Resultados de la auditoría a 27 juegos DeFi durante 2023

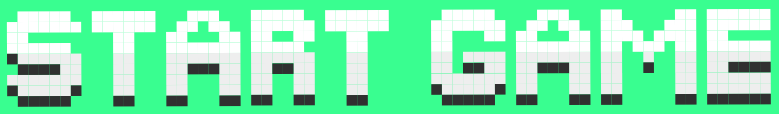
Resultado de la auditoría a 27 juegos de finanzas descentralizadas DeFi realizada por Sofistic durante el 2023

480

SOFISTIC
CYBERSECURITY

En colaboración con:

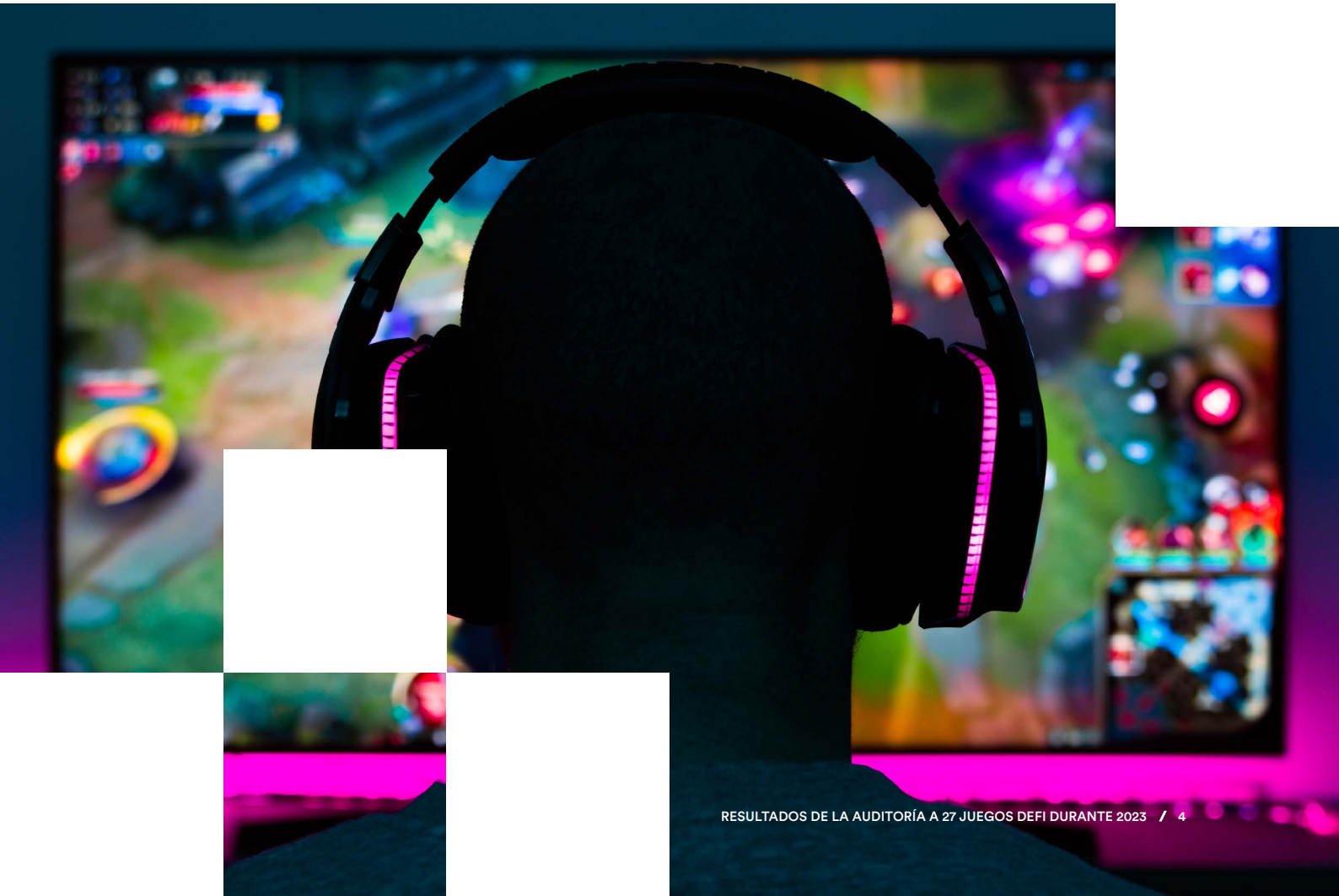
bit 2 me


START GAME

Índice

1. Introducción

En los últimos años, los videojuegos DeFi *(Decentralized Finance)* han ganado popularidad y se han convertido en una opción atractiva para muchos jugadores y entusiastas de la tecnología blockchain. Sin embargo, la seguridad sigue siendo una preocupación importante en este campo, especialmente en lo que respecta a la protección de los activos de los usuarios, y la prevención de posibles ciberataques. En este artículo de investigación, se analizará la seguridad actual de los videojuegos DeFi y se explorarán las medidas de seguridad implementadas por los desarrolladores para proteger a los jugadores y garantizar la integridad de los juegos. Además, se hablará también sobre la transparencia de las empresas que desarrollan este tipo de juegos, y que criterios o condiciones hay que tener en cuenta a la hora de tomar decisiones que vayan a poner en juego nuestro dinero.



2. Justificación

A lo largo de 2021, varios juegos comenzaron a salir al mercado autodenominándose como P2E *(Play To Earn)*, un concepto del que se hablará más adelante, y que se utiliza para referirse a aquellos videojuegos en los que los usuarios pueden llegar a ganar dinero jugando. Todos estos, tienen una característica en común, y es que se hacen valer de la blockchain para gestionar toda la economía del juego, dándoles a los usuarios el control de sus activos. Esto hizo que muchos de los títulos que se publicaron, comenzasen a ganar muchos usuarios y un alto volumen de transacciones en un periodo de tiempo muy corto. Por ejemplo, 'Axie Infinity', uno de los primeros títulos en lanzarse al mundo del P2E, mueve diariamente unos 74 millones de dólares [1]. Su éxito provocó que muchas empresas y desarrolladoras se subiesen al carro, haciendo que los juegos P2E ocuparan el 49% de toda la actividad de aplicaciones descentralizadas en 2022. Aunque no todo fue tan bien como parecía, en marzo de 2022, este mismo título sufrió un ataque en el que se consiguieron sustraer aproximadamente 620 millones de dólares en criptomonedas.

Ante este tipos de ataques y sabiendo que jugar a este tipo de juegos implica que los usuarios tienen que añadir dinero real para poder comprar tokens, mejorar a sus personajes, hacer compras dentro del mercado del juego, etc, se ha decidido realizar una investigación acerca de la seguridad de este tipo de títulos, analizando a fondo la transparencia que ofrecen de cara a los usuarios, qué tipo de vulnerabilidades presentan, como evaluar el nivel de fiabilidad, y otros aspectos interesantes que cualquier persona que vaya a probar este tipo de títulos debería conocer.

2.1. Metodología

Teniendo en cuenta la duración de la investigación, acotada únicamente a dos semanas, se decidió comenzar con un listado de varios juegos P2E que siguiesen los siguientes requisitos:

- ✓ Que hubiesen sido lanzados recientemente.
- ✓ Que contasen con la menos un token ERC20 o NFT.
- ✓ Que sus contratos sean implementados en el lenguaje de programación Solidity.
- ✓ Que se pueda interactuar a través de alguna billetera como podría ser Metamask.

Lo primero que se hizo fue buscar la información disponible relativa al funcionamiento de cada uno de los juegos en sus respectivas páginas web, siendo el recurso más común el *whitepaper*, del que se hablará más adelante. También se revisaron todas las direcciones de los contratos que implementan la lógica de los juegos, necesitando en muchos casos analizar el código fuente de la página web o las interacciones en la blockchain con los contratos de los tokens.

Además, hay que tener en cuenta que la seguridad de los juegos P2E o web3 no se limita solamente a los *Smart Contracts*, sino que tanto el cliente como el servidor también pueden contener vulnerabilidades. Por lo tanto, se consideró evaluar la seguridad de estos tres componentes (contratos, cliente y servidor), ya que forman la base sobre la que se construyen este tipo de juegos. Todo esto, con el objetivo de poner en común todos los fallos que puedan afectar a cada uno de ellos, así como la transparencia de las empresas que se encargan de desarrollar estos videojuegos de cara a los futuros usuarios e inversores.

3. Muestras

Con el fin de tener una variedad de juegos P2E, se han revisado un total de 27 juegos a lo largo de 2 semanas aproximadamente. De estos, el 70% ya habían sido lanzados, y los restantes estaban en versión beta o desarrollo (que no implica que no sean auditables). Entre los juegos elegidos, 8 de ellos utilizaban Polygon, 7 Binance Smart Chain, 2 Ethereum, 6 son *multichain* y los otros 4 utilizaban otras tecnologías. Como fuente de información de cada uno de los juegos, se ha revisado la página web, su repositorio de Github, y toda la información pública de la blockchain y su *whitepaper*, ya que todos los juegos revisados, menos uno, contaban con uno.

