

# Bash Script

De Silva K.R.K.D

# Table of Contents

<b>Introduction To The Topic .....</b>	<b>3</b>
<b>Methodology .....</b>	<b>4</b>
<b>Script .....</b>	<b>4</b>
<b>Nmap .....</b>	<b>7</b>
<b>Nslookup .....</b>	<b>9</b>
<b>Nikto .....</b>	<b>10</b>
<b>Sqlmap .....</b>	<b>11</b>
<b>Skipfish.....</b>	<b>11</b>
<b>Conclusion .....</b>	<b>12</b>
<b>References .....</b>	<b>12</b>

## Introduction To The Topic

Bash scripting is a flexible and essential tool in the vast world of programming and automation. The terminal shell and scripting syntax known as Bash or “Bounty again shell,” is firmly established in Unix and Linux-like operating systems. In essence, Bash scripts are collections of commands written in the Bash programming language that are performed in a predetermined order in order to automate processes, manage files, change data, and communicate with the underlying operating system that is underlying. Bash scripts offer a potent route for boosting productivity and mastering computing environments, regardless of whether you play the part of a careful system manager performing routine upkeep, a software developer coordinating the installation of intricate programs, or a curious beginner looking to delve into the fascinating field of scripting.

“A bash script is a simple text file with a set of commands inside of it. These commands are a combination of ones we typically type on a terminal by ourselves (like `ls` or `cp`, for instance) and others that we might use on the task line but typically wouldn’t (you’ll learn about these over the following pages.)”[1]

In a wide range of industries and professions, bash scripting is of utmost importance. First off, Bash scripts are essential tools for people in charge of managing computer systems in the field of system administration. Second, Bash scripts are crucial to the lifecycle of software development in the field of software development. These scripts are used by developers to quickly create, verify, and deploy apps. Thirdly, Bash scripts are excellent at handling and analyzing data when it involves data manipulation and analysis. Additionally, Bash programming expands into the customization space, giving users the flexibility to tailor their computer settings. Finally, Bash scripting offers a simple way for students to begin learning about programming and automation.

