Project config.Play a Turn-based Strategy Security Board Game

Hugo Enriquez, Youki Kadobayashi and Doudou Fall

Nara Institute of Science and Technology (奈良先端科学技術大学院大学), Ikoma, Japan

ugohenriquez25@gmail.com
youki-k@is.naist.jp
doudou-f@is.naist.jp

Abstract: Games can be effectively used in an educational setting to improve motivation and learning. *Project config.Play* was designed using cybersecurity terminology and concepts to build upon game mechanics, the result is a two players strategy turn-based board game in which players must attack each other's vulnerabilities while protecting their own territory. The goal of the project is to introduce new people to internet security and computer science in general, the game was designed so all players can enjoy it without the need of previous computer science knowledge. Players takes the roll of both, attacker and defender, and must come up with a strategy using the available cards while taking cautious action with their movement after throwing the dices. The game simulates a basic cybersecurity scenario involving two systems with authentication requirements, configuration conditions, race conditions and conceptualizes the CVSS (Common Vulnerability Scoring System). The Common Vulnerability Scoring System (CVSS) provides a way to evaluate a system's security by assigning numerical scores to principal characteristics. The CVSS is a standardized public tool which is currently being worked on by the CVSS Special Interest Group (SIG) to make improvements for the next version. We designed a game based on the CVSS v2.0 utilizing the core concepts to create a simulated scenario to teach the CVSS functionality in action. The educational goal is to familiarize students with the concepts of Common Vulnerabilities and Exposures (CVE) and CVSS in a simulated scenario, although the game was designed with computer science students in mind, it's use isn't limited to students and can be played by anyone. Moreover, another project goal is to introduce students to computer science and make them aware of cybersecurity. An important project goal is to reach a wide audience, thus the game can be easily printed and built with card board, yet the game has not been publicly distributed.

Keywords: gamification, game design, board game, educational, cybersecurity, internet security

1. Introduction:

The use of novel approaches in education, specially game-related work, is gaining popularity and is being incorporated in creative ways with the objective of improving engagement and motivation (Enriquez, Lemus, Ajin, 2015). Gamification involves the application of game mechanics to non-game contexts, it has been widely used in higher education to cover and asses the academic content, it has been proved to provide a positive student engagement (Schreuders, Butterfield, 2016).

It is known that majority of users lack basic knowledge and awareness in internet security which leads to rushed and risky decisions, different approaches has been used to increase user awareness and promote safer security practices (Olano et al, 2014). To achieve this, Computer games like *SecurityEmpire* and *CyberCIEGE* (Michael et al, 2014) integrate security concepts into actual gameplay to provide continuous feedback allowing the players to understand the consequences to security events. Taking example of card games and board games like *Control-Alt-Hack* (Denning et al, 2003) we designed a strategy game to introduce players to cybersecurity in a fun and entertaining way relying on the game experience itself.

The Common Vulnerability Scoring System (CVSS) provides a way to evaluate a system's security by assigning numerical scores to principal characteristics (FIRST — Forum of Incident Response and Security Teams, 2018). The CVSS is a standardized public tool which is currently being worked on by the CVSS Special Interest Group (SIG) to make improvements for the next version. We designed a game based on the CVSS v2.0 utilizing the core concepts to create a simulated scenario to teach the CVSS functionality in action.

Project Config.Play is a two-players strategy board game in which players take turns to advance through the board with the goal of compromising the other player's system. The game utilizes cards similar to those in trading card games and *Control-Alt-Hack* (Denning et al, 2003). Players takes the roll of both, attacker and defender, and must come up with a strategy using the available cards while taking cautious action with their movement after throwing the dices.

Our game simulates a basic cybersecurity scenario involving two systems with authentication requirements, configuration conditions, race conditions and conceptualizes the CVSS (FIRST — Forum of Incident Response and Security Teams, 2018). The educational goal is to familiarize students with the concepts of CVE and CVSS in a simulated scenario, although the game was designed with computer science students in mind, it's use isn't limited to students and can be played by anyone. Moreover, another project goal is to introduce students to computer science and make them aware of cybersecurity.

