UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Jiménez Núñez, Lucas; Carpio Miranda, Miguel Eduardo, tut. Implementació de solucions en la ciberseguretat a aplicacions web utilitzant AWS. 2025. (Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/308791

under the terms of the  license

Implementació de solucions en la ciberseguretat a aplicacions web utilitzant AWS

Lucas Jiménez Núñez

Resum—Aquest projecte se centra en la implementació de solucions de ciberseguretat en aplicacions web utilitzant Amazon Web Services (AWS). El treball comença amb un estudi teòric de les vulnerabilitats més comunes que afecten a les aplicacions digitals, com ara Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Command Injection, etc. Explicant com es presenten aquestes i les millors pràctiques per mitigar-les. Posteriorment, es duen a terme proves d'explotació en entorns controlats utilitzant les aplicacions web DVWA (Damm Vulnerable Application) i Juice Shop. Aquestes proves ens permeten analitzar els riscs i danys que poden causar aquestes vulnerabilitats. Finalment, es detallen les solucions de seguretat implementades mitjançant AWS per protegir els entorns i evitar l'explotació de les vulnerabilitats, destacant la importància d'AWS i els seus avantatgess en termes d'escalabilitat, flexibilitat en preus i els seus serveis de seguretat.

Paraules clau—Ciberseguretat, AWS, Vulnerabilitat, Explotació, Pentesting, Proxy, Kali Linux, DVWA, Juice Shop, HTTP, Petició.

Abstract—This project focuses on implementing cybersecurity solutions in web applications using Amazon Web Services (AWS). The work begins with a theoretical study of the most common vulnerabilities affecting digital applications, such as Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Command Injection, etc., explaining how they manifest and the best practices to mitigate them. Subsequently, exploitation tests are conducted in controlled environments using the web applications DVWA (Damn Vulnerable Web Application) and Juice Shop. These tests allow us to analyze the risks and damages these vulnerabilities can cause. Finally, the security solutions implemented through AWS to protect the environments and prevent vulnerability exploitation are detailed, highlighting the importance of AWS and its advantages in terms of scalability, pricing flexibility, and security services.

Index Terms—Cybersecurity, AWS, Vulnerability, Exploitation, Pentesting, Proxy, Kali Linux, DVWA, Juice Shop, HTTP, Request.

1 Introducció - Context del treball

En el món actual, pràcticament tothom disposa d'un telèfon mòbil, ordinador, o qualsevol dispositiu capaç d'emmagatzemar tota mena d'informació personal, com ara fotografies, missatges, contrasenyes, etc., connectar-se a internet, comprar en línia, comunicar-se amb altres usuaris i treballar en diferents àmbits. Aquestes noves tecnologies presenten nous problemes a afrontar que es tornen imprescindibles si es vol assegurar la nostra privacitat, i en alguns casos fins i tot la nostra seguretat física o financera.

Una part molt important per a la seguretat informàtica està relacionada amb les aplicacions web i la xarxa. Aquestes són les encarregades de proporcionar-nos informació i portar-nos a través de tota la xarxa, però hi ha casos en què una pàgina web, com podria ser DVWA (Damn Vulnerable Web Application, aplicació web feta per a practicar l'explotació de vulnerabilitats) [1] siguin vulnerables, fent així que es puguin realitzar accions en el nostre ordinador/navegador sense el nostre consentiment ni coneixement. Dit això, ens interessa saber com protegir-nos davant d'aquestes vulnerabilitats. Per aquest motiu, en aquest projecte es busca explicar amb detall aquestes vulnerabilitats i, posteriorment, mitigar-les utilitzant Amazon Web Service, el servei de cloud d'Amazon que ha guanyat una certa popularitat en els darrers anys.

Per a veure de primera mà les accions dels atacants, a la secció 7 s'han executat diferents proves d'explotació de vulnerabilitats en entorns controlats, amb l'objectiu d'analitzar els riscs i els danys que pot causar un desenvolupament deficient. Seguidament, a la secció 10 s'han implementat mesures de seguretat mitjançant AWS per assegurar la protecció d'aquests entorns i evitar l'explotació de les vulnerabilitats prèviament analitzades. Les seccions 5, 6, 8 i 9 són purament teòriques i ens permeten conneixer i entendre les vulnerabilitats i eines de seguretat de les quals disposem.

Per a estalviar feina en el TFG s'ha optat per a no crear les aplicacions vulnerables i utilitzar d'altres ja creades específicament amb l'objectiu de poder practicar amb elles l'explotació de vulnerabilitats i protecció. Algunes d'aquestes aplicacions, cal recalcar que aquestes poden variar, són, DVWA (Damn Vulnerable Web Application) [1] i OWASP Juice Shop [2]. Aquestes aplicacions han estat dissenyades per a permetre a estudiant o entusiastes de la

- *E-mail de contacte: 1606727@uab.cat*
- *Menció realitzada: Tecnologies de la Informació*
- *Treball tutoritzat per: Miguel Carpio Miranda(DEiC)*
- *Curs 2024/25*

ciberseguretat a descobrir i prevenir erros, dur a terme proves de pentesting i realitzar activitats de hacking ètiques. Per aquests motius, i pel fet que són públiques i es pot trobar informació en cas d'estar encallat, he decidit utilitzar aquests dues aplicacions.

2 Objectius

Per tal d'assolir una securització efectiva de les aplicacions, aquest treball persegueix un objectiu principal: implementar solucions de ciberseguretat en una sèrie d'aplicacions utilitzant serveis d'AWS. No obstant això, per aconseguir aquest objectiu, és necessari complir una sèrie de subobjectius que permetin abordar de manera sistemàtica els problemes de seguretat identificats.

2.1 Estudiar vulnerabilitats i eines d'explotació

El primer subobjectiu consisteix en estudiar de manera teòrica les vulnerabilitats més comunes que afecten les aplicacions digitals actuals. Aquesta part proporcionarà el coneixement necessari per comprendre com es manifesten aquestes vulnerabilitats, quines conseqüències poden tenir si no es gestionen adequadament i les millors pràctiques i metodologies per a garantir la seguretat de les aplicacions.

2.2 Explotar vulnerabilitats

El segon subobjectiu inclou la posada en pràctica d'atacs controlats sobre les aplicacions per tal d'explotar les vulnerabilitats identificades prèviament. Aquest procés permetrà observar de manera pràctica com es poden comprometre les aplicacions quan aquestes no estan degudament protegides.

2.3 Implementar solucions pràctiques amb AWS

Finalment, el tercer subobjectiu implica protegir les aplicacions utilitzant serveis d'AWS. Aquesta fase del treball es dedicarà a la configuració d'eines i mecanismes de seguretat dins de la infraestructura d'AWS per garantir la protecció de les dades i la integritat de les aplicacions.

3 Planificació i Metodologia

Per al desenvolupament d'aquest projecte, es segueix una metodologia Scrum [3]. Aquesta metodologia destaca per la seva flexibilitat i permet, en cas de ser necessari, reorganitzar tasques per a adequar l'estat del projecte amb les tasques.

El projecte s'ha dividit en 5 sprints de 26 dies especificats a la Figura 1 de l'apèndix A1, on cada sprint està format per una sèrie de tasques a desenvolupar. Com es pot veure el primer sprint és purament teòrica, on l'objectiu és investigar vulnerabilitats més comunes i les eines de Kali Linux per a l'explotació d'aquestes. El següent sprint ja és pràctica, i es posen a prova els coneixements obtinguts en el sprint anterior, on s'intentaran explotar les vulnerabilitats de les aplicacions en un entorn controlat. El següent sprint torna a ser teòrica i s'investiguen en profunditat les eines d'AWS per a protegir les aplicacions. De nou, el següent sprint és pràctica, i es basa en la protecció d'aquestes aplicacions prèviament explotades. Finalment, el sprint final del projecte se centra a acabar de polir tota la documentació del projecte. Les tasques pròpies de cada sprint es poden veure a l'apèndix A1, Figures 2 i 3.

4 Estat de l'art

En el context actual de la ciberseguretat, existeixen diverses alternatives per a la securització d'aplicacions i infraestructures al núvol.

Microsoft Azure és una de les principals plataformes competidores d'AWS. Azure ofereix un conjunt d'eines, com Azure Security Center o Microsoft Entra ID, que permeten un control segur d'usuaris, a més d'oferir monitoratge en temps real de dades del sistema i amenaces. [4]

Google Cloud Platform també és una opció molt interessant, gràcies a la seva infraestructura escalable i l'ús d'eines de seguretat com Google Cloud Armor, que protegeix els nostres serveis de diferents amenaces a la web. [5]

En aquest projecte, s'ha escollit AWS per tres motius principals.