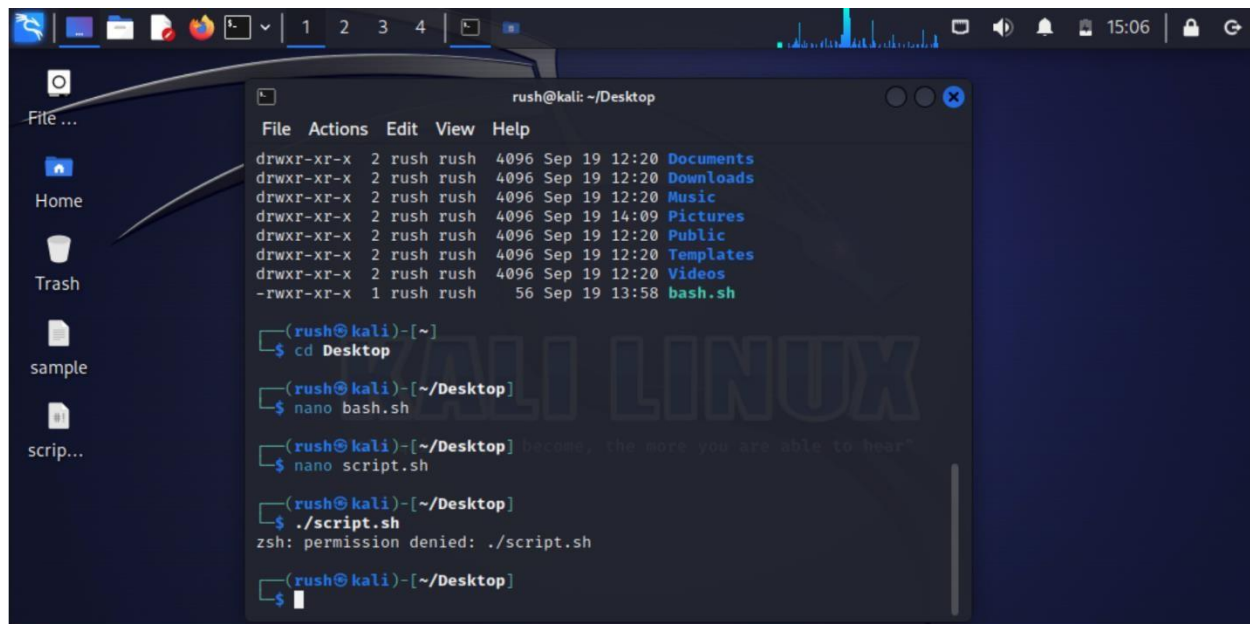
## Methodology

One can carefully go through a number of steps to scan an Ip address or URL by applying a Bash script. The script asks the user to provide the IP address or URL to be scanned at the beginning, enabling dynamic target selection. After that, it's critical to carry out input validation to make sure that the supplied input follows the anticipated format and satisfies any requirements for the scan, increasing the script's dependability. The script should make a special directory to methodically keep track of the results. This step of creating a directory creates a tidy and organized environment in which to store the scan findings and associated files.

The script can use a network scan tool like Nmap to do net-related scans, like identifying open ports. The outcomes are saved in records within the earliercreated directory. When a DNS lookup is necessary, it provides information about the desired hosts and IP addresses, enhancing the data from the overall scan. The script can also include any extra scans or tests that are necessary to achieve the objectives, such as pings tests, requests via HTTP, or other pertinent actions.

# Script

First, you need to navigate the desktop using “cd” command. Next, create the text editor using “nano” command