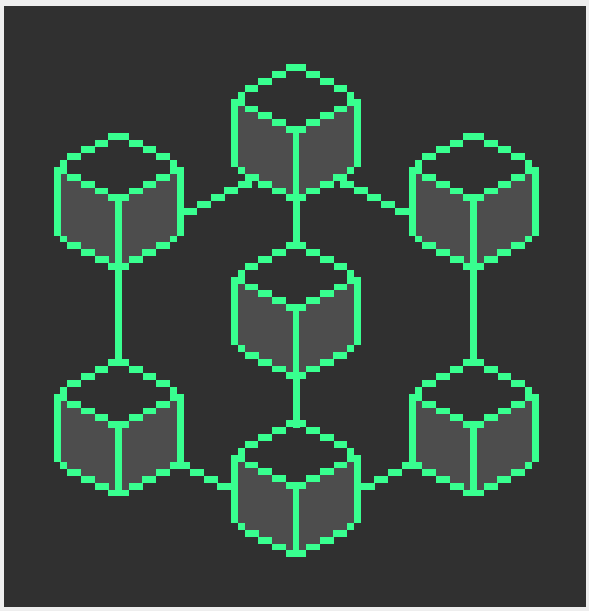
Como se comentará más adelante, todos estos juegos, de normal, tienen un Marketplace donde los usuarios pueden intercambiar sus tokens por objetos para sus personajes o ciertas mejoras. Estos pueden ser implementados por la propia desarrolladora o se puede recurrir a terceros que ya lo tengan implementado. Un 70% de los proyectos recurre a la primera opción, la cual se explicará más adelante ya que no ni la mejor opción ni la más segura.

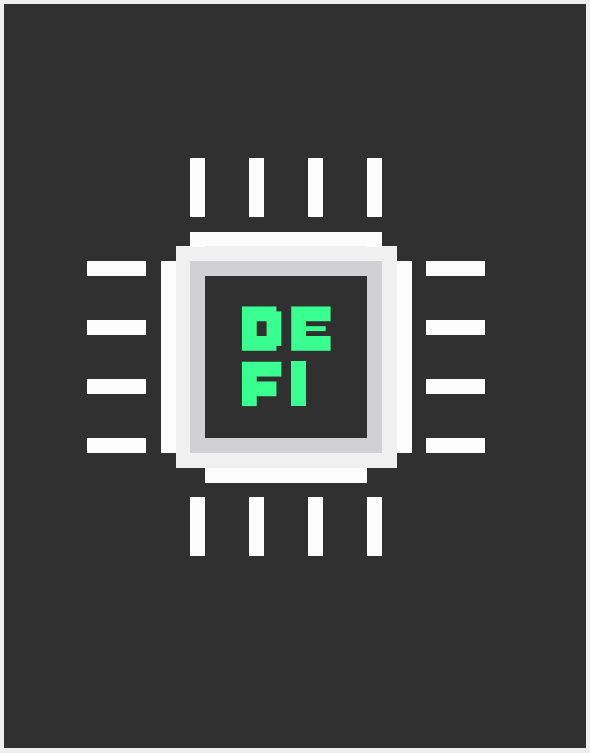
En cuanto a los contratos, un 65% de los proyectos, no ponía fácil la tarea de acceder a sus direcciones, lo cual ya añade un gran problema a la hora de auditarlos y, además, deja ver la falta de transparencia de este tipo de proyectos, algo que se detallará en puntos posteriores. Por último, añadir algo de lo que se hablará más adelante, que tiene que ver con la implementación de la lógica fuera de la blockchain, y es el hecho de emplear sistemas de autenticación mediante usuario y contraseña. Esto es algo que realmente se puede evitar identificando a los usuarios por sus billeteras, evitando así posibles fallos de autenticación. Aunque es verdad que más de la mitad de los juegos utilizan la opción de la identificación mediante billeteras, hay bastantes juegos que aún siguen obligando a sus jugadores a utilizar usuario y contraseña.

4. Definiciones

4.1. Web3

Web3 o Web 3.0 es un concepto utilizado para referirse a la web como la conocíamos hasta ahora, pero que incorpora algunos conceptos clave propios de la blockchain tales como la descentralización, la economía de tokens o la lectura y escritura de datos sobre la blockchain. Haciendo uso de esta tecnología, los usuarios pueden tener propiedades digitales como pueden ser tokens, imágenes, música o cualquier tipo de fichero, inclusive certificados o títulos de propiedad de valores físicos pudiendo venderlos, intercambiarlos o gestionarlos, sin depender de una entidad centralizada, que normalmente restringe el uso de estos activos digitales dentro de su ecosistema. Todo esto, claro está, dentro de una aplicación web que puede correr en cualquier navegador, como lo haría cualquier página web a la que estamos acostumbrados.





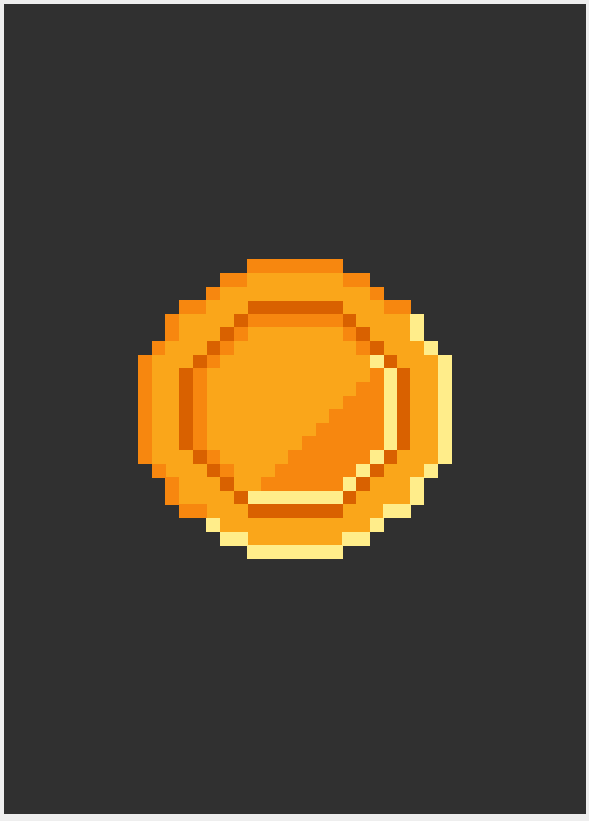
4.2. Juegos DeFi

El término “DeFi” viene de las palabras: ‘descentralizado’ y ‘finanzas’, es decir, la administración de finanzas de manera descentralizada, donde todas las reglas y condiciones quedan definidas en los contratos inteligentes, sin necesidad de tener que depositar la confianza en una entidad centralizada y que, además, se gestiona de manera automática.

Por lo tanto, teniendo este concepto en cuenta, un juego DeFi es aquel que implementa un sistema de gestión de finanzas de manera descentralizada según sus necesidades. Un claro ejemplo de esto, y por lo que apuestan muchos juegos, son los mercados en los que puedes comprar equipamiento y mejoras para tu personaje dentro del juego mediante tokens ERC20. A su vez, estos objetos son representados mediante NFTs (ERC721) y ambos pertenecen al wallet del usuario, que tiene la libertad de usarlos tanto en el juego como en cualquier otra plataforma compatible.

4.3. P2E

P2E (Play To Earn), traducido al español ‘jugar para ganar’, es un concepto al que muchas desarrolladoras se han unido a lo largo de 2021. Básicamente, consiste en ganar tokens con un valor real en el mercado por conseguir logros dentro de un videojuego. Además, suelen ser juegos en los que invertir dinero real es necesario si se quiere sacar algún beneficio jugando. Estos juegos no necesariamente tienen que ver con la blockchain, llevan ya muchos años en el mercado. Un claro ejemplo son los juegos de apuestas o casinos online. La diferencia es que con la blockchain se ha conseguido implementar funcionalidades muy interesantes que le añaden valor a los videojuegos. Como se ha comentado anteriormente, algo que suele hacerse es implementar mercados dentro de los juegos donde puedes comprar y vender objetos y accesorios para mejorar a tu personaje. Esto incita a muchos jugadores a probar estos juegos, ya que ya no estarían jugando a cambio de nada, sino que podrían llegar a ganar dinero, por ejemplo, vendiendo objetos de gran valor dentro del mercado.



5. Resultados

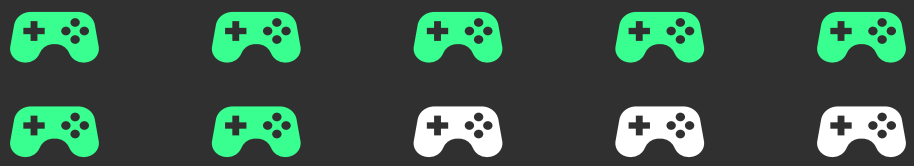
5.1. Transparencia

5.1.1. Whitepaper

¿Qué es un whitepaper?

Un *whitepaper*, dentro del ámbito de las criptomonedas y blockchain, se usa para dar nombre a todos aquellos documentos técnicos que explican todos los aspectos necesarios para entender el funcionamiento del sistema o aplicación al que están haciendo referencia. Por ejemplo, Bitcoin tiene su propio *whitepaper* en el que se explica con todo detalle y a nivel técnico su funcionamiento. Esto es esencial en cualquier proyecto blockchain y sirve como base para los nuevos usuarios o desarrolladores que quieran involucrarse.

7 de cada 10 juegos no muestran las direcciones de sus contratos en el whitepaper



Dentro del ámbito de los juegos P2E, como se ha mencionado anteriormente, una de las cosas más importantes y esenciales tanto para la confianza de los usuarios como para la seguridad del proyecto, es añadir un apartado, normalmente, dentro del propio *whitepaper*, con las direcciones de los *smart contracts* que se empleen para el funcionamiento del juego. Un problema bastante habitual es que la mayoría de los usuarios tienden a pensar que dentro de un juego P2E solo existen los contratos ERC20 y ERC721, utilizados para los tokens y NFTs, respectivamente. Esto no es así, ya que hay mucha más lógica que gestionar por *smart contracts* aparte de los propios tokens. Que más de la mitad de los juegos estudiados, no tengan un apartado donde muestren las direcciones de los contratos que gestionan la economía del juego, deja ver una clara falta de transparencia hacia los usuarios.