The game has not yet been publicly distributed since it hasn't been fully tested yet. An important project goal is to reach a wide audience, thus the game can be easily printed and built with card board. A detailed explanation of the design process and implementation is provided in further sections, how game mechanics were mapped to cybersecurity concepts is explained in the next section, the game flow and general rules are discussed as well.

2. Related Work:

Designing games based on cybersecurity is widely discussed in modern literature. Creating a simulated scenario is an effective way to express security policies and use it as a tool for training and awareness as described in (Michael et al, 2014). In the construction and management resource simulation game*, CyberCIEGE*, Thompson and Irvine describe the various elements of scenario construction and methods for interaction with students. They discuss the process for scenario construction, suggesting that the audience and security policies to be covered must be identified to include circumstances that illustrate policy enforcement implications and consequences.

Denning et al (2003) designed and evaluated the effectiveness of a recreational table top card game created to raise computer security awareness. For game mechanics, they chose to license a system from a pre-existing game, an approach that let them focus on play-testing and game balance. They targeted people with an affinity for computer science and undergraduate students in the engineering disciplines, reportedly 14 classroom educators who used the game indicated that their student's awareness of computer security was increased.

Olano et al (2014) base their work on evidence that students are more willing to learn when engaged in hand-on experiences they can interact with. They designed *Security Empire*, a multiplayer videogame that teaches Information Assurance (IA) concepts to address the lack of user awareness of basic information. They conducted evaluation sessions consisting of two gameplay periods of fifteen minute each, then players were asked to complete a gameplay experience survey.

3. The Game Foundation:

Deciding the teaching material is needed to start designing the game, mapping the educational elements with desired game mechanics implies that creating new game mechanics based on the teaching material needs the theoretical or practical foundation of the studying material.

Designing a proper scenario helps players associate the game mechanics with real life applications (Koch, Schneider, Nordholz, 2012). With *config.Play*, the goal is to build a scenario that reflects the use of the CVSS.

3.1 The CVE (Common Vulnerabilities and Exposures):

The CVE contains a description of known vulnerabilities and it serves as a reference and baseline for evaluating the security tool of a system.

3.2 The CVSS (Common Vulnerability Scoring System):

The CVSS provides specific metrics to evaluate the security of a system and its goal is to provide a universal measure for a system’s security. The board game *config.Play* is based on the CVSS and all game mechanics are built on this system.

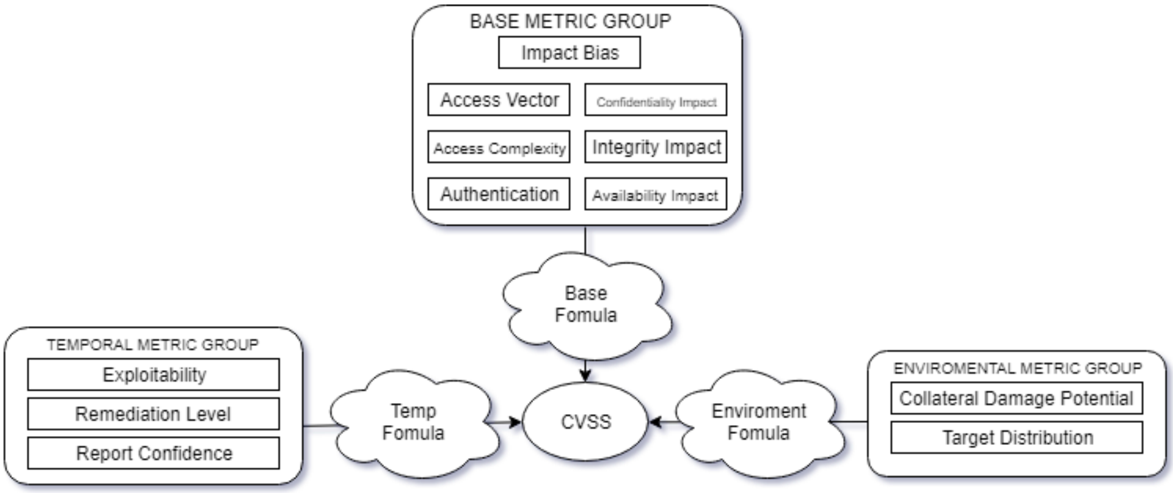

Figure 1: Diagram of the CVE functionality and interoperability.