1. L'augment de la demanda d'enginyers especialitzats en AWS en el mercat laboral ha motivat l'elecció d'aquesta plataforma, amb l'objectiu de conèixer i provar les seves característiques. [6]
2. AWS destaca pels seus avantatges d'escalabilitat i flexibilitat en preus. La plataforma permet ajustar la capacitat dels serveis contractats (potència de còmput, emmagatzematge, etc.) segons la demanda de l'aplicació, possibilitant una gestió eficient dels recursos. Per exemple, en cas d'un augment de trànsit a l'aplicació, AWS pot escalar automàticament els recursos per mantenir el rendiment, i reduir-los un cop el trànsit torna a la normalitat. [7]
3. AWS ofereix una àmplia gamma de serveis de seguretat per protegir tant les aplicacions com les dades en el núvol. Entre aquestes eines s'inclouen IAM (Identity and Access Management), KMS (Key Management Service), VPC (Virtual Private Cloud), AWS Shield i altres, que es detallen més endavant. [8]

5 Eines per a l'explotació web

Per a l'explotació de vulnerabilitats web, es poden utilitzar diferents eines que simplifiquen considerablement la feina. Ara bé, aquestes eines acostumen a ser força intrusives, de manera que, en un cas real, si el servidor web està ben configurat, probablement bloquejaria l'adreça IP, impedint el seu us. Això faria que la cerca i explotació de vulnerabilitats s'hagués de fer de forma manual. Per al projecte, s'han utilitzat aquestes eines però també s'han buscat vulnerabilitats manualment per representar un escenari més proper a la realitat.

La majoria d'aquestes eines les he conegut i après a utilitzar gràcies al curs **Curso completo de Hacking Ético y Ciberseguridad** [9]. En aquest curs he après les bases de la ciberseguretat, les diferents fases que es duen a terme en una prova de pentesting i, finalment, les eines essencials per a cada fase. En aquest cas, com que s'està treballant amb una aplicació web, s'han utilitzat eines per al descobriment de hosts i ports, així com eines de detecció i

explotació de vulnerabilitats web, aquestes eines s'especifiquen a les subseccions següents. És important remarcar que en un cas real caldria completar diverses fases prèvies per a recollir informació sobre l'empresa objectiu de l'atac.

5.1 Descobriment de hosts i ports

Per al descobriment de hosts i ports s'ha utilitzat Nmap [10]. Nmap es una eina que es centra en la detecció de hosts en una xarxa, descobriment de direccions IP i ports, entre d'altres opcions. S'utilitza aquesta eina ja que per a explotar diferents vulnerabilitats es necessari saber els ports o serveis que utilitza el servidor.

5.2 Detecció i explotació de vulnerabilitats

Per a la detecció i explotació de vulnerabilitats s'han utilitzat les següents eines:

- **Metasploit** es una eina de codi obert que serveix per al descobriment i explotació de vulnerabilitats de manera automàtica. En aquest cas s'ha fet servir per a realitzar atacs DoS i veure si l'aplicació pateix algun problema de rendiment. [11]
- **Burp suite** es un proxy d'intercepció de peticions web que permet analitzar el contingut de les peticions que s'envien al servidor i de les respostes d'aquest. Al ser un programa de pagament amb versió gratuïta hi ha moltes funcionalitats que no puc utilitzar. Per a suplir això també s'ha fet servir la següent eina. [12]
- **ZAP** (Zed Proxy Attack) també es un proxy d'intercepció de peticions web, però al ser gratuïta disposa de diferents eines com aranyes (web crawler) que descarreguen i indexen el contingut de la web per a descobrir les pagines d'aquesta i poder realitzar diferents proves. A més d'això disposa d'un escaneig de vulnerabilitats automàtic molt potent que permet llistar les vulnerabilitats trobades d'un servidor web. [13]
- **SQLmap** es una eina de codi obert que serveix per a detectar i explotar vulnerabilitats SQL injection i aconseguir accés a les diferents bases de dades del servidor. [14]

6 Vulnerabilitats i atacs web

Com s'ha comentat a la secció anterior, al buscar explotar una aplicació web es va centrar en explotar vulnerabilitats que es centraven en el servidor que allotjava la pàgina web, i en vulnerabilitats lligades directament a aquesta.

- **Denial of Service** es un atac el qual es centra en sobrecarregar al servidor de web de diferents paquets, sobrecarregant-lo i aconseguint aleshores que els serveis no funcionin de la manera esperada. La vulnerabilitat associada a aquest atac seria una mala configuració del servidor, no verificant les connexions que es creen. [15]
- **Cross Site Request Forgery** (CSRF) es un atac que força a un usuari a executar certes accions en una aplicació web en la qual està autenticat. Aquest atac acostuma a realitzar-se juntament amb ajuda d'enginyeria social. [16]
- **Command Injection** es un atac que permet al atacant executar orderes directament al sistema operatiu del host. Aquest atac es realitza a traves de la injecció de comandes en formularis o inputs no securitzats. [17]
- **Cross-Site Scripting**, o XSS, es basa en la injecció de scripts maliciosos en pàgines web verificades. Aquests scripts serien executats per a usuaris verificats on comunament enviarien les seves dades privades al creador del script. [18]
- **Path Traversal**, també conegut com directory traversal, té com a objectiu accedir a fitxers i directoris que es troben fora de la carpeta arrel de la web. Aquest atac està limitat pel control de l'accés operatiu del sistema. [19]
- **L'atac de força bruta** consisteix en provar diverses combinacions de valors per aconseguir accés no autoritzat a un sistema per obtenir informació sensible. [20]

7 Explotació de vulnerabilitats

Com s'ha comentat abans, a l'hora d'explotar les vulnerabilitats, s'han utilitzat dues pàgines web per a practicar. La primera DVWA, és més senzilla i permet entendre les vulnerabilitats, mentre que la segona, Juice Shop [2], mostra un cas real, que permet treballar de manera més seriosa.

Primer de tot, per la part de l'atac de DoS, cal recalcar que aquest es centra a atacar la màquina host, no la pròpia pàgina web, per tant, es fa un anàlisis de vulnerabilitats utilitzant l'eina Nessus [21]. Aquesta anàlisi no va retornar cap vulnerabilitat coneguda, ja que les màquines virtuals que van ser instal·lades, estaven actualitzades amb les últimes versions, por això, no es va poder dur a terme l'atac de DoS.

7.1 DVWA

A l'hora de practicar amb DVWA, cal tenir en compte un parell de coses. En primer lloc, aquesta web s'instal·la sobre la màquina host, fent servir el servei HTTP d'apache2 [22] que ve instal·lat per defecte a Kali Linux. En segon lloc, com aquesta web està orientada a una pràctica senzilla de pentesting, les vulnerabilitats estan dividides en seccions per a practicar.

7.1.1 CSRF

Per a practicar la vulnerabilitat de CSRF a DVWA, es troba el següent apartat dedicat especificat a l'apèndix A3 Figura 1. Es veu que el formulari fa l'acció de canviar la contrasenya de l'usuari autenticat, on, si amb BurpSuite s'intercepta la petició, aquesta tenia el format especificat en l'apèndix A3 Figura 2. Aquest formulari envia una petició GET amb dos atributs, la nova contrasenya, i una confirmació. Aleshores, si un altre usuari autenticat, executa aquesta petició GET, a través d'un link de, per exemple, un correu phising, se li canviaria la contrasenya a, en aquest cas 'admin1234'. En la part de Juice Shop es veu més en detall aquesta tècnica.

7.1.2 Command Injection

A la secció de Command Injection, es troba el formulari de l'apèndix A3 Figura 3. Aquest formulari demana que s'introdueixi una adreça IP, on, si es fa, s'obté l'output de la Figura 4 de l'apèndix A3. Es pot veure que aquest output és exactament el que torna la comanda ping d'un sistema Ubuntu/Debian, per tant, es pot suposar que el formulari està executant la comanda ping de la direcció IP que se li enviava. Per a executar més d'una comanda en un sistema Linux, s'utilitza el caràcter "|", aleshores, si enlloc d'enviar una adreça IP, s'envia " | cat /etc/passwd ", per a veure si es pot imprimir el fitxer passwd, s'obté el resultat de la Figura 5 de l'apèndix A3. Es veu aleshores que aquest camp és vulnerable a Command Injection, i es poden realitzar diferents accions que no haurien d'estar permeses.

7.1.3 Cross-Site Scripting

La vulnerabilitat XSS, es pot dividir en 3 apartats, basada en DOM, reflectit i guardada en el servidor. Basada en DOM busca executar el script maliciós en el DOM del navegador, sense passar pel servidor. Reflectit, executa el script com a part de la sol·licitud HTTP i es reflecteix a la resposta del servidor. Finalment, XSS guardat en el servidor, busca guardar el script en el servidor web, perquè així, aquest s'executi en carregar la pàgina, juntament amb el script, de qualsevol usuari [23]. Aquesta tercera opció és la més perillosa, i la que exposaré en aquest projecte.

