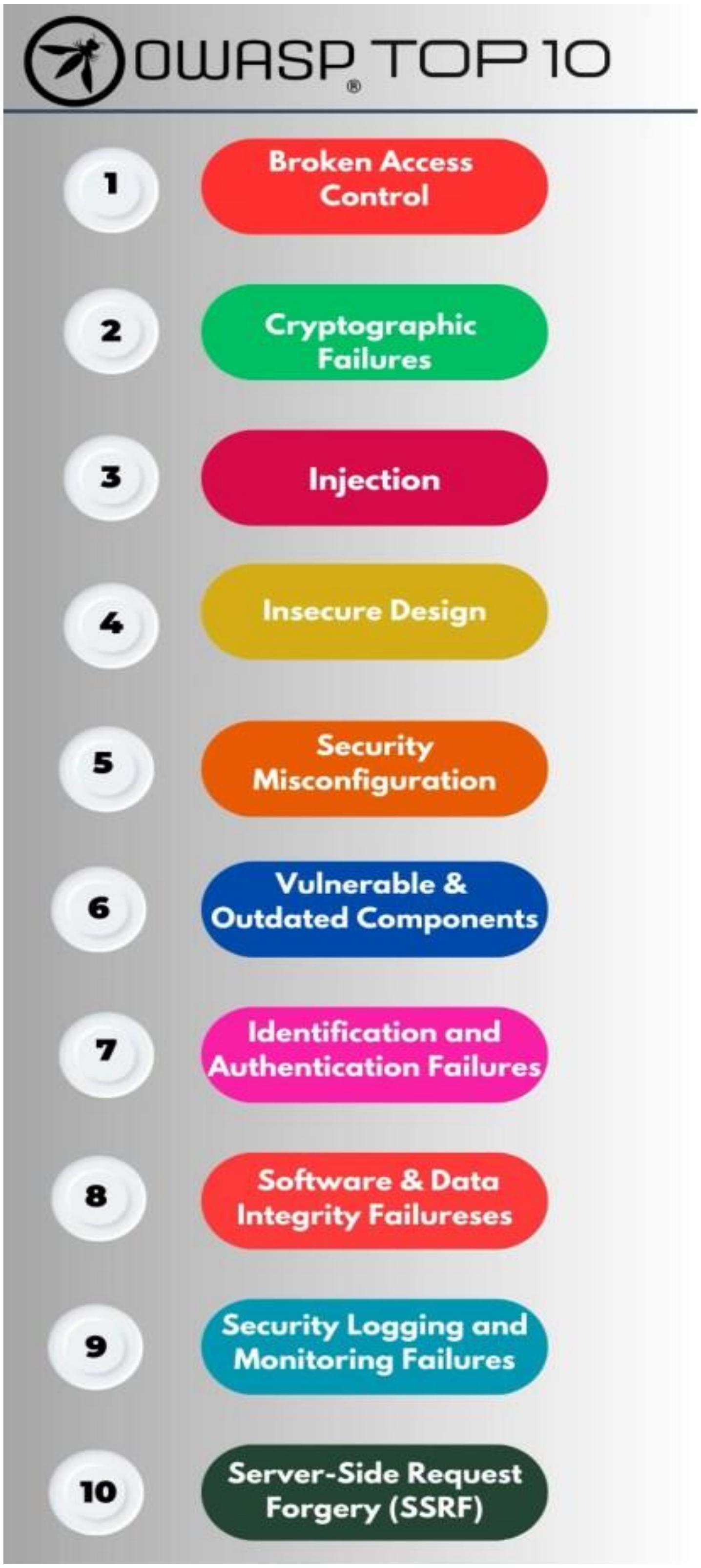
Muhammad Tayab's report 3rd day of 3rd week

OWASP TOP 10 Vulnerabilities



OWASP® TOP 10

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable & Outdated Components
7. Identification and Authentication Failures
8. Software & Data Integrity Failureses
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

Muhammad Tayab's report 3rd day of 3rd week

What is OWASP top 10?

It is a research-backed standard application security awareness document gathered from 40 voluntary organizations across the globe. The priority order in the listing is based on the occurrence of incidents reported from each bug (the highest reported vulnerability takes the first place). The project is led by The OWASP Foundation, which is a non-profit organization that distributes application security-related free and open resources, forums, documentation, technologies, tools, and methodologies. And is updated once every three to four years, with the latest updations based on.

Rather than being a mere vulnerability list, the OWASP top 10 vulnerabilities list helps to assess every flaw with the OWASP Risk Rating methodology. And provides examples, guidelines, and best practices to prevent cyber-attacks. With this, developers and security experts can build cyber resilience to keep their applications safe from threat actors.

List of the OWASP top 10 Vulnerabilities:

Following are the OWASP top 10 2024 vulnerabilities list:

A01:2021—Broken Access Control

This vulnerability allows bad actors to bypass authentications and gain access to sensitive data and systems. They can exploit this vulnerability simply by modifying the URL or by changing the parameter within the browser.

And it occurs due to the lack of proper access control configuration.

Impacts of Broken Access Control:

- Admin privilege even without being logged in
- Add, modify, or remove the data from the user record
- Reputational loss

How to prevent Broken Access Control?

- Implement deny-by-default for resources that are not to be made public
- Minimize Cross-Origin Resource Sharing (CORS)
- Implement monitoring and alert for admins for suspicious user access
- Implement rate limiting to prevent brute forcing
- Ensure metadata and backup files are not present in web roots

Muhammad Tayab's report 3rd day of 3rd week

A02:2021—Cryptographic Failures

This vulnerability enables bad actors to bypass encryptions implemented over sensitive data such as passwords, financial records, credit card numbers, health records, personal information, and even business-related classified information.

This vulnerability is exploited in scenarios in which either automatic database encryption is used or a lack of proper encryption of network requests or when simple or unsalted hashes are used for encryption for data storage.

Impacts of Cryptographic Failures

- Sensitive data breach
- Hefty fines from legal authorities due to lack of data privacy compliance

How to prevent Cryptographic Failures

- Encrypt sensitive data in transit and storage with secure protocols and standards
- Delete sensitive data that is stored unnecessarily
- Ensure cryptographic randomness wherever possible and ensure it is not predictable
- Store passwords with strong and adaptive hashing functions
- Avoid outdated cryptographic mechanisms

A03:2021 — Injection

This vulnerability allows attackers to exploit an application or even gain access to its infrastructure when it does not properly sanitize user input. It can be executed by uploading unintended data or pieces of code along with the web request, which makes the interpreter output sensitive information stored in the database server.

Also, injection is a group of vulnerabilities that contains:

- SQL/NoSQL Injection
- Command Injection
- Server Side Template Injection
- Header Injection
- Cross-Site Scripting (XSS)
- HTML injection
- CSS injection

Impacts of Injection attacks:

- Data leak
- Partial or complete access to the server
- Response manipulation
- Loss of user integrity

Muhammad Tayab's report 3rd day of 3rd week

How to prevent Injection attacks?

- Source code review
- User input sanitation and filtering
- Output encoding
- Implementing limit over output and connection timeout

A04:2021 — Insecure Design

Secure design is a process for evaluating threats continually and ensuring that code is robustly built and tested to prevent known attack methods. A lack of inefficient control design leads to this vulnerability and it fails to choose the level of security architecture for the application or the organization.

Impacts of Insecure Design

- Access to sensitive data stored in the vulnerable system or server
- Altering the functionality of the application

How to prevent insecure design

- Implementing security checks from the initial phase of SDLC (Software Development Lifecycle)
- Validate all important flows that are immune to the threat model and create use-cases and misuse-cases for each layer of your application.
- For important authentication, access control, business logic, and key flows, use threat modeling.
- Implement and use secure design patterns and libraries

A05:2021—Security Misconfiguration

- Indeed we must configure security measures in our systems and applications. But what if they are not properly configured?
- It can be misconfigured or unchanged default (common) credentials, enable unnecessary features such as (ports, services, privileges, pages, etc.), outdated software, etc.

Impacts of Security Misconfiguration

- Complete access over the server or the system and the data stored
- Functionality manipulation of the application, which affects the user

How to prevent Security Misconfiguration

- Make the application minimal with just the necessary features and frameworks used
- Share security practices and directives with the clients

- Implement separate credentials for each phase of development (development, QA, production) environments

A06:2021-Vulnerable and Outdated Components

Software service providers fix vulnerabilities in their products with each software security upgrade. This includes the operating system, web and application servers, database management systems, runtime environments, libraries, and all APIs and related parts. This vulnerability takes place if we haven't updated or implemented the latest version of the secure software. It results in making the entire application vulnerable from the vulnerable framework or software we used within. Also, this vulnerability takes place when you implement software and its related components in your application for unreliable or untrusted sources.

Impacts of Vulnerable and Outdated Components

- Server compromise
- Data breach
- Reputational damage over the firm

How to prevent Vulnerable and Outdated Components

- Update the software and framework patches
- Remove unnecessary dependencies
- Only rely on components and dependencies from secure sources

A07:2021 – Identification and Authentication Failures

This vulnerability is related to security weakness in the login module. It occurs due to a lack of restrictions for automated attacks, unchanged default passwords, improper session validation or expiry, and a lack of restrictions for weak or well-known passwords.

Impacts of Identification and Authentication Failures

- User account takeover
- Identity theft

How to prevent Identification and Authentication Failures

- Restrict weak or default passwords
- Implement Multi-Factor Authentication
- Log failed password attempts and implement an admin alert system
- Implement delay for numerous failed login attempts
- Enforce password length, complexity, and password standards

Muhammad Tayab's report 3rd day of 3rd week

A08:2021 – Software and Data Integrity Failures

This vulnerability occurs due to a lack of integrity in either or both of the code and infrastructure of the software being used. It can be due to using plugins, modules, or libraries from illegitimate sources. Also, a lack of proper integrity checks of software updates would lead to the same.