3.3 The CVSS Base Metric Group:

This group includes the vulnerability characteristics that don't change through time, how it can be accessed and whether it requires special conditions to be exploited. The impact level determines to what degree the vulnerability is affected (FIRST — Forum of Incident Response and Security Teams, 2018).

3.3.1 (CVSS) Base GROUP: 1. Access Vector:

This metric indicates if a vulnerability must be exploited locally or remotely, possible values are:

- Local
- Remote

3.3.2 (CVSS) Base GROUP: 2. Access Complexity:

A metric that measures the complexity of the attack required to exploit the vulnerability once the system have been accessed, a low value is always exploitable while high values could require special conditions.

- Race Condition: specific window of time to exploit the vulnerability.
- Non-default configuration: specific circumstances are required to exploit the vulnerability.
- Victim interaction could be necessary in some cases.

3.3.3 (CVSS) Base GROUP: 3. Authentication:

This metric indicates whether the attacker needs to be authenticated to the target system, possible values are:

- Required
- Not required

3.3.4 (CVSS) Base GROUP: 4. Confidentiality impact:

The impact on confidentiality when the attack succeeds, levels are:

- None: No impact.
- Partial: There is a considerable amount of information disclosure.
- Complete: Total compromise of critical system information.

3.3.5 (CVSS) Base GROUP: 5. Integrity impact:

The impact on integrity when the attack succeeds, levels are:

- None: No impact.

- Partial: Considerable breach in integrity.
- Complete: Total compromise of system integrity.

3.3.6 (CVSS) Base GROUP: 6. Availability Impact:

The impact on availability when the attack succeeds, levels are:

- None: No impact.
- Partial: Considerable lag or interruptions in resource availability.
- Complete: Total shutdown of the affected resource.

3.3.7 (CVSS) Base GROUP: 7. Impact Bias:

This metric compares the weight of an impact to the system over the other two impact types to determine the risk value.

3.4 The CVSS Temporal Metric Group:

This group includes the vulnerability characteristics that change through time, it describes the time dependent qualities of the vulnerability.

3.4.1 (CVSS) Temporal GROUP: 1. Exploitability:

The metric that indicates the state of known techniques available to public to exploit a vulnerability, levels are:

- Unproven: No exploit code exists.
- Proof of concept: Proof of concept exploit code available.
- Functional: Functional exploit code available.
- High: Exploitable by autonomous code or no exploit required at all.

3.4.2 CVSS) Temporal GROUP: 2. Remediation level:

This metric measures the level of solution available in case of an attack, possible levels are:

- Official Fix: Complete solution exists.
- Temporary Fix: Only temporary fix available.
- Workaround: Unofficial solution available.
- Unavailable: No solution at all.

3.4.3 (CVSS) Temporal GROUP: 3. Report Confidence:

Degree of confidence in the existence of the vulnerability, possible levels are:

- Unconfirmed: There may be conflicting reports but still not confirmed.
- Uncorroborated: Multiple non-official sources.
- Confirmed: Officially confirmed by the company itself.

3.5 The CVSS Environmental Metric Group:

This group measures the potential risk of a vulnerability according to the current IT environment being used.

3.5.1 (CVSS) Environmental GROUP: 1. Collateral Damage Potential:

This metric measures the weight of the damage caused to equipment, revenue or even life loss, values are:

- None: No potential for damage.
- Low: Light property damage.
- Medium: Significant property damage.
- High: Catastrophic property damage.

3.5.2 (CVSS) Environmental GROUP: 2. Target Distribution:

This metric measures the scope of the exploit in terms of how many systems can be affected. Values are:

- None: No target systems.
- Low: 1%-15% of targets exist.
- Medium: 16%-49% of targets exist.
- High: 50% - 100% of targets exist.

4. What is config.Play?

4.1 A Board Game:

A two players board game that utilizes cybersecurity as its base for game mechanics and assign players with “attacker” and “defender” roles in order to experience a cybersecurity related scenario. The most basic game rules can be described as follows:

- A turn-based board game.
- Utilizes tokens and/or cards as player interface.
- No player has advantage over each other.
- First player to completely deplete the other player’s health wins.