Si es veu el formulari de la vulnerabilitat XSS Stored de la Figura 6 de l'apèndix A3, aquest tracta d'una pàgina on es pengen comentaris. Per veure si aquests camps són vulnerables, s'introdueix al input de 'Message' el comentari de la Figura 7 de l'apèndix A3, i com es pot veure, el comentari (que es mostra una mica més avall) es presentava com un títol gràcies al tag 'h1'. Imprimir 'canario' en format títol no aporta res com a atacants, aleshores, si enlloc del comentari anterior, s'introdueix el comentari de la Figura 8 de l'apèndix A3, s'afegeix el comentari marcat amb la fletxa. Aquest comentari el que fa és que, quan es carrega la pàgina de nou, apareix l'alerta de la Figura 9 de l'apèndix A3, on, en aquest cas, s'imprimeix per pantalla les cookies de l'usuari, però també es podrien enviar a un servidor sense que l'usuari legítim se n'adonés.

7.2 Juice Shop

Per a facilitar l'enteniment de les vulnerabilitats explotades, cada una disposa d'un diagrama de fluxe, el qual mostra els passos d'explotació.

7.2.1 Brute Force Attack

Un atac de força bruta (veure diagrama 1 de l'apèndix 2), s'utilitza per intentar desxifrar usuari/contrasenya dels perfils de la pàgina web. Per a fer això, com a atacant interessa aconseguir els usuaris amb més privilegis. Després d'analitzar la pàgina web, es troba un producte amb una ressenya d'un usuari 'admin@juice-shop.op'. Aquest usuari, a priori, sembla de l'administrador de la pàgina, per tant s'intenta realitzar un atac de força bruta per a obtenir la contrasenya de l'administrador. Per a fer això s'utilitza servir Burp Suite, on després de capturar una petició de login de l'admin, es pot realitzar un atac de força bruta amb un llistat de les contrasenyes més comunes de 2023, on en acabar la prova de contrasenyes, es troba que aquest és 'admin123' (veure Figura 1).

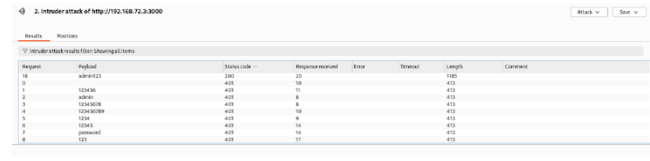

Fig. 1: Contrasenya administrador

Finalment, s'intenta iniciar sessió, obtenint accés al comte d'administrador.

7.2.2 SQL Injection

La vulnerabilitat SQLInjection (veure diagrama 2 de l'apèndix 2), es basa en veure si els camps de la web, són vulnerables a comandes SQL. Aquests camps són aquells que interactuen amb la base de dades, com pot ser un formulari d'inici de sessió. Si un camp és vulnerable a SQL injection, pot permetre utilitzar eines com SQLMap, que poden aconseguir informació sobre la base de dades. [24]

La manera més comú de veure si els camps són vulnerables, és introduir la següent comanda 'or 1=1 -- ' (veure Figura 2) on, després d'introduir-la en el formulari de login, juntament amb una contrasenya inventada i enviar la petició, s'inicia sessió al compte de l'administrador. Aquesta comanda retorna true en tot cas, per tant, si no es sanetizat el input de l'usuari abans de processar-lo per la pàgina web, aquest pot portar a executar codi no fiable a la base de dades.

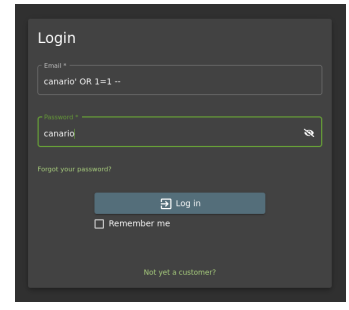

Fig. 2: SQL Injection

Utilitzant aleshores SQLMap, s'obté informació privada, com, el tipus de base de dades, SQLite, o per exemple el nom de totes les taules. Això es pot veure en les Figures 10 i 11 de l'apèndix A3.

7.2.3 CSRF

A Juice Shop s'ha explotat la vulnerabilitat CSRF (veure diagrama 3 de l'apèndix 2) per a forçar a l'usuari a canviar-se el nom del compte a un diferent. Per a fer això primer s'ha de capturar la petició de canvi de nom a 'Test', veure Figura 12 de l'apèndix A3. Com la petició és de tipus POST, s'utilitza Burp Suite per convertir-la a GET i provar de "passar-li" el nou enllaç a un usuari autenticat. Si l'usuari accedeix a aquest enllaç maliciós, el seu nom d'usuari no es modificava, per tant, canviar el tipus de la petició no explotava la vulnerabilitat.

L'altra opció és fer que l'usuari realitzi una acció, en la majoria de casos clicar un botó, per a enviar una petició de

canvi de nom sense que l'usuari o sàpiga. Per a fer això, a traves de la petició POST de la Figura 12 de l'apèndix A3, Burp Suite permet crear el CSRF HTML de la Figura 13 del mateix apèndix. El codi representa un formulari POST que en ser clicat envia la petició POST de la Figura 12 de l'apèndix A3, amb el valor 'HasSidoHackeado', i si l'usuari autenticat accedeix a aquest codi HTML i prem el botó 'Submit request' de la Figura 14 de l'apèndix A3, se li carregaria la pàgina de perfil de Juice Shop, mostrant el nom d'usuari modificat com a la Figura 15 de l'apèndix A3.

Cal destacar que en un cas real, la web maliciosa creada hauria de ser prou convincent per a poder enganyar les víctimes.

7.2.4 Cros-Site Scripting Persisted

A l'hora d'explotar la vulnerabilitat XSS Persisted (veure diagrama 4 de l'apèndix 2), Juice Shop dona diferents opcions, pel que s'ha escollit el tipus persistent. Explotar aquesta vulnerabilitat és relativament senzill. En primer lloc s'ha d'anar a la pantalla d'enregistrament de nou usuari, aquesta disposa de diferents camps, on destaca el camp de 'email'. Aquest camp retorna error en cas que no es segueixi el format tradicional d'una adreça electrònica (email@domini.com/es), però aquesta mesura de seguretat és fàcil de saltar utilitzant BurpSuite. Es captura una petició d'enregistrament, amb un correu fictici, i en el camp de correu, dins de Burp Suite, s'afegeix el codi maliciós. La pàgina retorna un missatge de 'usuari creat correctament', indicant que s'ha creat un usuari amb aquesta "adreça".

Per a comprovar si s'havia inserit el codi maliciós correctament, s'ha d'anar a la pantalla d'administrador, on apareixen tots el usuaris de l'aplicació, on al accedir, es carregaria l'alerta que representa el robatori d'informació, que s'ha incerit.

Tot aquest procés es pot comprovar en el següent enllaç [25], on hi ha penjat un vídeo gravat per mi.

8 Serveis de desplegament d'AWS

A l'hora de securitzar l'aplicació, he utilitzat Amazon Web Services (AWS). AWS és una de les plataformes més populars per a la implementació de mesures de seguretat en aplicacions cloud, gràcies a la seva àmplia oferta de serveis sota demanda. Aquests poden gestionar recursos com la potència de computació, l'emmagatzematge, bases de dades, xarxes, seguretat i molts d'altres, proporcionant una infraestructura flexible i escalable. [7]

- **Amazon Elastic Beanstalk:** Aquest servei permet desplegar i gestionar de manera ràpida i eficient aplicacions sense haver de preocupar-se de la infraestructura on s'executaran aquestes. AWS Elastic Beanstalk redurix la complexitat de la gestió de l'aplicació, aconseguint així una gran velocitat en el procés de producció. Pel que fa a la securització d'una aplicació com Juice Shop, no és un servei ideal, principalment perquè aquest està orientat a la creació de codi, i Juice Shop ja està desenvolupada (i, per a securitzar la web, tot i que moltes vulnerabilitats es solucionen de manera senzilla al codi, vull explotar al màxim les eines d'AWS, per això, el codi no es modificarà). A més, aquest servei no permet gestionar la seguretat a nivell de servidor. [26]
- **Amazon S3 amb CloudFront**: Amazon Simple Storage Service és un servei d'emmagatzematge d'objectes que ofereix gran escalabilitat, disponibilitat de dades i mesures de seguretat. Aquests objectes d'emmagatzematge poden variar entre pàgines estàtiques d'HTML, com a fitxers de dades csv o bases de dades [26]. Aquest servei, juntament amb CloudFront, que és un servei web que agilitza la distribució de contingut en pàgines web estàtiques o dinàmiques (com arxius HTML, CSS, JS), és una gran opció si es vol desplegar una pàgina web estàtica, però com que Juice Shop és dinàmica, aquesta opció no serveix. [27]
- **AWS Elastic Container Service (ECS) amb Fargate:** És un servei que és centra en la gestió de contenidors, cosa que permet desplegar-los de manera senzilla, gestionar recursos i escalar de forma eficient. Aquest fet, facilita una integració senzilla que permet als desenvolupadors centrar-se a construir les aplicacions sense haver de configurar l'entorn [28]. Juntament amb ECS, podem utilitzar AWS Fargate, una eina que ens allibera de la necessitat d'escollir el tipus de servidor, decidir quan escalar clústers o optimitzar-los, ja que aquestes tasques es gestionen automàticament [29]. Com he dit anteriorment, aquesta opció tampoc permet gestionar la seguretat a nivell de sistema operatiu, per tant, no faré servir aquesta opció.
- **Amazon Elastic Compute Cloud (EC2):** EC2 ens proporciona capacitat informàtica escalable sota demanda, reduint a més els costos hardware per a fer això. Ens permet crear tants servidors virtuals com vulguem, configurar la seguretat i les xarxes i gestionar l'emmagatzematge del servidor. Cada instància EC2 és un servidor virtual creat al Cloud d'AWS (un servidor virtual es podria comparar com a una màquina virtual, allotjada al hardware d'Amazon). Com veiem, aquest servei es gairebé com gestionar una màquina virtual amb tot el control d'aquesta. Per tant, aquesta opció ens permet, utilitzar eines d'Amazon per a gestionar peticions i eines, no pròpies d'Amazon, però que poden ser útils instal·lar en el sistema operatiu/servidor. [30]

Aquestes són algunes de les opcions disponibles per a desplegar una aplicació web utilitzant AWS, on cada una d'aquestes té els seus punts forts i fluixos. En el nostre cas, obtenim control total sobre el servidor, amb les facilitats d'escalabilitat que ens proporciona Amazon. Això sí, hem de configurar tot l'entorn, la qual cosa ens trauria temps per a desenvolupar.

