Alumno: Rodrigo Vila

Ejercicios Modulo 3 – Unidad 4

Ejercicio Número 1 Unidad 4:

Hagan pruebas para que noten la diferencia entre 4 palabras simples concatenadas (nueva + clave) y el uso de caracteres especiales.

Por ejemplo, probar las siguientes claves:

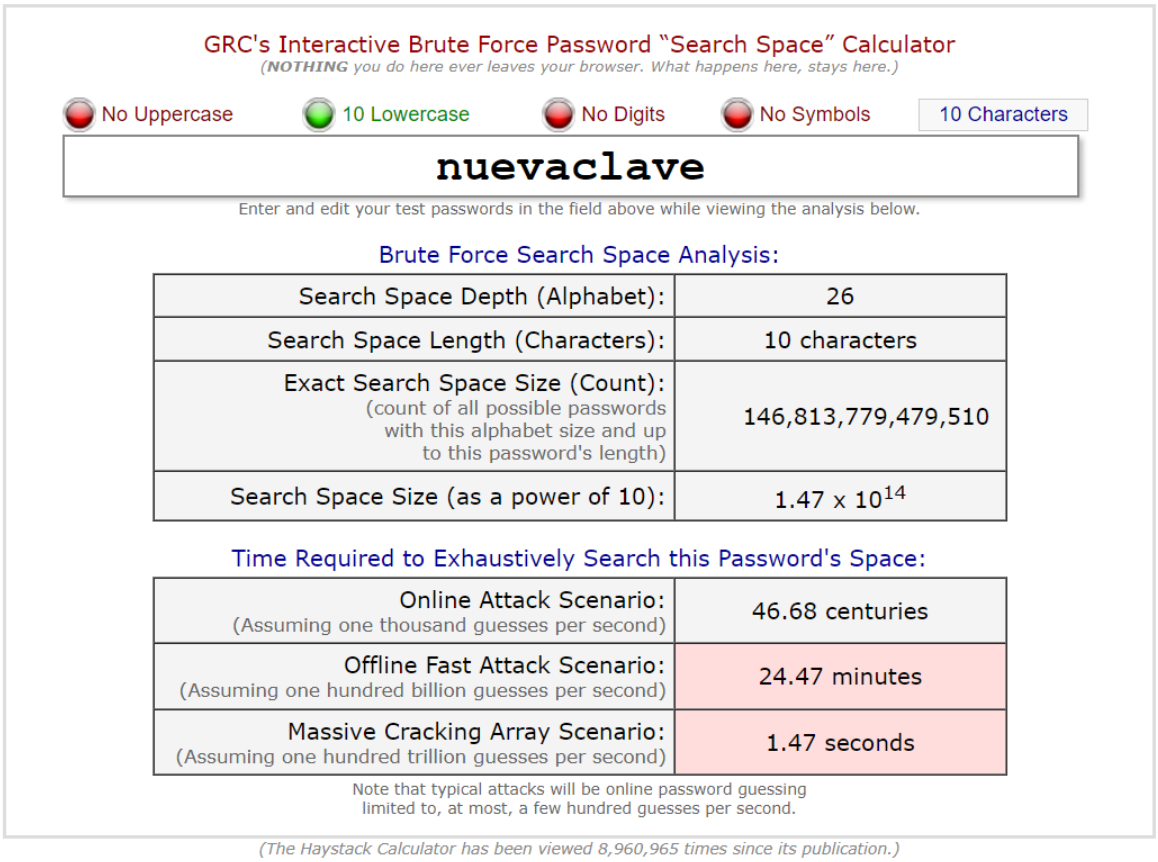
“nuevaclave”

“nu3v@claVe”

“NueVaCLaVe”

“nUEvAclAvE”

Prueba con “nuevaclave”:



GRC's Interactive Brute Force Password “Search Space” Calculator

(*NOTHING* you do here ever leaves your browser. What happens here, stays here.)

No Uppercase | 10 Lowercase | No Digits | No Symbols | 10 Characters

nuevaclave

Enter and edit your test passwords in the field above while viewing the analysis below.

Brute Force Search Space Analysis:

Search Space Depth (Alphabet):	26
Search Space Length (Characters):	10 characters
Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to this password's length)	146,813,779,479,510
Search Space Size (as a power of 10):	1.47×10^{14}

