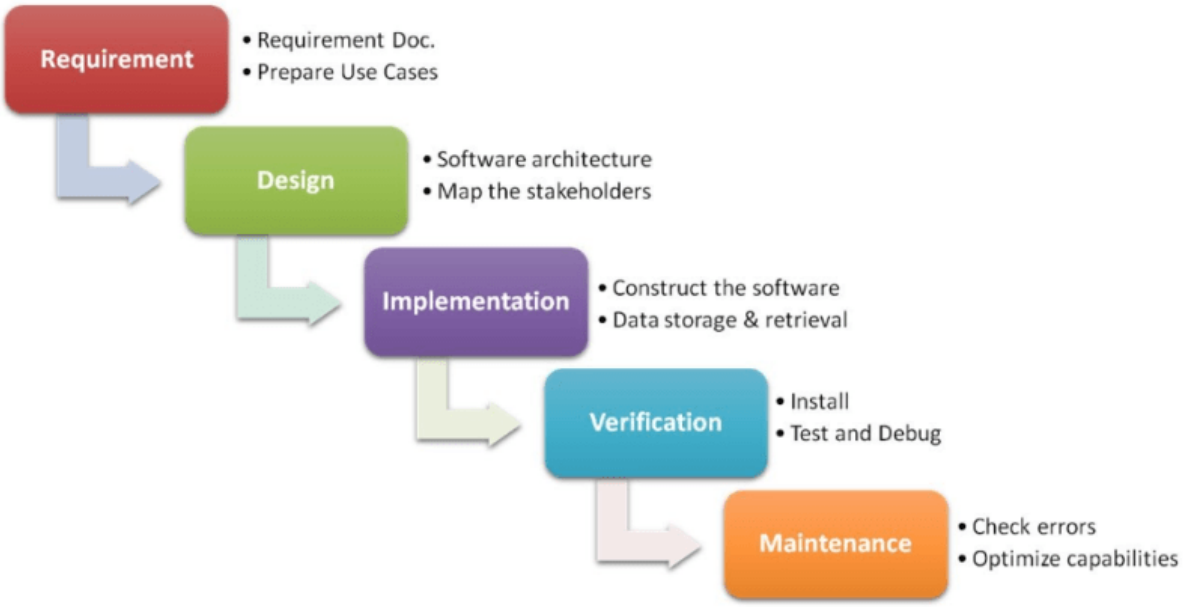
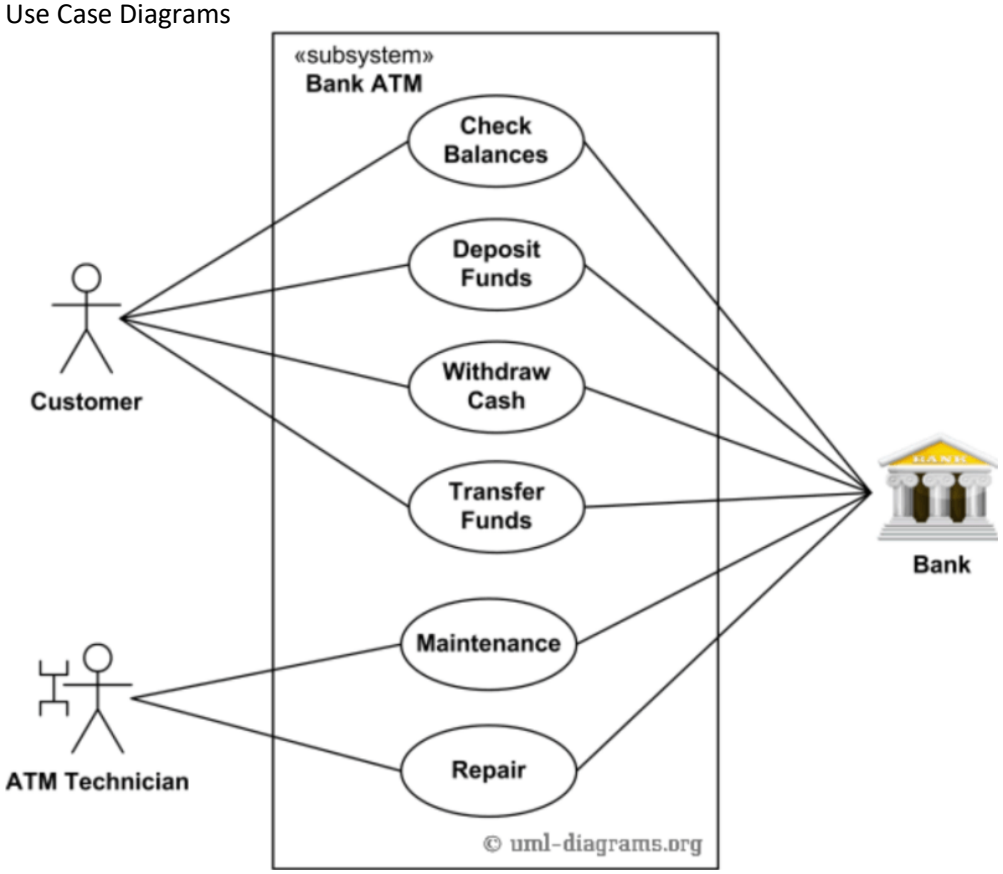
Software Development Life Cycle and Waterfall Model

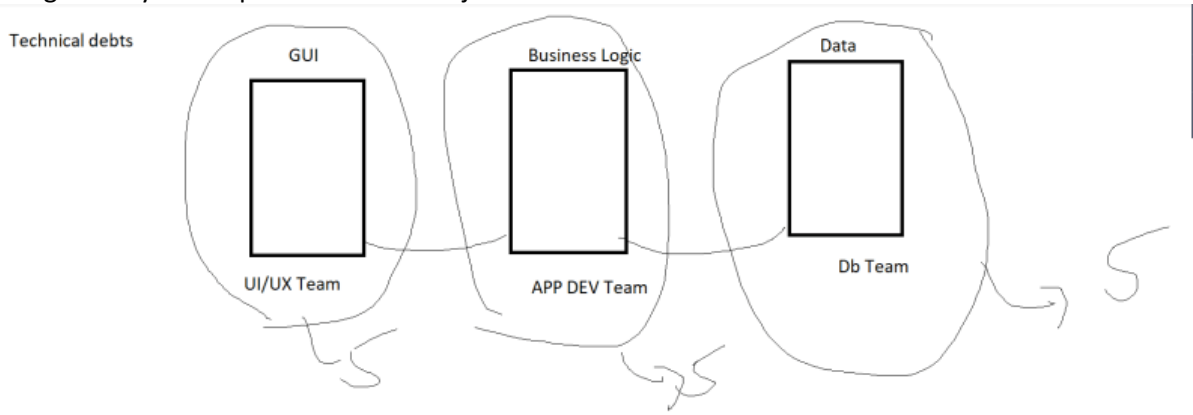
- Waterfall Model



- Requirements:
  - Create a High-Level Requirements Document
  - Use Case Diagrams



- Big Bang Integrations:
    - Integration of applications/components developed individually in the last phases of project
    - This generally ends up with issues & major reason for technical debt's



- Continuous Integration:
    - Here we integrate different components of the application from day 1 and ensure we run some integration tests
    - The basic idea is to fail fast and know the errors upfront.

- Agile Software Methodologies:
    - https://agilemanifesto.org/ for the Agile Manifesto
    - DevOps enhances the idea of agile by approach which is shift left
- The idea of DevSecOps is to make security shift left.
- What is that we can do for this shift left of Security
    - Code to be scanned for securiy vulnerabilities
    - Scanning application deployed in various test environments for security vulnerabilities
    - Reporting the security issues and eventually breaking build when security issues are reported.