Un 88% de los juegos
no daban una mínima descripción de los contratos que controlan la economía del juego

La mayoría de los proyectos tomados como muestra incluyen en su *whitepaper* una descripción del funcionamiento del juego, pero en cuanto a los contratos, solo un 10% dan una breve explicación de la mecánica de estos, y teniendo en cuenta que son el principal núcleo de la gestión de capital del proyecto, es algo de lo que se debería hablar en profundidad. Esto, junto a las direcciones de los contratos, es esencial para que, cualquier persona que lea el documento, tenga una idea de cómo el juego gestiona a nivel técnico la economía de sus activos.

Únicamente uno de los juegos auditados daba una pequeña descripción de que parte del juego se había implementado off-chain, y que parte estaba implementada en la blockchain

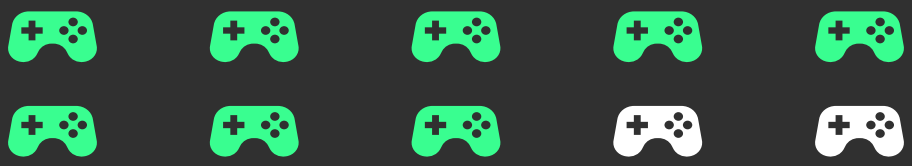
Algo también muy importante a tener en cuenta en este tipo de proyectos, es saber qué parte del juego está implementada en la blockchain, y qué otra parte está implementada *off-chain*. Es decir, si un juego promete ser descentralizado, pero en la realidad, el único componente implementado en la blockchain es el token ERC20, está engañando a los usuarios, ya que toda la lógica que se encargue de manejar la economía del juego va a estar centralizada en unos servidores. Es importante tener esto en cuenta, ya que mucha gente podría llegar a pensar que tener algún que otro token ERC20 ya implica que todo el juego esté implementado de manera descentralizada, cuando en realidad, no es así.

5.1.2. Tokenomics

¿Qué son los tokenomics?

Tokenomics, traducido al español como 'economía de los tokens', es un concepto utilizado dentro del mundo de las criptomonedas que, entre otras cosas, define la manera en la que se van a distribuir y administrar los tokens de un proyecto a lo largo del tiempo. Esto es muy importante por dos cosas: en primer lugar, que un proyecto de estas características necesita distribuir sus tokens de manera inteligente, para que comiencen a tener un valor en el mercado y que no se diluya con el tiempo y, en segundo lugar, presentar esta información a los inversores para generar confianza y poder proyectar las expectativas a futuro.

8 de cada 10 juegos tienen una sección de tokenomics en su whitepaper, pero de estos, solo un 10% lo implementa mediante contratos

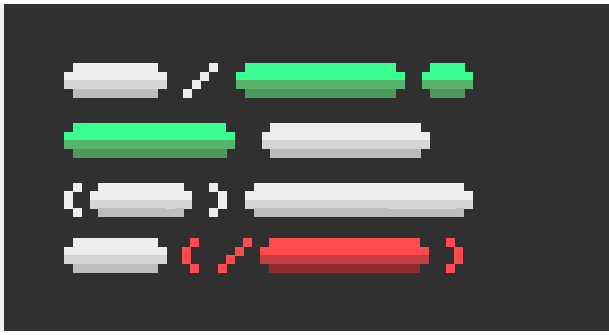


Lo correcto a la hora de definir los *tokenomics* de un proyecto es que aparezcan descritos en el *whitepaper*, justificando su distribución y que, además, se vea reflejada su implementación en el código de los *smart contracts*, de manera que la distribución sea automática y que cualquier persona pueda verificar que se están repartiendo los tokens tal y como consta en el *whitepaper*. En la mayoría de los proyectos que hemos tomado como muestra esto no ocurre así, y se suele delegar esta responsabilidad a una persona que lo gestiona manualmente, haciendo que todo el capital del proyecto dependa de la confianza de un tercero y exponiéndolo a la posible pérdida o filtración de las claves privadas de la cuenta del administrador.

5.1.3. Contratos

¿Se verifican los smart contracts?

Los Smart Contracts, son trozos de código que se ejecutan en la blockchain y son los encargados de gestionar la economía de este tipo de juegos. Para que cualquier persona que vaya a invertir dinero en el juego pueda leer el código de estos contratos necesita dos cosas: la dirección del contrato, y que este contrato esté verificado, lo que permite que se pueda leer su código en el explorador de bloques.



Como se ha comentado en apartados anteriores, muchos de los proyectos no daban acceso a los contratos utilizados para implementar la lógica del juego, por lo tanto, es difícil dar datos sobre qué cantidad de proyectos verifican sus contratos o no, ya que la muestra que tenemos de smart contracts es muy pequeña en comparación al número de juegos auditados. Aun así, se quiere hacer hincapié en la importancia de verificar los contratos ya que esto afecta a la confianza de los usuarios, pues hay que tener en cuenta que se trata del código que gestiona de manera directa sus inversiones. Además, que el código sea visible, ayuda a que otros desarrolladores puedan revisarlo y alertar a tiempo posibles vulnerabilidades, o en el peor de los casos, avisar a los inversores de funcionalidades dentro de los contratos que permitan la malversación de los fondos del proyecto.

5.1.4. Auditorías

Uno de los puntos más importantes a tener en cuenta para saber si un proyecto se ha preocupado de verdad por la seguridad de su economía, es que haya contratado una auditoría a una empresa externa. El alcance de esta debería contener todos los contratos utilizados en el proyecto, así como los demás componentes que interaccionan con la blockchain, como podría ser el cliente o el servidor.

Únicamente un 30% de los juegos revisados auditan sus contratos, y en la mayoría de casos, solo se limitan a los tokens ERC20 y ERC721

Algo que se ha visto de manera muy frecuente en los proyectos revisados es que se contratan auditorías solo para el contrato del token (ERC20), dejando sin auditar los demás contratos que implementa la lógica del juego como podría ser marketplace, pujas, staking, etc. Estos contratos suelen gestionar la interacción de los tokens, por lo que sus vulnerabilidades afectan a la economía del juego de la misma manera que lo pueda hacer el token. Además, hay que tener en cuenta que son justo estos contratos los más susceptibles a tener fallos, ya que, a diferencia del contrato del token ERC20, que suele ser más estándar, tienen una implementación hecha a medida que requiere de mucha más atención.

Ninguno de los juegos ha pasado auditorías ni del lado del cliente ni del servidor

Hay que tener en cuenta, que a pesar de ser juegos que, en principio, su lógica debería estar implementada en la blockchain, tal y como ya se ha explicado anteriormente, necesitan de un cliente y servidor para la página web, para almacenar ciertos datos de los usuarios, gestión de cuentas, etc. El problema radica en que, al tratarse de juegos, no existe ninguna regulación sobre estos, así que no es necesario que pasen ningún tipo de control, como sí lo deben pasar entidades bancarias o cualquier empresa que se dedique a la gestión de capital. Por esto, muchas de las desarrolladoras se ahorran el paso de tener que auditar tanto el cliente como el servidor.

5.2. Fallos

A continuación, se van a presentar una serie de fallos que se han encontrado en algunos de los juegos auditados y que podrían poner en riesgo el capital de dichos proyectos. Los hemos dividido en tres apartados diferentes teniendo en cuenta la comunicación entre cliente, servidor y blockchain. Antes de nada, cabe mencionar, que muchas de las vulnerabilidades que se han encontrado no son referentes a la blockchain o *smart contracts*. Esto es importante tenerlo en cuenta ya que se suele asociar blockchain con seguridad, y si bien es cierto que la blockchain se sustenta en una base criptográfica sólida, sí que puede haber vulnerabilidades tanto en el código de los *smart contracts* como en la comunicación de la blockchain con el exterior y viceversa.

5.2.1. Smart contracts

5.2.1.1. Gestión de capital

Un 72% de los juegos auditados
implementan funciones abusivas o tienen roles de administrador con un exceso de poder

Uno de los aspectos más importantes a la hora de conocer el nivel de fiabilidad de estos videojuegos es saber cómo gestionan el dinero internamente. A continuación, se describen algunas prácticas que se han encontrado y que pueden poner en duda la credibilidad de las desarrolladoras. El problema es que estas son muy fáciles de implementar, pero pueden tener un impacto muy grande en el valor del token y en el balance de los tokens de los usuarios y, además, pueden pasar desapercibidas a la vista de un usuario común:

Uso incorrecto de las *black lists*. Se pueden añadir direcciones arbitrarias a una black list congelando de esta forma sus tokens.

Cambiar las comisiones de algunas transacciones hasta el 100% haciendo que todos los tokens se envíen al beneficiario de las comisiones y el destinatario de la transacción no reciba nada. Las comisiones deberían ser fijas, o tener un pequeño rango de variación.

Crear un numero indefinido de tokens por parte del administrador sin ninguna restricción, diluyendo así el valor de los tokens de los demás inversores.