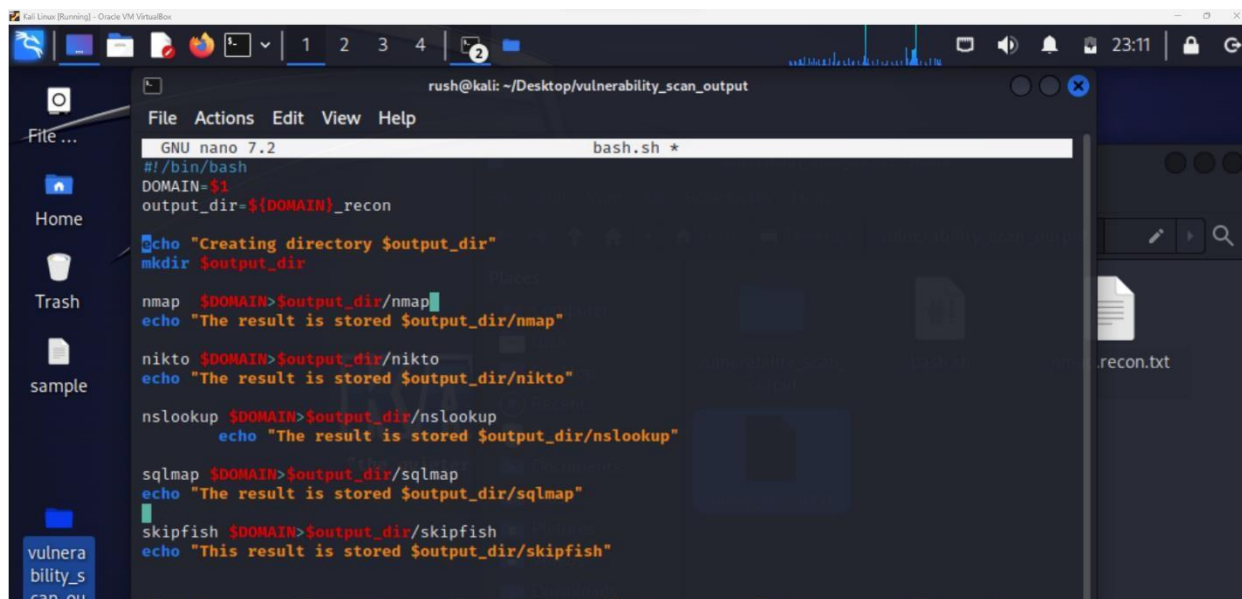


The screenshot shows a Kali Linux desktop environment. On the left is a file manager sidebar with icons for Home, Trash, sample, and scrip... The main window is a terminal titled 'rush@kali: ~/Desktop'. It displays the following commands and output:

```
File Actions Edit View Help
drwxr-xr-x 2 rush rush 4096 Sep 19 12:20 Documents
drwxr-xr-x 2 rush rush 4096 Sep 19 12:20 Downloads
drwxr-xr-x 2 rush rush 4096 Sep 19 12:20 Music
drwxr-xr-x 2 rush rush 4096 Sep 19 14:09 Pictures
drwxr-xr-x 2 rush rush 4096 Sep 19 12:20 Public
drwxr-xr-x 2 rush rush 4096 Sep 19 12:20 Templates
drwxr-xr-x 2 rush rush 4096 Sep 19 12:20 Videos
-rwxr-xr-x 1 rush rush 56 Sep 19 13:58 bash.sh

(rush@kali)-[~]
$ cd Desktop
(rush@kali)-[~/Desktop]
$ nano bash.sh
(rush@kali)-[~/Desktop]
$ nano script.sh
(rush@kali)-[~/Desktop]
$ ./script.sh
zsh: permission denied: ./script.sh
(rush@kali)-[~/Desktop]
$
```

Go to that file and write the bash code into that file. The whole line should be the initial line of each bash script you create. It states which translator to use for the script at the beginning with a hash(#) and (!) mark. This makes it possible to run a text file like binary. It will serve as a reminder that we use bash. In this code, you need to assign a value using the (\$) sign. Create an output directory file to save the output using “mkdir”. I used 5 tools to scan IP addresses or URLs. They are, “Nmap”, “Nikto”, “nslookup”, “sqlmap” and “skipfish”.



```
rush@kali: ~/Desktop/vulnerability_scan_output
GNU nano 7.2 bash.sh *
#!/bin/bash
DOMAIN=$1
output_dir=${DOMAIN}_recon

echo "Creating directory $output_dir"
mkdir $output_dir

nmap $DOMAIN>$output_dir/nmap
echo "The result is stored $output_dir/nmap"

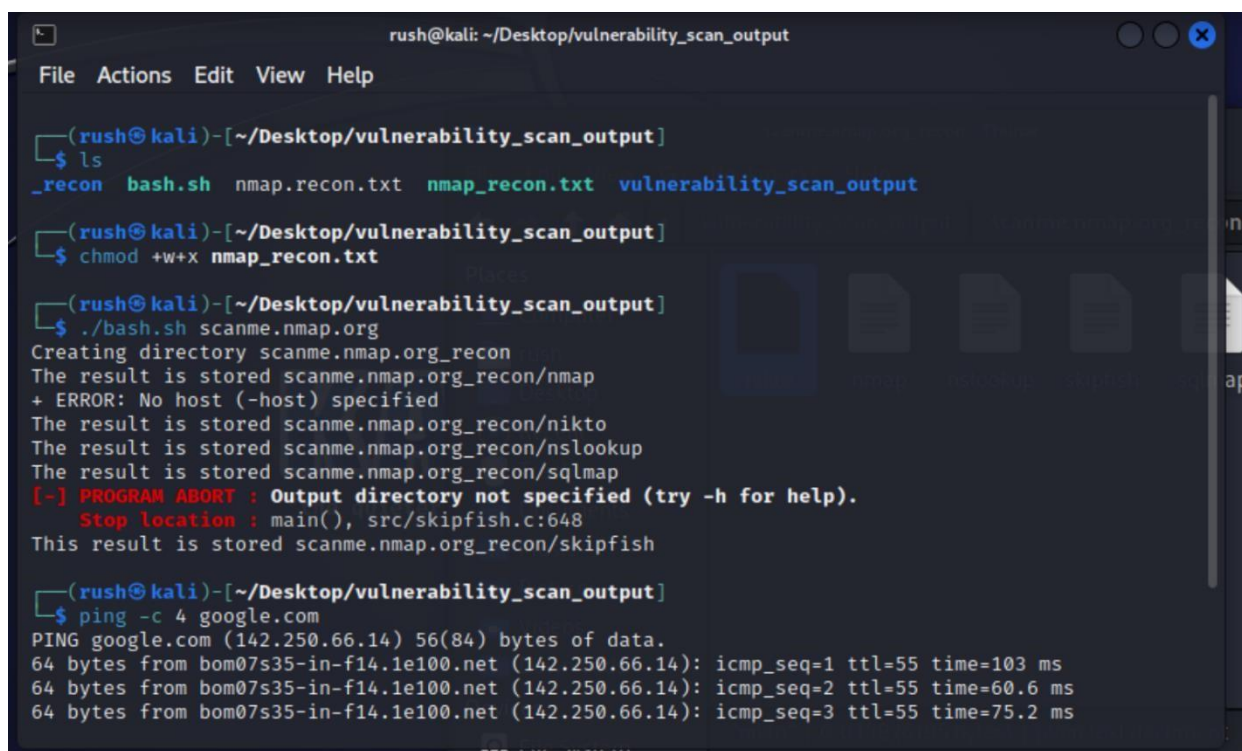
nikto $DOMAIN>$output_dir/nikto
echo "The result is stored $output_dir/nikto"

nslookup $DOMAIN>$output_dir/nslookup
echo "The result is stored $output_dir/nslookup"

sqlmap $DOMAIN>$output_dir/sqlmap
echo "The result is stored $output_dir/sqlmap"

skipfish $DOMAIN>$output_dir/skipfish
echo "This result is stored $output_dir/skipfish"
```