- DevSecOps Manifesto

Leaning in over Always Saying "No"

Data & Security Science over Fear, Uncertainty and Doubt

Open Contribution & Collaboration over Security-Only Requirements

Consumable Security Services with APIs over Mandated Security Controls & Paperwork

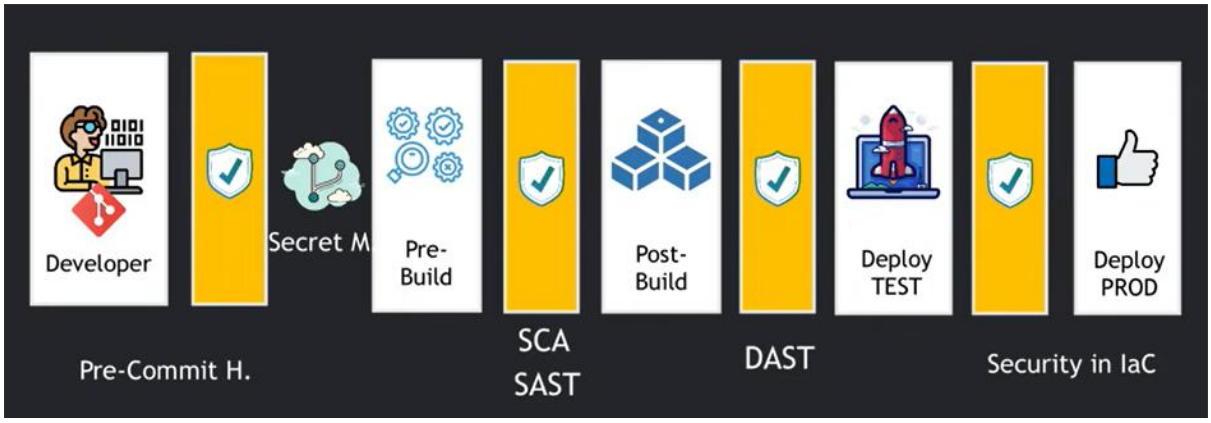
Business Driven Security Scores over Rubber Stamp Security

Red & Blue Team Exploit Testing over Relying on Scans & Theoretical Vulnerabilities

24x7 Proactive Security Monitoring over Reacting after being Informed of an Incident

Shared Threat Intelligence over Keeping Info to Ourselves

Compliance Operations over Clipboards & Checklists

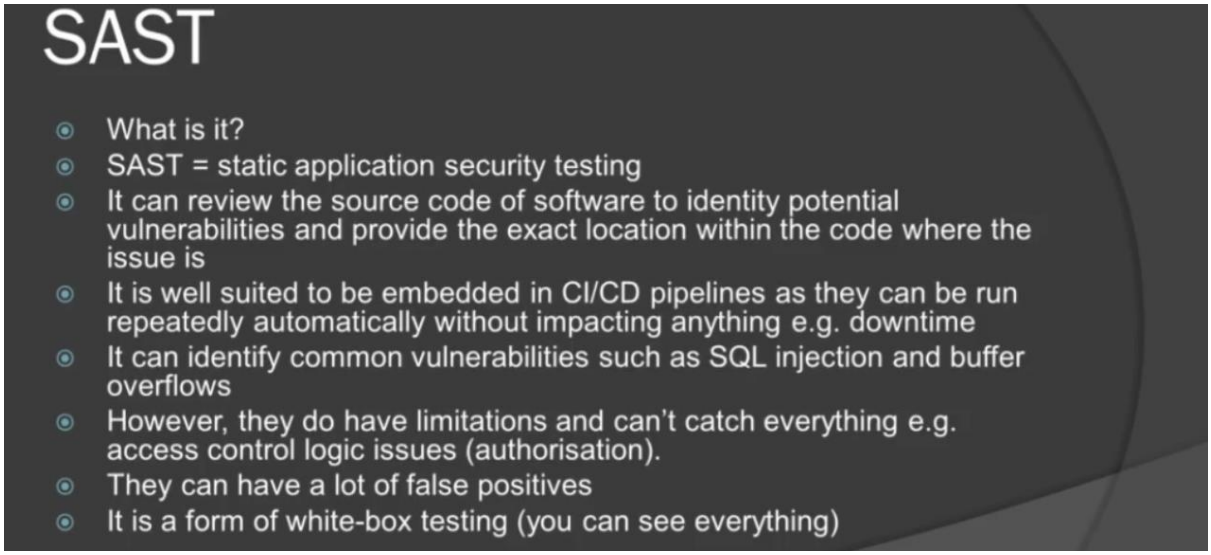

Developer
Secret M
Pre-
Build
Post-
Build
Deploy
TEST
Deploy
PROD
Pre-Commit H.
SCA
SAST
DAST
Security in IaC

Dependency in Code

- Any application which developers code, they rely on open-source libraries or packages or frameworks for
    - web applications
    - Database connections and query executions (ORM Frameworks)
    - Logging
    - Authentication and Authorizations
    - Notifications.
- Scanning Dependencies for security risks is a mandatory operation as part of DevSecOps.
- If we need to scan dependencies from security issues, we need a database of possible vulnerabilities => CVE and NVD

Static vs Dynamic Security Testing

- Two possible ways to test for security
    - Static:
        - When the tool scans the application with the knowledge of code and reports vulnerability
        - This is called as SAST
    - Dynamic:
        - When the tool scans the application which is running and doesn't have access to source code.
        - This is called as DAST

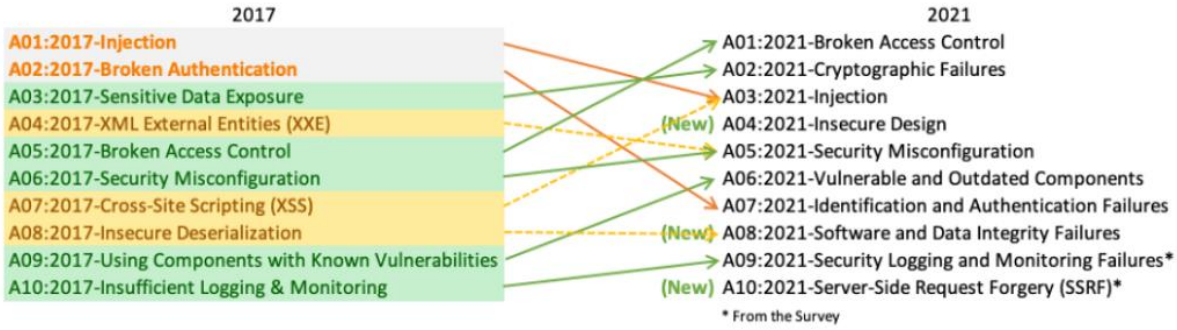


DAST

- What is it?
- DAST = dynamic application security testing
- It can test a running application through its UI or API for common vulnerabilities such as SQL injection and buffer overflows
- It *can* be embedded into CI/CD pipelines, but it would more be for completeness and are primarily aimed at manual testing as part of a dedicated penetration test
- It is a form of black-box testing (it can't see everything behind the scenes)

OWASP (Open Web Application Security ProjectÂ® (OWASP))

- This organization publishes top issues to be concerned with early.
- They also give necessary tools to scan
- https://owasp.org/Top10/ for OWASP 10:2021
- As a DevSecOps Engineer, We will be
    - performing SAST during packaging/building the application
    - performing DAST post application deployment
    - Ensuring our application doesnot have any issues mentioned/listed in OWASP TOP 10 (latest year)



```
public Person find_Unsecure(String inputFromUser) {
    // select * from person where name=Ranga or 1 = 1
    return jdbcTemplate.queryForObject
            ("select * from person where name="
                    +inputFromUser,
            new BeanPropertyRowMapper<Person>(Person.class));
}

public Person find_secure(String inputFromUser) {
    // select * from person where name='Ranga or 1 = 1'
    return jdbcTemplate.queryForObject
            ("select * from person where name=?"
            ,new Object[] { inputFromUser },
            new BeanPropertyRowMapper<Person>(Person.class));
```

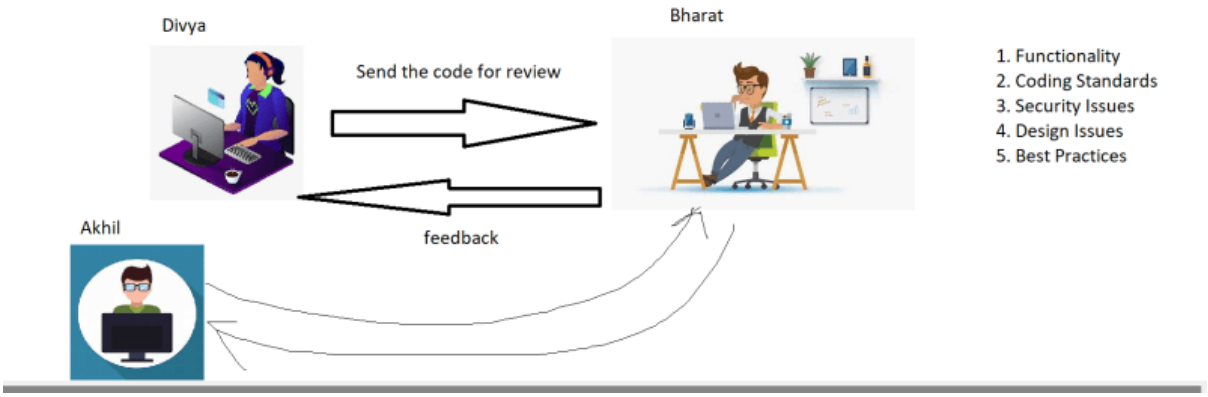
GDPR: The **General Data Protection Regulation** (**GDPR**) is a European Union regulation on data protection and privacy in the EU and the European Economic Area (EEA)

PCI-DSS: The Payment Card Industry Data Security Standard (PCI DSS) is a set of security standards formed in 2004 by Visa, MasterCard, Discover Financial Services, JCB International and American Express. Governed by the Payment Card Industry Security Standards Council (PCI SSC), the compliance scheme aims to secure credit and debit card transactions against data theft and fraud.