También se ha observado que algunos de los NFTs tienen características definidas en un fichero JSON, el cual se almacena fuera de la blockchain, guardando su ruta en una variable registrada en el contrato, denominada tokenURI. Si el fichero no está identificado por su hash, este es fácilmente manipulable permitiendo, además, que haya múltiples copias del mismo y ambas cosas afectan al propietario del mismo. Los NFT permiten usar URLs que apuntan a un fichero que representa el token, el cual podría ser una imagen, vídeo, sonido, etc. Este tipo de ficheros suelen ser muy grandes para almacenarlos en la blockchain, debido al coste que esto supone. En cambio, la información que no requiere mucho espacio en memoria es mejor guardarla de forma segura en el propio contrato y consultarla desde ahí.

Para evitar que se puedan manipular estos ficheros se puede usar un servicio de alojamiento descentralizado como, por ejemplo, IPFS. Esto permite usar un hash único para cada fichero almacenado impidiendo que existan copias, aunque, no garantiza la persistencia del fichero, ya que, para ello, los administradores del proyecto deberían ejecutar un nodo IPFS y alojar los ficheros en el mismo.

5.2.1.2. Administración y privilegios

Otro caso común de falta de compromiso y confianza hacia los usuarios es el de exceso de privilegios. Está bien que un *smart contract* tenga la posibilidad de retirar rápidamente sus fondos y redistribuirlos en caso de ataque, pero esto no implica que haya funciones como las que se van a describir a continuación que permitan realizar ciertas acciones que pongan en juego el dinero de los usuarios:

Implementación de métodos que permiten a los administradores transferir o destruir tokens de cualquiera de los usuarios sin su consentimiento, dejando en duda la propiedad de los activos.

Métodos que permiten retirar todos los fondos de los usuarios a una dirección solo con una llamada a una función. Estas funciones probablemente se diseñaron para que, en caso de emergencia, se puedan sacar los activos de manera rápida. Sin embargo, se debería cambiar su implementación de manera que los activos sean devueltos a sus respectivos usuarios ya que, en caso contrario, cabe la posibilidad de que el administrador, aprovechando sus privilegios, se quede los fondos de los usuarios.

Este tipo de funciones no solo perjudica al proyecto a nivel de transparencia y confianza de los usuarios, si no que, también implica un impacto muy grande en caso de que la cuenta del administrador sea robada, ya que se le estaría "facilitando" la extracción de los fondos al hacker. Al igual que un banco centralizado, los administradores son capaces de modificar el estado de las cuentas, pero a diferencia de estos, los cambios en la blockchain no se pueden revertir ni se puede pedir responsabilidades al autor, ya que, según el contrato, entra dentro de sus privilegios y que además este puede ser anónimo.

5.2.1.3. Implementación de lógica compleja

Como en cualquier videojuego, es necesario manejar cierta lógica que se encargue de algunas funcionalidades del juego como un sistema de pujas, aumento de nivel de un personaje, un *marketplace*, etc. Un buen diseño debería separar toda esta lógica permitiendo, por ejemplo, actualizar un contrato con nuevas funcionalidades, sin necesidad de modificar el token y facilitando, así, el mantenimiento y las actualizaciones del juego. Si bien es cierto que la mayoría de los proyectos cuentan con un diseño acertado en este aspecto, dos de los proyectos auditados implementaban en el mismo contrato funcionalidades relativas a la lógica del juego y los tokens (ERC20, ERC721 o ERC1155), es decir, a la vez que había funciones propias del token como la transferencia o creación de los mismos, había también métodos encargados de gestionar un sistema de ventas y pujas, propios del *marketplace*, por lo que una actualización importante obligaría a cambiar el token generando desconfianza entre los inversores. Uno de estos, tenía un fallo que afectaba de manera directa al sistema de pujas, permitiendo el reseteo del precio de las pujas o la denegación de servicio del *marketplace*.

La gran mayoría de las vulnerabilidades tienen como origen fallos en la lógica de la implementación, o simples despistes los cuales se podrían evitar con una amplia batería de tests que cubra todas las funcionalidades, así como comprobar que la lógica de negocio esperada concuerda con la implementación. Por ejemplo, en uno de los proyectos, dos de los contratos tenían funciones administrativas sin control de acceso permitiendo que cualquiera pudiese cambiar un contrato que se encargaba de calcular el precio de compra o puja de un NFT, por lo que se permitía inyectar un contrato malicioso dando lugar, entre otras cosas, a compras a precio cero o el robo de todos los tokens del comprador. En otro caso, sucedía lo contrario, una función requería que el usuario fuese a la vez el *owner* y administrador y, por lo tanto, a no ser que la dirección tuviese los dos roles no podía acceder, haciendo que sean roles redundantes.

5.2.1.4. Retrocompatibilidad y lógica off-chain

A continuación, se hablará de cómo a veces, la implementación de cierta lógica fuera de la blockchain puede tener efectos negativos en el proyecto, poniendo en riesgo la seguridad y confianza de los fondos y usuarios respectivamente. En alguno de los juegos auditados, para facilitar el uso del juego a usuarios con poca experiencia DeFi, permiten que el *wallet* sea opcional, dejando registrarse en el juego solo con usuario y contraseña sin enlazar una billetera. Esto conlleva ciertos aspectos negativos dependiendo de la implementación, pero dos de los más comunes tienen que ver con la lógica duplicada *offchain* o que la empresa encargada de desarrollar el proyecto sea la que custodia las *wallets* de estos usuarios.

En el caso de tener una lógica duplicada *offchain*, que es cuando calculan y gestionan los cambios de estado de los valores tanto en la *blockchain* como en el servidor, puede haber discrepancias entre la contabilidad de las dos partes. Además,

esta solución es propensa a tener condiciones de carrera dando lugar al problema de doble gasto. Por otra parte, si se tiene la custodia de los *wallets* es necesario que las comisiones de las transacciones las pague quien tiene la custodia de los mismos. En ambos casos se pierde la descentralización por completo, aunque en el segundo caso se puede implementar soluciones para que un usuario pueda reclamar los *assets* en otra cuenta a posteriori.

Es habitual que la lógica de parte de los *assets* que se gestionan en el juego se implemente *off-chain* de forma centralizada, por diferentes motivos como por ejemplo la retrocompatibilidad web2, un diseño inadecuado a los requisitos, o el uso de redes blockchains menos eficientes. Sin embargo, al no tener la lógica básica del juego ligada a un contrato se pierde la transparencia y parte de la descentralización haciendo que sea un juego híbrido.

La integración de la tecnología blockchain se hace en mayor o menor medida dependiendo del juego, debido a varios motivos que pueden condicionarlo como pueden ser:

Lógica *off-chain*:

- Dependiendo de la red las transacciones tienen un coste y requieren un lapso de tiempo para que se confirmen, lo cual puede tener un impacto negativo en el juego.
- Si el juego requiere de cambios de estado frecuentes en la lógica.
- Requiere de más experiencia en el desarrollo de *smart contracts* implementar toda la lógica que solo la parte básica.
- Es más fácil hacer cambios en la lógica para añadir funcionalidades.

Lógica *on-chain*:

- Es más transparente de cara a los usuarios dando mucha más confianza en el proyecto.
- Da más libertad a los usuarios para gestionar sus tokens de manera descentralizada.
- Es más seguro ya que, se eliminan muchas de las vulnerabilidades que pueda haber en el *frontend* por la comprobación en los contratos.
- Evita que se pueda manipular la base de datos guardando el estado en la blockchain.
- Todos los cambios quedan registrados y se pueden consultar a posteriori para estadísticas o en el caso de un fallo.

5.2.1.5. Calidad del código

El 60% del código auditado
no contiene comentarios descriptivos sobre su funcionalidad

Algunos de los contratos auditados presentan implementaciones con un uso excesivo de gas, código repetido o sin usar, uso de funciones no seguras, falta de comentarios sobre todo en funciones más complejas, y otros aspectos que denotan falta de madurez por parte de los desarrolladores. Teniendo en cuenta que, a diferencia de otro tipo de software, generalmente la implementación de los *smart contracts* es difícil o imposible de actualizar, hace que estos aspectos negativos se mantengan a lo largo del tiempo.

5.2.1.6. Marketplace propio vs de terceros

7 de cada 10 juegos que tienen marketplace lo implementan de manera independiente y no recurren a terceros

Mientras algunos de los juegos usan *marketplace* de terceros para vender NFTs o swaps para intercambiar los tokens otros usan implementaciones propias. Estas además de ahorrarse las comisiones que puedan tener los *marketplace* externos, ofrecen un mayor grado de libertad para personalizar el proceso de compra/venta/swap.

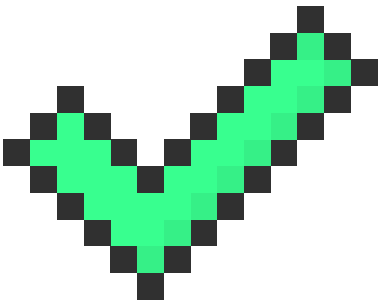
En contrapartida, usar una implementación propia es más propensa a vulnerabilidades que las implementaciones de terceros las cuales suelen estar mucho más probadas y auditadas. Estos servicios externos también pueden proporcionar una mayor visibilidad del proyecto por el mayor número de usuarios.

Si lo que se pretende es gestionar todos los contratos internamente sin depender de servicios externos y ahorrarse sus comisiones se puede usar, como solución intermedia y siempre que la licencia lo permita, las mismas implementaciones para publicar nuevos contratos propios.