After saving the command file we want to execute, we can run it after giving “chmod”. Now want to connect wo the google using “ping” command.



```
rush@kali: ~/Desktop/vulnerability_scan_output
File Actions Edit View Help

(rush@kali)-[~/Desktop/vulnerability_scan_output]
$ ls
_recon  bash.sh  nmap.recon.txt  nmap_recon.txt  vulnerability_scan_output

(rush@kali)-[~/Desktop/vulnerability_scan_output]
$ chmod +w+x nmap_recon.txt

(rush@kali)-[~/Desktop/vulnerability_scan_output]
$ ./bash.sh scanme.nmap.org
Creating directory scanme.nmap.org_recon
The result is stored scanme.nmap.org_recon/nmap
+ ERROR: No host (-host) specified
The result is stored scanme.nmap.org_recon/nikto
The result is stored scanme.nmap.org_recon/nslookup
The result is stored scanme.nmap.org_recon/sqlmap
[-] PROGRAM ABORT : Output directory not specified (try -h for help).
Stop location : main(), src/skipfish.c:648
This result is stored scanme.nmap.org_recon/skipfish

(rush@kali)-[~/Desktop/vulnerability_scan_output]
$ ping -c 4 google.com
PING google.com (142.250.66.14) 56(84) bytes of data.
64 bytes from bom07s35-in-f14.1e100.net (142.250.66.14): icmp_seq=1 ttl=55 time=103 ms
64 bytes from bom07s35-in-f14.1e100.net (142.250.66.14): icmp_seq=2 ttl=55 time=60.6 ms
64 bytes from bom07s35-in-f14.1e100.net (142.250.66.14): icmp_seq=3 ttl=55 time=75.2 ms
```

