Linux commands and utilities for security testing

By Swapnil

Cat

cat - concatenate files and print on the standard output

Cat Usage

- **Display Contents of a File**
  `cat test1.txt`
- **Redirect Contents of a File**
  `cat test1.txt > test3.txt`
- **To display content of all txt files**
  `cat *.txt`
- **To display the contents of a file with line number**
  `cat -n file1.txt`

Find

Find command basically finds the things for you

find Usage

- **Find files in a directory**

  `find /`
- **Specific files in a directory**

  `find ~ -name ‘*.jpg’`
- **”OR”**

  `find  ~ ( -iname 'jpeg' -o -iname 'jpg' )`
- **Find world-readble files**

  `find ~ -perm -o=r`

PARALLEL

Parallel is a shell utility for executing jobs in parallel

parallel Usage

- **From serial to parallel**

```
find . -name "*jpeg" | parallel -I% --max-args 1 convert % %.png
```

- **Multiple Inputs**

```
ls -l | parallel --max-args=2 echo
```

Cut

cut is a command-line utility that allows you to cut parts of lines from specified files or piped data and print the result to standard

Cut Usage

- **Specify a field**
  `Cut -f`
- **Bytes**
  `Cut -b`
- **Characters list**
  `Cut -c`
- **Delimiter**
  `Cut -d`

SORT

Sort sorts its input

sort Usage

- **Numeric sort**
  `Sort -n`
- **Human sort**
  `Sort -h`
- **Uniq values**
  `Sort -u`

AWK

Awk is a general-purpose scripting language designed for advanced text processing.

AWK Usage

- **AWK patterns**

  ```
  Awk ‘{print $ 3}’ test.txt
  ```

- **Awk regex**

  ```
  Awk ‘/reg/ {print $4}’ test.txt
  ```

- **AWK field separator**

  ```
  Awk ‘BEGIN {FS = “.”}{ print $1}’ test.txt
  ```

Echo

echo is one of the most commonly and widely used built-in command for Linux bash and C shells, that typically used in scripting language and batch files to display a line of text/string on standard output or a file.

Echo Usage

- **Display a line of text on standard output**

  ```
  Echo Hello world
  ```
- **Pattern matching characters**

  ```
  echo The PHP files are: *.php
  ```
- **Redirect to a file**

  ```
  echo -e 'The test file' >> /tmp/file.txt
  ```
- **Displaying output of a command**

  ```
  echo "The date is: $(date +%D)"
  ```

Some more command

- **Reverse command**

  `rev`
- **Grep command**

  `Grep -r`
- **SED - edit the input stream**

  `Sed -n 1-4p`
- **Delimiter**

  `Cut -d`

Lets make cocktail of above commands

Processing data for Recon

- **Get javascript files from domains list**

  ```
  Cat domains list | gau | grep “.js”
  ```

- **Get v1 api enpoints from URL list**

  ```
  printf yahoo.com | gau | grep -w "v1" | head -10
  ```

- **Find URL with admin keyword in it**

  ```
  Cat domains.txt | grep “admin”
  ```

- **With staus code 200**

  ```
  cat domains.txt| gau | hakcheckurl | grep -w '200' | head -10
  ```

- **Extract subdomains from output**

  ```
  gau -subs example.com | cut -d / -f 3 | sort -u
  ```

-

- **Pull Root Subdomains from Final.txt**

```
cat final | rev | cut -d . -f 1-3 | rev | sort -u | tee root.subdomains
```

- **Extract URLs from junk data**

```
cat file | grep -Eo "(http|https)://[a-zA-Z0-9./?=_-]*"*
```

-

Some bonus commands

- **Command injection to File inclusion**
  ```
  echo "<?php include($ GET['page'])| ?>" > rfi.php
  ```
- **Command Injection bypass**
  ```
  Cat /etc/passwd
  Cat /e”t”c/pass”w”d
  Cat /etc/pass*d
  ```
- **Echo and rev**
  ```
  Echo “dwssap/cte/ tac” | rev
  ```
- **AWK and shell**
  ```
  awk 'BEGIN {system("/bin/sh")}'
  ```
- **Find and AWK**
  ```
  find / -name blahblah -exec /bin/awk 'BEGIN {system("/bin/sh")}' \;
  ```
- **Echo and tee**
  ```
  echo "evil script code" | tee script.sh
  ```

Thank you