2.1.7. Validaciones manuales

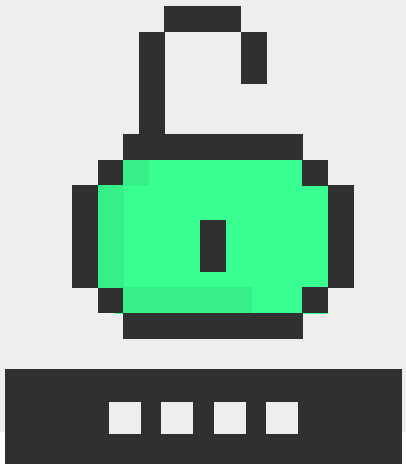
Se ha podido verificar que, en algunos casos, se hacen validaciones manuales de transacciones críticas como reclamar tokens, es decir, cambiar tokens por dinero real. Este tipo de acciones son propensas al error humano y susceptibles a manipulaciones por parte de la persona física que se encarga de esto, sin tener en cuenta la demora que esto pueda suponer.

Por lo general este tipo de interacciones se pueden evitar automatizando todas las validaciones que una persona puede hacer antes de aceptar la transacción de manera que se ejecute con las mismas garantías, pero evitando todos los aspectos negativos mencionados anteriormente.



5.2.2. Servidor

5.2.2.1. Autenticación y gestión de sesiones



Los sistemas de autenticación, autorización y manejo de sesiones son componentes esenciales en la seguridad de la información en aplicaciones y sistemas en línea. La autenticación se refiere al proceso de verificar la identidad del usuario, lo que puede incluir la verificación de credenciales como nombre de usuario y contraseña, tokens de seguridad o biométricos. El manejo de sesiones se refiere a la gestión del tiempo de inicio y finalización de una sesión de usuario en una aplicación, lo que puede incluir la gestión de tokens de autenticación, la expiración de la sesión y la revocación de accesos.

Tan solo 2 de los 27 juegos auditados implementaban doble factor de autenticación

Existen dos métodos de autenticación principales: usuario y contraseña o firma mediante las claves de la billetera. Cerca de un 65% de los proyectos analizados, implementaban autenticación mediante firma. Este método de autenticación se basa en la firma de algo conocido (nonce) mediante la clave privada del usuario, la cual podrá ser verificada del lado de servidor usando la clave pública. Generalmente se usan ambos tipos de autenticación y se vinculan ya que, la aplicación móvil no suele tener implementación web3 por lo que se inicia sesión con usuario y contraseña, mientras en la web se suele iniciar sesión con la billetera o de las dos formas.

En estos procesos se ha de prestar especial atención en esta verificación final. De no realizarse, cualquier usuario podría autenticar en nombre de otro. De hecho, varios de los proyectos auditados fallaron en este punto y permitieron acceder a la plataforma suplantando la identidad de cualquier otro usuario, con todas las consecuencias que ello conlleva.

En uno de ellos no se verificaba del lado de servidor la firma generada por el usuario, permitiendo enviar cualquier dirección de billetera para hacerse pasar por ese usuario. En el segundo ejemplo, se identificaron tres parámetros en la petición de autenticación al servidor: el primero indicando la dirección de la billetera con la que se pretende autenticar; el segundo el nonce firmado, y el tercero la clave pública del que firma. Al realizarse la comprobación, se tenían en cuenta estos tres parámetros enviados por el usuario para todo el proceso, que podían ser falseados para hacer que el servidor verificara una firma y autenticara como un tercer usuario.

Por otra parte, los proyectos que implementaban autenticación con usuario y contraseña no eran capaces en un 70% de los casos, de implementar una política de contraseñas seguras y, en su mayoría, no implementaban tampoco un método para protegerse de ataques de fuerza bruta, simplificando el proceso de obtención de cuentas por prueba y error.

Además, tan sólo dos de los proyectos tenían disponible algún tipo de segundo factor de autenticación que proteja las cuentas contra robos de credenciales. Este tipo de medidas de seguridad debería ser un standard de facto, sobre todo en proyectos donde se gestiona dinero. En cualquier caso, este sistema debe estar correctamente implementado, realizando las verificaciones necesarias del lado de servidor. Hacemos especial hincapié en esto, puesto que en uno de los dos proyectos que implementaban esta protección, la implementación no estaba realizada de forma segura y fue posible sortear esta protección simplemente eliminando un parámetro en la petición web.

En cuanto a la gestión de sesiones, cerca de un 80% de los proyectos implementaban JWT y el resto se apoyaban en cookies de sesión.

El punto donde más problemas se detectaron es en la parte del cierre de sesiones. **Cerca de un 60% de los proyectos no implementaban una funcionalidad de cierre de sesión** y parte de las que lo implementaban, lo hacían de forma que no invalidaba los tokens, simplemente los eliminaba del lado de cliente. Además, gran parte de las sesiones se creaban con un tiempo de caducidad demasiado alto (cerca de un mes). De esta forma, un atacante con acceso a uno de estos tokens, podría ser capaz de realizar acciones en la plataforma en nombre del usuario, sin que existiera un mecanismo para invalidarlo y durante un periodo realmente largo de tiempo.

5.2.2.2. Control de accesos y gestión de información sensible

El control de accesos y la gestión de información sensible son elementos cruciales de la seguridad del lado del servidor. El control de accesos se refiere al conjunto de técnicas y políticas que se utilizan para restringir el acceso a recursos de la aplicación o datos a usuarios no autorizados. La gestión de información sensible se centra en la protección de datos confidenciales, como información personal del usuario o datos financieros. Juntos, el control de accesos y la gestión de información sensible, ayudan a garantizar la privacidad y seguridad de los datos y recursos del lado del servidor, lo que es esencial para prevenir vulnerabilidades y posibles amenazas a la seguridad de la aplicación.

El 39% de los proyectos
tenían algún tipo de fallo en el control de acceso

Un aspecto preocupante es la cantidad de proyectos afectados por vulnerabilidades en el control de accesos, lo que permitía en muchos casos exponer información sensible de otros usuarios. Algunos permitían obtener información de otros usuarios simplemente manipulando los parámetros de la petición, esto es, pidiendo los datos de otro usuario mediante, por ejemplo, su identificador o dirección de correo.

En otros casos ni siquiera se implementaban controles de acceso en llamadas que contenían información sensible. Por ejemplo, en uno de los proyectos implementaba una funcionalidad para obtener información de otros usuarios sin requerir credenciales. De esta forma, fue posible obtener datos sensibles de terceros, simplemente preguntando al servidor por la información de una dirección de correo.

Por último, había una serie de juegos en que las peticiones devolvían mucha más información de la necesaria. Un ejemplo podría ser una petición que devolvía información sobre los usuarios en un ranking, que también incluía datos como los correos, direcciones de blockchain e incluso direcciones IP o hashes de contraseñas.

En muchos proyectos se permitía la enumeración de usuarios, ya fuera por diferencia entre las peticiones de usuarios existentes y no existentes, o debido al uso de identificadores numéricos incrementales, que permitían conocer que previo al usuario 63, había otros 62 usuarios con esos identificadores. Hay que tener en cuenta que estos datos pueden servir de apoyo a otros ataques, como algunos de los fallos en el control de acceso mencionados anteriormente, pero también pueden suponer un riesgo para la privacidad, puesto que la mera pertenencia a uno de estos proyectos puede suponer problemas para los usuarios.

5.2.2.3. Validación de datos de entrada

La validación de datos del lado del servidor es un paso crítico en la seguridad de las aplicaciones web y móviles. Este proceso implica la verificación de los datos recibidos por la aplicación desde un cliente antes de procesarlos y almacenarlos en una base de datos. La validación del lado del servidor es importante porque evita que los datos malintencionados o erróneos se procesen y se almacenen en la base de datos, lo que puede provocar vulnerabilidades y posibles amenazas a la seguridad. Además, la validación del lado del servidor puede ayudar a proteger la integridad de los datos al garantizar que se cumplan los requisitos de formato, longitud y tipo de datos. Esto reduce el riesgo de errores de programación y explotación de vulnerabilidades de la aplicación, lo que puede ayudar a prevenir ataques como inyección de SQL y ataques de cross-site scripting (XSS). En resumen, la validación de datos del lado del servidor es fundamental para garantizar la seguridad y la integridad de los datos en aplicaciones web y móviles.

Cerca de un 65% de los proyectos analizados
tenían validaciones incorrectas o insuficientes

Gran parte de los proyectos analizados permitían insertar datos erróneos o potencialmente dañinos en campos como nombres de usuario, direcciones de correo o, incluso, direcciones de billetera. Esta información podría afectar a procesos internos de la plataforma de diversas maneras, incluido inyecciones de código o ejecución de comandos.

Además, se identificaron varios proyectos que hacían uso de los datos introducidos por los usuarios para realizar acciones críticas sin validar su veracidad. Por ejemplo, fue posible adquirir activos digitales (NFT) por un precio cercano a 0. Pese a que los autores nos indicaron que los procesos de validación se hacían manualmente de forma posterior, fue posible usar estos activos mientras se realizaba el proceso, influyendo en la mecánica del juego.

El caso más grave fue el de un juego donde, al finalizar una partida online, era posible manipular la llamada al servidor para indicarle qué jugador había ganado y en qué modo de juego. De esta forma, fue posible obtener tokens por valor de varios miles de euros en cuestión de minutos, simulando partidas en modo P2E.