Impact of Software and Data Integrity Failures

- Database compromise
- Unauthorized updates making it run over all the installations

How to prevent Software and Data Integrity Failures

- Ensure packages, libraries, and dependencies are utilizing trusted repositories
- Verify the authenticity of the software or data with proper digital signatures
- Implement and ensure proper access control, configuration, and segregation in the CI/CD pipeline
- Use a software supply chain security tool, such as OWASP Dependency-Check or OWASP CycloneDX, to ensure that components do not contain known vulnerabilities.

A09:2021 – Security Logging and Monitoring Failures

Monitoring and logging incidents play a vital role in the detection of breaches. This category is to help find and respond to active breaches properly, with the help of logs. It occurs due to a lack of or unclear logs of failed or suspicious login attempts, high-value transactions, etc.

Impact of Security Logging and Monitoring Failures

- The source and the intensity of a data breach can’t be analyzed due to a lack of incident logs
- Database compromise

How to prevent Security Logging and Monitoring Failures

- Establish proper monitoring and alerting of suspicious activities
- Ensure logs are implemented
- Encode the logs to prevent injection attacks on the logging system
- Implement a standardized incident response and recovery plan

A10:2021 – Server-Side Request Forgery (SSRF)

SSRF vulnerabilities occur when a web application retrieves a remote resource from the server or database without verifying the user-supplied URL. It allows the bad actor to send a forged request to an unexpected network destination, even if it is secured by a firewall, VPN, or any sort of network access control measures.

Muhammad Tayab's report 3rd day of 3rd week

Impacts of Server-Side Request Forgery

- Compromising the infrastructure of the application including its internal services
- Sensitive data leakage
- Conduct further attacks such as RCE (Remote Code Execution) and DoS (Denial of Service)

How to prevent Server-Side Request Forgery

- Implement deny-by-default in firewall settings which block all except essential intranet traffic
- Sanitize and validate all the client-supplied input data
- Enforce URL schema, port, and destination with a positive allow-list
- Disable HTTP redirections
- Use network encryption

Automated scans with scripts refer to the use of scripting languages and tools to automate the process of scanning networks, systems, and applications for vulnerabilities. This approach allows security professionals to efficiently identify and address potential security issues without manual intervention. Here's a detailed explanation:

Automated Scans with Scripts:

Automated scans with scripts involve writing and executing scripts to perform security scans. These scripts can automate tasks such as port scanning, vulnerability assessment, configuration checking, and more. By leveraging scripts, security professionals can schedule scans, generate reports, and integrate scanning into continuous integration/continuous deployment (CI/CD) pipelines.

Tools and Techniques

1. **Nmap (Network Mapper)**
   - **Description**: Nmap is a powerful open-source network scanning tool that can be used to discover hosts and services on a computer network.
   - **Techniques**: Port scanning, service detection, OS detection, version detection.
   - **Example Script**:

     ```
     // Nmap script to scan a network range for open ports
     nmap -sS -p 1-65535 192.168.1.0/24 -oN scan_results.txt
     ```

2. **Nessus**
   - **Description**: Nessus is a widely used vulnerability scanner that identifies vulnerabilities in systems and applications.
   - **Techniques**: Vulnerability scanning, configuration auditing, patch management.
   - **Example Script** (using Nessus CLI):

```
// Nessus script to launch a scan and generate a report
nessus -q -x -T pdf -o report.pdf -i nessus_scan.nessus
```

3. OpenVAS

- **Description**: OpenVAS (Open Vulnerability Assessment System) is an open-source framework for network vulnerability scanning and management.
- **Techniques**: Full system vulnerability assessment, reporting.
- **Example Script**:

```
// OpenVAS script to run a scan
omp -u admin -w admin -h localhost -p 9390 -n -iX "<create_task>
<name>Network Scan</name> <comment>Full scan of the
network</comment> <config id='daba56c8-73ec-11df-a475-
002264764cea'/></create_task>"
```

4. Nikto

- **Description**: Nikto is an open-source web server scanner that performs comprehensive tests against web servers for multiple vulnerabilities.
- **Techniques**: Web server scanning, identifying outdated software, checking for configuration issues.
- **Example Script**:

```
// Nikto script to scan a web server
nikto -h http://example.com -o nikto_report.txt
```

5. Python Scripts

- **Description**: Python is a versatile scripting language that can be used to write custom scanning scripts.
- **Techniques**: Custom vulnerability scanning, integration with APIs, data parsing and reporting.
- **Example Script** (simple port scanner):

```python
import socket

def scan_port(ip, port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(1)
    try:
        s.connect((ip, port))
        print(f"Port {port} is open on {ip}")
    except:
        pass
    finally:
        s.close()

if __name__ == "__main__":
    target_ip = "192.168.1.1"
    for port in range(1, 1025):
        scan_port(target_ip, port)
```

Muhammad Tayab's report 3rd day of 3rd week

Scanning Best Practices

Effective scanning is critical to maintaining the security posture of an organization. Following best practices ensures that scans are comprehensive, non-intrusive, and actionable. Here are some key best practices for scanning:

1. **Plan and Define Scope**
   - **Identify Assets**: Clearly identify and document all assets that need to be scanned, including servers, workstations, network devices, and applications.
   - **Define Boundaries**: Clearly define the boundaries and exclusions for the scan to avoid unintended disruptions.
2. **Use Credentialed Scans**
   - **Credentials**: Use administrative credentials to perform scans. Credentialed scans provide deeper insights into the security posture of systems, such as patch levels and configuration settings.
3. **Schedule Regular Scans**
   - **Frequency**: Schedule regular scans to continuously monitor the network and systems. Weekly or monthly scans are common, but critical systems may require more frequent scanning.
   - **Off-Peak Hours**: Schedule scans during off-peak hours to minimize the impact on network performance and user productivity.
4. **Update Scanning Tools**
   - **Regular Updates**: Keep scanning tools and their vulnerability databases updated to ensure the latest vulnerabilities are detected.
   - **Patch Management**: Regularly apply patches and updates to scanning tools themselves to avoid any security flaws.
5. **Validate and Prioritize Findings**
   - **False Positives**: Review scan results to validate findings and eliminate false positives.
   - **Risk Assessment**: Prioritize vulnerabilities based on risk and impact to the organization. Focus on high-risk vulnerabilities first.
6. **Remediation and Verification**
   - **Remediation Plan**: Develop and implement a remediation plan to address identified vulnerabilities. Assign responsibility and deadlines for fixing issues.
   - **Rescan**: Perform rescans to verify that vulnerabilities have been successfully remediated.
7. **Documentation and Reporting**
   - **Detailed Reports**: Generate detailed reports that include the scope of the scan, findings, risk levels, and remediation steps.
   - **Compliance**: Ensure reports meet compliance requirements for regulatory standards such as PCI-DSS, HIPAA, and GDPR.
8. **Security Awareness and Training**
   - **Educate Staff**: Train IT staff and stakeholders on the importance of scanning and how to interpret and act on scan results.
   - **Security Policies**: Incorporate scanning practices into organizational security policies and procedures.
9. **Integration with Security Tools**

- **SIEM Integration**: Integrate scan results with Security Information and Event Management (SIEM) systems for centralized monitoring and alerting.
- **Automation**: Use automation tools to streamline the scanning and remediation processes.

10. **Regular Review and Improvement**
    - **Continuous Improvement**: Regularly review and update scanning policies, tools, and practices to adapt to new threats and improve overall security posture.

CIS Security

CIS (Center for Internet Security) is a non-profit organization focused on enhancing cybersecurity readiness and response by providing best practices, guidelines, tools, and services. The organization is known for its CIS Controls and CIS Benchmarks.

- **Website**: CIS Security

Key Offerings of CIS Security

1. **CIS Controls**
    - A prioritized set of actions that help protect organizations and data from known cyber-attack vectors. These controls are widely recognized and used as a standard for cyber defense.
2. **CIS Benchmarks**
    - Configuration guidelines for securing various technologies, including operating systems, software, and network devices. These benchmarks provide detailed, actionable guidance to help organizations configure systems securely.
3. **CIS-CAT Pro**
    - A tool that helps organizations assess the configuration of their systems against the CIS Benchmarks. It provides automated assessment capabilities and detailed reports.
4. **CIS SecureSuite Membership**
    - Membership program offering access to the full range of CIS tools and resources, including CIS Controls, CIS Benchmarks, CIS-CAT Pro, and more.
5. **MS-ISAC (Multi-State Information Sharing and Analysis Center)**
    - A collaboration between CIS and state, local, tribal, and territorial (SLTT) governments to improve the overall cybersecurity posture through information sharing and coordinated response efforts.
6. **EI-ISAC (Elections Infrastructure Information Sharing and Analysis Center)**
    - A resource to help protect election infrastructure in the United States through cybersecurity services, information sharing, and threat intelligence.
7. **CIS Community Defense Model**
    - A model that maps CIS Controls to common attack patterns and techniques to help organizations prioritize and implement effective security measures.

Muhammad Tayab's report 3rd day of 3rd week

Port-Scanning Analysis

Port-scanning analysis is a technique used to identify open ports and services running on a target system or network. It helps in assessing the security posture of systems by revealing potential entry points for attackers. Port-scanning is a crucial step in the information-gathering phase of a penetration test or vulnerability assessment

For Port-Scanning Analysis

1. **Initial Reconnaissance**: Identify live hosts and open ports.
2. **Service Enumeration**: Determine the services and versions running on identified open ports.
3. **Vulnerability Assessment**: Analyze the services for known vulnerabilities.
4. **Intrusion Detection**: Identify unusual patterns that might indicate malicious scanning activity.
5. **Compliance Monitoring**: Ensure that unnecessary ports are closed as per organizational security policies.

Techniques for Port-Scanning Analysis

1. **Banner Grabbing**: Collect information about services running on open ports by capturing the banners returned by them.
2. **Service Fingerprinting**: Use tools like Nmap to determine the exact version of the services running on open ports.
3. **Topology Mapping**: Visualize the network topology to understand the network architecture and identify potential security weaknesses.