9 Eines de securització

AWS disposa de diferents eines de securització, que es divideixen en 3 grups: securització de compte d'AWS, securització de recursos i securització web.

El primer grup, securització del compte AWS, es basa en eines que serveixen per monitorar i afegir mesures de seguretat a un compte d'AWS. Això pot interessar en cas que es treballés amb un equip i es volgués limitar els permisos de diferents usuaris per a evitar possibles problemes, aplicant el principi de menys privilegi. Algunes eines que es podiren posar en aquest grup serien AWS IAM (permet gestionar de forma segura els accessos als recursos mitjançant usuaris, rols i polítiques) [31] i Amazon GuardDuty (monitora i analitza el compte d'AWS per a evitar activitat maliciosa o inusual) [32].

El segon grup, securització de recursos, es basa en eines que permeten protegir, analitzar i monitorar els recursos, com per exemple instàncies de EC2 o S3. Aquestes eines permeten buscar vulnerabilitats o aturar recursos en cas de ser necessari. En aquest grup destaca una eina que s'ha utilitzat per al present treball, Amazon Inspector. Aquesta eina disposa d'una prova gratuïta de 15 dies, i després d'aquests 15 dies passa a ser de pagament, seguint un pla específic fet a mesura per a cada usuari que l'utilitzi. Aquesta eina es fa servir per a la gestió de vulnerabilitats, detectant vulnerabilitats software i de xarxa a les instàncies, per exemple EC2, avaluant les vulnerabilitats trobades i reanalitzant els recursos de manera automàtica per a mantenir un entorn segur i controlat. En el següent apartat s'explica més en detall el seu funcionament. [33]

El tercer i últim grup, securització web, es basa finalment en les eines que proporciona AWS per a protegir pàgines web. En aquest aparat destaquen dues eines molt útils d'AWS, AWS Web Application Firewall (WAF) i AWS Shield Advanced.

9.1 AWS WAF

AWS WAF és un servei de tallafoc d'aplicacions web que s'encarrega de protegir aplicacions web o APIs contra amenaces que podrien afectar la seva disponibilitat, privadesa o integritat. AWS WAF, permet definir regles de seguretat personalitzables que filtren i controlen el tràfic HTTP/HTTPS (en el nostre cas HTTP).

WAF funciona filtrant el trànsit d'entrada a través d'unes regles personalitzades, definides en un ACL web (Access Control List). Aquestes ACL determinen si les sol·licituds d'entrada seran permeses, bloquejades o monitoritzades.

Finalment, per a decidir si utilitzar o no aquesta eina en la securització de Juice Shop s'ha de saber el que costarà utilitzar-la. El cost varia en funció del número de ACL creades, el número de regles i el número de sol·licituds inspeccionades, sent la mitjana mensual per una aplicació petita, de 5€. Per tant, el cost benefici és molt alt, i resulta interessant utilitzar aquesta eina en la aplicació web plantejada.

9.2 AWS Shield Advanced

AWS Shield Advanced és la versió de pagament d'AWS Shiled (versió bàsica que ofereix AWS sense cost addicional). Aquest és un servei per a protegir els recursos contra atacs de denegació de serveis distribuïts (DDoS). Aquest servei ofereix protecció avançada, detecció en temps real i mitigació automàtica contra atacs DDoS a nivell de xarxa, transport i d'aplicació.

Les característiques principals d'AWS Shield Advanced són:

1. Protecció avançada contra DDoS: Detectant i mitigant atacs més sofisticats, a les capes de xarxa, transport i aplicació.
2. Detecció en temps real: Monitoratge continu del trànsit, detectant patrons sospitosos.
3. Resposta personalitzada: A diferència de la seva opció gratuïta, AWS Shield Advance disposa d'accés a un equip de resposta d'AWS Shield (SRT, Shield Response Team), un grup d'experts de seguretat que ajuden a respondre atacs DDoS.

Finalment, de nou, s'ha d'analitzar el preu d'aquesta eina, on aquí si que es troba una gran diferència amb l'eina anterior. El preu mensual d'aquesta eina és de 3000$, preu que, per a protegir una aplicació web petita, és desorbitat. Per això, s'arriba a la conclusió que aquesta eina va orientada a serveis molt sensibles, que requereixin de la màxima seguretat, que no és el cas de la aplicació web a protegir, per tant, no s'utilitzarà aquesta eina.

10 Securització Juice Shop

En aquest apartat, s'han utilitzat les eines explicades anteriorment per a desplegar l'aplicació web Juice Shop, configurar-la i securitzar-la. Un cop fet això, s'han realitzat els mateixos atacs de la secció 7 per a veure el comportament de l'aplicació.

10.1 Creació de la instància

Com s'ha comentat abans, per a desplegar l'aplicació s'utilitza l'eina d'AWS EC2. El primer que es fa és crear una instància d'EC2. Per a fer-ho, s'han de seguir els següents passos: primer de tot, s'accedeix a la pàgina de EC2; seguidament, es fa clic al botó de crear instància, i s'és redirigit a la pàgina per a especificar les dades de l'instància. En aquest punt, s'especifica el nom, el tipus (Ubuntu, Windows), la potencia i l'emmagatzematge de la instància. Es selecciona també un parell de claus de seguretat, que permetran connectar-se a través de SSH, i es configuren les propietats de la xarxa i els diferents usuaris que es poden connectar a la instància. En aquest cas, com que el servei actua com a pàgina web, hauria de ser accessible des de tot el món, però per a evitar problemes de seguretat, i com que s'està fent només una prova de securització, i no aportant un servei amb la web, es selecciona l'opció de permetre només la connexió des de l'adreça IP pròpia. Finalment, per evitar un problema de creació d'instància, s'ha de seleccionar el tipus de crèdit de la instància com a estàndard. Això passa perquè, tot i que les instàncies creades tenen una CPU base, aquesta pot augmentar depenent de les necessitats. "Unlimited" permet que la instància utilitzi més CPU de la que té instal·lada i pot portar a costos addicionals, mentre que "Strandard" es limita únicament a la CPU del sistema. Tots aquests apartats es poden visualitzar a

l'apèndix A4, a les Figues 1, 2, 3, 4, 5, 6 i 7.

10.2 Configuració de la instància

Un cop creada la instància, si es clica el botó "Connectar" es mostren els passos a seguir per a connectar-se a través de SSH i començar a configurar-la.

Per a provar les eines que aporta AWS, no s'actualiza el sistema operatiu i només s'instal·len els programes necessaris per al funcionament de la web. Aquests programes són Docker i Caddy.

Docker es necessita per a instal·lar de manera senzilla Juice Shop, utilitzant la seva imatge contenidora. Per a instal·lar-lo simplement s'han de seguir els passos d'instal·lació de Docker de la seva pàgina principal [34]. Després d'instal·lar Docker, s'han de seguir els passos per a instal·lar Juice Shop a través de la seva imatge contenidora [2]. Això permet aleshores accedir a la adreça IP, al port 3000 per accedir a Juice Shop, però si s'intenta, no permet accedir al servei, degut a que les regles d'entrada de la instància (creades per defecte) no permeten accedir al port 3000, com es pot veure a la Figura 8 de l'apèndix A4.