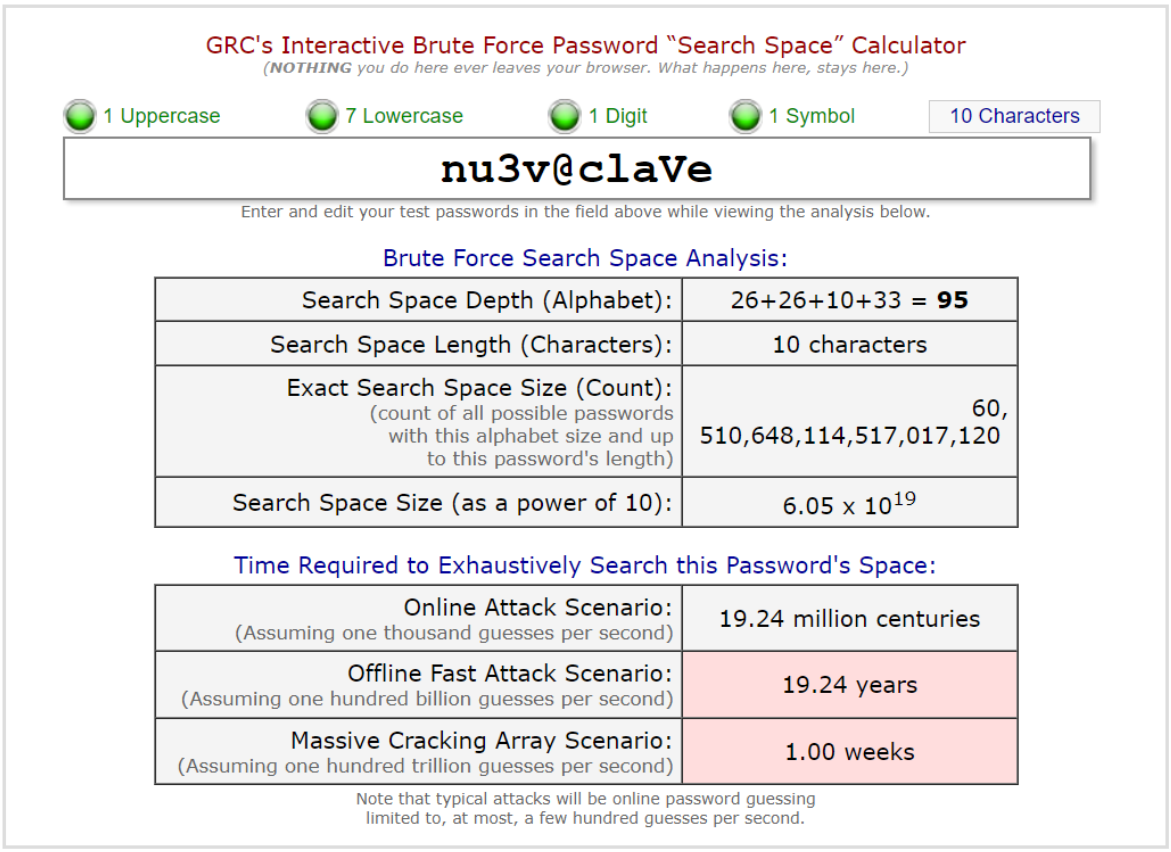
Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: (Assuming one thousand guesses per second)	46.68 centuries
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	24.47 minutes
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	1.47 seconds

Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.

(The Haystack Calculator has been viewed 8,960,965 times since its publication.)

Prueba con “nu3v@claVe”:



GRC's Interactive Brute Force Password "Search Space" Calculator

*(**NOTHING** you do here ever leaves your browser. What happens here, stays here.)*

1 Uppercase | 7 Lowercase | 1 Digit | 1 Symbol | 10 Characters

nu3v@claVe

Enter and edit your test passwords in the field above while viewing the analysis below.

Brute Force Search Space Analysis:

Search Space Depth (Alphabet):	26+26+10+33 = **95**
Search Space Length (Characters):	10 characters
Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to this password's length)	60, 510,648,114,517,017,120
Search Space Size (as a power of 10):	6.05×10^{19}

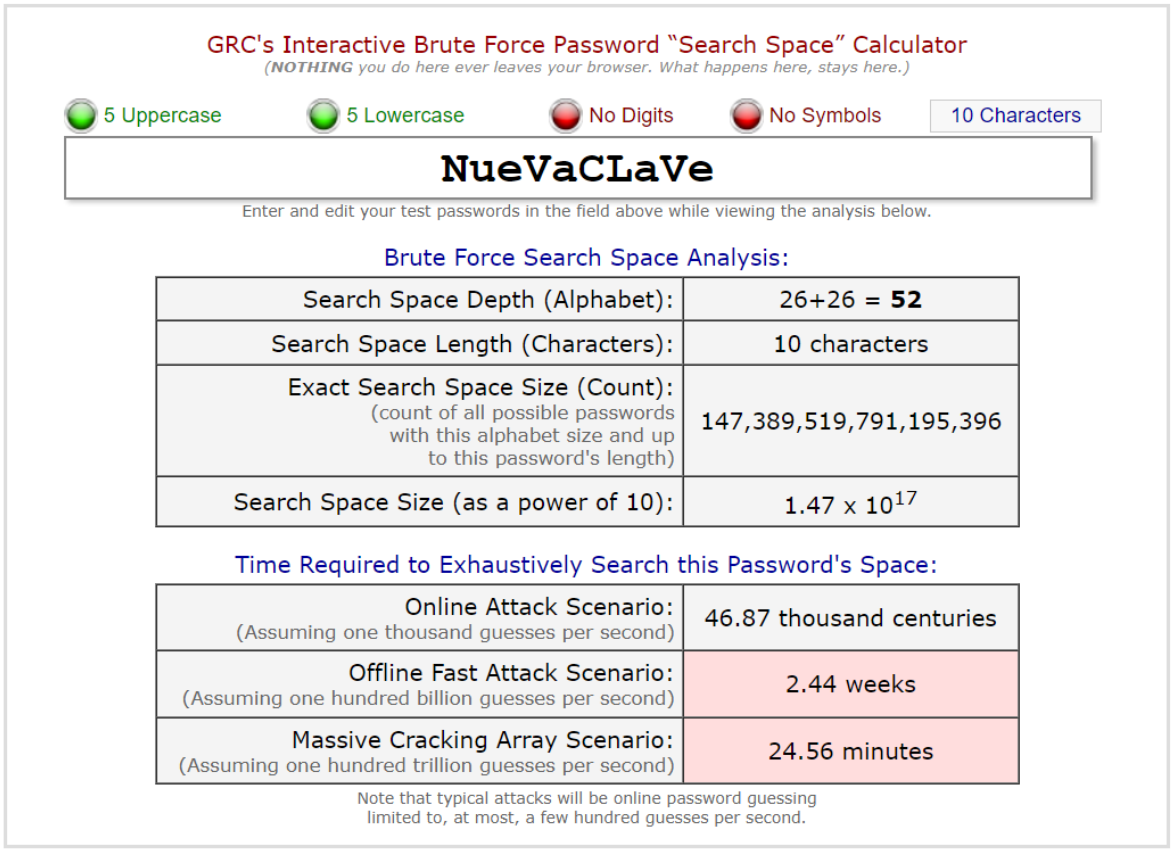
Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: (Assuming one thousand guesses per second)	19.24 million centuries
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	19.24 years
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	1.00 weeks