Websites for Post-Scanning Analysis

1. **Shodan**
   - **Description**: A search engine for Internet-connected devices, useful for finding information about the devices exposed by port scans.
   - **Website**: Shodan
2. **Censys**
   - **Description**: Provides data on all devices accessible on the Internet, similar to Shodan, and useful for understanding the exposure of scanned devices.
   - **Website**: Censys
3. **GreyNoise**
   - **Description**: Aggregates data on Internet-wide scan activity, allowing users to differentiate between benign and malicious scanning.
   - **Website**: GreyNoise
4. **VirusTotal**
   - **Description**: Allows uploading and analysis of files and URLs to detect malicious content, useful for analyzing suspicious files found during scans.
   - **Website**: VirusTotal
5. **Have I Been Pwned**
   - **Description**: A service that allows you to check if your email address has been compromised in a data breach.

- **Website**: Have I Been Pwned

6. **ThreatCrowd**
   - **Description**: A search engine for threat intelligence that provides information about IP addresses, domains, emails, and more.
   - **Website**: ThreatCrowd

Reporting Tools for Post-Scanning Analysis

1. **Dradis Framework**
   - **Description**: An open-source collaboration and reporting tool used by security professionals to create comprehensive reports.
   - **Website**: Dradis Framework
2. **Faraday IDE**
   - **Description**: A collaborative penetration test and vulnerability management platform that integrates with various tools for automated reporting.
   - **Website**: Faraday
3. **Metasploit Pro**
   - **Description**: Offers automated reporting and collaboration features, alongside its extensive penetration testing capabilities.
   - **Website**: Metasploit Pro
4. **Cobalt Strike**
   - **Description**: Provides advanced threat emulation and reporting capabilities, allowing for detailed post-exploitation reporting.
   - **Website**: Cobalt Strike
5. **OWASP ZAP (Zed Attack Proxy)**
   - **Description**: A popular open-source web application security scanner with automated reporting features.
   - **Website**: OWASP ZAP
6. **Nessus**
   - **Description**: A vulnerability scanner with extensive reporting capabilities, including customizable templates and compliance checks.
   - **Website**: Nessus
7. **OpenVAS**
   - **Description**: An open-source vulnerability scanner that includes detailed reporting features for post-scanning analysis.
   - **Website**: OpenVAS
8. **Burp Suite**
   - **Description**: A comprehensive web vulnerability scanner that provides detailed reports on identified vulnerabilities.
   - **Website**: Burp Suite
9. **Acunetix**
   - **Description**: A web vulnerability scanner that provides detailed reports on web application vulnerabilities.
   - **Website**: Acunetix

Muhammad Tayab's report 3rd day of 3rd week

RAPID7:

Rapid7 is a cybersecurity company that provides a range of security solutions aimed at improving the security posture of organizations. They offer products and services for vulnerability management, penetration testing, incident detection and response, and more.

Rapid7 Website

- **Website URL**: Rapid7

Key Offerings

1. **Vulnerability Management**
   - **InsightVM**: A comprehensive vulnerability management solution that helps organizations identify, prioritize, and remediate vulnerabilities in their network.
2. **Penetration Testing**
   - **Metasploit**: A popular penetration testing framework used by security professionals to test the security of systems and applications by simulating real-world attacks.
3. **Incident Detection and Response**
   - **InsightIDR**: A detection and response solution that helps organizations identify and respond to threats more effectively.
4. **Application Security**
   - **InsightAppSec**: A dynamic application security testing (DAST) solution that scans web applications for vulnerabilities.
5. **Security Information and Event Management (SIEM)**
   - **InsightOps**: A cloud-based log management and SIEM solution that provides visibility into security and operations data.
6. **Cloud Security**
   - **DivvyCloud**: A cloud security posture management (CSPM) solution that helps organizations secure their cloud infrastructure.
7. **Security Orchestration, Automation, and Response (SOAR)**
   - **InsightConnect**: A SOAR solution that automates repetitive security tasks, allowing security teams to focus on higher-priority activities.

Services

- **Consulting Services**: Rapid7 offers consulting services to help organizations with their security strategy, assessments, and compliance requirements.
- **Managed Services**: They provide managed detection and response (MDR) services to help organizations monitor and respond to threats.

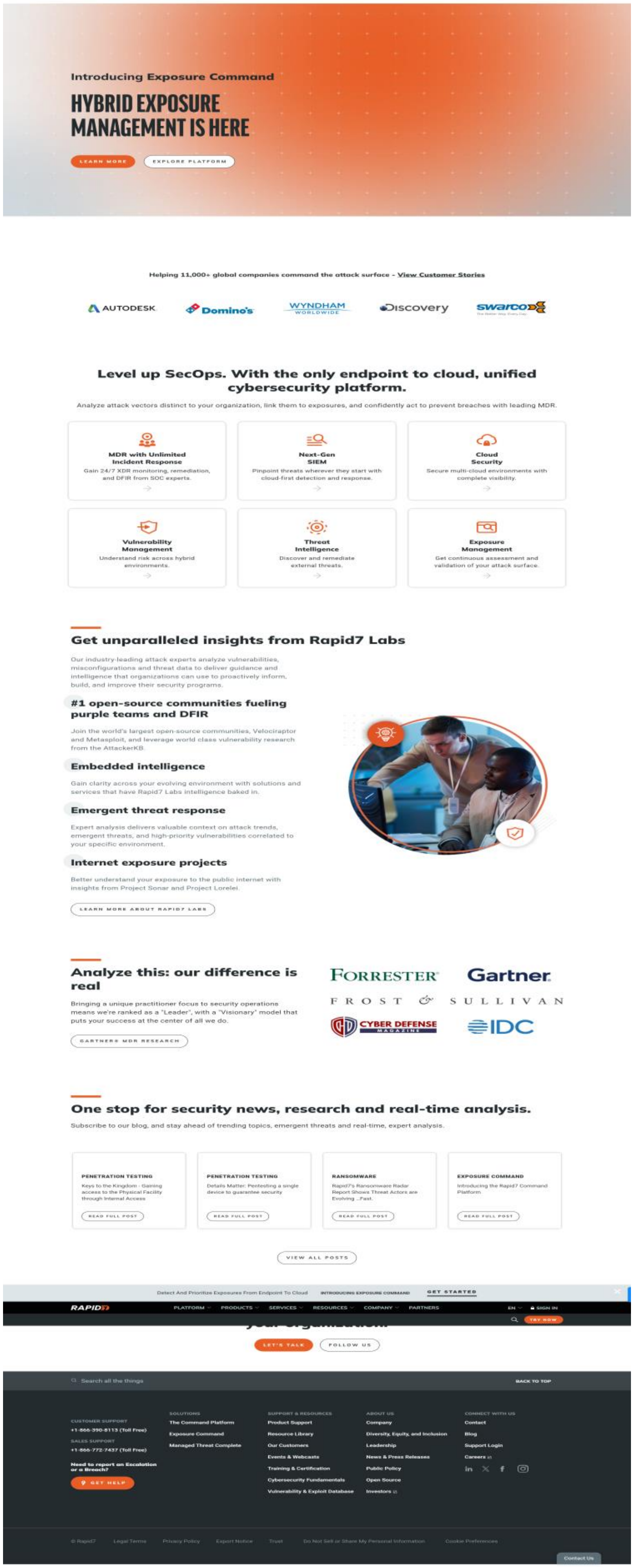
Resources

- **Blog**: Rapid7 maintains a blog with insights and updates on cybersecurity trends and best practices.
- **Webinars**: They offer webinars on various topics related to cybersecurity.
- **Whitepapers and Reports**: Rapid7 publishes whitepapers and research reports on security-related topics.

Community

- **Rapid7 Community**: A platform where security professionals can share knowledge, ask questions, and collaborate on security challenges.

Muhammad Tayab's report 3rd day of 3rd week



Introducing Exposure Command

HYBRID EXPOSURE MANAGEMENT IS HERE

Helping 11,000+ global companies command the attack surface - View Customer Stories

Level up SecOps. With the only endpoint to cloud, unified cybersecurity platform.

Analyze attack vectors distinct to your organization, link them to exposures, and confidently act to prevent breaches with leading MDR.

MDR with Unlimited Incident Response
Gain 24/7 XDR monitoring, remediation, and DFIR from SOC experts.

Next-Gen SIEM
Pinpoint threats wherever they start with cloud-first detection and response.

Cloud Security
Secure multi-cloud environments with complete visibility.

Vulnerability Management
Understand risk across hybrid environments.

Threat Intelligence
Discover and remediate external threats.

Exposure Management
Get continuous assessment and validation of your attack surface.

Get unparalleled insights from Rapid7 Labs

Our industry-leading attack experts analyze vulnerabilities, misconfigurations and threat data to deliver guidance and intelligence that organizations can use to proactively inform, build, and improve their security programs.

#1 open-source communities fueling purple teams and DFIR

Join the world's largest open-source communities, Velociraptor and Metasploit, and leverage world class vulnerability research from the AttackerKB.

Embedded intelligence

Gain clarity across your evolving environment with solutions and services that have Rapid7 Labs intelligence baked in.

Emergent threat response

Expert analysis delivers valuable context on attack trends, emergent threats, and high-priority vulnerabilities correlated to your specific environment.

Internet exposure projects

Better understand your exposure to the public internet with insights from Project Sonar and Project Lorelei.

Analyze this: our difference is real

Bringing a unique practitioner focus to security operations means we're ranked as a "Leader", with a "Visionary" model that puts your success at the center of all we do.

GARTNER® MDR RESEARCH

One stop for security news, research and real-time analysis.

Subscribe to our blog, and stay ahead of trending topics, emergent threats and real-time, expert analysis.

PENETRATION TESTING
Keys to the Kingdom - Gaining access to the Physical Facility through Internal Access

PENETRATION TESTING
Details Matter: Pentesting a single device to guarantee security

RANSOMWARE
Rapid7's Ransomware Radar Report Shows Threat Actors are Evolving ...Fast.