Aquestes regles permeten l'entrada a través dels ports 22, 443 i 80 per al protocol TCP, però no pel port 3000, que és en el que es desplega Juice Shop. Per a poder-hi accedir aleshores hi ha 2 opcions. La primera és afegir una regla d'entrada que permeti accedir al port 3000, aquesta és una solució fàcil i ràpida, però no és gaire segura, ja que no permet filtrar el tràfic per aquest port, i utilitzant una aplicació no segura, deixa oberta una porta perquè possibles hackers intentin accedir al compte d'AWS. La segona opció es basa a configurar un reverse Proxy, que redirigeixi el tràfic del port 3000 al 80, per així poder accedir a la pàgina web a través d'un port verificat, aquesta solució requereix instal·lar Caddy. [35]

Per a instal·lar i configurar Caddy, que com s'ha dit és un servidor web i Proxy HTTP, només s'han de seguir els passos de la documentació oficial [35]. Caddy és una eina prou útil, però al mateix temps senzilla, per tant, en un cas real interessaria més utilitzar un altre proxy amb més capacitats de configuració.

Un cop instal·lat i configurat Caddy, si s'accedeix a la adreça IP de la instància creada, es visualitza, finalment, Juice Shop funcionant.

10.3 Securització

Un cop instanciat tot, es fa servir l'eina Amazon Inspector perquè analitzi el recurs EC2 cercant vulnerabilitats. Per a fer això primer es busca el servei a la pàgina principal d'AWS i s'inicia la prova gratuïta. Després d'acceptar una sèrie de permisos, es mostra la pantalla principal d'aquesta eina, on apareixen tots els recursos que analitzarà. Això ho fa de forma automàtica, per la qual cosa, començarà a analitzar els recursos. A la Figura 9 de l'apèndix A4, es pot veure l'anàlisis del recurs EC2 efectuat.

Si es navega per al panell principal, es pot trobar la taula de vulnerabilitats de la Figura 10 de l'apèndix A4, on es pot veure que la instància EC2 creada té una vulnerabilitat crítica que s'hauria de solucionar.

En clicar a la vulnerabilitat, AWS redirigeix l'usuari a una pàgina amb el llistat de totes les vulnerabilitats, on permet visualitzar informació del CVE d'aquesta vulnerabilitat. Aquesta informació també indica com solucionar la vulnerabilitat, en cas que tingui solució. En el cas de la vulnerabilitat de la Figura 10 de l'apèndix A4, és necessari actualitzar el sistema.

Aquesta informació és molt important en el context de la seguretat informàtica, ja que permet identificar vulnerabilitats dels recursos i solucionar-les ràpidament.

Després de comprovar i solucionar les vulnerabilitats de la instància EC2, s'ha de protegir activament. Primerament, s'ha de comprovar que funcioni correctament.

Si s'accedeix a la IP de la instància, s'obre la web Juice Shop. Per a comprovar que tot estigui correcte, s'explota la vulnerabilitat SQL injection, que permet accedir com a administrador a la web, veient que s'aconsegueix accedir-hi satisfactoriament. Un cop comprovat que l'aplicació és vulnerable, s'ha de configurar AWS WAF per a evitar aquest tipus d'atacs.

AWS WAF no permet protegir directament l'instancia EC2, sinó que protegeix els recursos que es connecten a la xarxa per tal de gestionar el trànsit HTTP(S). Aleshores, per a poder configurar el WAF, primer s'ha de configurar un Balancejador de Càrrega de la instància EC2. Aquesta és una feina relativament senzilla i molt útil, ja que l'objectiu del balancejador és, en cas que hi hagi massa carrega per una instància, enviar les peticions a un altre per a distribuir la càrrega de treball. En aquest cas, el balencejador només disposa d'una instància a la qual passar-li les peticions. Cal destacar que el balancejador ens proporciona un nom DNS, el qual s'utilitza per a accedir a la instància, en cas que hi hagi més d'una que s'encarregui de fer la gestió de peticions.

Un cop fet, s'ha de configurar el tallafoc a la secció d'AWS WAF. Per a fer-ho s'han de seguir els passos següents.

Primer cal dirigir-se a la secció de ACL (Access Control Lists) per a crear el conjunt de normes encarregades de filtrar el trànsit. A l'hora de crear la ACL, s'ha d'especificar la regió (la mateixa en la qual estan desplegats els recursos), el nom, una descripció (opcional), el nom de la mètrica de CloudWatch (servei de visualització de logs) i els recursos objectius a protegir. Seguidament, es configuren les regles de gestió de trànsit. Aquestes poden ser o bé de pagament/privades, creades per Amazon o altres entitats, o pròpies, creades per un mateix usuari. Per a protegir Juice Shop, es necessiten dues regles: una per a protegir la web contra atacs XSS i un altre contra SQL Injections. Per al primer atac, s'especifica que es revisa el cos de la petició (dades) en busca d'atacs XSS (aquesta és una opció pròpia d'AWS que guarda cadenes típiques d'aquest atac), s'especifica la sensibilitat de detecció a alta i s'especifica que, en cas que es trobi una petició com la descrita, la rebutgi. Per a l'atac de SQL Injection, tots els passos són els mateixos a excepció que en el cos es busquen atacs SQL Injection (de nou es una opció pròpia d'AWS) en lloc de XSS. Cal destacar que per evitar que bots accedeixin a la web, s'ha creat una norma extra en què bloqueja tota connexió que no vingui d'Espanya. Finalment, després de crear les normes, s'especifica la prioritat d'aquestes i s'acaba de crear el ACL.

Tots els passos de la creació de la ACL i les normes es poden veure a les Figures 11, 12, 13, 14 i 15 de l'apèndix A4.

Un cop fet això, la creació del ACL comença, i, passats 2-3 minuts, ja està operativa. Un cop l'ACL està actiu, és el moment de posar a prova la seguretat de la web.

10.4 Atac

En aquest punt, es prova d'explotar totes les vulnerabilitats que s'han explicat anteriorment.

En realitzar l'**atac de força bruta**, no s'aconsegueix desxifrar la contrasenya del compte administrador, ja que les peticions que s'envien a través de Burp Suite han estat bloquejades (codi 401) com es pot veure a la Figura 16 de l'apèndix A4. Aquest és el funcionament esperat de la pàgina web, perquè gràcies a AWS Shield (gratuït) s'està protegit contra eines automàtiques.

Per explotar la vulnerabilitat **SQL Injection**, s'han de seguir els mateixos passos que en l'apartat d'explotació, però, com es pot veure a la següent figura, en el moment en què s'hauria d'iniciar sessió, apareix un missatge de "Forbidden" (veure Figura 3), el qual és retornat pel WAF, ja que ha bloquejat la petició. Cal recalcar que aquest comportament no hauria de ser l'esperat d'una pàgina web, ja que està gestionant les respostes de la web d'una manera errònia.

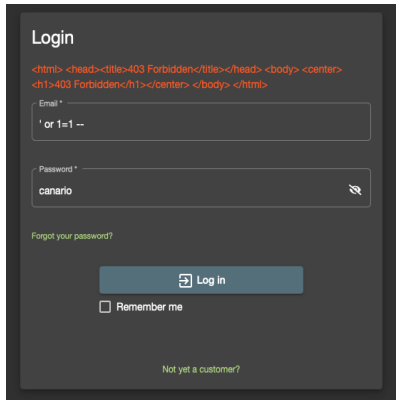

Fig. 3: Bloqueig SQL Injection

En explotar la vulnerabilitat **CSRF**, aquesta, a diferència de les anteriors, no ha estat bloquejada per AWS. Això es deu a que com la vulnerabilitat es centra en "replicar" una petició legitima de canvi de nom, AWS no la detecta com a dolenta. La solució per a una vulnerabilitat del tipus CSRF seria implementar un token CSRF. Aquest procés d'explotació es pot veure al següent vídeo de YouTube [36].

Finalment per a explotar la vulnerabilitat de **XSS Persisted**, de nou es segueixen els mateixos passos que anteriorment, amb la diferència, que, en aquest cas no es registra l'usuari, ja que WAF bloqueja la petició d'enregistrament i no es genera un nou usuari a la base de dades. Tot aquest procés es pot veure al següent vídeo [37].

11 Conclusions

En aquest treball, s'ha portat a terme un estudi sobre la implementació de solucions de ciberseguretat en aplicacions web utilitzant AWS. S'han atacat les principals vulnerabilitats web i s'han presentat solucions pràctiques per a la seva mitigació.

Els aprenentatges obtinguts a l'hora d'estudiar les vulnerabilitats i diferents eines d'explotació, han contribuït de gran manera a l'hora de dur a terme les proves de penetració a les dues webs especificades. Com a enginyer sense un gran coneixement en la ciberseguretat, m'he familiaritzat amb les vulnerabilitats web més conegudes, veient com s'exploten i quines són les pràctiques a seguir per a evitarles. També he analitzat la gran quantitat d'eines d'explotació i de recerca d'informació de les que disposen els hackers, tan ètics com no, i el gran avantatge que donen per aconseguir informació, que de manera manual, potser, no seria possible.