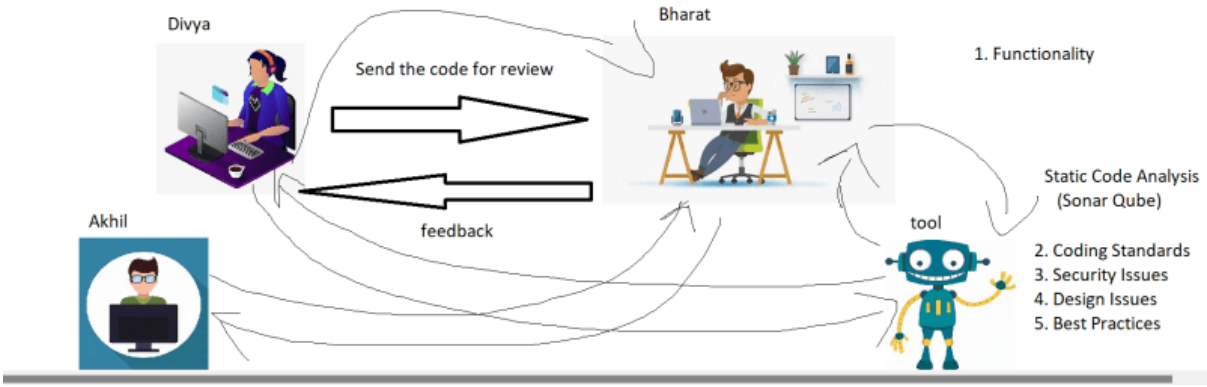
OWASP Organisation list down all the security attack happen for the year, categorized them, and provide guideline to develop software.

Static Code Analysis

- Peer Code Review



- Since a person is doing the review, mistakes can happen.
- Static Code Analysis tools take over certain responsibilites from Bharat
    - Coding Standards
    - Best Practices
    - Securiy Issues
    - Design Issues
    - Test Case Quality



- If we can run this static code analysis with every change in the pipeline and
    - Phase-1 : Show the report generated
    - Phase-2: Fail the build if the agreed criteria is not met (Quality Gate)

Testing By DevTeams

- Unit Tests: Developers are expected to write tests to check the code developed by team
- When developers perform unit testing, there should be a way to measure the quality of unit tests.
- Line Coverage
- Branch Coverage
- Note: As a devops Engineer, we are expected to create Quality Gates around
    - Test Coverage
    - Static Code Analysis issues

DevSecOps

- This is the practice that involves security earlier in SDLC.
- To implement DevSecOps, Organizations consider variety of applications security tools (AST) to integrated with various stages of CI/CD Process. Commonly used AST tools include
- SCA (Software Composition Analysis)
- SAST (Static Application Security Testing)
- DAST (Dynamic application security Testing)

Software Compostion Analysis (SCA)

- SCA tools scan source code and binaries to identify known vulnerabilities in open source and third-party components.
- They also provide insight into security and license risks.

Static application Security Testing (SAST)

- These tools scan propietary code or custom code for coding errors and design flaws that could lead to exploitable weakness.

Dynamic Application Security Testing (DAST)

- DAST is automated opaque black box testing technology that mimics how a hacker could interact with your web application or API.
- This tests application over a network connection & by examining the client side rendering of application.

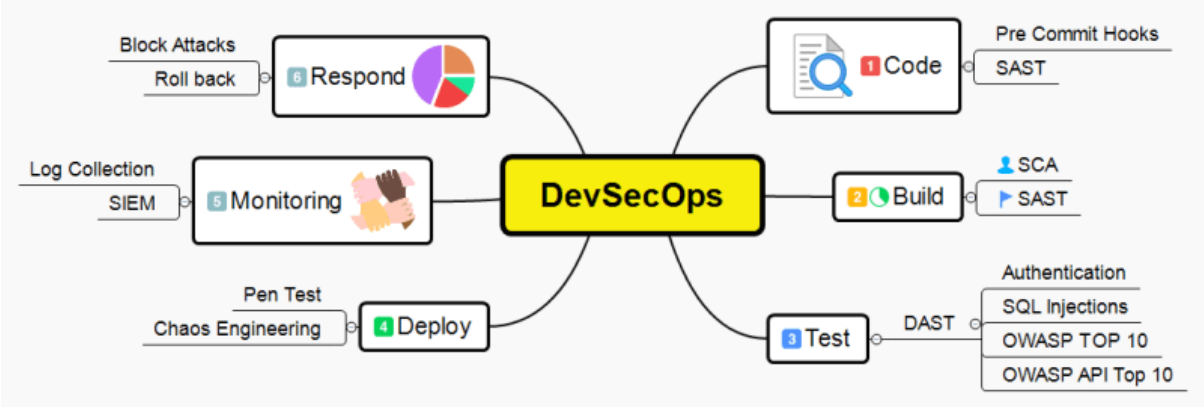
DevSecOps Tools

- Aqua Security:
    - Used with cloud-native applications i.e cloud native application protection platform (CNAPP).
    - This is very popular for kubernetes, serverless, container security etc
    - https://www.aquasec.com/ for the offical web page for aqua security
- Checkmarx:
    - This is very popular is application security testing (AST).
    - We can perform
        - SCA
        - SAST
        - Interactive Application Secirity testing
    - https://checkmarx.com/ for the official web page for CheckMarx
- Micro Focus Cyber Res Fortify:
    - This is very popular in IDE scanning of the code and they offer different products around
        - SAST
        - DAST
        - SCA
    - https://www.microfocus.com/en-us/cyberres/application-security for the official web page for Fortity

- Synopsys:
    - AST tools include SCA, interactive,DAST and SAST
    - https://www.synopsys.com/software-integrity/solutions/devsecops.html for the official web page
- Veracode:
    - This is cloud solution provider for SAST
    - https://www.veracode.com/ for veracode
- WhiteSource:
    - This offers SAST, dependecy scanning and risk exposure
    - https://www.mend.io/ for official web page
- OWASP ZAP:
    - This is from OWASP community which is opensource.
    - Automated active and passive scanning of web applications for vulnerabilities
    - This is DAST testing
    - https://owasp.org/www-project-zap/ for the official pages for OWASP ZAP

Integrating Security To CI/CD Pipelines

- Overview of Integration



Terms To Be Understood

- OWASP
- OWASP TOP 10
- SIEM - Security information and event management
- NVD - National Vulnerability Database
- CVE- Common Vulnerabilities and Exposures

NVD (National Vulnerability Database)

- This is list of all known vulnerabilities

CVE

- This is a number given to vulnerability and we can search vulnerabilities by technology/platform which we use

Vulnerability Sources

- Proprietary Code
- Dependencies/libraries/frameworks your application is using
- Network
- Hardware
- Operating Systems.
- Container

OWASP

- https://owasp.org/ for the official website
- OWASP TOP 10 https://owasp.org/www-project-top-ten/
- OWASP API TOP 10 https://owasp.org/www-project-api-security/

DevSecOps (Contd)

- Here is the list of some opensource free tools for SAST, SCA and DAST https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools

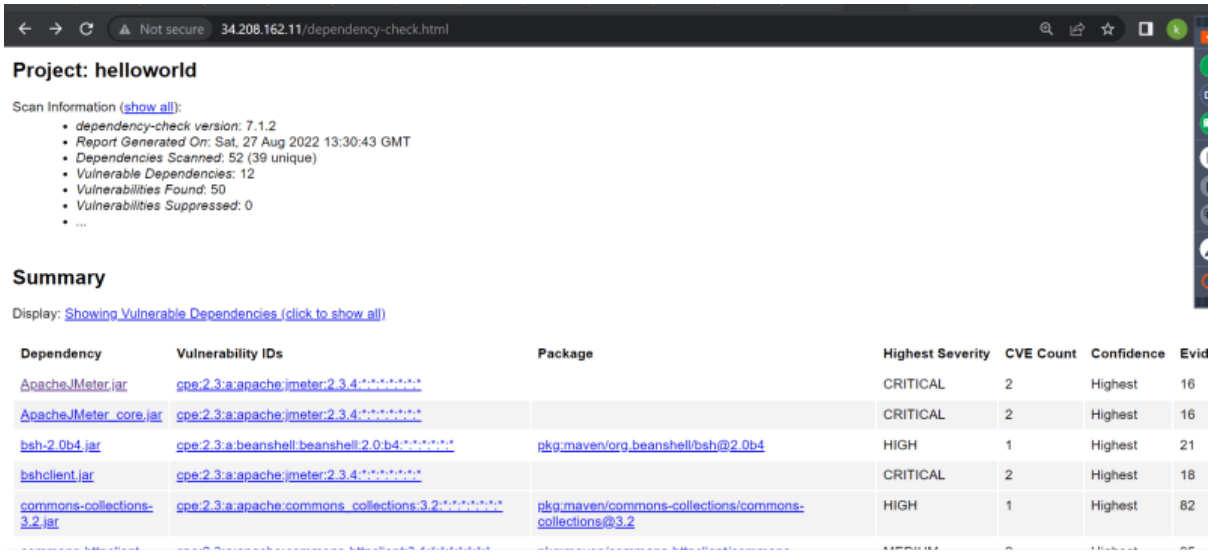
OWASP Dependency Check

- https://owasp.org/www-project-dependency-check/ for the official web site
- https://jeremylong.github.io/DependencyCheck/ for the documentation of dependency check
- Installation:
    - Ensure JAVA is installed

sudo apt update
sudo apt install openjdk-11-jdk -y

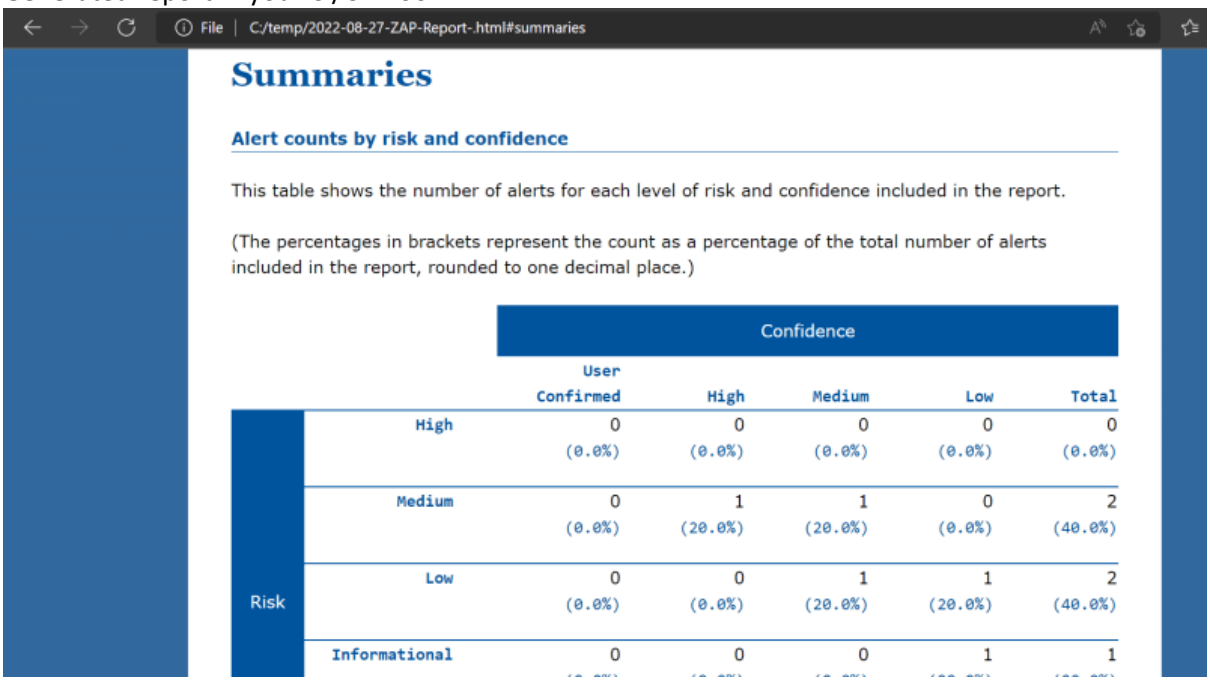
- Download the dependency check https://objects.githubusercontent.com/github-production-release-asset-2e65be/5663857/3535be4b-a468-41e0-9979-1215580abe52?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220827%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220827T131135Z&X-Amz-Expires=300&X-Amz-Signature=0714be8bdc48d95b955b506985ea7a39ec975e849448832c379ceae656282ecc&X-Amz-SignedHeaders=host&actor_id=2438317&key_id=0&repo_id=5663857&response-content-disposition=attachment%3B%20filename%3Ddependency-check-7.1.2-release.zip&response-content-type=application%2Foctet-stream
- To upload this into linux use sftp
    - Get into the directory where you have zip downloaded
    - Figure out ssh command to connect to ubuntu instance, replace ssh with sftp `sftp -i ~/Downloads/ansiblelearning.pem ubuntu@100.100.100.100
    - upload using put <filename>
- Now install unzip sudo apt install unzip -y

unzip ~/dependency-check-7.1.2-release.zip
cd ~
git clone https://github.com/wakaleo/game-of-life.git
cd game-of-life/
~/dependecy-check/dependency-check/bin/dependency-check.sh --project "helloworld" --scan ~/game-of-life/



OWASP ZAP (DAST)

- Install OWASP ZAP either from chocolatey or manual install and then scan any url
- Generated report in your CI/CD Tool



- When we work with CI/CD we generate this report by running ZAP from Command line https://www.zaproxy.org/docs/desktop/cmdline/

Activities

1. Create a Synk account https://snyk.io/plans/
2. Ensure you have a Github account and docker hub account.

Docker Image Scanning

- In CI/CD Pipelines we build docker images, so we are expected to scan images for vulnerabilities.
- Docker has its own scan as part of its command line
- To perform extensive Scanning Organizations opt for third party tools
    - Aqua Security
    - ECR Scanning
    - Microsoft Defender for Image Scanning
    - Qualysis
- Refer the below two part articles
    - part 1 https://www.prplbx.com/resources/blog/docker-part1/
    - part 2 https://www.prplbx.com/resources/blog/docker-part2/
- CIS benchmark for docker
- Scanning for vulnerabilities for docker local images
- Docker images are store in Registries (Docker Hub, Elastic Container Registry, Azure Container Registry) and all of them support image scanning.

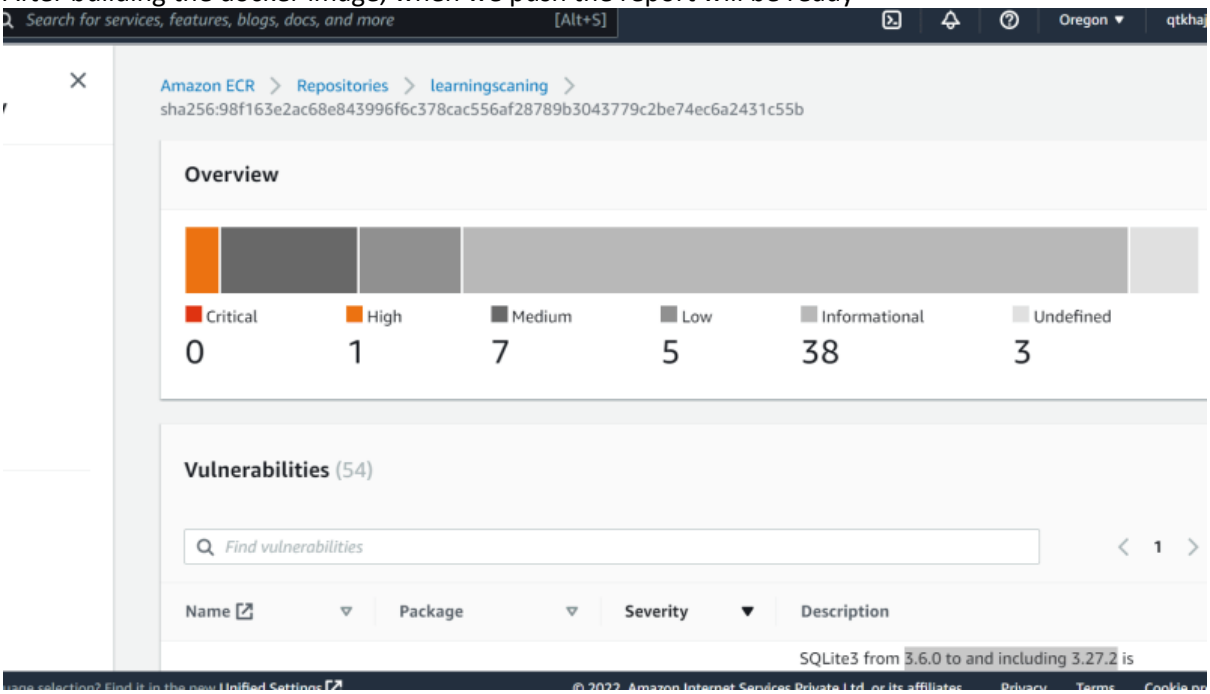
Sample Scanning of Docker images in Registries

AWS ECR

- Sample Dockerfile

FROM httpd
EXPOSE 80

- After installing docker on the build server
- Create a repository in AWS ECR and select the option to scan on push
- After building the docker image, when we push the report will be ready



- Configuration required on your build server
    - aws cli

    - Create an IAM user and configure
    - Execute aws configure
- In Azure, lets do this activity after some time.