EXPOSURE COMMAND
Introducing the Rapid7 Command Platform

Detect And Prioritize Exposures From Endpoint To Cloud   INTRODUCING EXPOSURE COMMAND

your organization.

CUSTOMER SUPPORT
+1-866-390-8113 (Toll Free)
SALES SUPPORT
+1-866-772-7437 (Toll Free)
Need to report an Escalation or a Breach?

SOLUTIONS
The Command Platform
Exposure Command
Managed Threat Complete

SUPPORT & RESOURCES
Product Support
Resource Library
Our Customers
Events & Webcasts
Training & Certification
Cybersecurity Fundamentals
Vulnerability & Exploit Database

ABOUT US
Company
Diversity, Equity, and Inclusion
Leadership
News & Press Releases
Public Policy
Open Source
Investors

CONNECT WITH US
Contact
Blog
Support Login
Careers

© Rapid7   Legal Terms   Privacy Policy   Export Notice   Trust   Do Not Sell or Share My Personal Information   Cookie Preferences

Muhammad Tayab's report 3rd day of 3rd week

Acunetix

Definition

- Automated web application security scanner designed to identify vulnerabilities in websites, web applications, and APIs.

Key Features

- **Comprehensive Scanning**: Detects vulnerabilities like SQL injection, XSS, etc.
- **DeepScan Technology**: Analyzes modern web applications, including those using frameworks like React and Angular.
- **Authenticated Scanning**: Tests areas behind login forms.
- **Compliance Reporting**: Supports PCI DSS, HIPAA, ISO 27001 compliance.
- **Integration**: Compatible with CI/CD pipelines, and integrates with issue trackers (e.g., Jira, GitLab).
- **Risk Prioritization**: Categorizes vulnerabilities by severity and provides remediation guidance.

How It Works

- **Crawling**: Maps out the website or application by discovering pages, forms, and inputs.
- **Vulnerability Detection**: Tests for security issues based on discovered resources.
- **Analysis and Reporting**: Generates detailed reports with information on vulnerabilities, their severity, and recommendations for remediation.
- **Remediation**: Provides guidance on fixing vulnerabilities and allows for retesting.

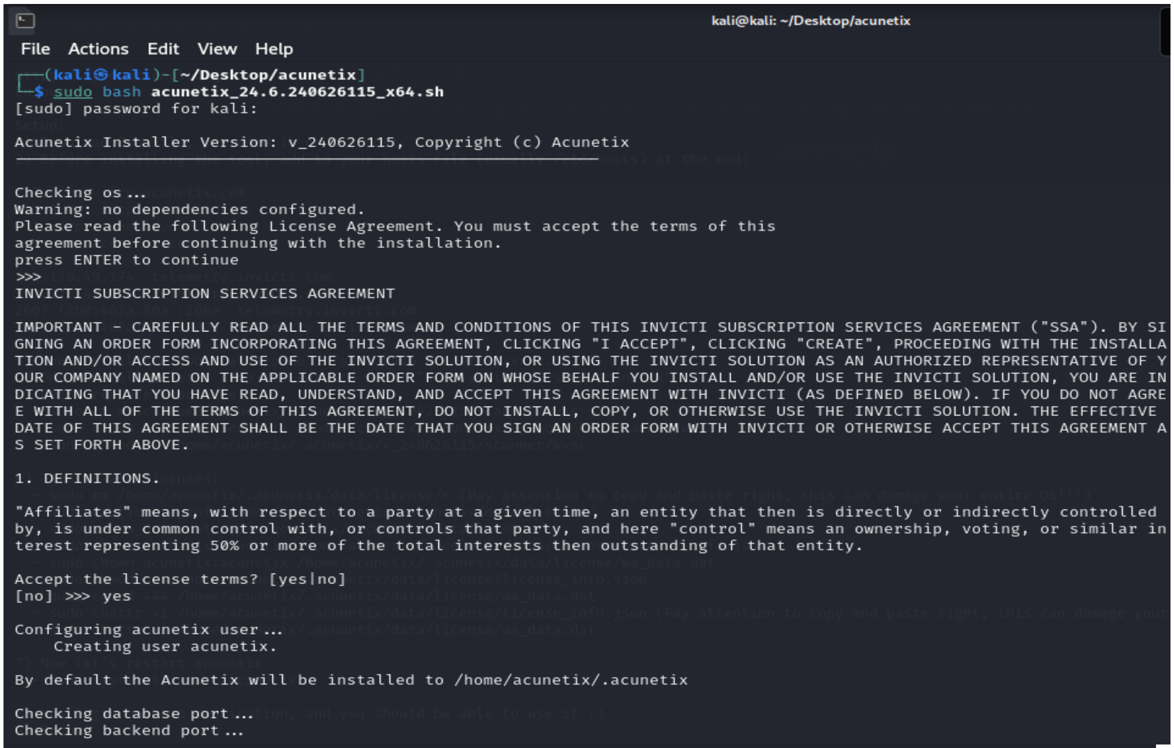
How to Use It

1. **Install Acunetix**: Choose between on-premises or cloud-based deployment.
2. **Configure Scanning**: Set up targets and authentication settings if needed.
3. **Run a Scan**: Initiate a scan to discover vulnerabilities.
4. **Review Reports**: Analyze the detailed reports generated after scanning.
5. **Remediate Issues**: Follow the recommendations to fix identified vulnerabilities.
6. **Retest**: Perform additional scans to ensure issues have been resolved.

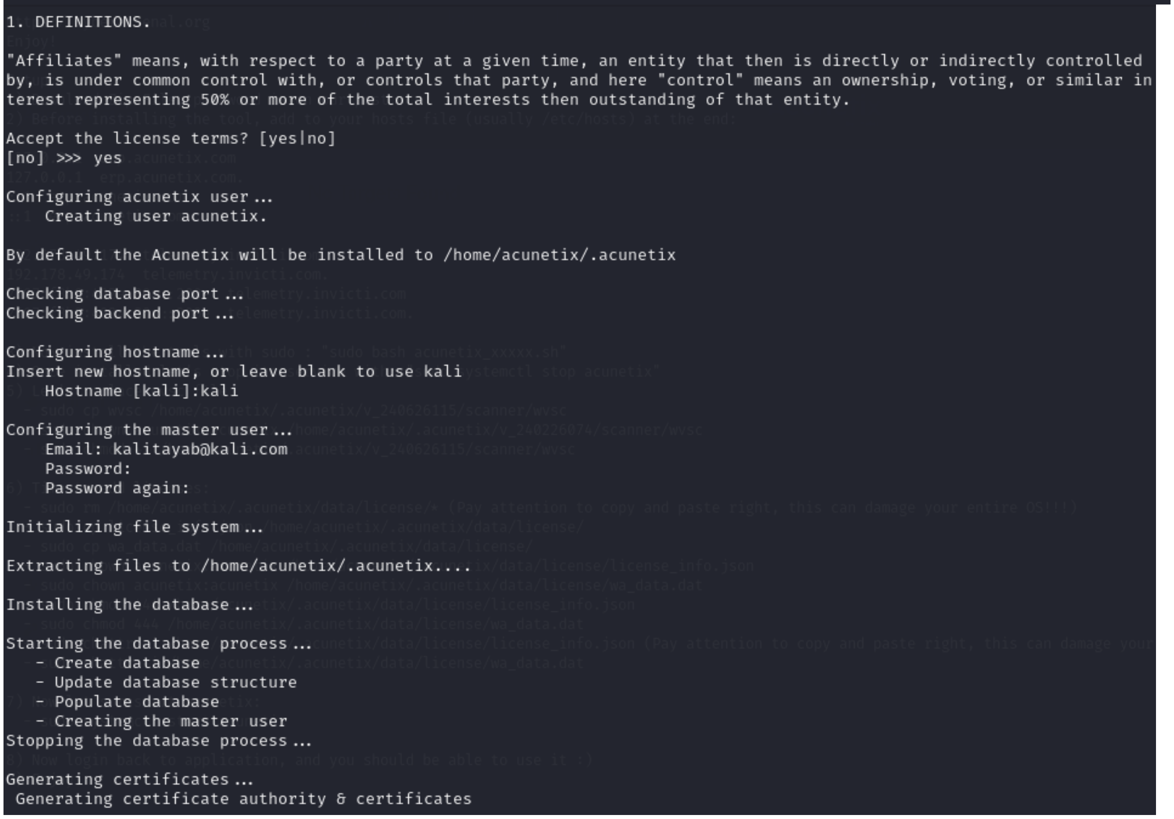
Muhammad Tayab's report 3rd day of 3rd week

INSTALLATION OF THE ACUNETIX

Here you can see the installation of the acunetix:



