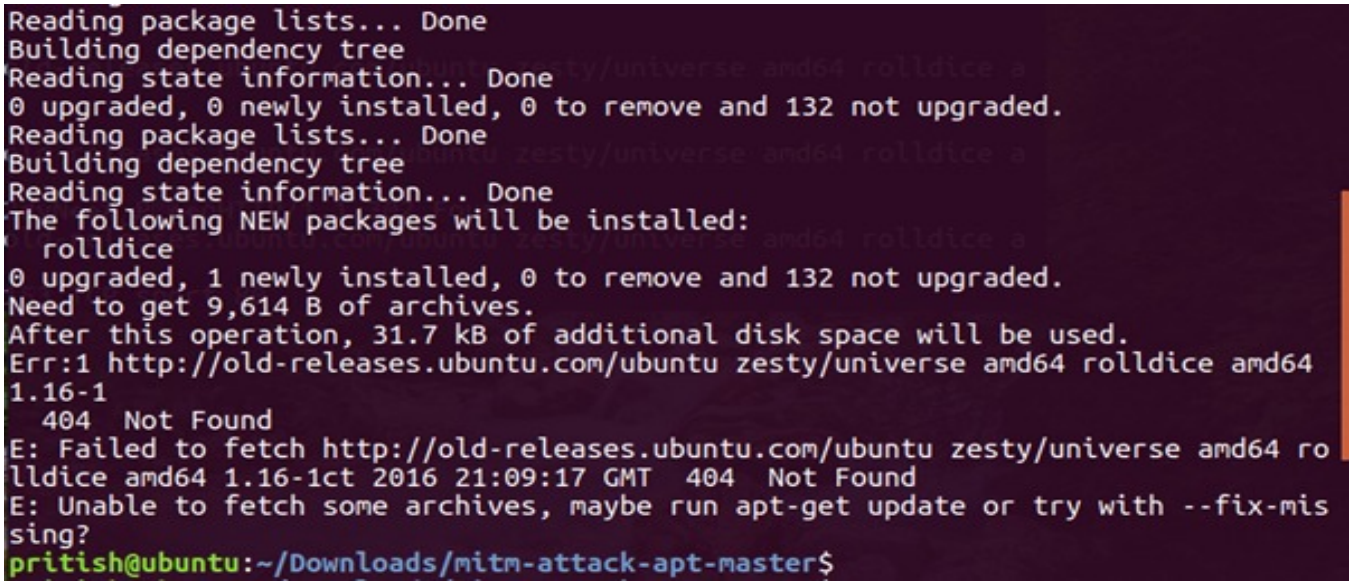

Fig. 6: Attack on apt 1.4 - The vulnerable version

packages getting installed. This is because open source codes are available for most of such packages. Although we can confgure our apt to use https to prevent this types of attacks.

REFERENCES

[1] Akash, A., Yadnesh, Salvi.: Man-in-the middle attack on APT/APT-GET
[2] Mina Hao: APT/APT-GET RCE Vulnerability (CVE-2019-3462) Handling Guide
[3] https://lists.debian.org/debian-security-announce/2019/msg00010.html