4.2 Project Goals:

As an educational game, *config.Play* has several objectives that were kept in mind when designing the game mechanics and rules. The game has three main goals as explained below.

4.2.1 For Future Learning:

- Prepare computer science students to learn more about common vulnerabilities, attacks, exploits, the CVSS and CVEs.
- Make the study of cybersecurity more appealing and engaging to students.
- Provide students the opportunity to experience a cybersecurity scenario through gameplay.

4.2.2 For Introduction to Cybersecurity:

- Make a good first impression about cybersecurity.
- Make players interested in computer science and cybersecurity, provide the initial push towards cybersecurity.

4.2.3 For Entertainment:

- Create a board game that casual players can enjoy together without need of previous computer science or cybersecurity knowledge.
- Make the game as accessible as possible to anyone, as fun as any other popular board game.
- Avoid making gameplay sessions a chore.

4.3 Target Audience:

The game was designed to introduce players to cybersecurity in mind, therefore all game mechanics were built sustaining this concept. However, the game is not limited to people with previous knowledge in the computer science field e.g. students and people interested in computer science. *config.Play* can be played by anyone without previous knowledge of cybersecurity or computer science concepts, the objective is to introduce them to these new basic concepts through gameplay and hopefully make them interested. Either way, *config.Play* is first and foremost a game and it aims to be an enjoyable experience for everyone.

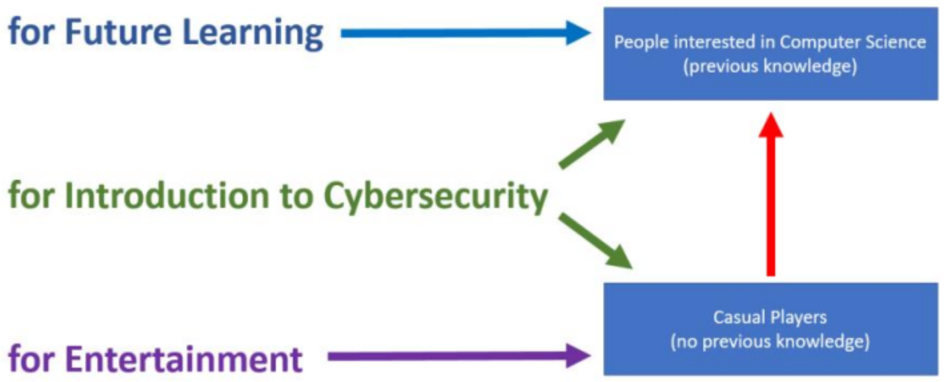
for Future Learning
for Introduction to Cybersecurity
for Entertainment
People interested in Computer Science (previous knowledge)
Casual Players (no previous knowledge)

Figure 2: The target audience.

5. Game Goals and Mechanics:

5.1 Game Goals:

The objective of the game is to come up with the best possible strategy to configure security settings and try to reach the other player's vulnerabilities to successfully attack (exploit vulnerabilities). When all three vulnerabilities are attacked the opponent player's health will be completely depleted.

5.2 Game Mechanics:

The game board is a grid of 8 x 16 divided in two parts, from the player one perspective the green part represents the local territory while the red part is the remote or enemy territory. The middle line is used to separate each player's territory.

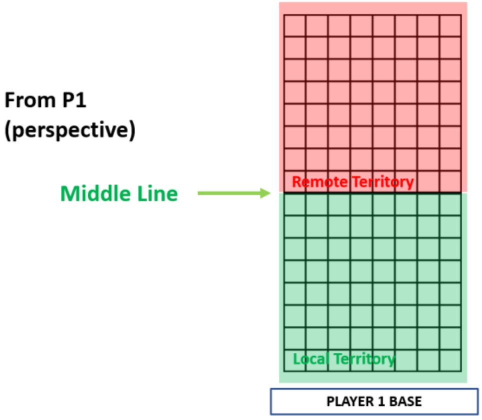

Figure 3: View of the game board divided by territory.

The game interface consists of three basic elements, the configuration module, the authentication indicator and the health gauges. The interface is used to keep track of the current state of the game and indicate the changes on each player's system.