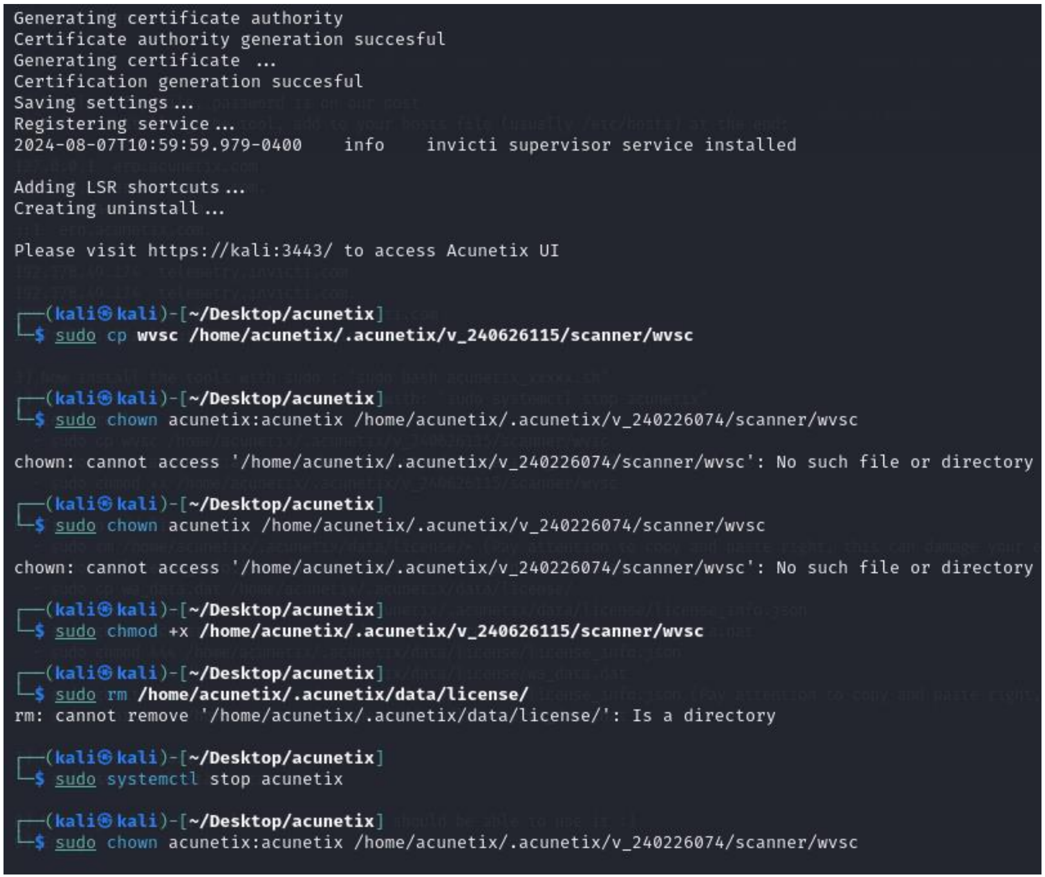
```
┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo bash acunetix_24.6.240626115_x64.sh
[sudo] password for kali:

Acunetix Installer Version: v_240626115, Copyright (c) Acunetix
────────────────────────────────────────────────────────────────

Checking os ...
Warning: no dependencies configured.
Please read the following License Agreement. You must accept the terms of this
agreement before continuing with the installation.
press ENTER to continue
>>>
INVICTI SUBSCRIPTION SERVICES AGREEMENT

IMPORTANT - CAREFULLY READ ALL THE TERMS AND CONDITIONS OF THIS INVICTI SUBSCRIPTION SERVICES AGREEMENT ("SSA"). BY SI
GNING AN ORDER FORM INCORPORATING THIS AGREEMENT, CLICKING "I ACCEPT", CLICKING "CREATE", PROCEEDING WITH THE INSTALLA
TION AND/OR ACCESS AND USE OF THE INVICTI SOLUTION, OR USING THE INVICTI SOLUTION AS AN AUTHORIZED REPRESENTATIVE OF Y
OUR COMPANY NAMED ON THE APPLICABLE ORDER FORM ON WHOSE BEHALF YOU INSTALL AND/OR USE THE INVICTI SOLUTION, YOU ARE IN
DICATING THAT YOU HAVE READ, UNDERSTAND, AND ACCEPT THIS AGREEMENT WITH INVICTI (AS DEFINED BELOW). IF YOU DO NOT AGRE
E WITH ALL OF THE TERMS OF THIS AGREEMENT, DO NOT INSTALL, COPY, OR OTHERWISE USE THE INVICTI SOLUTION. THE EFFECTIVE
DATE OF THIS AGREEMENT SHALL BE THE DATE THAT YOU SIGN AN ORDER FORM WITH INVICTI OR OTHERWISE ACCEPT THIS AGREEMENT A
S SET FORTH ABOVE.

1. DEFINITIONS.

"Affiliates" means, with respect to a party at a given time, an entity that then is directly or indirectly controlled
by, is under common control with, or controls that party, and here "control" means an ownership, voting, or similar in
terest representing 50% or more of the total interests then outstanding of that entity.

Accept the license terms? [yes|no]
[no] >>> yes

Configuring acunetix user ...
    Creating user acunetix.

By default the Acunetix will be installed to /home/acunetix/.acunetix

Checking database port ...
Checking backend port ...
```



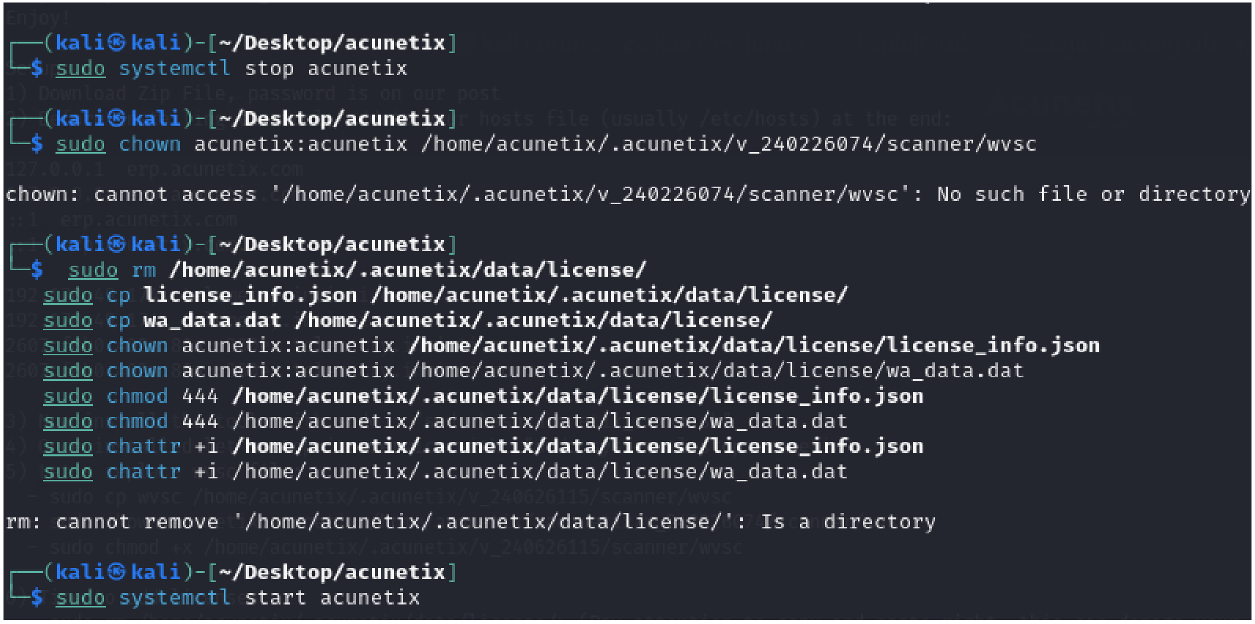
```
1. DEFINITIONS.

"Affiliates" means, with respect to a party at a given time, an entity that then is directly or indirectly controlled
by, is under common control with, or controls that party, and here "control" means an ownership, voting, or similar in
terest representing 50% or more of the total interests then outstanding of that entity.

Accept the license terms? [yes|no]
[no] >>> yes

Configuring acunetix user ...
    Creating user acunetix.

By default the Acunetix will be installed to /home/acunetix/.acunetix

Checking database port ...
Checking backend port ...
```

```
Configuring hostname ...
Insert new hostname, or leave blank to use kali
    Hostname [kali]:kali

Configuring the master user ...
    Email: kalitayab@kali.com
    Password:
    Password again:

Initializing file system ...

Extracting files to /home/acunetix/.acunetix....

Installing the database ...

Starting the database process ...
   - Create database
   - Update database structure
   - Populate database
   - Creating the master user
Stopping the database process ...

Generating certificates ...
 Generating certificate authority & certificates
```