Linux Vulnerability Scanning

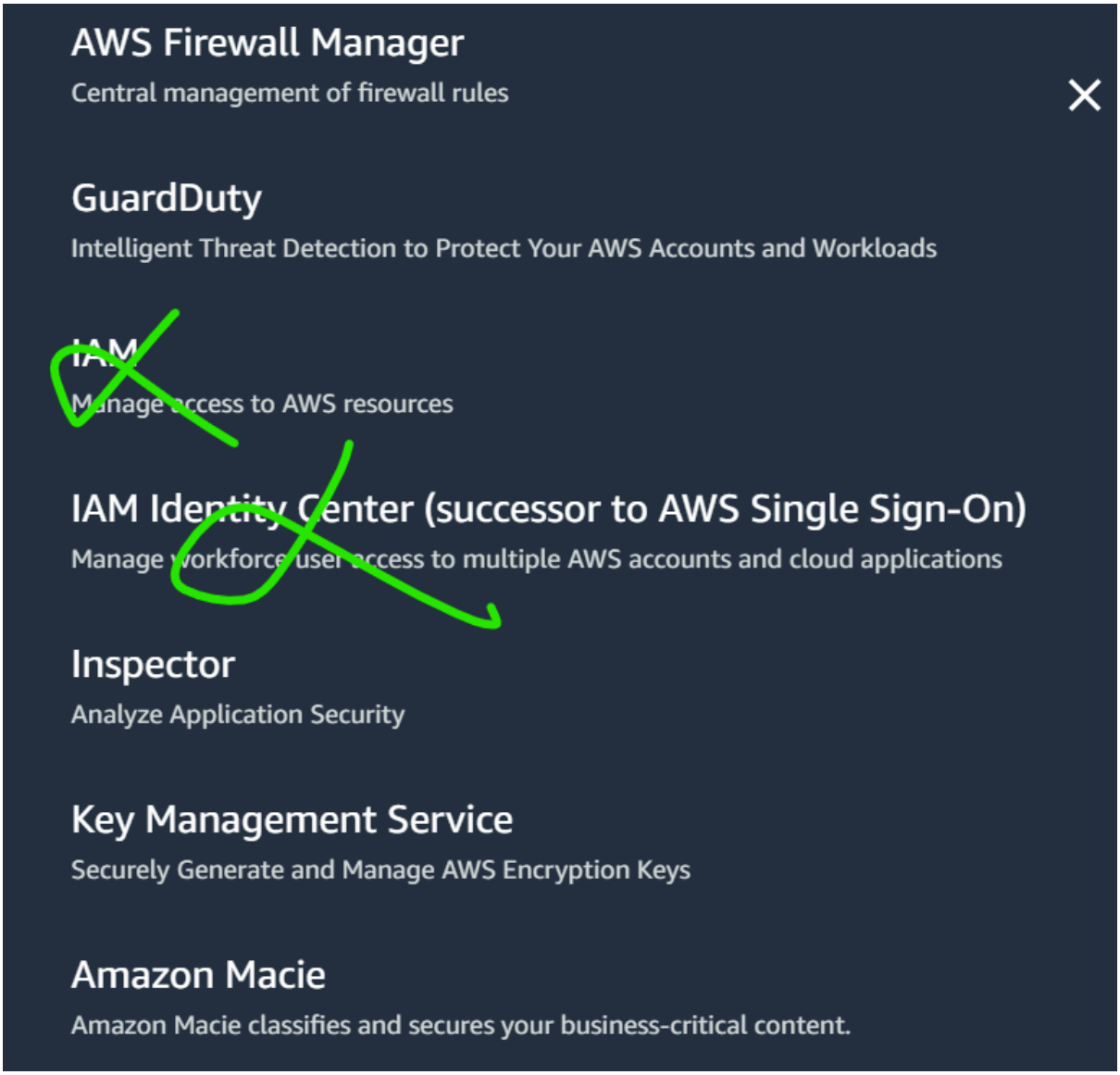
- Scanning Linux For Vulnerabiliteis https://geekflare.com/linux-security-scanner/
- Nessus https://www.tenable.com/products/nessus
- Open VAS https://www.openvas.org/ ss

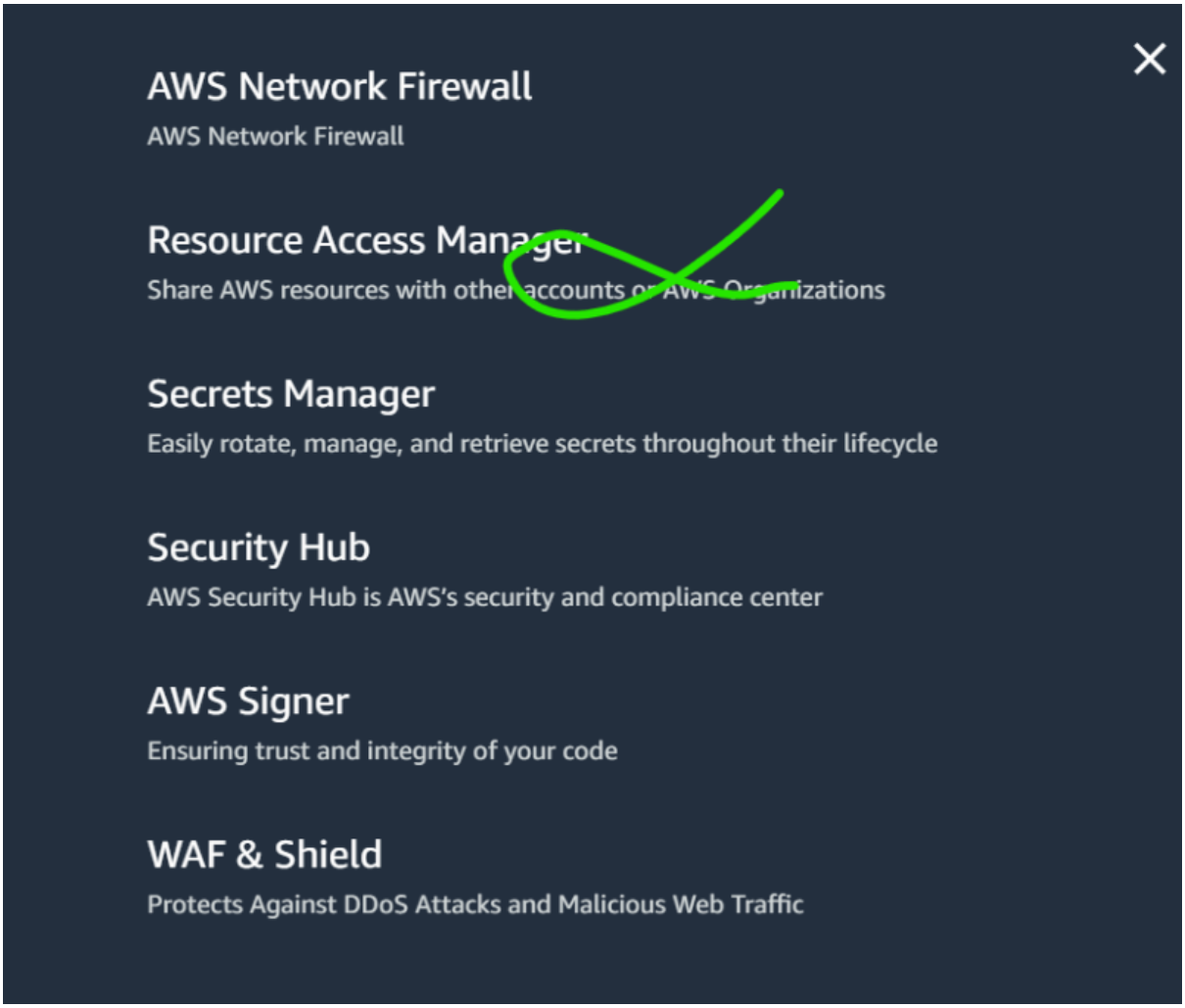
Attack Surface and Attack Vector

- https://www.techtarget.com/whatis/definition/attack-surface
- Attack surfaces are where unauthorized entry can be done. The attack surfaces to be protected are
    - Operating Systems
    - Ports
    - Code
    - Servers
- An attack vector is a path or means by which an attacker or hacker can gain access…