Each player has a configuration module that represents the actual configuration of the system. The state of the configuration switches might be changed depending on the vulnerability settings, some vulnerabilities require a special condition to be exploited which means that the configuration switches must match the vulnerability description. The default configuration for each system is shown in FIGURE 4.

The authentication indicator panel shows whether a player is logged in into the target system (enemy system) or not. Some vulnerabilities require the attacker to be logged in to be exploited. Each player is logged out from the target system at the beginning of the game. Health is represented by three gauges, the gauges are composed of confidentiality points, integrity points and availability points. The game is over when a player loses all health points. The health gauges are full at the beginning of the game.

A vulnerability is any weakness that could be exploited to violate a system or the information it contains. Players must attack each other's vulnerabilities to deplete the health gauges. Each player must configure a total of three vulnerabilities, A, B and C. Figure 4 represents a vulnerability completely configured, in this case the vulnerability name is A. For each vulnerability, a total of three points can be used to adjust configuration, each attribute consumes the specified amount of points (pt). Default values are set for all attributes unless the player uses points to change the configuration.

A vulnerability configuration is basically structured as follows (default values are in parenthesis):

- Access Vector [1 pt]: This indicates whether the vulnerability must be attacked locally or remotely. A local vulnerability must be attacked by getting to the vulnerability's spot, a remote vulnerability must be attacked using a specific attack card.
  (Default value: remotely)

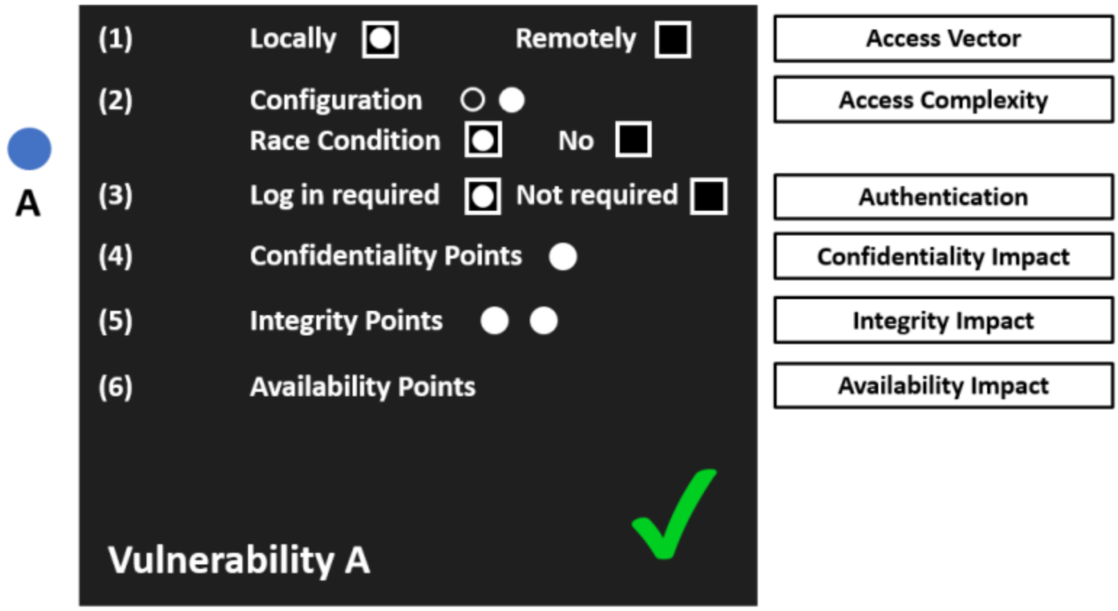

Figure 4: Fully configured system vulnerability representing the CVSS base metric group.

- Access Complexity - Configuration [2 pts]: A vulnerability can be set to require a specific configuration of switches to be exploited, the player cannot attack the vulnerability until the switch configuration matches the vulnerability's configuration.
  (Default value: switch 1 off, switch 2 off)

- Access Complexity – Race Condition [1 pt]: A race condition can be set in a vulnerability, when active, the attacker has five turns after crossing the middle line of the game board to attack the vulnerability. After the race condition ends the vulnerability can no longer be attacked and the player must exit the enemy territory and try the attack again.
  (Default value: no)

- Authentication [1 pt]: If activated, the attacker must log in into the target system before attempting an attack.
  (Default value: not required)

- Confidentiality – Availability – Integrity Tokens: These values represent the damage inflicted to the specific health gauge when attacked. One confidentiality token depletes the confidentiality heath gauge by a third, the same goes for availability and integrity.