5.2.2.4. Configuraciones y software

Un 90% de los juegos auditados
no tenían configuradas de manera correcta las cabeceras HTTP de seguridad básicas

Las configuraciones de seguridad del lado del servidor y las actualizaciones de software son fundamentales para mantener la seguridad y la integridad de los datos y aplicaciones alojadas en un servidor. Las configuraciones de seguridad adecuadas ayudan a proteger el servidor y su contenido de posibles amenazas externas, como ataques de hackers o malware. Las actualizaciones de software, por otro lado, son importantes para corregir errores, vulnerabilidades y brechas de seguridad que podrían ser explotadas por hackers. A medida que los desarrolladores descubren y corrigen estas vulnerabilidades, las actualizaciones garantizan que los servidores estén actualizados y sean menos propensos a ser atacados. En resumen, las configuraciones de seguridad del lado del servidor y las actualizaciones de software son medidas importantes para proteger la información y los recursos alojados en el servidor.

Prácticamente un 90% de los proyectos que usaban la web, no tenían configuradas correctamente (o de ninguna forma) **las cabeceras HTTP de seguridad básicas**, como CSP, gestión de caché o la preferencia de conexiones seguras, entre otras.

Además, **cerca de un 40% filtraban información** sobre las tecnologías en uso, principalmente debido a cabeceras o mensajes de error generados por el servidor. De esta forma, un atacante puede obtener información sobre tecnologías y versiones en uso que usar como soporte para otros ataques.

Así, fue posible contrastar información sobre sistemas operativos, servidores o servicios que soportaban la infraestructura de los diferentes proyectos. De ahí, se pudo comprobar como cerca de un 70% hacían uso de tecnologías desactualizadas de las que se conocían vulnerabilidades públicas, algunas de ellas tan graves que podrían poner en riesgo toda la seguridad de los servidores.

5.2.2.5. Comunicaciones y sistemas de protección

Casi un 85% de los proyectos revisados
no implementaba ningún sistema de protección en el servidor

Algo que debe tener una infraestructura si queremos que esté protegida ante todo el tráfico que llega desde fuera es una buena protección perimetral. Este tipo de seguridad por lo general se encarga de analizar todo el tráfico que entra o sale de nuestro servidor en busca de anomalías o llamadas maliciosas. Entre algunas de las protecciones que se pueden añadir a un servidor tenemos:

Firewall

Permite filtrar el tráfico entrante y saliente entre 2 redes, es decir, bloquea todos aquellos paquetes que entran o salen de la red y cumplen con dichos filtros

WAF

Web Application Firewall, viene a ser el equivalente de un cortafuegos pero para aplicaciones web, y en vez de filtrar paquetes, filtra llamadas http.

IDS

Intrusion Detection System. Lanza alertas de cualquier evento sospechoso, avisando así a los administradores del sistema.

Anti DDoS

Como bien dice el nombre, son sistemas de protección diseñados para mitigar ataques DDoS.

Muchos de estas protecciones se pueden implementar mediante uso de software, otras mediante hardware. Cloudfare es un buen ejemplo de empresa que proporciona algunas de las protecciones mencionadas anteriormente, así como otras. El problema aquí radica en que casi un 85% de los juegos auditados no contaban con ninguna de estas protecciones en los servidores que 'hosteaban' sus páginas web.

Cabe mencionar que, a la hora de realizar comunicaciones, todos los proyectos hacían uso de comunicaciones cifradas. Esto, principalmente, suele hacerse mediante el uso de https, que es algo con lo que todas las aplicaciones web contaban.

5.2.3. Cliente

5.2.3.1. Validación de datos de entrada y codificación de datos de salida

La validación de datos de entrada implica verificar que los datos ingresados por los usuarios sean válidos y seguros antes de procesarlos o almacenarlos en la base de datos. Esto ayuda a prevenir posibles errores o vulnerabilidades en el sistema. Por otro lado, la codificación de datos de salida implica transformar los datos almacenados en un formato que sea seguro y adecuado para su uso por parte de los usuarios. Esto ayuda a prevenir la divulgación de información confidencial o la manipulación no autorizada de los datos. En conjunto, la validación de datos de entrada y la codificación de datos de salida son prácticas esenciales para garantizar la seguridad y la integridad de los datos en cualquier aplicación o sistema informático.

Durante las diferentes pruebas se pudo averiguar que todas las aplicaciones hacían una excelente labor codificando los datos de salida, en gran parte debido al uso extendido de *frameworks*, como angular o react.

Por el contrario, aunque simplemente sirva como apoyo, no se realizaba una validación activa de los datos de entrada, tanto sintácticamente como semánticamente:.

Análisis sintáctico

Si se trata de una fecha, por ejemplo, que esta sea introducida de una manera específica (e.g dd/mm/yyyy). Si se introduce un mail, que este siga la estructura típica 'user@gmail.com', etc.

Análisis semántico

Mirar que la fecha de inicio sea menor a la de fin, que cierto valor este dentro de un rango, etc.

5.2.3.2. Actualizaciones de software

Múltiples proyectos incluían librerías de terceros que no estaban actualizadas a sus últimas versiones, evidenciando la falta de controles sobre este aspecto. Alguna de ellas contenía fallos conocidos, pero no pudieron ser explotados durante la auditoría

5.2.3.3. Gestión de datos sensibles

Es muy importante que, para toda la información que se guarda de los usuarios, se diferencien claramente aquellos datos que son sensibles, de los que no lo son, ya que todo aquello que se considere así, tendrá que ser guardado y gestionado de la manera más segura posible, con el fin de que no se pueda filtrar nada.

Al menos 3 de los 27 proyectos filtraban información sensible en los ficheros de log

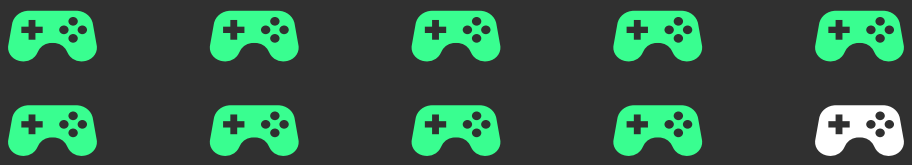
En unos pocos proyectos se ha identificado que las aplicaciones almacenaban información sensible sin cifrar en el directorio de datos de la aplicación o en la carpeta de instalación del programa. Estos directorios suelen ser accesibles para todos los usuarios y las aplicaciones y, por ello, cualquier aplicación maliciosa o un atacante con acceso al dispositivo o a los archivos de copia de seguridad podrá recuperar información como el correo electrónico o el *refresh* token que permite generar otros tokens de acceso.

Además, se descubrió que al menos 3 de los 27 proyectos filtraban información sensible en los ficheros de log, tanto de web, como de escritorio o móvil. La información filtrada contenía desde tokens de acceso a usuarios, contraseñas o direcciones tanto de billetera como móviles.

Pero, por lo general, todas las aplicaciones guardaban los datos correctamente cifrados en la memoria del proceso o en emplazamientos de seguridad.

5.2.3.4. Protecciones de seguridad

9 de cada 10 juegos que contaban con aplicación para móvil no comprobaban si se estaba ejecutando en un dispositivo rooteado



Varias de las aplicaciones de escritorio tenían componentes que no estaban compilados para usar las medidas de seguridad necesarias, lo que permitiría una más fácil explotación en caso de encontrar errores de memoria, como desbordamientos de buffer. Aunque en muchos casos sí que se implementaron otro tipo de medidas, como las de *anti-debug* u ofuscamiento, debido en muchos casos al uso de *frameworks* que automatizan estas funcionalidades.

Por otra parte, cerca de 95% de las aplicaciones móviles no comprobaban si el dispositivo donde se estaba corriendo la aplicación estaba rootead/jailbreak. En proyectos donde se gestiona capital, debería evitarse su uso o, al menos, informarse al usuario de que la seguridad puede estar degradada. De igual forma sucedería con la validación de certificados (*certificate pinning*). Aunque estas medidas pueden ser eventualmente saltadas, siempre se recomienda implementarlas para ofrecer una mejor protección por capas.

5.2.3.5. Web3 – Seguridad de las transacciones

Para finalizar con la parte de cliente, se ha añadido este punto en el que se van a mencionar algunos fallos que tienen una implicación directa con la blockchain, es decir, algunas vulnerabilidades que provienen del cliente, pero que afectan a su comunicación con la blockchain.

Implicaciones de la integración web3

Cuando hablamos de aplicaciones web, ciertos fallos de seguridad que en web2 no llegan a ser críticos, sí que lo son en aplicaciones web3, debido a que se tiene acceso a la blockchain, y esto puede llegar a tener un gran impacto. Los XSS cobran especial relevancia ya que hacerse con la ejecución de javascript permite, por ejemplo, falsear llamadas a *metamask* cambiando los valores de las llamadas o haciendo que apunten a contratos maliciosos. En este caso, el usuario firmaría una transacción no deseada que puede pasar fácilmente por una válida y enviar todos los tokens a un determinado contrato o dirección sin que lo sepa.

En uno de los juegos auditados se ha podido comprobar que en la tienda oficial de NFTs se podía inyectar en la URL la dirección de una colección de NFTs diferente a la oficial, haciendo que en la página web oficial se vendiese una colección diferente a la esperada. Dado que la página oficial del juego en su tienda de NFTs, usaba como servicio externo un Marketplace de terceros que gestionaba tanto el listado de los NFTs como la venta y pago de los mismos se ha podido crear una nueva colección clonando las imágenes y las características de la oficial dando lugar a que se pudiese listar esa colección falsa en la tienda oficial y que se pudiese vender desde la misma y sin que los usuarios notasen ninguna diferencia respeto a la oficial. Con esto, se pudo comprobar que es muy fácil hacer una campaña de phishing y hacer que los usuarios comprasen los NFTs clonados en la página oficial del juego y pagándole el precio de los NFTs comprados al atacante.