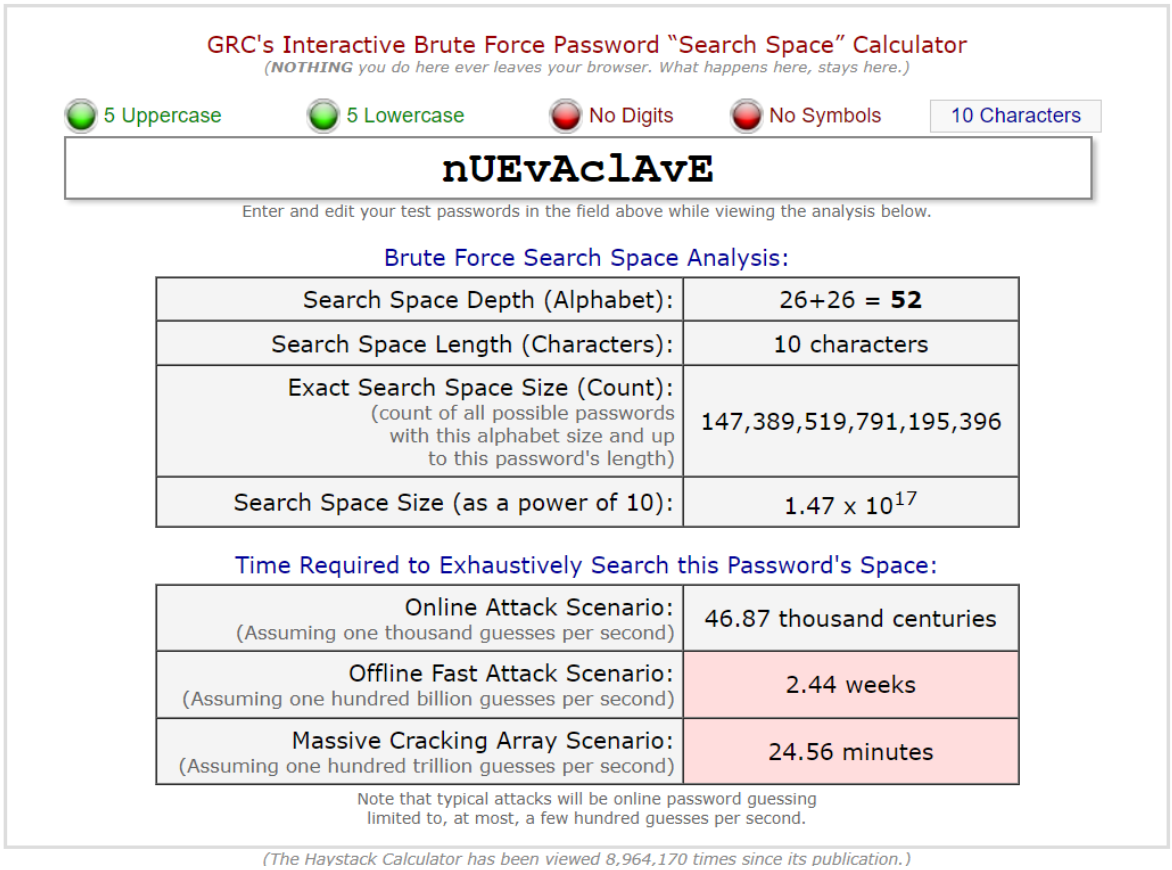
Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.

(The Haystack Calculator has been viewed 8,964,170 times since its publication.)