# Nmap

Run the nmap tool within IP address or URL.

```
rush@kali: ~/Desktop/vulnerability_scan_output
File Actions Edit View Help
(rush@kali)-[~/Desktop/vulnerability_scan_output]
$ nmap -sV 35.208.78.12
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-19 23:29 +0530
Nmap scan report for 12.78.208.35.bc.googleusercontent.com (35.208.78.12)
Host is up (0.31s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Pure-FTPd
25/tcp    open  smtp?
80/tcp    open  http         nginx
110/tcp   open  pop3         Dovecot pop3d
143/tcp   open  imap         Dovecot imapd
443/tcp   open  ssl/http     nginx
465/tcp   open  ssl/smtp
587/tcp   open  smtp
993/tcp   open  ssl/imap     Dovecot imapd
995/tcp   open  ssl/pop3     Dovecot pop3d
3306/tcp  open  mysql        MySQL (unauthorized)
5432/tcp  open  postgresql   PostgreSQL DB 14.1 - 14.5
3 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port25-TCP:V=7.94%I=7%D=9/19%Time=6509E1B6%P=x86_64-pc-linux-gnu%r(NULL
SF:,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(Hello,22
SF:,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(Help,22,"42
SF:1\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(GenericLines,22
```

```
rush@kali: ~/Desktop/vulnerability_scan_output
File Actions Edit View Help
SF:ns,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(RTSPRE
SF:quest,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(RPC
SF:Check,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(DNS
SF:VersionBindReqTCP,22,"421\x204\2\1\x20please\x20try\x20again\x20later
SF:\r\n")%r(DNSStatusRequestTCP,22,"421\x204\2\1\x20please\x20try\x20aga
SF:in\x20later\r\n")%r(SSLSessionReq,22,"421\x204\2\1\x20please\x20try\x
SF:20again\x20later\r\n")%r(TerminalServerCookie,22,"421\x204\2\1\x20ple
SF:ase\x20try\x20again\x20later\r\n")%r(TLSSessionReq,22,"421\x204\2\1\x
SF:20please\x20try\x20again\x20later\r\n")%r(Kerberos,22,"421\x204\2\1\x
SF:20please\x20try\x20again\x20later\r\n")%r(SMBProgNeg,22,"421\x204\2\1
SF:\x20please\x20try\x20again\x20later\r\n")%r(FourOhFourRequest,22,"421\x
SF:204\2\1\x20please\x20try\x20again\x20later\r\n")%r(LPDString,22,"421\
SF:x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(LDAPSearchReq,22,
SF:"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(LDAPBindReq,
SF:22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(SIPOption
SF:s,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(LANDesk
SF:-RC,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(Termi
SF:nalServer,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r
SF:(NCP,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(Note
SF:sRPC,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(WMSR
SF:equest,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(or
SF:acle-tns,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(
SF:afp,22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n")%r(giop,
SF:22,"421\x204\2\1\x20please\x20try\x20again\x20later\r\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port465-TCP:V=7.94%T=SSL%I=7%D=9/19%Time=6509E1C4%P=x86_64-pc-linux-gnu
```



```
rush@kali: ~/Desktop/vulnerability_scan_output

File Actions Edit View Help

SF:L,A5,"220-c104313\sgvps\.\net\x20ESMTP\x20#2\x20Tue,\x2019\x20Sep\x2020
SF:23\x2018:16:03\x20\+0000\x20\r\n220-We\x20do\x20not\x20authorize\x20the
SF:\x20use\x20of\x20this\x20system\x20to\x20transport\x20unsolicited,\x20\