Muhammad Tayab's report 3rd day of 3rd week



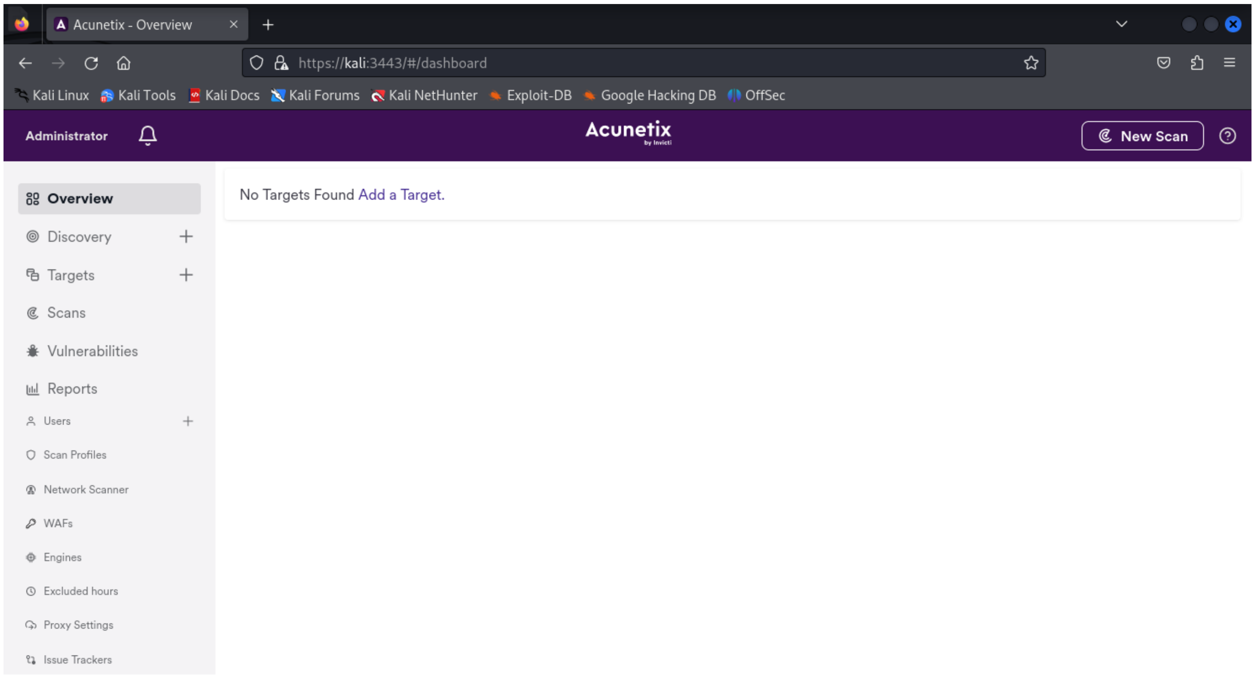
```
Generating certificate authority
Certificate authority generation succesful
Generating certificate ...
Certification generation succesful
Saving settings ...
Registering service ...
2024-08-07T10:59:59.979-0400    info    invicti supervisor service installed

Adding LSR shortcuts ...
Creating uninstall ...

Please visit https://kali:3443/ to access Acunetix UI

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo cp wvsc /home/acunetix/.acunetix/v_240626115/scanner/wvsc

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo chown acunetix:acunetix /home/acunetix/.acunetix/v_240226074/scanner/wvsc

chown: cannot access '/home/acunetix/.acunetix/v_240226074/scanner/wvsc': No such file or directory

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo chown acunetix /home/acunetix/.acunetix/v_240226074/scanner/wvsc

chown: cannot access '/home/acunetix/.acunetix/v_240226074/scanner/wvsc': No such file or directory

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo chmod +x /home/acunetix/.acunetix/v_240626115/scanner/wvsc

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo rm /home/acunetix/.acunetix/data/license/
rm: cannot remove '/home/acunetix/.acunetix/data/license/': Is a directory

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo systemctl stop acunetix

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo chown acunetix:acunetix /home/acunetix/.acunetix/v_240226074/scanner/wvsc
```



```
┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo systemctl stop acunetix

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo chown acunetix:acunetix /home/acunetix/.acunetix/v_240226074/scanner/wvsc

chown: cannot access '/home/acunetix/.acunetix/v_240226074/scanner/wvsc': No such file or directory

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$  sudo rm /home/acunetix/.acunetix/data/license/
sudo cp license_info.json /home/acunetix/.acunetix/data/license/
sudo cp wa_data.dat /home/acunetix/.acunetix/data/license/
sudo chown acunetix:acunetix /home/acunetix/.acunetix/data/license/license_info.json
sudo chown acunetix:acunetix /home/acunetix/.acunetix/data/license/wa_data.dat
sudo chmod 444 /home/acunetix/.acunetix/data/license/license_info.json
sudo chmod 444 /home/acunetix/.acunetix/data/license/wa_data.dat
sudo chattr +i /home/acunetix/.acunetix/data/license/license_info.json
sudo chattr +i /home/acunetix/.acunetix/data/license/wa_data.dat

rm: cannot remove '/home/acunetix/.acunetix/data/license/': Is a directory

┌──(kali㉿kali)-[~/Desktop/acunetix]
└─$ sudo systemctl start acunetix
```

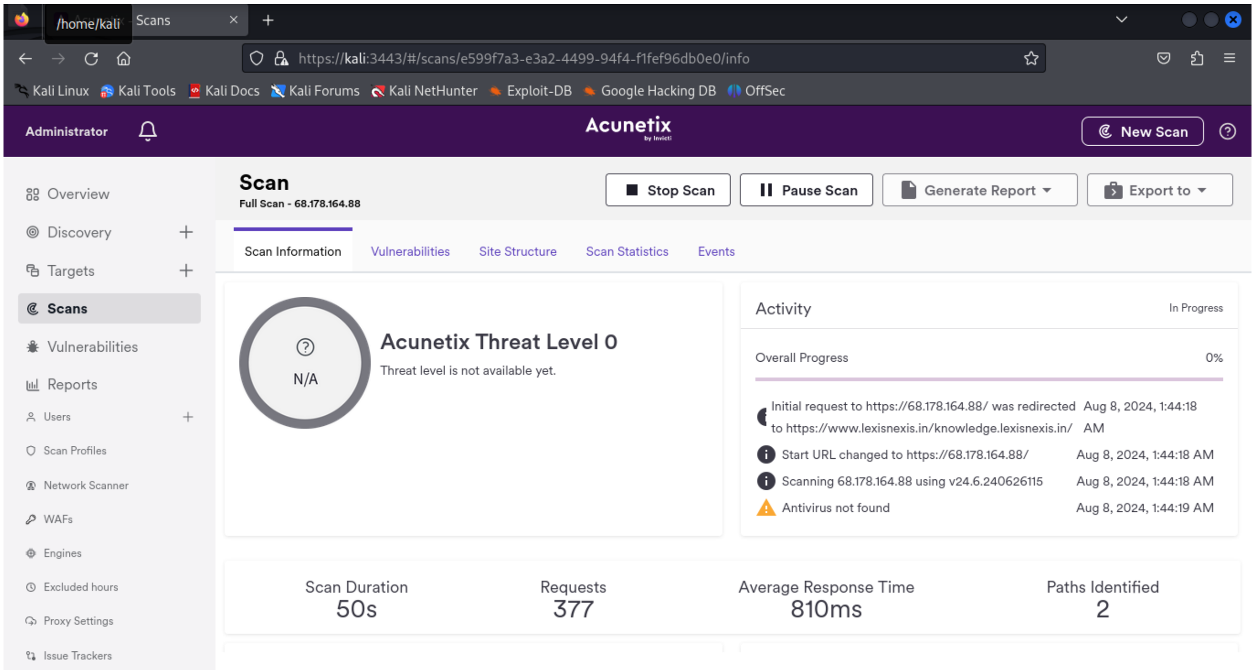
Muhammad Tayab's report 3rd day of 3rd week

Now Acunetix have been installed..so goto https://kali:3443/#/ then the page.



No Targets Found Add a Target.

I have added a target which is { 68.178.164.88} the website name is lexisnexis. It has started scanning.



Scan
Full Scan - 68.178.164.88

N/A

Acunetix Threat Level 0

Threat level is not available yet.

Activity — In Progress

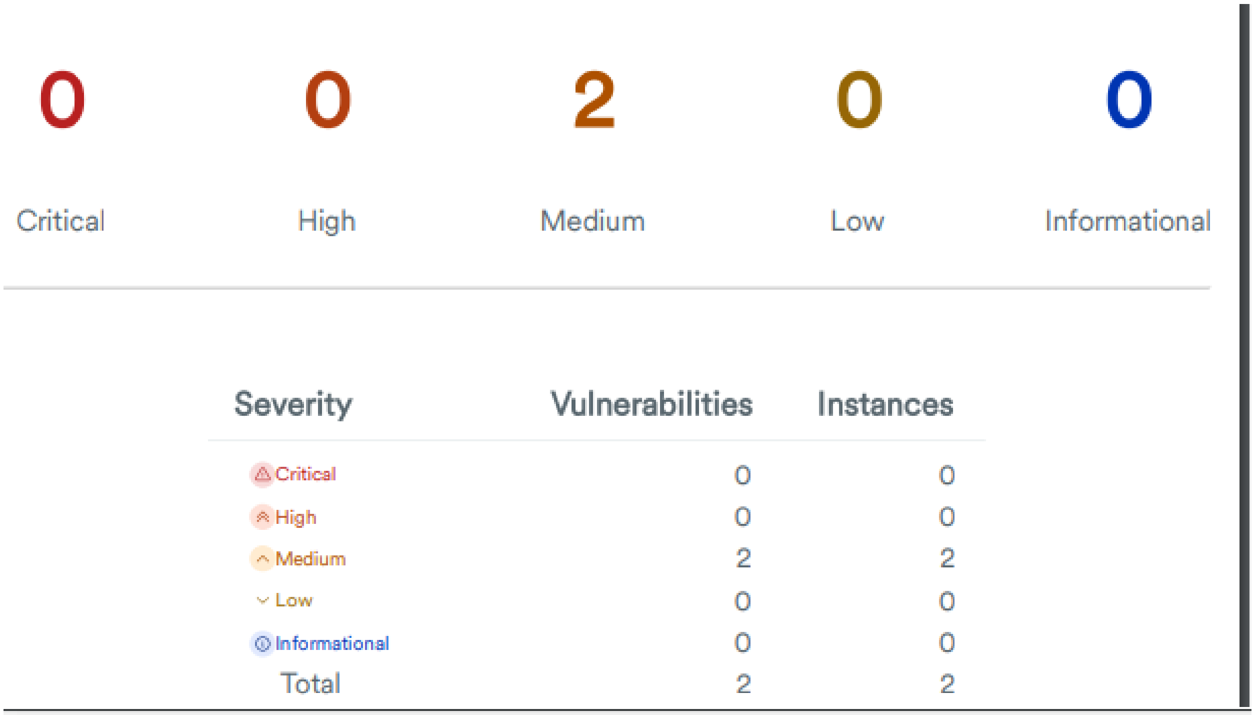
Overall Progress 0%

Initial request to https://68.178.164.88/ was redirected to https://www.lexisnexis.in/knowledge.lexisnexis.in/	Aug 8, 2024, 1:44:18 AM
Start URL changed to https://68.178.164.88/	Aug 8, 2024, 1:44:18 AM
Scanning 68.178.164.88 using v24.6.240626115	Aug 8, 2024, 1:44:18 AM
Antivirus not found	Aug 8, 2024, 1:44:19 AM