Prueba con “NueVaCLaVe”:



Prueba con “nUEvAclAvE”:

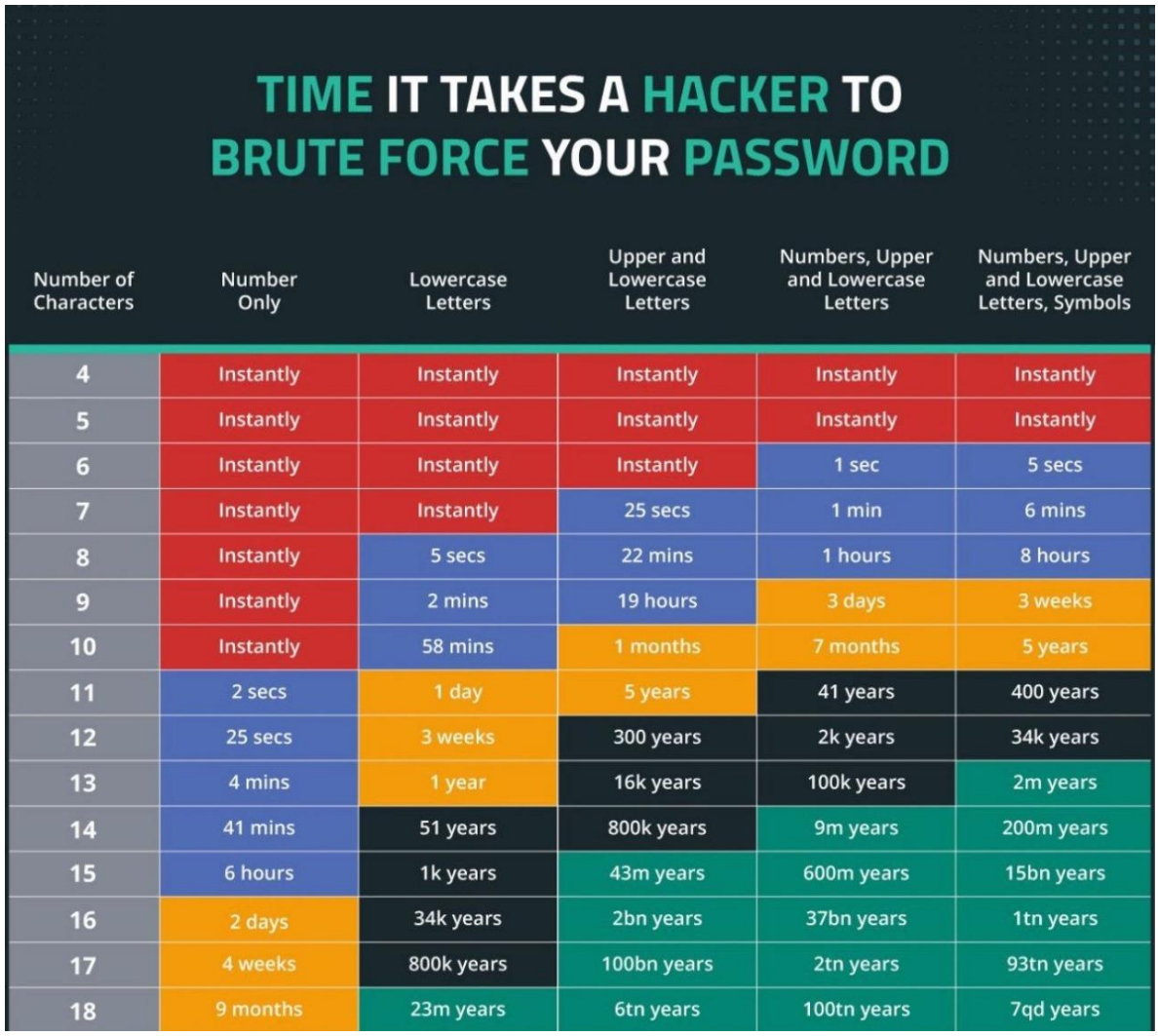


En resumen:

Se realizaron pruebas para evaluar la seguridad de diferentes tipos de contraseñas, específicamente comparando contraseñas simples con aquellas que incorporan caracteres especiales y variaciones en el uso de mayúsculas y minúsculas.

El análisis demostró que las contraseñas simples son más vulnerables a ataques, mientras que las contraseñas que incorporan una combinación de caracteres especiales, números, y variaciones de mayúsculas y minúsculas ofrecen una mayor seguridad. Este ejercicio resalta la importancia de utilizar contraseñas complejas para proteger la información personal y sensible.

Adjunto otro "mapa" estimativo sobre cuánto tiempo tardaría aproximadamente un hacker para vulnerar una contraseña utilizando Fuerza bruta. Obviamente esto es estimativo y muy vago, ya que todo depende de distintos factores como el listado de diccionarios que utilizan, la capacidad de cómputo de su hardware y muchos otros factores:



Number of Characters	Number Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 sec	5 secs
7	Instantly	Instantly	25 secs	1 min	6 mins
8	Instantly	5 secs	22 mins	1 hours	8 hours
9	Instantly	2 mins	19 hours	3 days	3 weeks
10	Instantly	58 mins	1 months	7 months	5 years
11	2 secs	1 day	5 years	41 years	400 years
12	25 secs	3 weeks	300 years	2k years	34k years
13	4 mins	1 year	16k years	100k years	2m years
14	41 mins	51 years	800k years	9m years	200m years
15	6 hours	1k years	43m years	600m years	15bn years
16	2 days	34k years	2bn years	37bn years	1tn years
17	4 weeks	800k years	100bn years	2tn years	93tn years
18	9 months	23m years	6tn years	100tn years	7qd years

Tema adicional: Administración de Software (Linux)

Ejercicios

Hacer listados de:

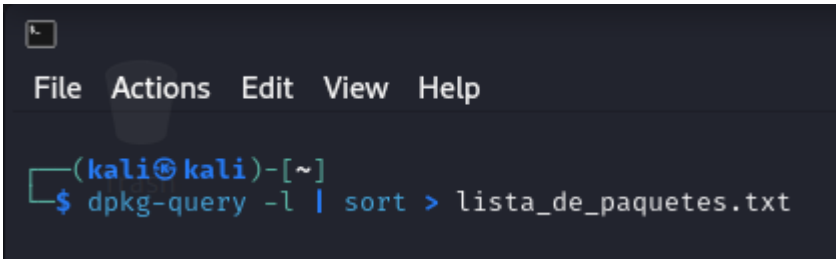
1. Una lista ordenada de todos los paquetes de su sistema
2. Un recuento de todos los paquetes de su sistema
3. Un recuento de todos los archivos en todos los paquetes de su sistema
4. Un recuento de todos los archivos de documentación instalados con RPMs
5. Una búsqueda de todos los paquetes en cuyo nombre aparece "perl"

Partamos de la base que el módulo está diseñado sobre ejercicios sobre la distribución CentOs, que utiliza el gestor de paquetes YUM (Yellowdog Updater, Modified) y los paquetes RPM (Red Hat Package Manager).

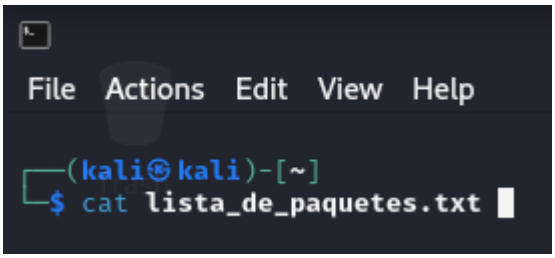
En esta ocasión realizaré el ejercicio sobre la distribución Kali, que utiliza “dpkg” (Debian Package Manager) y apt (Advanced Package Tool) para manejar los paquetes .deb.

1- Una lista ordenada de todos los paquetes de su sistema:

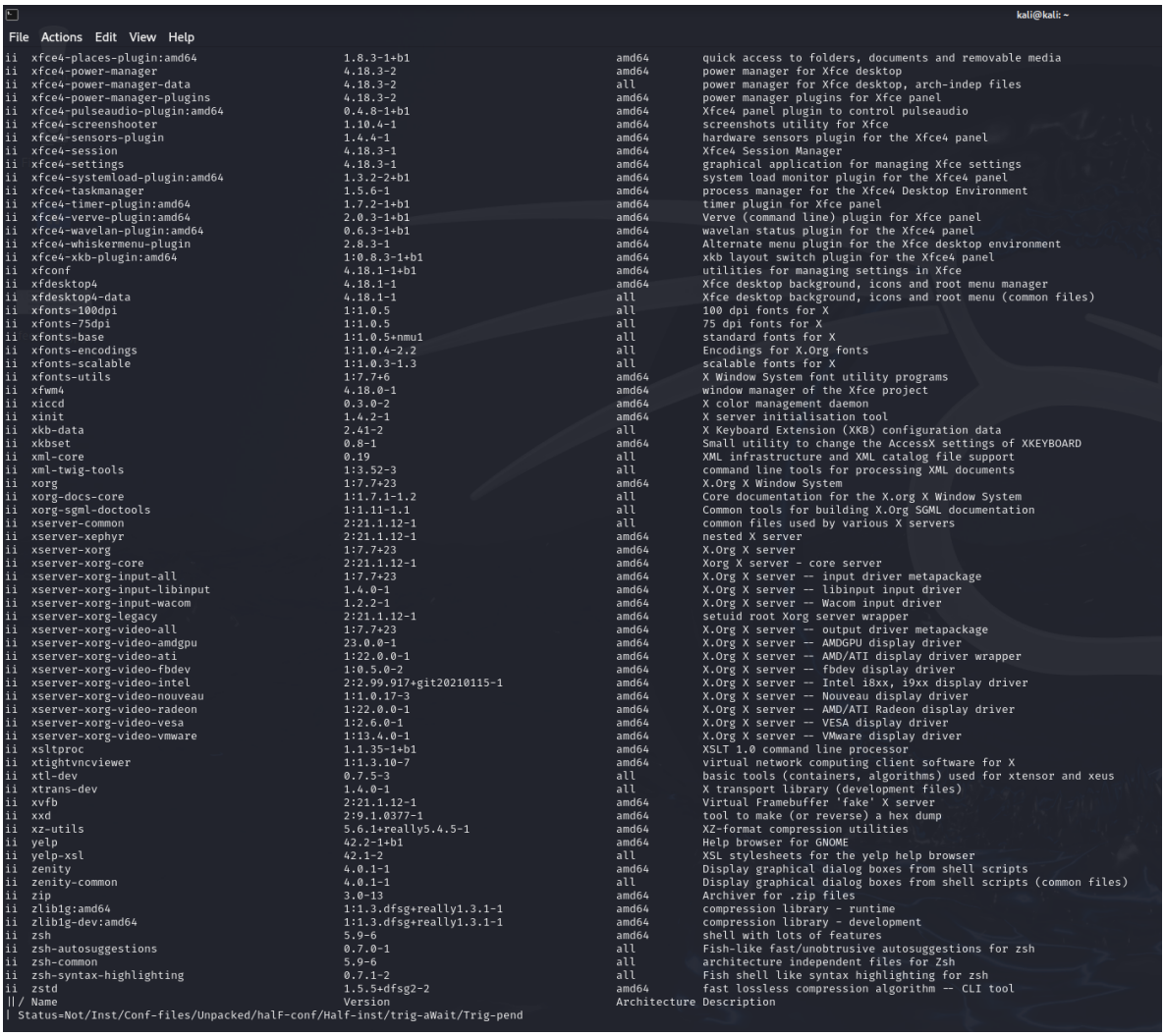
Ejecutamos el comando: *dpkg-query -l | sort > lista_de_paquetes.txt*