AWS Cloud Security

- The tools available in Aws are

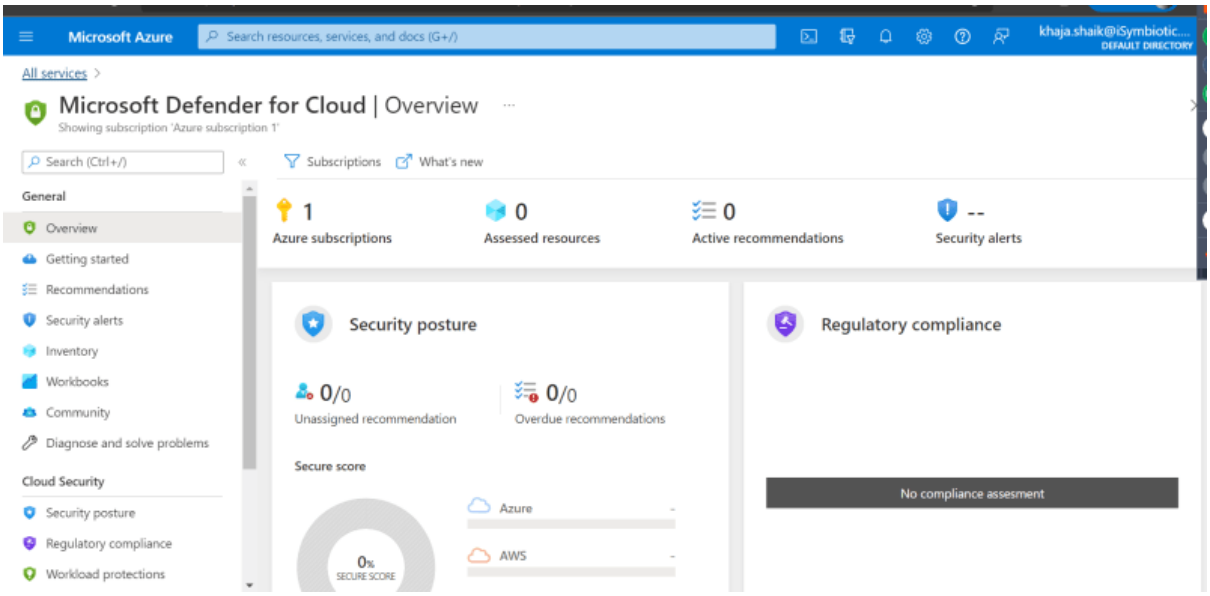

AWS Firewall Manager
Central management of firewall rules
GuardDuty
Intelligent Threat Detection to Protect Your AWS Accounts and Workloads
IAM
Manage access to AWS resources
IAM Identity Center (successor to AWS Single Sign-On)
Manage workforce user access to multiple AWS accounts and cloud applications
Inspector
Analyze Application Security
Key Management Service
Securely Generate and Manage AWS Encryption Keys
Amazon Macie
Amazon Macie classifies and secures your business-critical content.



- Gaurd Duty: We can enable security scanning for some services in AWS
- Inspector: Vulnerability Management
- AWS WAF: Provides firewall for your APIs
- AWS Security Hub
- AWS Sheild: Protects for DDOS

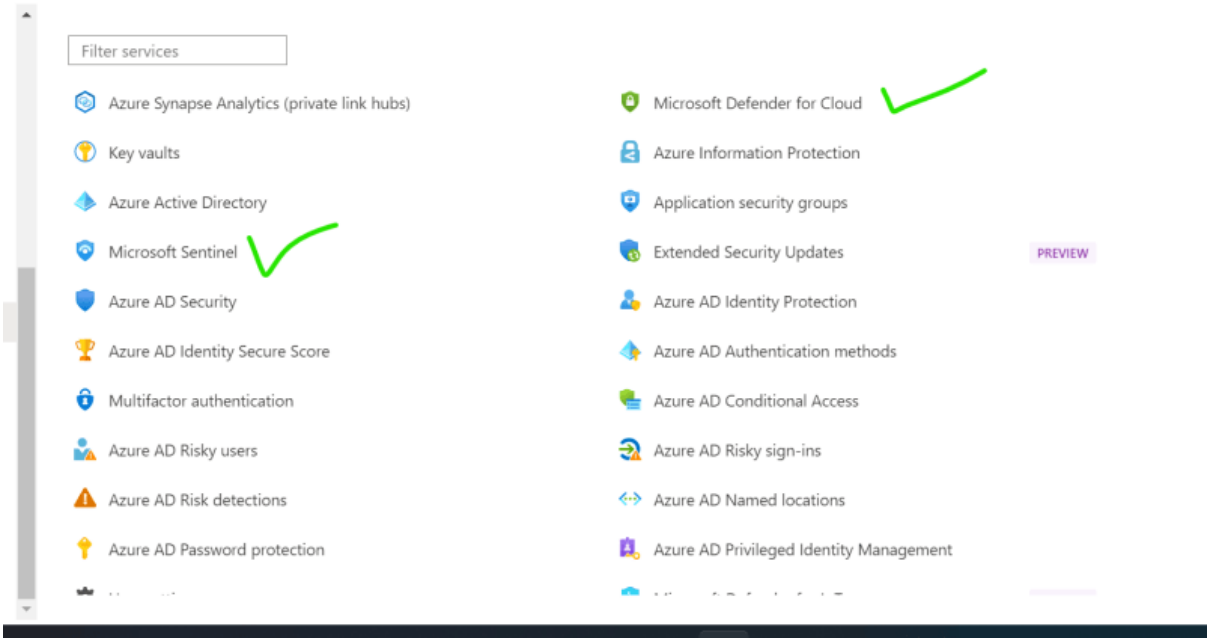
Azure Cloud Security

- Microsoft Defenders



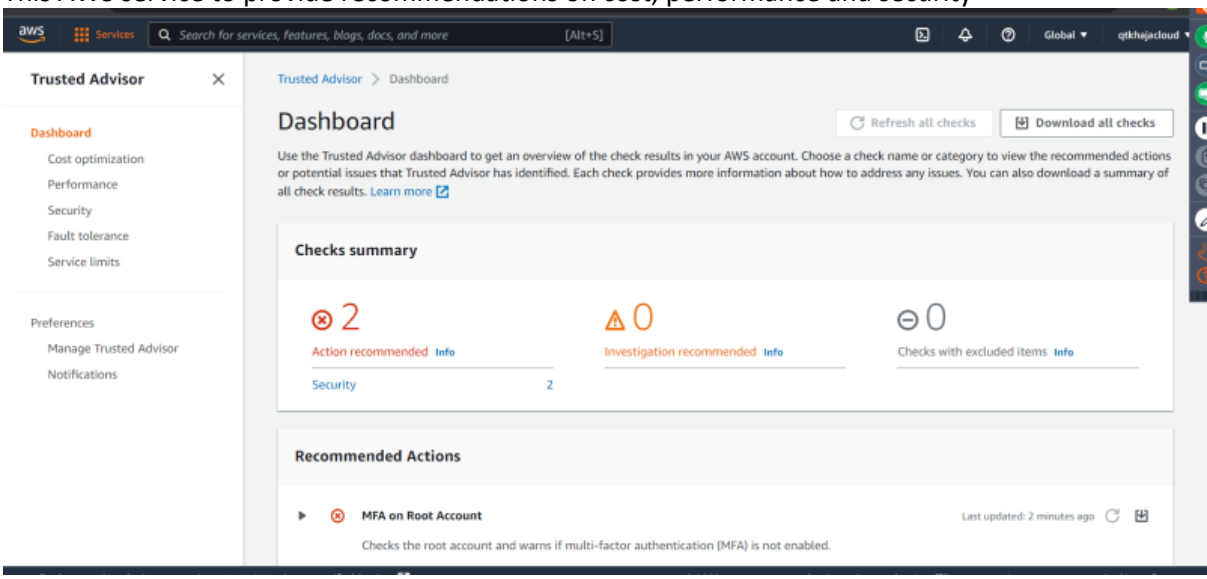
-
- The tool used is Microsoft Defender for Azure