Scan Duration	Requests	Average Response Time	Paths Identified
50s	377	810ms	2

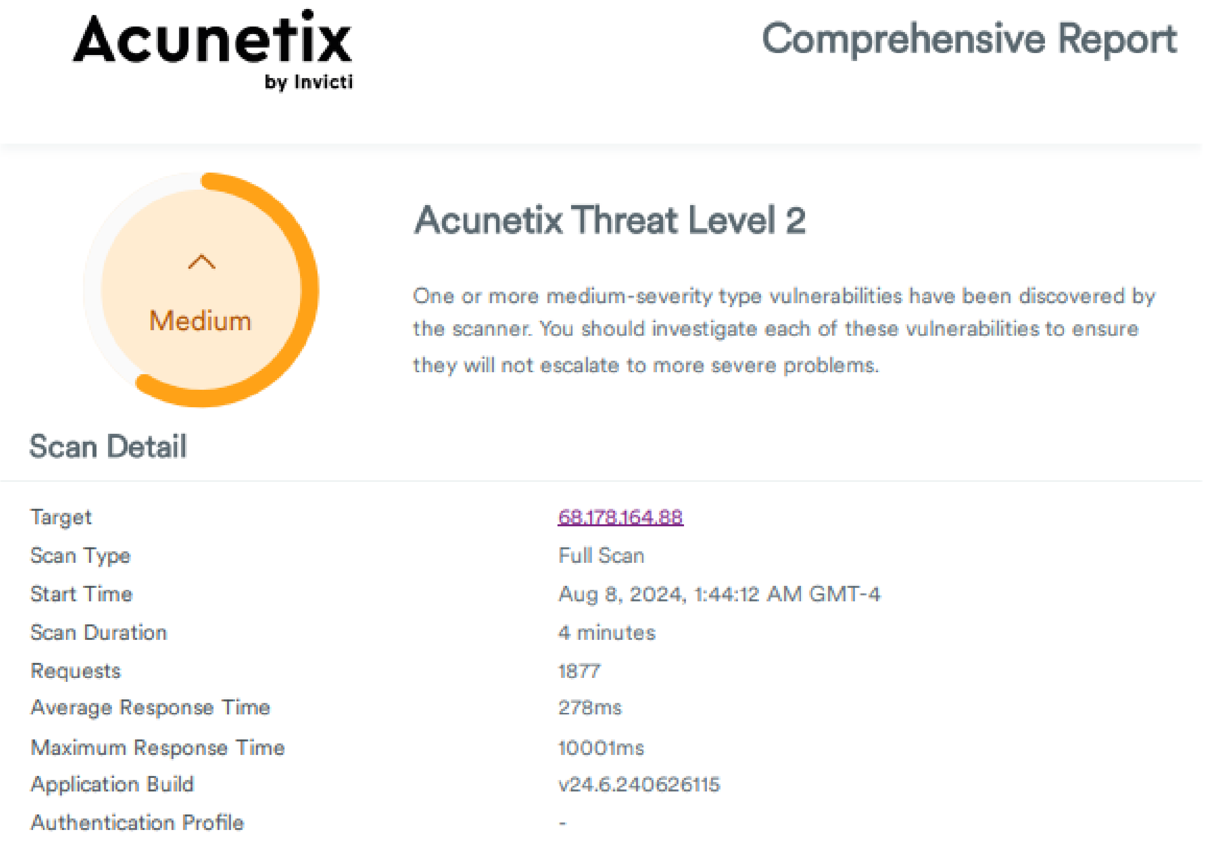
Muhammad Tayab's report 3rd day of 3rd week

Here is the report



0	0	2	0	0
Critical	High	Medium	Low	Informational

Severity	Vulnerabilities	Instances
⚠ Critical	0	0
High	0	0
Medium	2	2
Low	0	0
ⓘ Informational	0	0
Total	2	2



Acunetix by Invicti

Comprehensive Report

Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

Scan Detail

Target	68.178.164.88
Scan Type	Full Scan
Start Time	Aug 8, 2024, 1:44:12 AM GMT-4
Scan Duration	4 minutes
Requests	1877
Average Response Time	278ms
Maximum Response Time	10001ms
Application Build	v24.6.240626115
Authentication Profile	-

Muhammad Tayab's report 3rd day of 3rd week

OWASP ZAP

Definition

- OWASP ZAP (Zed Attack Proxy) is an open-source web application security scanner designed to find vulnerabilities in web applications.

Features

- **Automated Scanning**: Automated discovery of security vulnerabilities in web applications.
- **Manual Testing**: Tools for manual security testing.
- **Active Scanning**: Actively scans and tests the web application for security issues.
- **Passive Scanning**: Analyzes HTTP requests and responses without altering the traffic.
- **Spidering**: Automatically discovers site content and structure.
- **Fuzzer**: Tests for unexpected behavior using fuzzing techniques.
- **Proxying**: Allows manual inspection of HTTP requests and responses.
- **Session Management**: Handles different user sessions.
- **Reporting**: Generates detailed security reports.
- **API**: Provides an API for integration with other tools and CI/CD pipelines.

How It Works

1. **Proxy Setup**: ZAP acts as a proxy, intercepting HTTP/S traffic between your browser and the web application.
2. **Spidering**: ZAP spider crawls through the web application to discover all its pages and forms.
3. **Passive Scanning**: During spidering, ZAP passively scans HTTP responses for vulnerabilities.
4. **Active Scanning**: Actively probes discovered pages and forms to find security issues.
5. **Fuzzing**: Sends invalid or unexpected data to inputs to discover vulnerabilities.
6. **Reporting**: Generates a report detailing discovered vulnerabilities and their severity.

How to Use It

1. **Installation**: Download and install OWASP ZAP from the official website.
2. **Launch ZAP**: Start the application.
3. **Set Up Proxy**: Configure your browser to use ZAP as a proxy.
4. **Explore Your Application**:
   - **Manual Exploration**: Browse through your application with your browser while ZAP records the traffic.
   - **Automated Exploration**: Use ZAP’s spider to automatically discover pages.
5. **Run Scans**:
   - **Passive Scan**: Automatically runs while you explore the application.
   - **Active Scan**: Initiate an active scan on the discovered resources.
6. **Fuzz Testing**: Use the fuzzer tool to test for vulnerabilities by sending varied inputs to application fields.
7. **Analyze Results**: Review the list of discovered vulnerabilities, their details, and suggested remediation steps.
8. **Report Generation**: Generate detailed security reports for documentation and further analysis.

Muhammad Tayab's report 3rd day of 3rd week

Features and Functions in Detail

1. **Automated Scanning**:
   - **Quick Start**: Automatically scans a target URL for common vulnerabilities.
   - **Customizable Scans**: Configure scan settings to target specific parts of the application.
2. **Manual Testing**:
   - **Break Points**: Set breakpoints to intercept and modify requests and responses.
   - **Forced Browsing**: Test application access controls by manually navigating to restricted resources.
3. **Active Scanning**:
   - **Attack Modes**: Automatically probes the application for vulnerabilities.
   - **Targeted Scans**: Focus scans on specific URLs or parameters.
4. **Passive Scanning**:
   - **Background Analysis**: Analyzes traffic for security issues without affecting the application.
5. **Spidering**:
   - **Content Discovery**: Finds all URLs and forms by crawling the web application.
   - **Ajax Spider**: Handles applications that use heavy JavaScript/AJAX.
6. **Fuzzer**:
   - **Customizable Payloads**: Send varied data inputs to discover vulnerabilities.
   - **Input Validation**: Test how the application handles unexpected or invalid data.
7. **Proxying**:
   - **Manual Inspection**: Inspect and modify HTTP requests and responses.
   - **Session Handling**: Manage different user sessions during testing.
8. **Session Management**:
   - **User Authentication**: Handle login mechanisms and maintain user sessions.
   - **Session Tokens**: Test for session fixation and hijacking issues.
9. **Reporting**:
   - **Vulnerability Reports**: Detailed reports on discovered security issues.
   - **Export Options**: Export reports in various formats (HTML, XML, JSON).
10. **API**:
    - **Automation**: Integrate ZAP with CI/CD pipelines for automated security testing.
    - **Scripting**: Extend ZAP functionality using custom scripts.

Muhammad Tayab's report 3rd day of 3rd week

Scanning for Insider Threats

Definition

- Insider threats involve risks posed by employees, contractors, or other trusted individuals who have access to an organization’s systems and data. These threats can be intentional, such as data theft or sabotage, or unintentional, such as accidental data leaks.

Tools and Techniques

1. **User Behavior Analytics (UBA)**
    - **Description**: Analyzes user behavior patterns to detect anomalies that could indicate insider threats.
    - **Tools**:
        - **Splunk User Behavior Analytics**: Monitors user activity and detects deviations from normal behavior.
        - **Exabeam**: Uses machine learning to identify abnormal user behavior.
    - **Example**: Detecting an employee downloading large amounts of sensitive data at unusual times.
2. **Data Loss Prevention (DLP)**
    - **Description**: Monitors and controls the flow of sensitive information to prevent unauthorized access or leaks.
    - **Tools**:
        - **Symantec DLP**: Prevents data breaches by monitoring and protecting data in use, in motion, and at rest.
        - **McAfee Total Protection for DLP**: Provides comprehensive data protection across devices and cloud services.
    - **Example**: Blocking an attempt to send confidential documents to an external email address.
3. **Identity and Access Management (IAM)**
    - **Description**: Manages user identities and their access to resources, ensuring only authorized users have access to sensitive information.
    - **Tools**:
        - **Okta**: Provides secure identity management and single sign-on solutions.
        - **Azure Active Directory**: Manages user identities and access permissions in Microsoft environments.
    - **Example**: Implementing multi-factor authentication (MFA) for accessing critical systems.
4. **Endpoint Detection and Response (EDR)**
    - **Description**: Monitors and responds to threats on endpoints (devices) within a network.
    - **Tools**:
        - **CrowdStrike Falcon**: Provides real-time endpoint monitoring and threat detection.
        - **Carbon Black**: Offers advanced threat detection and response capabilities for endpoints.
    - **Example**: Detecting and responding to malware introduced by a malicious insider.
5. **SIEM (Security Information and Event Management)**

- **Description**: Collects, analyzes, and correlates security event data from across the organization to detect and respond to threats.
- **Tools**:
  - **Splunk Enterprise Security**: Provides real-time security monitoring and incident response.
  - **IBM QRadar**: Analyzes and correlates data to identify security incidents.
- **Example**: Correlating log data to detect patterns indicative of insider threats, such as repeated access attempts to restricted files.