Validación resultado transacciones web3

Es muy común encontrar errores en las validaciones por ejemplo hacer una transacción de con un precio inferior, o la comprobación de que los cambios esperados en la blockchain se han producido ya que, si se manipula la llamada que se hace desde el *frontend*, puede haber una confirmación de la transacción, pero haciendo una llamada la diferente a la esperada.

También se ha comprobado que manipulando la respuesta de *metamask* en el cliente es posible simular la adquisición de NFTs que, aunque no se quede registrada en la blockchain si permite usarlos en el juego permitiendo generar otros tokens reales.

Seguridad del wallet

Actualmente, los juegos no suelen interaccionar directamente con la blockchain sobre todo si son aplicaciones móviles, aunque esta es una situación que tiende a cambiar ya que van apareciendo tecnologías que buscan la interacción de los *wallets* en las librerías de los juegos como es el caso de ChainSafe Gaming de Unity o WalletConnect Sign. Este es un componente crítico ya que, guarda las claves privadas y que puede contener fallos de seguridad. Otro aspecto a considerar es el nodo al que se conecta el *wallet* y los demás componentes para hacer las llamadas ya que, generalmente se usa uno proporcionado por un servicio externo, susceptible a censura o manipulación.

Remarcar que teniendo en cuenta el corto periodo de tiempo dedicado a las auditorías este apartado queda pendiente como futuro trabajo de investigación ya que este módulo requiere una especial atención en cuanto a la seguridad y sus implicaciones siendo este crítico.

Inicio de sesión en el juego

En algún juego de escritorio se ha comprobado que se iniciaba sesión a través de la sesión del cliente web el cual tenía una vulnerabilidad que permitía el robo de sesión permitiendo aprovecharla también en la aplicación del juego y jugar con la cuenta de otro usuario.

6. Desarrollo Seguro y Mejores Prácticas

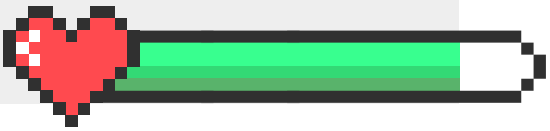
La seguridad es un pilar fundamental en la creación de cualquier producto digital, más aún cuando nos referimos a juegos DeFi, donde los activos y la información de los usuarios están en constante riesgo. El desarrollo seguro, en este contexto, hace referencia a la aplicación de una serie de buenas prácticas y principios de diseño que buscan minimizar la aparición de fallos de seguridad o errores en la implementación que podrían ser aprovechados por actores malintencionados.

En primer lugar, es importante recalcar que la seguridad debe comenzar en etapas tempranas del desarrollo y acompañar todo el ciclo de vida del mismo. Esto implica que los desarrolladores deben tener las herramientas que les permitan testear la seguridad del código desarrollado y que los equipos de DevSecOps cumplen un rol fundamental, integrando la seguridad y definiendo umbrales de riesgos en los pipelines durante las etapas de despliegue de estas aplicaciones. La detección de vulnerabilidades en etapas tempranas no solo ahorra tiempo y dinero a las empresas, sino que también ayuda al negocio a fortalecer la imagen y confianza del producto ofrecido por la empresa a sus clientes.

Entre las prácticas más comunes en el desarrollo seguro encontramos el uso de análisis estático de código (SAST), si bien este tipo de análisis son muy importantes, **en los desarrollos actuales el 85% utiliza librerías de código abierto**, por lo cual es casi imprescindible poder contar con una herramienta que nos permita realizar SAST y SCA. Ahora bien, ¿Qué es SCA? SCA (Software Composition Analysis) es un software que nos permite identificar componentes de código abierto utilizados en nuestras aplicaciones y evaluar el riesgo (vulnerabilidades, licencias, etc) de estas dependencias. Otra práctica que debemos tener presente es la realización de pruebas de penetración previas a la puesta en producción de nuestras aplicaciones, que permitan emular los pasos que realizará un actor malicioso una vez que nuestro producto esté disponible.

Muchas veces se valora más la parte gráfica o de marketing que la seguridad, priorizando un lanzamiento temprano del juego. Al dirigir la mayor parte de los recursos a estos aspectos en detrimento de la seguridad tal como ya ha ocurrido en muchas ocasiones, se consiguen que el proyecto tenga un gran éxito inicialmente, pero después del lanzamiento los actores malintencionados que están a la espera de proyectos vulnerables, explotan esas brechas haciendo que al final ese juego pierda tanto el capital como los usuarios y acabe fracasando.

Un punto muy importante es tener siempre presente el ciclo de vida, se suele poner mucho énfasis en cumplir fechas para su puesta en producción, pero luego solemos olvidarnos de las etapas que siguen en el ciclo, como es la etapa de mantenimiento. En esta etapa se deberán reservar recursos para poder mantener el producto, lanzar nuevas actualizaciones, aplicación de parches de seguridad, corrección de bugs, realizar pentest, escaneos de vulnerabilidades, etc.



Un aspecto crítico para la ciberseguridad es la adopción de una serie de principios clave que guían el diseño e implementación de nuestras soluciones. Entre ellos, destacan el principio de mínimo privilegio, según el cual cada componente del sistema solo debe tener los permisos mínimos necesarios para realizar su función; el principio de defensa en profundidad, que promueve la implementación de múltiples capas de seguridad para proteger contra un fallo en cualquier capa individual; y el principio de negación por defecto, que establece que se debe negar el acceso a cualquier recurso a menos que se conceda explícitamente.

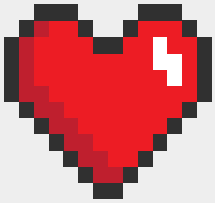
Principio de mínimo privilegio — **Principio de defensa en profundidad** — **Principio de negación por defecto**

Además de estas prácticas y principios, es vital adoptar una metodología de gestión de riesgos que permita identificar, evaluar y priorizar los riesgos de seguridad. Esto puede involucrar el uso de técnicas como la Evaluación de Riesgos y el Análisis de Impacto Empresarial, así como la creación de un plan de mitigación de riesgos y un plan de respuesta a incidentes. Una gestión de riesgos efectiva puede ayudar a anticipar problemas de seguridad antes de que ocurran y a responder de manera efectiva cuando se presenten.

El desarrollo seguro en los juegos DeFi requiere una combinación de buenas prácticas, principios sólidos de diseño, una metodología de gestión de riesgos y aplicar una mentalidad de seguridad desde el inicio. Al adoptar estas estrategias, los desarrolladores pueden contribuir significativamente a proteger los activos y la información de los usuarios, y a garantizar la sostenibilidad y el éxito a largo plazo de sus juegos.

7. Discusión y conclusiones

Economía de los Juegos y su impacto en la Seguridad



En el corazón de los juegos DeFi y Play-to-Earn (P2E) se encuentra una economía de juego inherentemente digital y dinámica que vincula directamente las acciones y logros de los jugadores con las recompensas monetarias tangibles. Sin embargo, equilibrar estas economías de juego es un desafío que a menudo se pasa por alto, pero tiene implicaciones directas en la seguridad y la longevidad de un juego.

En la mayoría de los juegos P2E, se implementa algún tipo de economía de tokens. Los jugadores ganan tokens a través de sus acciones en el juego, y estos tokens pueden ser intercambiados por bienes, servicios o incluso retirados como criptomoneda real. La clave aquí es el equilibrio. Si las recompensas son demasiado generosas en relación con el flujo de liquidez que alimenta el juego, puede resultar en un agotamiento de la liquidez y acortar la vida del juego. Por otro lado, si las recompensas son demasiado escasas, puede disuadir a los jugadores de participar.

Este equilibrio no es solo importante para la economía del juego, sino también para la seguridad de los usuarios y sus activos. Si una economía de juego está mal equilibrada y resulta en una rápida disminución de la liquidez, puede conducir a una caída precipitada en el valor de los tokens del juego. Esto puede poner en riesgo los activos de los usuarios, ya que su valor puede disminuir rápidamente y sin previo aviso. Esto es particularmente problemático en el caso de los juegos P2E, donde los jugadores a menudo invierten su propio dinero en el juego para comprar equipos, mejorar los personajes y participar en misiones y desafíos.

La estrategia económica de un juego también debe considerar el valor a largo plazo de los activos de los jugadores. Los desarrolladores de juegos deben asegurarse de que los jugadores sientan que sus inversiones y esfuerzos en el juego están protegidos y tienen un valor duradero. Si los jugadores sienten que sus activos podrían perder valor repentinamente debido a cambios en la economía del juego, es probable que estén menos dispuestos a invertir tiempo y dinero en el juego. Esta es una consideración de seguridad vital, ya que puede impactar la confianza de los jugadores en el juego y su disposición para participar.

En este sentido, la seguridad de los juegos DeFi no es solo una cuestión de proteger los datos de los usuarios y prevenir el fraude y la manipulación. También se trata de proteger la economía del juego y, por lo tanto, el valor de los activos de los jugadores. Los desarrolladores de juegos necesitan adoptar enfoques sólidos y bien pensados para el diseño de la economía del juego, y asegurarse de que estén equilibrados de manera que protejan la seguridad y la confianza de los jugadores.

El diseño y mantenimiento de una economía de juego segura y equilibrada es una tarea desafiante y en constante evolución. Requiere una comprensión profunda de las dinámicas económicas, así como de la psicología del jugador y el paisaje cambiante de la tecnología blockchain. Los juegos que logren este equilibrio prosperarán en el creciente ecosistema de juegos DeFi, mientras que aquellos que no lo hagan, corren el riesgo de perder la confianza de los jugadores y de desaparecer.