SF:r\n220\x20and/or\x20bulk\x20e-mail\.\r\n")%r(GenericLines,D9,"220-c1043
SF:13\sgvps\.\net\x20ESMTP\x20#2\x20Tue,\x2019\x20Sep\x202023\x2018:16:03\
SF:x20\+0000\x20\r\n220-We\x20do\x20not\x20authorize\x20the\x20use\x20of\x20
SF:20this\x20system\x20to\x20transport\x20unsolicited,\x20\r\n220\x20and/o
SF:r\x20bulk\x20e-mail\.\r\n500\x20unrecognized\x20command\r\n500\x20unrec
SF:ognized\x20command\r\n")%r(Hello,12B,"220-c104313\sgvps\.\net\x20ESMTP\
SF:x20#2\x20Tue,\x2019\x20Sep\x202023\x2018:16:14\x20\+0000\x20\r\n220-We\
SF:x20do\x20not\x20authorize\x20the\x20use\x20of\x20this\x20system\x20to\x20
SF:20transport\x20unsolicited,\x20\r\n220\x20and/or\x20bulk\x20e-mail\.\r\
SF:n550-You\x20can\x20not\x20sent\x20a\x20blank\x20HELO/EHLO\x20request\.\
SF:x20123\.\231\.\108\.\143\x20sent\x20[\]]\r\n550\x20EHLO/HELO\x20greeting\
SF:\x20Please\x20see\x20RFC\x202821\x20section\x204\.\1\.\1\.\1\r\n")%r(Help,
SF:108,"220-c104313\sgvps\.\net\x20ESMTP\x20#2\x20Tue,\x2019\x20Sep\x20202
SF:3\x2018:16:15\x20\+0000\x20\r\n220-We\x20do\x20not\x20authorize\x20the\
SF:x20use\x20of\x20this\x20system\x20to\x20transport\x20unsolicited,\x20\r
SF:\n220\x20and/or\x20bulk\x20e-mail\.\r\n214-Commands\x20supported:\r\n21
SF:4\x20AUTH\x20STARTTLS\x20HELO\x20EHLO\x20MAIL\x20RCPT\x20DATA\x20BDAT\x20
SF:20NOOP\x20QUIT\x20RSET\x20HELP\x20VRFY\r\n")%r(GetRequest,D9,"220-c1043
SF:13\sgvps\.\net\x20ESMTP\x20#2\x20Tue,\x2019\x20Sep\x202023\x2018:16:22\
SF:x20\+0000\x20\r\n220-We\x20do\x20not\x20authorize\x20the\x20use\x20of\x20
SF:20this\x20system\x20to\x20transport\x20unsolicited,\x20\r\n220\x20and/o
SF:r\x20bulk\x20e-mail\.\r\n500\x20unrecognized\x20command\r\n500\x20unrec
SF:ognized\x20command\r\n");
```

```
rush@kali: ~/Desktop/vulnerability_scan_output

File Actions Edit View Help

SF:20this\x20system\x20to\x20transport\x20unsolicited,\x20\r\n220\x20and/o
SF:r\x20bulk\x20e-mail\.\r\n500\x20unrecognized\x20command\r\n500\x20unrec
SF:ognized\x20command\r\n")%r(Hello,12B,"220-c104313\sgvps\.\net\x20ESMTP\
SF:x20#2\x20Tue,\x2019\x20Sep\x202023\x2018:16:14\x20\+0000\x20\r\n220-We\
SF:x20do\x20not\x20authorize\x20the\x20use\x20of\x20this\x20system\x20to\x20
SF:20transport\x20unsolicited,\x20\r\n220\x20and/or\x20bulk\x20e-mail\.\r\
SF:n550-You\x20can\x20not\x20sent\x20a\x20blank\x20HELO/EHLO\x20request\.\
SF:x20123\.\231\.\108\.\143\x20sent\x20[\]]\r\n550\x20EHLO/HELO\x20greeting\
SF:\x20Please\x20see\x20RFC\x202821\x20section\x204\.\1\.\1\.\1\r\n")%r(Help,
SF:108,"220-c104313\sgvps\.\net\x20ESMTP\x20#2\x20Tue,\x2019\x20Sep\x20202
SF:3\x2018:16:15\x20\+0000\x20\r\n220-We\x20do\x20not\x20authorize\x20the\
SF:x20use\x20of\x20this\x20system\x20to\x20transport\x20unsolicited,\x20\r
SF:\n220\x20and/or\x20bulk\x20e-mail\.\r\n214-Commands\x20supported:\r\n21
SF:4\x20AUTH\x20STARTTLS\x20HELO\x20EHLO\x20MAIL\x20RCPT\x20DATA\x20BDAT\x20
SF:20NOOP\x20QUIT\x20RSET\x20HELP\x20VRFY\r\n")%r(GetRequest,D9,"220-c1043
SF:13\sgvps\.\net\x20ESMTP\x20#2\x20Tue,\x2019\x20Sep\x202023\x2018:16:22\
SF:x20\+0000\x20\r\n220-We\x20do\x20not\x20authorize\x20the\x20use\x20of\x20
SF:20this\x20system\x20to\x20transport\x20unsolicited,\x20\r\n220\x20and/o
SF:r\x20bulk\x20e-mail\.\r\n500\x20unrecognized\x20command\r\n500\x20unrec
SF:ognized\x20command\r\n");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 68.89 seconds

(rush@kali)-[~/Desktop/vulnerability_scan_output]
$
```