ırity

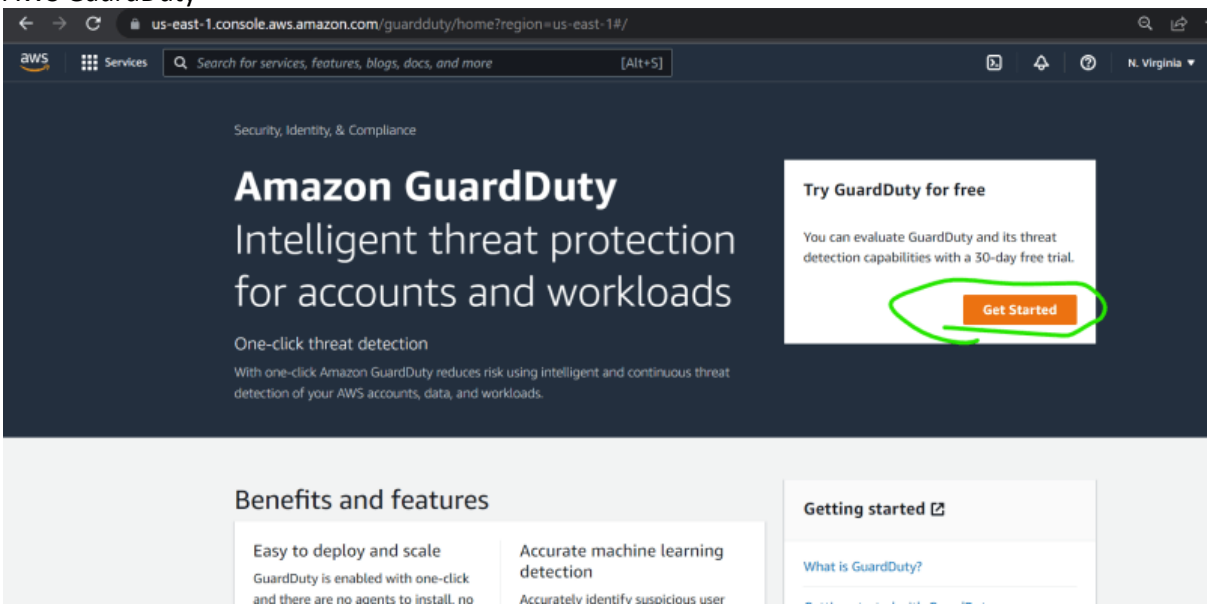


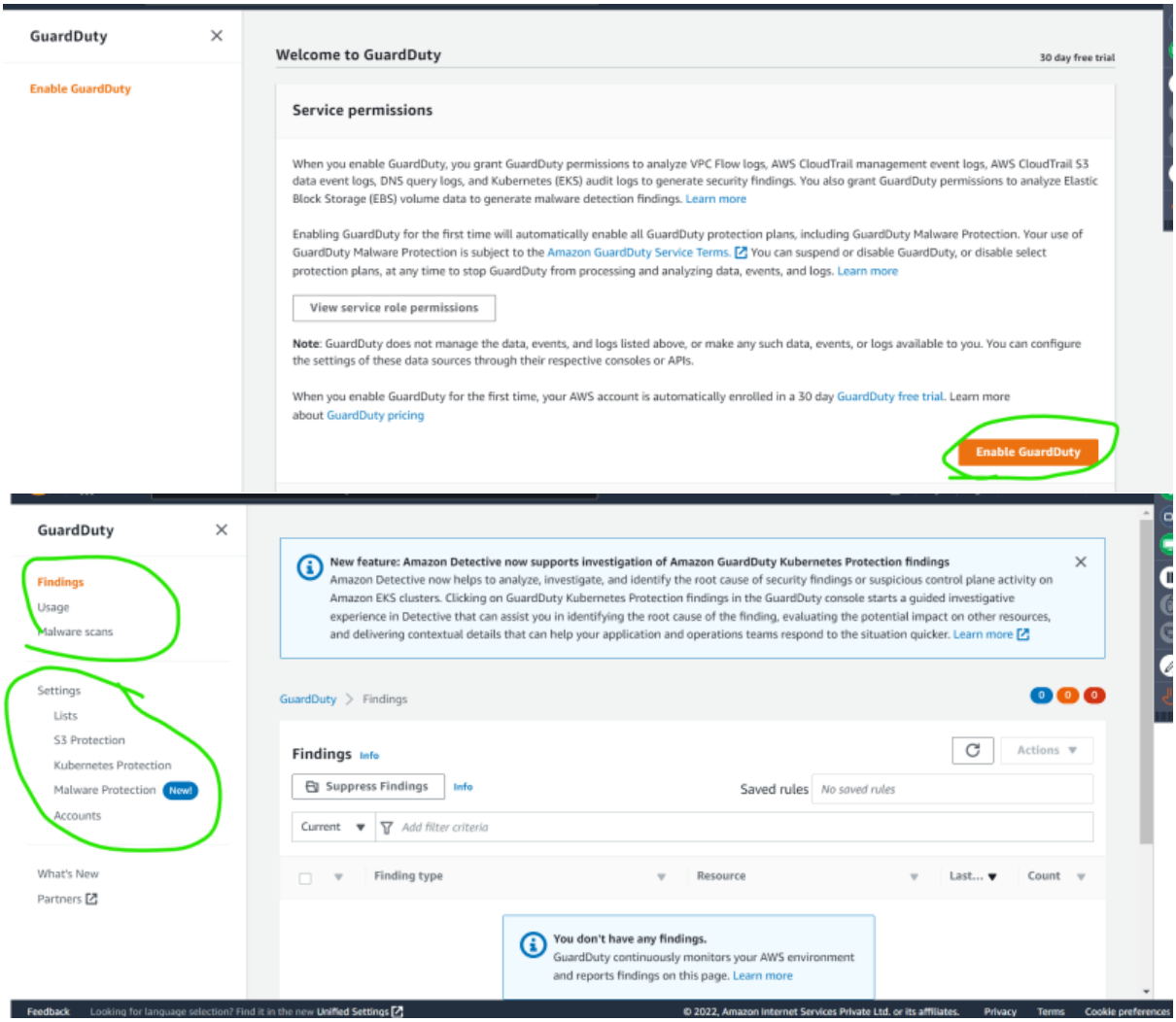
Security on AWS

- Trusted Advisor:
  - This AWS service to provide recommendations on cost, performance and security



- AWS GaurdDuty





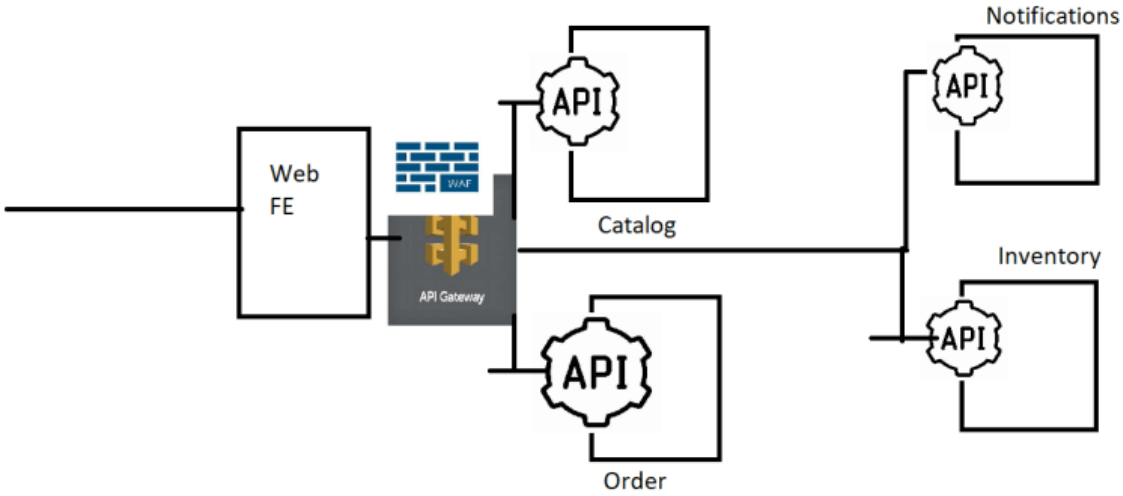
- A small script to access home page continuously

```
#!/bin/bash
while true
do
  curl http://18.212.20.237
done
```