A total of nine tokens must be distributed among all three vulnerabilities, point distribution is up to each player and the total number of tokens must be used when configuring vulnerabilities.

The game has a time limit of one hour, after this period the player with the highest amount of remaining health wins. After configuring all three vulnerabilities, the players must place the location of these vulnerabilities on any spot in the first row of the game board within their respective territory. The configuration of each vulnerability is written on a piece of paper which must be folded and kept hidden from the other player, a vulnerability will be eventually revealed to the other player after using an expose card.

6. Action Cards:

Action cards allow the player to perform special tasks to progress through the game, cards provide both defensive and offensive tools. Cards are in decks separated by colours, when players roll the colour dice they're allowed to take a card from the deck of the same colour, players must take the card at the top of the deck. Each card has the same structure including the name, the card type which determines the colour, an image and a detailed description. Some cards have a blue image which indicates it is a special card, specials cards may be available in other card colours.

Players can only keep five cards in their hand, they will be required to discard one card when drawing a new one and the hand is full, discarded or used cards must be returned to the bottom of the deck. As a rule, players must keep their cards hidden from each other during the game. When a vulnerability is successfully attacked the player who made the attack must return all the actions cards in hand and return to the first row of the local territory on the game board.

Cards are organized into four different types; each type is represented by a colour and allows the player to perform similar actions. When returning cards to the deck, they must match the deck colour.

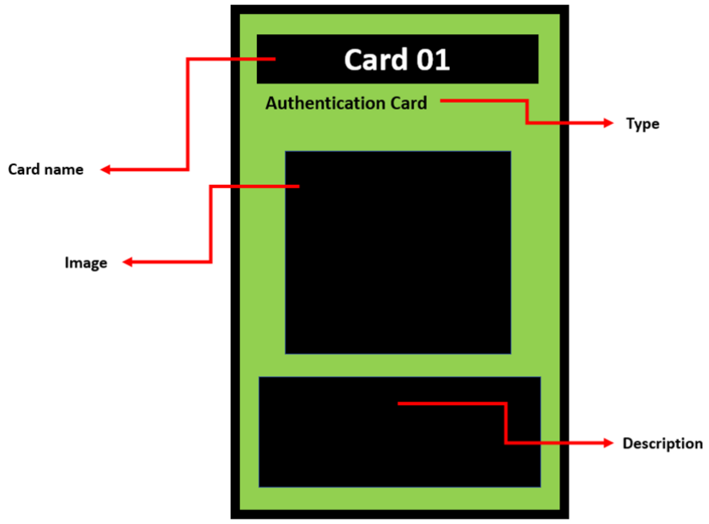

Figure 5: Design and parts of an action card.

- Green Cards: Provide authentication tools.
- Yellow Cards: Allow the player to change the configuration switches.
- Purple Cards: Can change race condition settings.
- Black Cards: Can change the game flow and turn order.

7. Game Flow and Movement:

A number dice and a colour dice are used to determine movement for each turn. Both dices must be rolled each turn, the number dice indicates the number of spaces the player can advance, the colour dice indicates the colour of the card the player must draw from the card deck. Players must advance the exact number of spaces specified by the number dice, moving backwards and sideways is allowed, however diagonal movement is not allowed in any direction.

7.1 Trace territory:

This mechanic allows to set a defence structure in the local territory, each player must set their own blockades to defend against the other player's attack. A total of 24 blockades can be set, each blocked occupies one spot on the game board and it must not block the access to the vulnerabilities. Blockades can only be placed on the local territory and the attacker can't move across blocked spaces, only the owner of the blockade can move freely across the blocked spaces.

Treasure chest must be places as well, a total of four treasures chests are required. The only player who can open the treasure chests is the attacker, each player must place treasure chests in their local territory alongside the blockades. Treasure chests contain access to a free card which can be of any colour.