Visualizamos el documento de texto: *cat lista_de_paquetes.txt*

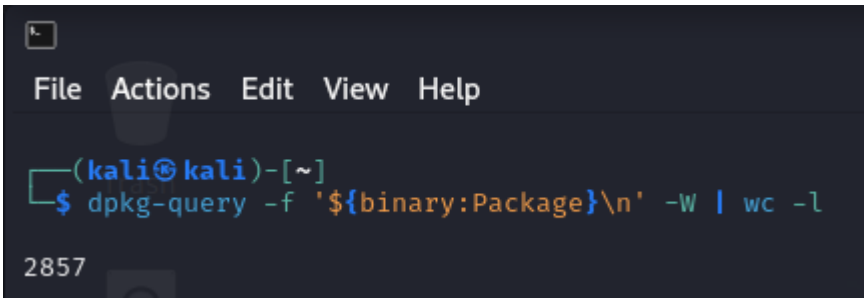


Resultado – Lista de paquetes instalados ordenados alfabéticamente:



2- Un recuento de todos los paquetes de su sistema:

Para realizar un recuento de todos los paquetes instalados en mi sistema Kali Linux utilizare el comando: *dpkg-query -f '${binary:Package}\n' -W | wc -l*



Desglose del Comando:

dpkg-query -f '${binary:Package}\n' -W:

dpkg-query es una herramienta que consulta la base de datos de paquetes gestionados por dpkg.

-f '${binary:Package}\n' especifica el formato de salida, en este caso, listando solo los nombres de los paquetes, uno por línea.

-W es una opción que indica que se deben listar todos los paquetes instalados.

wc -l:

wc (word count) es una herramienta que cuenta líneas, palabras y caracteres.

-l cuenta el número de líneas en la salida, que en este caso corresponderá al número total de paquetes instalados.

3- Un recuento de todos los archivos en todos los paquetes de su sistema:

Para realizar un recuento de todos los archivos en todos los paquetes del sistema utilizare el comando: *dpkg-query -f '${binary:Package}\n' -W | xargs dpkg -L | wc -l*

Explicación del Comando:

dpkg-query -f '${binary:Package}\n' -W:

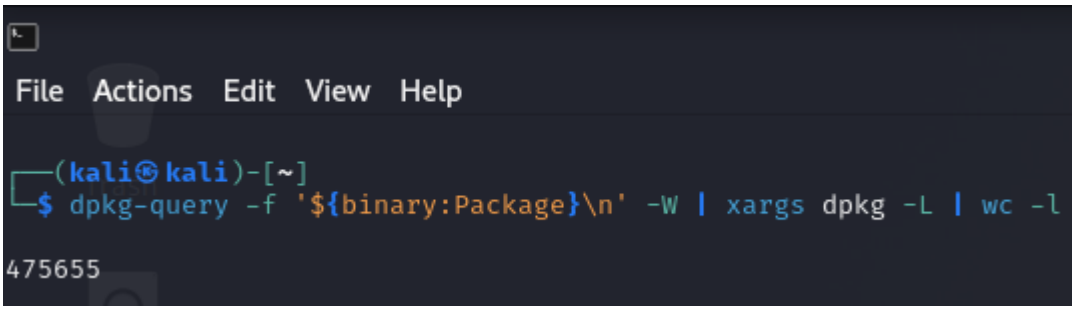
Lista todos los paquetes instalados.

xargs dpkg -L:

Toma la lista de paquetes y para cada paquete ejecuta dpkg -L que lista todos los archivos instalados por ese paquete.

wc -l:

Cuenta el número de líneas en la salida, que en este contexto corresponde al número total de archivos.



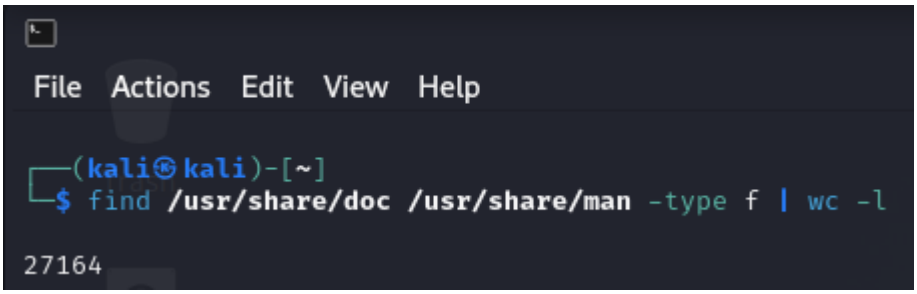
4- Un recuento de todos los archivos de documentación instalados con RPMs:

Aunque Kali Linux no usa RPMs (Red Hat Package Manager) nativamente (usa paquetes .deb y dpkg porque está basado en Debian), los archivos de documentación típicamente se encuentran en directorios estándar como /usr/share/doc.