## SqlMap

This is the sqlmap output

```
rush@kali: ~/Desktop/vulnerability_scan_output
File Actions Edit View Help
(rush@kali)-[~/Desktop/vulnerability_scan_output]
$ sqlmap -u "www.google.com" --data --dbms

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the
end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability
and are not responsible for any misuse or damage caused by this program. https://sqlmap.org

[*] starting @ 00:10:49 /2023-09-20/

[00:10:50] [INFO] testing connection to the target URL
[00:10:50] [WARNING] the web server responded with an HTTP error code (405) which could interfere with the resu
lts of the tests
[00:10:50] [INFO] checking if the target is protected by some kind of WAF/IPS
[00:10:51] [WARNING] reflective value(s) found and filtering out
[00:10:51] [INFO] testing if the target URL content is stable
[00:10:51] [INFO] target URL content is stable
[00:10:51] [CRITICAL] no parameter(s) found for testing in the provided data (e.g. GET parameter 'id' in 'www.s
ite.com/index.php?id=1')
[00:10:51] [WARNING] HTTP error codes detected during run:
405 (Method Not Allowed) - 3 times

[*] ending @ 00:10:51 /2023-09-20/
```

## Skipfish

This is the skipfish output

```
rush@kali: ~/Desktop/vulnerability_scan_output
File Actions Edit View Help
skipfish version 2.10b by lcamtuf@google.com
- 36.120.27.48 -

Scan statistics:
  Scan time : 0:00:42.949
  HTTP requests : 1 (0.0/s), 0 kB in, 0 kB out (0.0 kB/s)
  Compression : 0 kB in, 0 kB out (0.0% gain)
  HTTP faults : 1 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 1 total (1.0 req/conn)
  TCP faults : 0 failures, 1 timeouts, 0 purged
  External links : 0 skipped
  Reqs pending : 0

Database statistics:
  Pivots : 2 total, 2 done (100.00%)
  In progress : 0 pending, 0 init, 0 attacks, 0 dict
  Missing nodes : 0 spotted
  Node types : 1 serv, 1 dir, 0 file, 0 pinfo, 0 unkn, 0 par, 0 val
  Issues found : 0 info, 1 warn, 0 low, 0 medium, 0 high impact
  Dict size : 4 words (4 new), 0 extensions, 0 candidates
  Signatures : 77 total

[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 2
[+] Looking for duplicate entries: 2
```

```
rush@kali: ~/Desktop/vulnerability_scan_output
File Actions Edit View Help
TCP faults : 0 failures, 1 timeouts, 0 purged
External links : 0 skipped
Reqs pending : 0

Database statistics:
Pivots : 2 total, 2 done (100.00%)
In progress : 0 pending, 0 init, 0 attacks, 0 dict
Missing nodes : 0 spotted
Node types : 1 serv, 1 dir, 0 file, 0 pinfo, 0 unkn, 0 par, 0 val
Issues found : 0 info, 1 warn, 0 low, 0 medium, 0 high impact
Dict size : 4 words (4 new), 0 extensions, 0 candidates
Signatures : 77 total

[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 2
[+] Looking for duplicate entries: 2
[+] Counting unique nodes: 2
[+] Saving pivot data for third-party tools ...
[+] Writing scan description...
[+] Writing crawl tree: 2
[+] Generating summary views ...
[+] Report saved to '202/index.html' [0xee781202].
[+] This was a great day for science!

(rush@kali)-[~/Desktop/vulnerability_scan_output]
$
```

## Conclusion

The capacity to scan IP addresses and URLs using Bash scripts is a crucial skill in the field of computer programming and automation. Bash scripting, which has its roots in OSes similar to Unix, gives users the ability to automate processes, gather vital data, and communicate effectively with the system. The procedure for carrying out such scans has been explored throughout this exploration, with a focus on user input, verifying input, folder organization, and the usage of specialist tools like Nmap and nslookup. These scripts help system managers and programmers work point for beginners to understand the fundamentals of scripting.

## References

- [1] <https://ryanstutorials.net/bash-scripting-tutorial/bash-script.php> - What is a bash script (Ryans Tutorials)