Després d'estudiar sobre les vulnerabilitats i les eines d'explotació, l'objectiu va passar a ser explotar les webs designades. Aquest objectiu ha estat assolit, i no només això, sinó que com a enginyer de ciberseguretat considero que m'ha aportat molta experiència. M'ha permès, de primera mà, veure el mal que poden fer aquestes males pràctiques de programació, robant informació personal i, en el pitjor dels casos, suplantant la identitat dels usuaris. Analitzar diferents maneres en què es pot tractar amb una web, utilitzant proxies que permeten filtrar el trànsit, redirigirlo, o repetir una mateixa petició diferents cops. En resum, la posada en pràctica de l'explotació de vulnerabilitats m'ha permès afrontar diferents reptes, els quals m'han ajudat a guanyar experiència en pentesting, i a pensar de manera més analítica amb les meves accions.

Finalment, el tercer objectiu d'implementar solucions pràctiques amb AWS també ha estat assolit, i m'ha aportat molt de coneixement tècnic. M'ha permès treballar amb una de les eines més potents de Cloud Computing del mercat, creant instàncies, configurant webs, aprenents conceptes d'arquitectura web i sobretot, implementant mesures de seguretat web. Com s'ha vist anteriorment, s'ha aconseguit protegir activament la web, però no només això, sinó que també s'han utilitzat eines d'anàlisis de vulnerabilitats de recursos, que permeten mantenir l'entorn de treball i producció segur i actualitzat.

AWS és una eina molt potent, i, per a la majoria d'empreses, pot aportar molt de valor. Això sí, hi ha una sèrie de consideracions a tenir en compte.

El primer punt a tenir present és la implementació pròpia. Com he anat comentat al llarg del treball, la majoria d'aquestes vulnerabilitats tenen solució, a la qual s'ha d'arribar per a brindar un producte totalment segur. Tot i així, s'ha de tenir en compte que AWS no és perfecte i s'encarrega de donar-nos una capa de seguretat extra, però la nostra feina com a enginyers és desenvolupar el millor codi de la manera més segura possible.

El segon punt a tenir en compte és el preu de l'eina. Tot i ser una eina molt potent, també pot ser molt cara. En aquest treball s'ha utilitzat l'oferta "gratuïta" d'AWS per a comptes nous que dura 1 any. Aquesta oferta permet utilitzar certs recursos de manera gratuïta, però n'hi ha d'altres,

com els balancejadors de càrrega o AWS Shield Advanced, que són de pagament.

Finalment, considero que aquesta eina és molt útil, però no per a un particular, sinó en casos on és necessari disposar d'una gran infraestructura i no es té l'espai suficient, o el capital necessari, ja que el cost d'AWS resulta en funció de la quantitat d'ús i no dels recursos que es tinguin.

Agraïments

Agraeixo a tota la meva família el suport durant el desenvolupament del TFG. Als meus pares, que m'han ajudat i entès sempre que he tingut algun problema; al meu germà, que m'ha fet riure quan he estat angoixat i preocupat; i a la meva parella, que ha estat sempre al meu costat ajudant-me i donant-me suport incondicional.

A tots vosaltres, de tot cor, us dono les gràcies.

Bibliografía

[1] R. Wood, «Damn Vulnerable Web Application (DVWA),» GitHub, 5 Octubre 2015. [En línia]. Available: https://github.com/digininja/DVWA. [Últim accés: 6 Octubre 2024].

[2] OWASP, «OWASP Juice Shop,» OWASP, [En línia]. Available: https://github.com/juice-shop/juice-shop. [Últim accés: 5 10 2024].

[3] C. a. D. R. Rodríguez, «¿Por qué implementar Scrum?,» Ontare, 2015.

[4] Microsoft, «Azure Security,» Microsoft, 2024. [En línia]. Available: https://azure.microsoft.com/en-us/explore/security/. [Últim accés: 5 Octubre 2024].

[5] G. Cloud, «Google Cloud Armor,» Google, 2024. [En línia]. Available: https://cloud.google.com/security/products/armor. [Últim accés: 5 Octubre 2024].

[6] P. Rodríguez, «AWS busca 20.000 nuevos empleados, con 5.500 puestos para ingenieros de software: 200 de las vacantes son para España y 1.400 en remoto,» Xataka, 12 Agost 2021. [En línia]. Available: https://www.xataka.com/pro/aws-busca-20-000-nuevos-empleados-5-500-puestos-para-ingenieros-software-200-vacantes-para-espana-1-400-remoto. [Últim accés: 12 Juny 2024].

[7] «Descripción general de Amazon Web Services: AWS Documento técnico,» Amazon Web Service, 2024.

[8] A. Whitepaper, «Introduction to AWS Security: AWS Whitepaper,» Amazon Web Services, Inc., 2024.

[9] S. Hernandez, «Udemy, Curso Completo de Hacking Ético y Ciberseguridad,» Udemy, 1 12 2024. [En línia]. Available: https://www.udemy.com/course/curso-completo-de-hacking-etico-y-ciberseguridad/?couponCode=ST6MT103124. [Últim accés: 9 Novembre 2024].

[10] G. Lyon, «Nmap: Network Mapper - Documentation,» Nmap Project, [En línia]. Available: https://nmap.org/docs.html. [Últim accés: 10 Octubre 2024].

[11] M. Project, «Metasploit Documentation,» Metasploit Project, 2024. [En línia]. Available: https://docs.metasploit.com/. [Últim accés: 10 Octubre 2024].

[12] P. Ltd., «Burp Suite Documentation,» Burp Suite, [En línia]. Available: https://portswigger.net/burp/documentation. [Últim accés: 10 Octubre 2024].

[13] O. Foundation, «OWASP ZAP Documentation,» OWASP, [En línia]. Available: https://www.zaproxy.org/docs/. [Últim accés: 10 Octubre 2024].

[14] A. G. B. a. M. Stampar, «sqlmap Usage Documentation,» Sqlmap, [En línia]. Available: https://github.com/sqlmapproject/sqlmap/wiki/Usage. [Últim accés: 10 Octubre 2024].

[15] Nsrav, «Denial of Service (DoS),» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/Denial_of_Service. [Últim accés: 10 Octubre 2024].

[16] KirstenS, «Cross Site Request Forgery (CSRF),» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/csrf. [Últim accés: 10 Octubre 2024].

[17] W. Zhong, «Command Injection,» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/Command_Injection. [Últim accés: 10 Octubre 2024].

[18] OWASP, «Types of Cross-Site Scripting (XSS),» OWASP, [En línia]. Available: https://owasp.org/www-community/Types_of_Cross-Site_Scripting. [Últim accés: 10 Octubre 2024].

[19] «Path Traversal,» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/Path_Traversal. [Últim accés: 10 Octubre 2024].

[20] G. a. R. a. T. a. K. a. D. h. a. A. S. a. J. a. M. a. kingthorin, «Brute Force Attack,» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/Brute_force_attack. [Últim accés: 10 Octubre 2024].

[21] I. Tenable, «Tenable Documentation,» Tenable, [En línia]. Available: https://docs.tenable.com. [Últim accés: 10 Octubre 2024].

[22] A. S. Foundation, «Apache HTTP Server Documentation,» Apache, [En línia]. Available: https://devdocs.io/apache_http_server/. [Últim accés: 10 Octubre 2024].

[23] O. Foundation, «Cross Site Scripting (XSS),» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/xss/. [Últim accés: 10 Octubre 2024].

[24] K. a. Zbraiterman, «SQL Injection,» OWASP, [En

línia]. Available: https://owasp.org/www-community/attacks/SQL_Injection. [Últim accés: 10 Octubre 2024].

[25] L. J. Núñez, «Explotació d'una vulnerabilitat ByPass-Client Persisted XSS a Juice Shop,» YouTube, 8 Desembre 2024. [En línia]. Available: https://youtu.be/YKyPKx5Gny8. [Últim accés: 8 Desembre 2024].

[26] Amazon, «What is AWS Elastic Beanstalk?,» Amazon, [En línia]. Available: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html. [Últim accés: 10 Desembre 2024].

[27] Amazon, «¿Qué es Amazon CloudFront?,» Amazon, [En línia]. Available: https://docs.aws.amazon.com/es_es/AmazonCloudFront/latest/DeveloperGuide/Introduction.html. [Últim accés: 10 Desembre 2024].

[28] Amazon, «What is Amazon Elastic Container Service?,» Amazon, [En línia]. Available: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html. [Últim accés: 10 Desembre 2024].

[29] Amazon, «AWS Fargate for Amazon ECS,» Amazon, [En línia]. Available: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/AWS_Fargate.html. [Últim accés: 10 Desembre 2024].

[30] Amazon, «What is Amazon EC2?,» Amazon, [En línia]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html. [Últim accés: 10 Desembre 2024].

[31] Amazon, «AWS Identity and Access Management,» Amazon, [En línia]. Available: https://aws.amazon.com/es/iam/. [Últim accés: 2 1 2025].

[32] Amazon, «Amazon GuardDuty,» Amazon, [En línia]. Available: https://aws.amazon.com/es/guardduty/. [Últim accés: 2 1 2025].

[33] Amazon, «Amazon Inspector,» Amazon, [En línia]. Available: https://aws.amazon.com/es/inspector/. [Últim accés: 2 1 2025].

[34] Docker, «Install Docker Engine on Ubuntu,» Docker, [En línia]. Available: https://docs.docker.com/engine/install/ubuntu/. [Últim accés: 2 1 2025].