Primero vamos a listar todos los archivos de documentación:

Usamos find para buscar archivos en los directorios típicos de documentación, como /usr/share/doc y /usr/share/man, y le agregamos el comando *wc -l* para que nos devuelva el número de líneas para saber cuántos archivos de documentación tenemos instalados en esos directorios:

find /usr/share/doc /usr/share/man -type f | wc -l

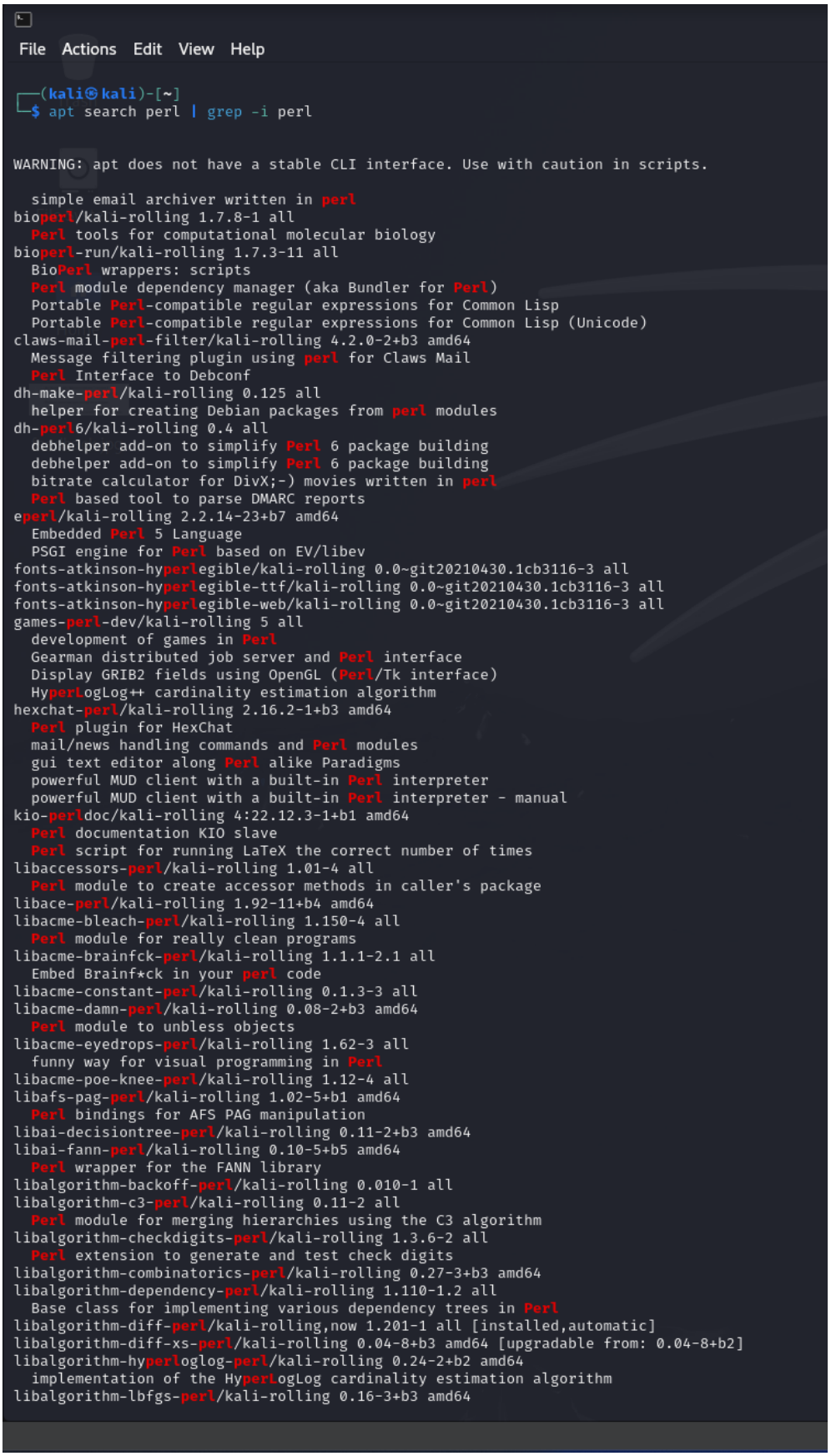


5- Una búsqueda de todos los paquetes en cuyo nombre aparece "perl":

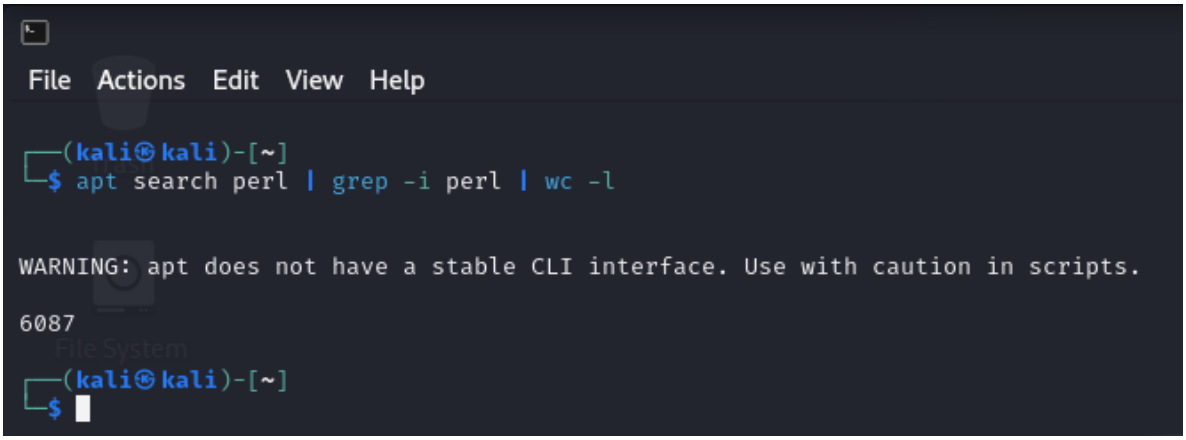
Para buscar todos los paquetes en cuyo nombre aparece "perl" en Kali Linux, voy a usar el comando apt que está diseñado para interactuar con los paquetes del sistema:

En este caso*: apt search perl* y le agrego: *grep -i perl* para que busque la palabra sobre en el nombre de los paquetes e ignore las descripciones:

apt search perl | grep -i perl


File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ apt search perl | grep -i perl
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
simple email archiver written in perl
bioperl/kali-rolling 1.7.8-1 all
Perl tools for computational molecular biology
bioperl-run/kali-rolling 1.7.3-11 all
BioPerl wrappers: scripts
Perl module dependency manager (aka Bundler for Perl)
Portable Perl-compatible regular expressions for Common Lisp
Portable Perl-compatible regular expressions for Common Lisp (Unicode)
claws-mail-perl-filter/kali-rolling 4.2.0-2+b3 amd64
Message filtering plugin using perl for Claws Mail
Perl Interface to Debconf
dh-make-perl/kali-rolling 0.125 all
helper for creating Debian packages from perl modules
dh-perl6/kali-rolling 0.4 all
debhelper add-on to simplify Perl 6 package building
debhelper add-on to simplify Perl 6 package building
bitrate calculator for DivX;-) movies written in perl
Perl based tool to parse DMARC reports
eperl/kali-rolling 2.2.14-23+b7 amd64
Embedded Perl 5 Language
PSGI engine for Perl based on EV/libev
fonts-atkinson-hyperlegible/kali-rolling 0.0~git20210430.1cb3116-3 all
fonts-atkinson-hyperlegible-ttf/kali-rolling 0.0~git20210430.1cb3116-3 all
fonts-atkinson-hyperlegible-web/kali-rolling 0.0~git20210430.1cb3116-3 all
games-perl-dev/kali-rolling 5 all
development of games in Perl
Gearman distributed job server and Perl interface
Display GRIB2 fields using OpenGL (Perl/Tk interface)
HyperLogLog++ cardinality estimation algorithm
hexchat-perl/kali-rolling 2.16.2-1+b3 amd64
Perl plugin for HexChat
mail/news handling commands and Perl modules
gui text editor along Perl alike Paradigms
powerful MUD client with a built-in Perl interpreter
powerful MUD client with a built-in Perl interpreter - manual
kio-perldoc/kali-rolling 4:22.12.3-1+b1 amd64
Perl documentation KIO slave
Perl script for running LaTeX the correct number of times
libaccessors-perl/kali-rolling 1.01-4 all
Perl module to create accessor methods in caller's package
libace-perl/kali-rolling 1.92-11+b4 amd64
libacme-bleach-perl/kali-rolling 1.150-4 all
Perl module for really clean programs
libacme-brainfck-perl/kali-rolling 1.1.1-2.1 all
Embed Brainf*ck in your perl code
libacme-constant-perl/kali-rolling 0.1.3-3 all
libacme-damn-perl/kali-rolling 0.08-2+b3 amd64
Perl module to unbless objects
libacme-eyedrops-perl/kali-rolling 1.62-3 all
funny way for visual programming in Perl
libacme-poe-knee-perl/kali-rolling 1.12-4 all
libafs-pag-perl/kali-rolling 1.02-5+b1 amd64
Perl bindings for AFS PAG manipulation
libai-decisiontree-perl/kali-rolling 0.11-2+b3 amd64
libai-fann-perl/kali-rolling 0.10-5+b5 amd64
Perl wrapper for the FANN library
libalgorithm-backoff-perl/kali-rolling 0.010-1 all
libalgorithm-c3-perl/kali-rolling 0.11-2 all
Perl module for merging hierarchies using the C3 algorithm
libalgorithm-checkdigits-perl/kali-rolling 1.3.6-2 all
Perl extension to generate and test check digits
libalgorithm-combinatorics-perl/kali-rolling 0.27-3+b3 amd64
libalgorithm-dependency-perl/kali-rolling 1.110-1.2 all
Base class for implementing various dependency trees in Perl
libalgorithm-diff-perl/kali-rolling,now 1.201-1 all [installed,automatic]
libalgorithm-diff-xs-perl/kali-rolling 0.04-8+b3 amd64 [upgradable from: 0.04-8+b2]
libalgorithm-hyperloglog-perl/kali-rolling 0.24-2+b2 amd64
implementation of the HyperLogLog cardinality estimation algorithm
libalgorithm-lbfgs-perl/kali-rolling 0.16-3+b3 amd64

Como podemos ver son un montón. Vamos a contar cuantas líneas tenemos para saber la cantidad de paquetes que tenemos que contienen la palabra “perl”:



Rodrigo Vila.-