6. **Network Traffic Analysis**
   - **Description**: Monitors network traffic to detect unusual patterns or behaviors that could indicate insider threats.
   - **Tools**:
     - **Darktrace**: Uses AI to detect and respond to network threats in real-time.
     - **NetWitness**: Provides visibility into network traffic and helps detect advanced threats.
   - **Example**: Identifying abnormal data transfer volumes or unusual access times.

Examples of Insider Threat Detection

1. **Unusual Access Patterns**:
   - **Scenario**: An employee accesses sensitive data during non-working hours.
   - **Tool**: User Behavior Analytics (UBA)
   - **Detection**: The UBA system flags the behavior as anomalous based on typical user activity patterns.
2. **Data Exfiltration**:
   - **Scenario**: A contractor attempts to download large amounts of data to an external device.
   - **Tool**: Data Loss Prevention (DLP)
   - **Detection**: The DLP system detects the unusual data transfer and blocks the action, alerting security personnel.
3. **Unauthorized Privilege Escalation**:
   - **Scenario**: An employee tries to gain administrative access to critical systems without authorization.
   - **Tool**: Identity and Access Management (IAM)
   - **Detection**: The IAM system denies the access attempt and logs the incident for further investigation.

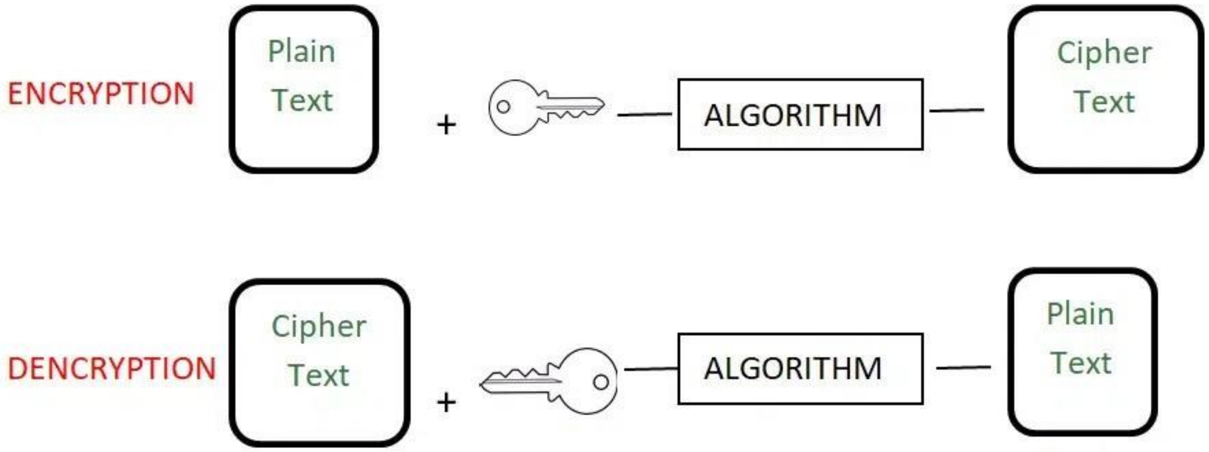
Encryption vs Encoding vs Hashing

Encryption, Encoding, and Hashing are similar kinds of things and have little difference between them. They all are used to change the format of the data or data transformation for different purposes.

Encryption

Encryption is an encoding technique in which a message is encoded by using an encryption algorithm in such a way that only authorized personnel can access the message or information. It is a special type of encoding that is used for transferring private data, for example sending a combination of username and password over the internet for email login. In encryption, data to be encrypted(called plain text) is transformed using an encryption algorithm like AES or RSA Encryption Algorithm using a secret key called a cipher. The encrypted data is called ciphertext, and finally, the secret key can be used by the intended recipient to convert it back to plain text. There are two types of encryption algorithms – symmetric and asymmetric encryption. In the case of symmetric encryption, data is encoded and decoded with the help of the same key whereas in the case of Asymmetric encryption, data is encoded and decoded with the help of different keys, that is public key and private key.

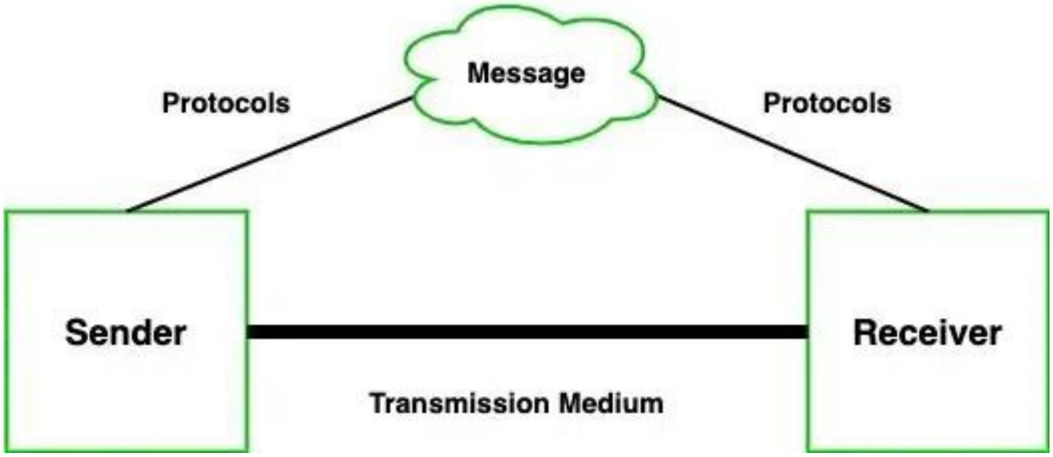
Example: AES Encryption Algorithm but in the case of the asymmetric encryption algorithm, data is encrypted with the help of two keys, namely the public and private keys, like the RSA algorithm.



Encoding

In the Encoding method, data is transformed from one form to another. The main aim of encoding is to transform data into a form that is readable by most of the systems or that can be used by any external process. It can’t be used for securing data, various publicly available algorithms are used for encoding. Encoding can be used for reducing the size of audio and video files. Each audio and video file format has a corresponding coder-decoder (codec) program that is used to code it into the appropriate format and then decodes it for playback.

Example: ASCII, BASE64, UNICODE

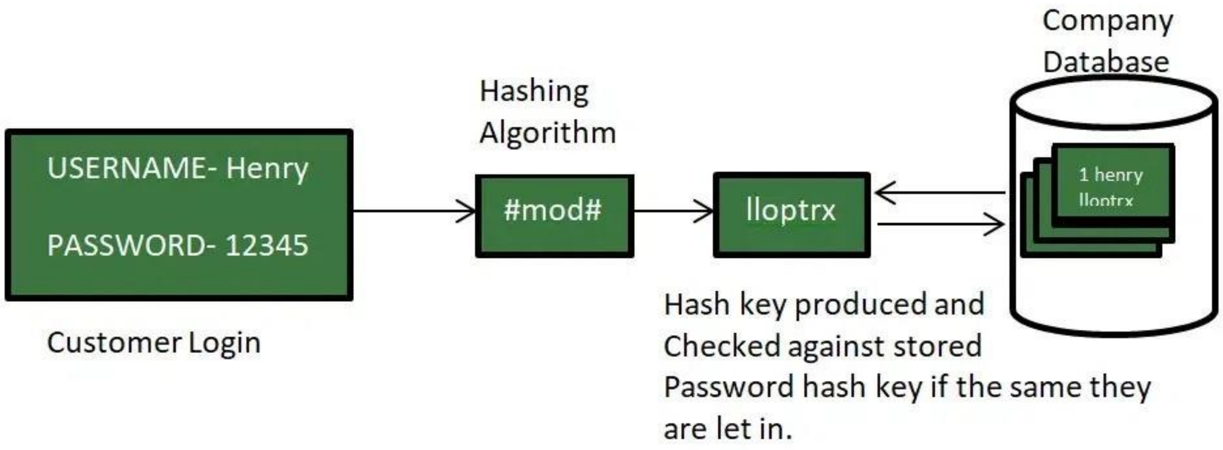


Muhammad Tayab's report 3rd day of 3rd week

Hashing

In Hashing, data is converted to the hash using some hashing function, which can be any number generated from a string or text. Various hashing algorithms are MD5 and SHA256. Data once hashed is non-reversible. The hash function can be any function that is used to map data of arbitrary size to data of fixed size. The data structure hash table is used for storing data.

Example: When you send pictures and text messages over WhatsApp over StackOverflow (posting in questions), images are sent to different servers, and text is sent to a different server for efficiency purposes. So for verifying images that the images are not tampered with between data transfers over the internet, a hashing algorithm like MD5 can be used. MD5 generates a message digest of 128 bits, while SHA1 generates a message digest of the 160-bit hash value. Hence, SHA1 is a relatively complex algorithm with better security than MD5. Another purpose for hashing is for verifying passwords for login on various websites, as shown in the image.



Muhammad Tayab's report 3rd day of 3rd week

Difference Between Encryption, Encoding, and Hashing

Encryption	**Encoding**	**Hashing**
Encryption is a type of encoding technique where the message is encoded using an encryption algorithm so that only authorized persons can access that information.	Encoding is a technique where the data is transformed from one form to another.	Hashing is a technique where the data is converted to hash using different algorithms present there.
Encryption is a technique used for protecting the confidentiality of the data.	Encryption is used for preserving the usability of the data.	Hashing is simply used for checking the integrity of the data.
Appropriate Keys are used in the Encryption.	No Keys are used in Encoding.	No Keys are used in Hashing.
Encryption can be reversed back to its original form by using appropriate keys.	Encoding can be reversed back to its original form.	The hashed one cannot be reversed back to its original form.
Example: AES Algorithm, RSA Algorithm, Diffie Hellman	**Example:** BASE64, UNICODE, ASCII, URL Encoding.	**Example:** MD5, SHA256, SHA – 3.