[35] Caddy, «Welcome to Caddy,» Caddy, [En línia]. Available: https://caddyserver.com/docs/. [Últim accés: 2 1 2025].

[36] L. J. Núñez, «CSRF not Blocked by AWS,» Youtube, 15 1 2025. [En línia]. Available: https://youtu.be/IP5V8SgD9iE. [Últim accés: 15 1 2025].

[37] L. J. Núñez, «Persisted XSS Block AWS,» Youtube, 15 1 2025. [En línia]. Available: https://youtu.be/guGldYKRz4s. [Últim accés: 15 1 2025].

[38] Microsoft, «Microsoft Entra ID,» Microsoft, 2024. [En línia]. Available: https://www.microsoft.com/en-us/security/business/identity-access/microsoft-entra-id. [Últim accés: 5 Octubre 2024].

[39] J. Security, «bWAPP: A buggy web application for testing,» GitHub, 5 Desembre 2016. [En línia]. Available: https://github.com/jehy-security/bwapp. [Últim accés: 6 Octubre 2024].

[40] s0md3v, «XSStrike,» GitHub, [En línia]. Available: https://github.com/s0md3v/XSStrike. [Últim accés: 10 Octubre 2024].

[41] H. Project, «Introduction to Hydra Documentation,» Hydra Project, [En línia]. Available: https://hydra.cc/docs/intro/. [Últim accés: 10 Octubre 2024].

[42] O. Foundation, «Cross Site Request Forgery (CSRF),» OWASP, [En línia]. Available: https://owasp.org/www-community/attacks/csrf. [Últim accés: 10 Octubre 2024].

[43] Amazon, «¿Qué es Amazon S3?,» Amazon, [En línia]. Available: https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/Welcome.html. [Últim accés: 10 Desembre 2024].