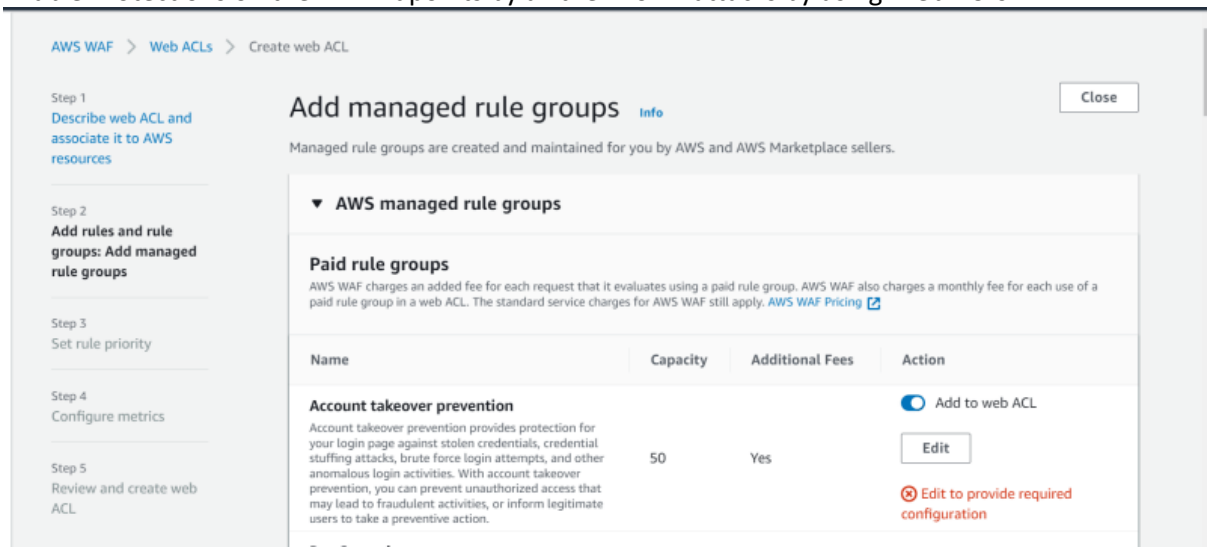
- Then we can add trusted ip list (Whitelist) and threat ip list (blacklist)
- AWS Macie:
    - This service scans for sensitive content on the S3 buckets
    - PII (Personally Identifiable Information) should not be stored in logs of your application.

AWS Security Tools

- AWS WAF (Web application Firewall): To prevent APIs from DDOS attacks and allow trusted IPs the WAF is enabled at API Gateway level.



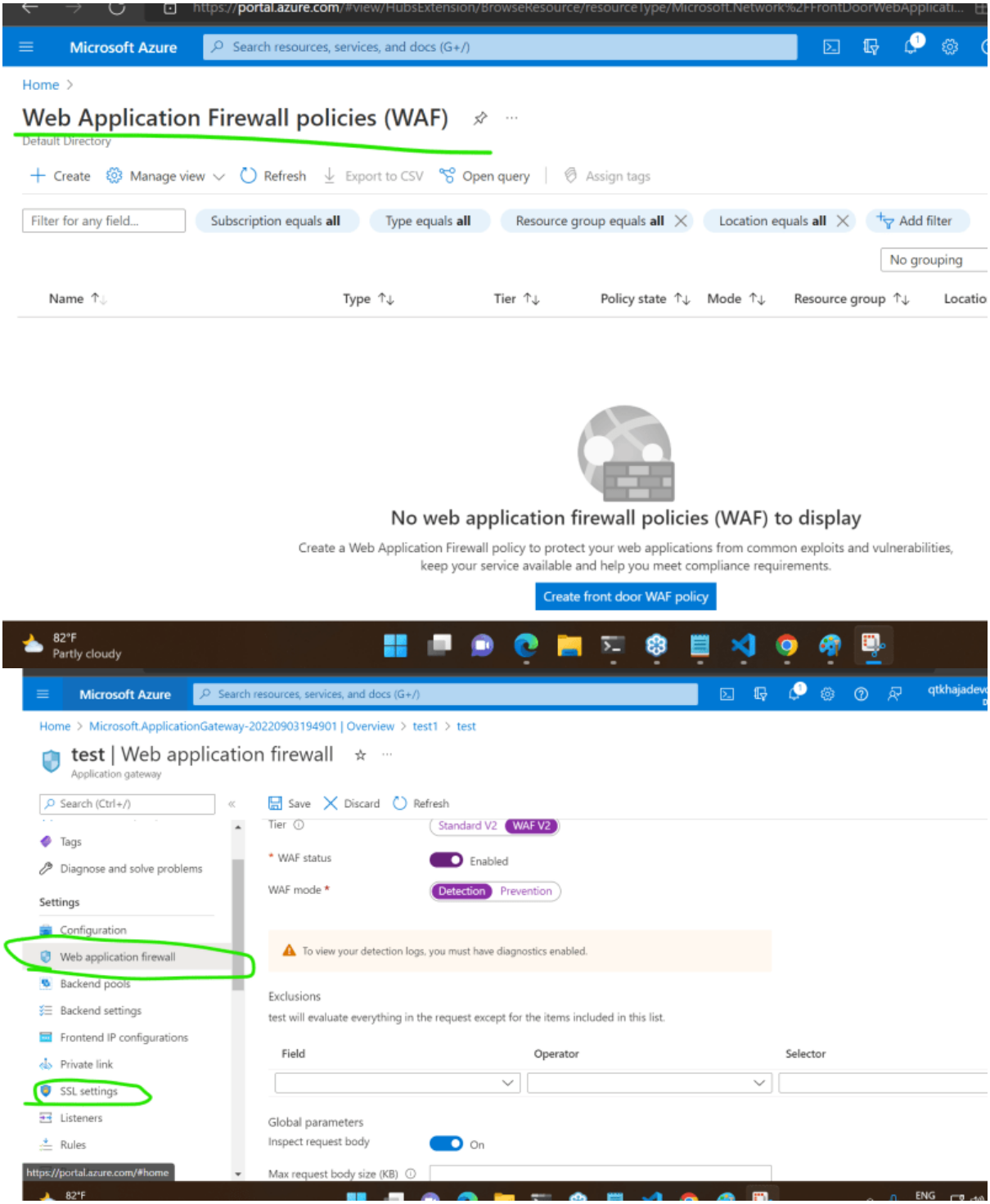
- Enable Protections on the API Endpoints by all the known attacks by using Web ACLs



Microsoft Azure => Microsoft Defender

- https://learn.microsoft.com/en-us/azure/defender-for-cloud/defender-for-cloud-introduction for the official docs.
- Enabling Defender in Azure Subscriptions https://learn.microsoft.com/en-us/azure/defender-for-cloud/get-started
- pricing https://azure.microsoft.com/en-us/pricing/details/defender-for-cloud/

- WAF Policies:



-

Fortify Scanning Docs

- https://www.microfocus.com/documentation/fortify-static-code/1720/ for the static code analysis tools docs from micro focus

1) X-ray from JFROG
2) Docker security for bench

1) SAST Hands on

==========

[Source code] https://github.com/juice-shop/juice-shop

Step1: git clone https://github.com/juice-shop/juice-shop

[SAST tool link] https://github.com/ZupIT/horusec

Step2: apt-get update ; apt install jq

Step3: curl -fsSL https://raw.githubusercontent.com/ZupIT/horusec/master/deployments/scripts/install.sh | bash -s latest

horusec version

Step3: horusec start -D -e="true" -p ./

2) DAST hands on

==========

Run application as container

Step1: docker run --rm -idt -p 3000:3000 bkimminich/juice-shop

Step2: docker run -t ghcr.io/zaproxy/zaproxy:stable zap-baseline.py -t <Application URL>

docker run -t ghcr.io/zaproxy/zaproxy:stable zap-baseline.py -t http://ip172-18-0-100-ckucfiufml8g009dm00g-3000.direct.labs.play-with-docker.com/#/

WARN-NEW: Dangerous JS Functions [10110] x 2

http://ip172-18-0-100-ckucfiufml8g009dm00g-3000.direct.labs.play-with-docker.com/main.js (200 OK)

http://ip172-18-0-100-ckucfiufml8g009dm00g-3000.direct.labs.play-with-docker.com/vendor.js (200 OK)

FAIL-NEW: 0 FAIL-INPROG: 0 WARN-NEW: 9 WARN-INPROG: 0 INFO: 0 IGNORE: 0 PASS: 56

3) SCA Hands on

==========

[Source code] https://github.com/juice-shop/juice-shop

Step1: git clone https://github.com/juice-shop/juice-shop

Step2: cd juice-shop/frontend/ ; vi package.json

in package.json we have defined our dependancy.

Step3: add dependacy "lodash": "0.5.0",

lodash: Lodash is a JavaScript library which provides utility functions for common programming tasks using the functional programming paradigm

Step4: apt-get update ; apt install npm

Step5: npm install

added 1479 packages from 2458 contributors and audited 1521 packages in 110.665s

260 packages are looking for funding

run `npm fund` for details

found 11 vulnerabilities (10 moderate, 1 high)

run `npm audit fix` to fix them, or `npm audit` for details