El papel de la seguridad en el diseño y la gestión de la economía de los juegos DeFi es cada vez más importante y complejo. Se necesita una combinación de sólidos mecanismos de seguridad, auditorías regulares y transparencia para garantizar que la economía de un juego se maneje de manera justa y segura. Los juegos, que priorizan la seguridad en su economía, serán más capaces de mantener a los jugadores comprometidos, proteger sus inversiones y garantizar la longevidad del juego.

Además, los juegos DeFi deben estar preparados para responder de manera flexible y rápida a los cambios en el mercado y en el comportamiento de los jugadores. Esto puede implicar ajustar las recompensas, modificar los costos y tomar otras medidas para mantener la economía del juego equilibrada. Sin embargo, estos ajustes deben hacerse con cuidado y considerando las posibles implicaciones de seguridad. Un cambio demasiado brusco podría desestabilizar la economía del juego, dañar la confianza de los jugadores y poner en riesgo sus activos.

La economía de los juegos es una faceta crítica de la seguridad en los juegos DeFi. Equilibrar las recompensas y mantener la liquidez es crucial para proteger los activos de los jugadores y garantizar la longevidad del juego. Los desarrolladores necesitan prestar una atención cuidadosa al diseño y la gestión de la economía del juego, y hacer de la seguridad una prioridad en todas las etapas del desarrollo y operación del juego.

Factores Regulatorios

Los juegos DeFi y Play to Earn, como cualquier actividad que involucre el uso de criptomonedas y blockchain, están sujetos a una variedad de factores regulatorios. Estos factores pueden variar ampliamente dependiendo del país y de la naturaleza específica del juego.

En muchos países, las criptomonedas son legales y sus actividades se encuentran bajo el amparo de un marco regulatorio específico, lo que significa que tanto jugadores como desarrolladores deben cumplir con una serie de requisitos establecidos en las leyes desarrolladas para tal fin. Esto puede incluir la necesidad de identificar y verificar la identidad de los jugadores (conocido como KYC, o Conozca a su Cliente), la necesidad de reportar ciertas transacciones a las autoridades fiscales, y la necesidad de cumplir con las leyes de blanqueo de dinero.

Además, algunos países pueden tener regulaciones específicas relacionadas con los juegos de azar. Si un juego DeFi o P2E es considerado como un juego de azar, entonces puede estar sujeto a leyes y regulaciones adicionales.

Por otro lado, algunos países pueden tener restricciones o prohibiciones en el uso de criptomonedas, lo que puede limitar la capacidad de los jugadores y desarrolladores para participar en juegos DeFi.

Es importante recalcar que aún hay muchos países que no tiene una regulación clara respeto a la industria de las criptomonedas, por tanto, en caso de que un ataque afecte a los fondos de los inversores, la responsabilidad de la empresa de cara a los usuarios acaba siendo voluntaria, y las medidas legales que se puedan tomar a posteriori dependen de la legislación de cada país y de cómo interpreta este tipo inversiones y que hace que, por lo general, sea muy difícil recuperar los fondos perdidos.

Además, se ha visto que, en algunos juegos, en el apartado de términos de uso, ya se desentienden de esta responsabilidad, añadiendo mensajes como el siguiente, sacado de la página de términos de uso de Axie Inifinity:

> ***"Existen riesgos asociados con el uso de una moneda basada en Internet, incluyendo, pero no limitado a, el riesgo de hardware, software y conexiones a Internet, el riesgo de introducción de software malicioso, y el riesgo de que terceros puedan obtener acceso no autorizado a la información almacenada en su monedero. El usuario acepta y reconoce que Axie Infinity no será responsable de los fallos de comunicación, interrupciones, errores, distorsiones o retrasos que pueda experimentar al utilizar la red Ronin, sea cual sea su causa."***

Implicaciones Éticas

Los juegos DeFi, al igual que cualquier actividad que implique transacciones económicas y relaciones entre personas, tienen una serie de implicaciones éticas. Estas implicaciones pueden abarcar una variedad de temas, incluyendo la equidad, la transparencia, y el impacto de estos juegos en las sociedades y economías.

Un aspecto clave de la ética en los juegos DeFi es la equidad. Este concepto se refiere a cómo los beneficios y los riesgos de estos juegos se distribuyen entre los jugadores. ¿Tienen todos los jugadores las mismas oportunidades para ganar y perder? ¿O hay algunas personas que tienen una ventaja injusta debido a su posición, conocimiento o acceso a recursos? Este es un tema especialmente crítico en los juegos que incluyen elementos de azar, donde la falta de equidad puede llevar a situaciones en las que algunos jugadores son explotados económicamente.

La transparencia es otro aspecto ético importante en los juegos DeFi. Los jugadores tienen el derecho de saber cómo funciona el juego, cómo se determinan las recompensas y los castigos, y cómo se gestiona y utiliza su información personal. La falta de transparencia puede llevar a la desconfianza y la insatisfacción entre los jugadores, y puede dar lugar a prácticas injustas o incluso fraudulentas.

Otra característica importante relacionada con la ética de estos juegos es referente a las medidas que se toman en caso de un ataque. Cuando esto ocurre, suele perjudicar tanto a la empresa como a los usuarios, ya que, si afecta a la empresa, repercute también en el precio de los activos que los jugadores poseen, lo que encadenaría una disminución de usuarios y, por lo tanto, una pérdida de capital. Lo que sí se ha observado es que, las empresas que implementan algunas opciones para tomar medidas en caso de algún ataque, priorizan la seguridad de los fondos de la empresa sobre los fondos de los usuarios. Esto lo hacen muchas veces gestionando los fondos de manera centralizada, haciendo que al usuario no tenga una manera segura de gestionar su capital.

Finalmente, los juegos DeFi pueden tener un impacto significativo en las sociedades y economías. Por un lado, pueden proporcionar nuevas oportunidades económicas y formas de entretenimiento para las personas. Por otro lado, pueden contribuir a problemas como la adicción al juego, la desigualdad económica y la evasión fiscal. Estos impactos deben ser cuidadosamente considerados y gestionados por los desarrolladores y reguladores de estos juegos.

Seguros para los fondos

Algo que suelen hacer grandes empresas dentro de la industria de las criptomonedas es tener fondos de garantía para poder cubrir posibles pérdidas causadas por ciberataques. En este caso, al tratarse muchas veces de desarrolladoras pequeñas, ninguna cuenta con este tipo de fondos. Aun así, sí que es habitual que este tipo de proyectos, cuenten con fondos dedicados a Bug Bounty, con el fin de mejorar la seguridad del mismo. Esto no es algo esencial, siempre y cuando estos proyectos sean lo suficientemente transparentes y estén correctamente auditados. Al final, el reclamo de este tipo de juegos es la descentralización y transparencia, donde cada usuario sea jugador o sea inversor pueda comprobar y analizar por sí mismo la seguridad y todas las características económicas del proyecto.

Tamaño de la muestra

Por último, añadir que toda esta información proporcionada en el artículo ha sido fruto del trabajo de unas dos semanas. Es decir, que todavía quedaría mucho en lo que profundizar ya que se ha tenido que acotar el estudio a un número reducido de proyectos y dedicar una parte de este tiempo a cada uno de los tres componentes, mencionados anteriormente, para tener una visión general de la seguridad de cada uno de los títulos. Como consecuencia, quedarían muchos fallos y detalles que objetar, pero que ha valido para dar a entender el panorama actual de este tipo de juegos dentro de la industria de las criptomonedas.

8. Futuro

La incursión de la tecnología blockchain en la industria del juego a través de los juegos DeFi está remodelando la forma en que los jugadores interactúan con los videojuegos. Los juegos DeFi ofrecen a los jugadores la oportunidad de ganar criptomonedas reales, propiedad de objetos en el juego, y la capacidad de interactuar y comerciar en un mercado de jugadores en un entorno descentralizado.

Sin embargo, con estas nuevas oportunidades también vienen nuevos desafíos y consideraciones. Los aspectos de desarrollo seguro, los factores regulatorios, las implicaciones éticas, la ingeniería social y el phishing, la justicia y equidad del juego, y la psicología del jugador, todos juegan un papel crítico en la salud a largo plazo de esta nueva industria.

Es imperativo que los desarrolladores de juegos estén a la altura de estos desafíos y sigan trabajando para construir juegos que sean seguros, justos y satisfactorios para los jugadores, y que cumplan con las leyes y regulaciones correspondientes. Del mismo modo, los jugadores necesitan estar informados y ser conscientes de los riesgos y las recompensas asociados con los juegos DeFi, y tomar decisiones informadas sobre su participación en estos juegos.

9. Referencias

[1] https://coinmarketcap.com/currencies/axie-infinity/

Audit Findings 101 - by Rajeev | Secureum - Secureum (substack.com)

Audit Findings 201 - by Rajeev | Secureum - Secureum (substack.com)

Overview · Smart Contract Weakness Classification and Test Cases (swcregistry.io)

GitHub - harendra-shakya/smart-contract-attack-vectors: A curated list of smart contract attack vectors

En la elaboración del informe han participado:

Pablo Vilallave Villach: Auditor de Seguridad Blockchain en Sofistic
Iulian Ghigheci: Auditor de Seguridad Blockchain en Sofistic
Manuel Ginés: Director Proyectos de Investigación en Sofistic

Washington Gómez: CISO en Bit2me
Jorge García: IT Security, Risk and Compliance en Bit2me

GAME OVER

SOFISTIC
CYBERSECURITY

480

En colaboración con:

bit 2 me