APÈNDIX

A1. Diagrama de Gantt

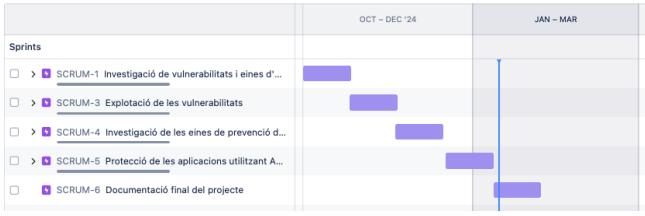

Fig. 1: Diagrama de Gantt

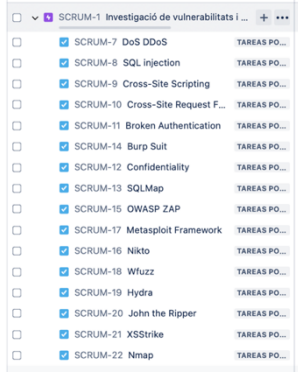

Fig. 2: Sub-tasques 1

Fig. 3: Sub-tasques 2, 3 i 4

A2. Diagrama de vulnerabilitats

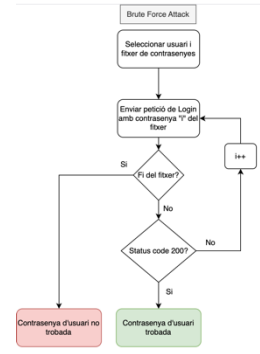

Diagrama 1: Brute Force Attack

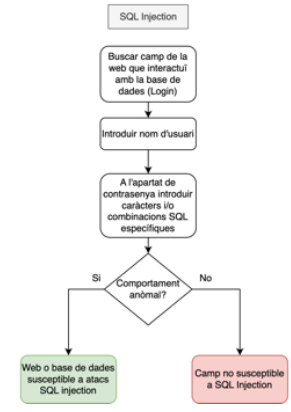

Diagrama 2: SQL Injection

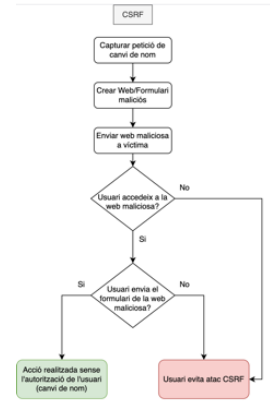

Diagrama 3: CSRF

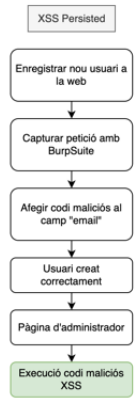

Diagrama 4: XSS Persisted

A3. Vulnerabilitats web

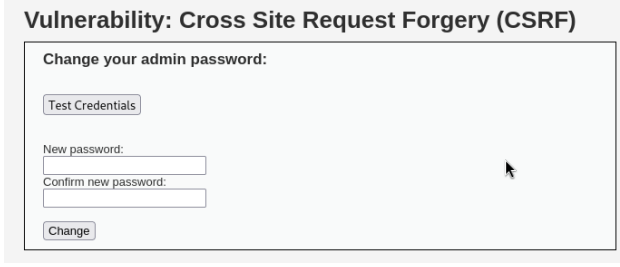

Fig. 1: CSRF Web Form

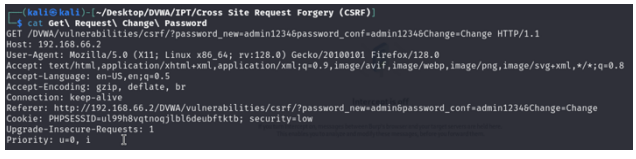

Fig. 2: Petició GET canvi de contrasenya

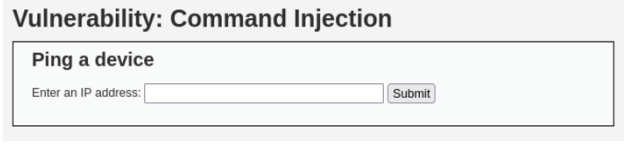

Fig. 3: Command Injection Web Form

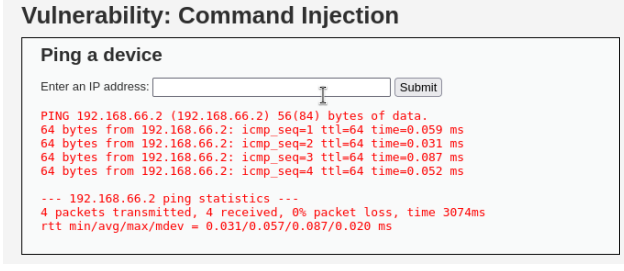

Fig. 4: Command Injection comportament esperat

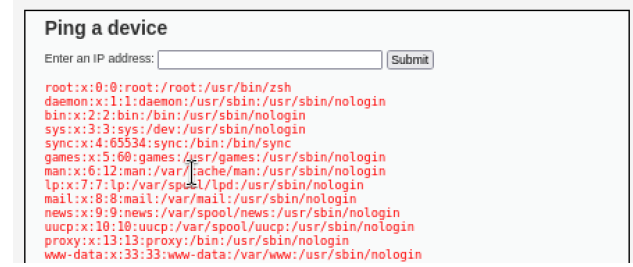

Fig. 5: Command Injection cat output

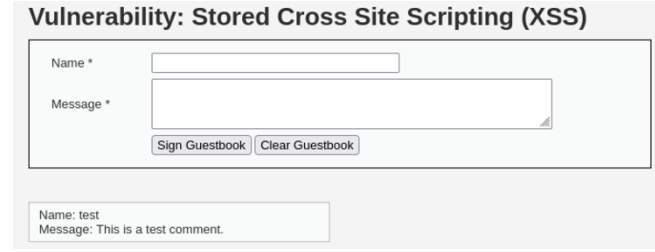

Fig. 6: XSS Formulari

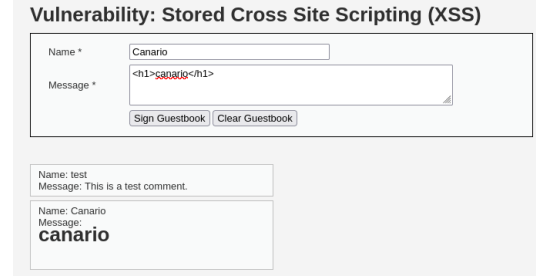

Fig. 7: XSS comentari h1

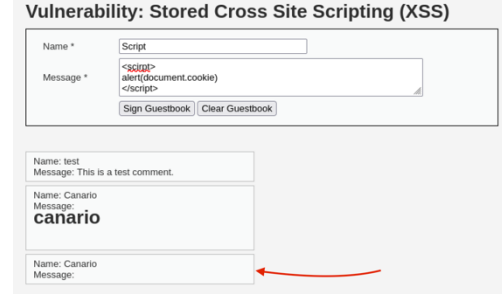

Fig. 8: XSS comentari script

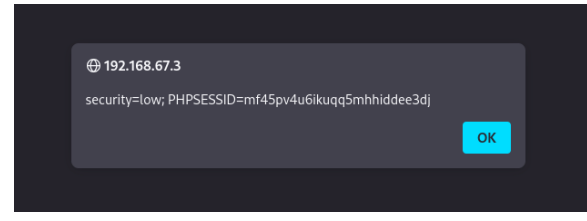

Fig. 9: Cookies de l'usuari mostrades per pantalla

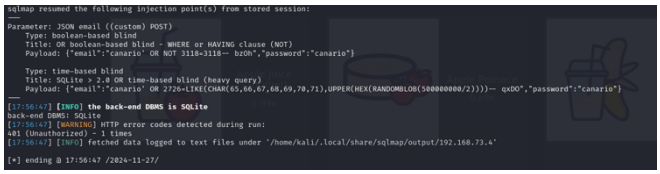

Fig. 10: SQLMap base de dades

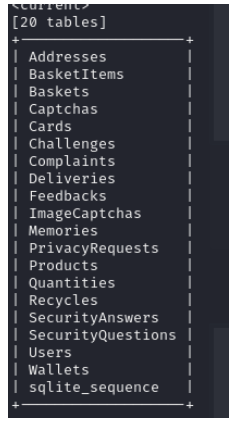

Fig. 11: SQLMap Taules BD

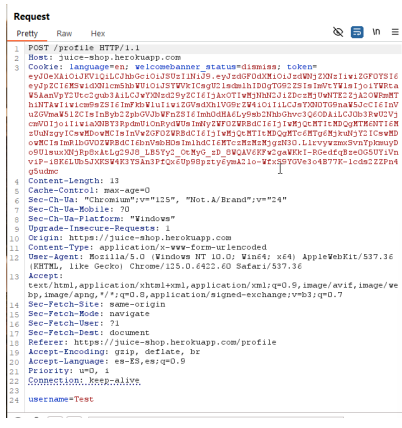

Fig. 12: Petició canvi de nom

Fig. 13: Html CSRF

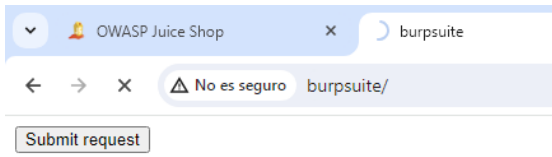

Fig. 14: Botó submit

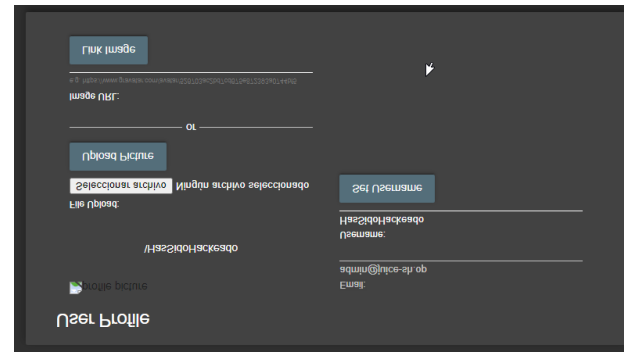

Fig. 15: Canvi de nom a 'HasSidoHackeado'

A4. AWS

Fig. 1: Nom instància EC2

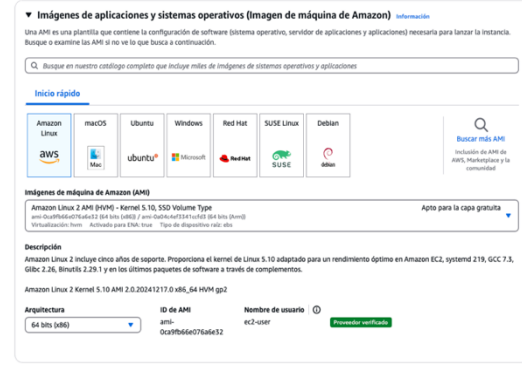

Fig. 2: Tipus de sistema operatiu

Fig. 3: Tipus d'instància (potencia)

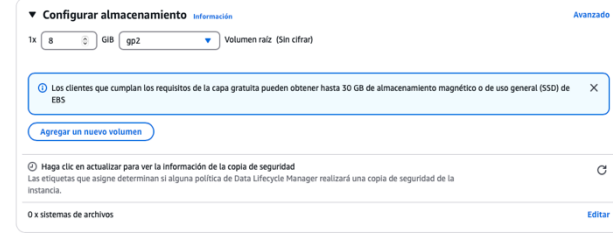

Fig. 4: Emmagatzemant de la instància

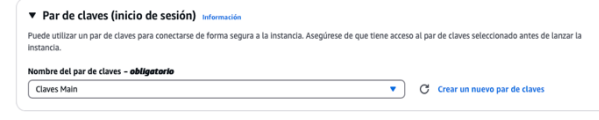

Fig. 5: Parell de claus

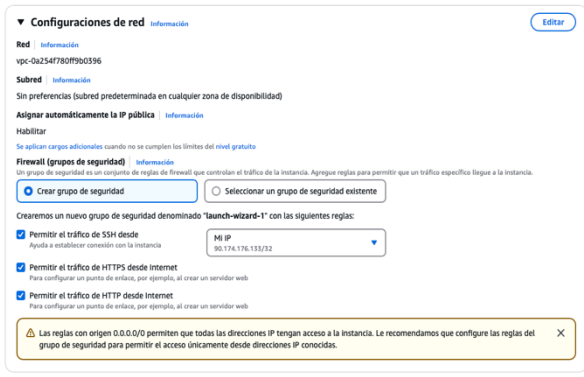

Fig. 6: Configuració de la xarxa i usuaris

Fig. 7: GPU Estàndard

Fig. 8: Regles d'entrada

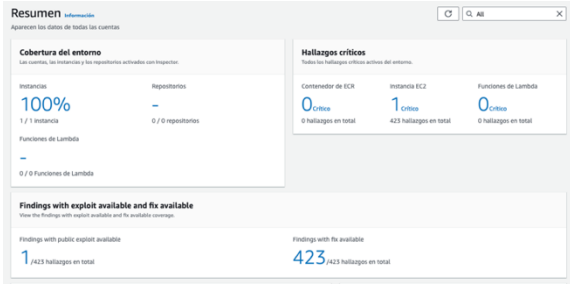

Fig. 9: Panel principal Amazon Inspector

Fig. 10: Taula de vulnerabilitats

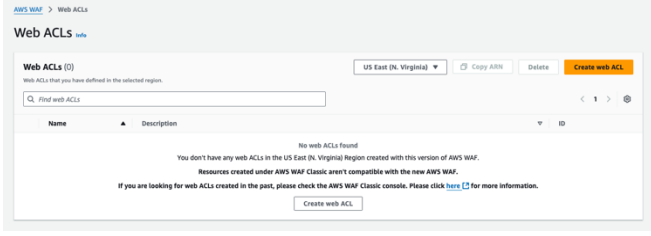

Fig. 11: Panel principal ACLs

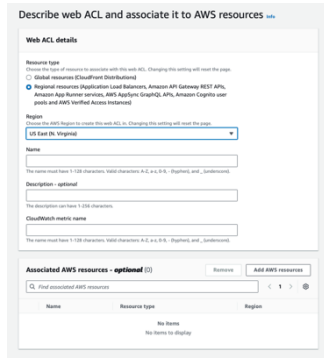

Fig. 12: Nom i recursos ACL

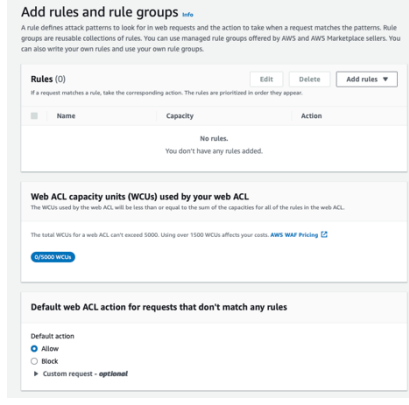

Fig. 13: Regles ACL no declarades

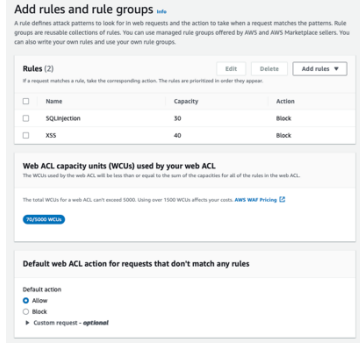

Fig. 14: Regles ACL declarades

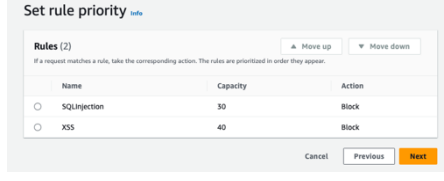

Fig. 15: Ordre de prioritat de les regles

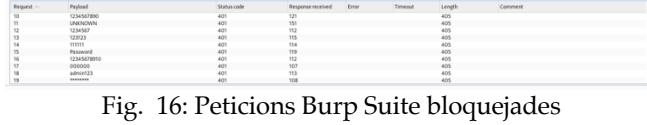

Fig. 16: Peticions Burp Suite bloquejades