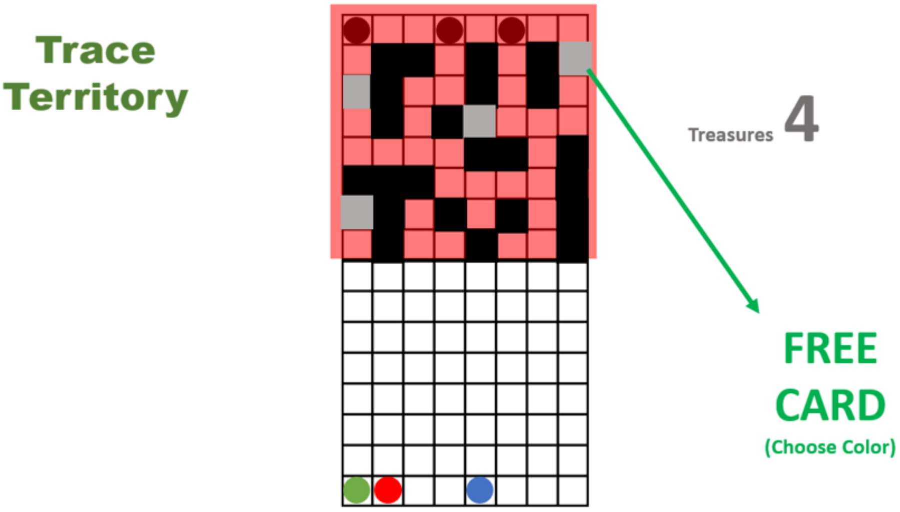

Figure 6: Diagram of the game board after tracing the territory with blockades and treasure chests.

7.2 Game Flow:

When starting a new game, players must first configure all three vulnerabilities and place them on the game board then trace their territory with blockades and treasure chests. The next step is to set the default values of the game interface which includes the configuration panel and authentication indicator, finally the turn order can be decided using the dices.

A turn is composed of three main actions, first is to roll the dices, after getting the number of spaces and the colour of the card the player can draw a card of the specified colour, the final action is to move the exact number of spaces. Only one action card can be used per turn and it can be used either before or after rolling the dices but only before drawing a new card from the deck.

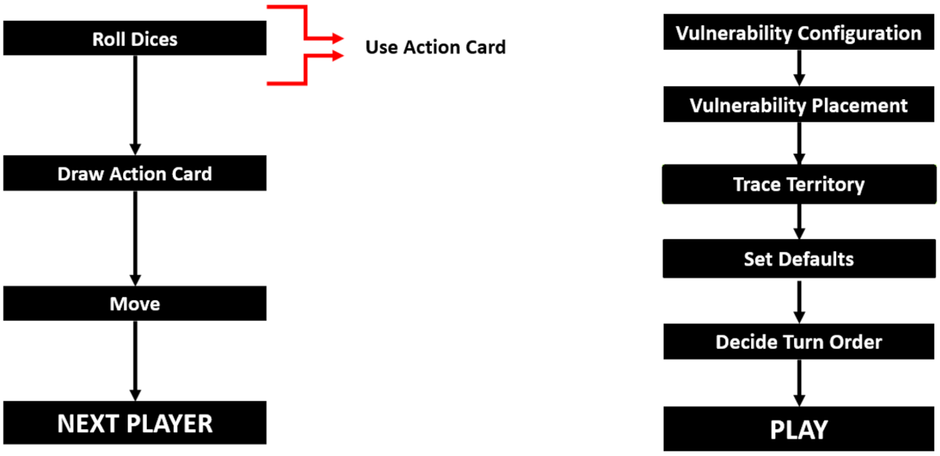

Figure 7: Game flow chart.

8. Conclusions and Future Work:

config.Play could be distributed online, it's easy to build and can be constructed with low cost materials. Whether it is an educational software or a board game, the game design process is the same, a specific subject need to be chosen to make it possible to start mapping game mechanics with teaching material. Several play testing sessions took place during development, similar to the debugging process of software development game mechanics were tested and modified when needed.

Possible future work includes the use of CVSS ve.3 and adjust game mechanics to match the ve.3 changes, that would guarantee updated teaching material. The other big step would be to start development of a software version of the game, the game design process has already been completed making software planning and development possible. The board game could benefit greatly from motivation theory and game design techniques to improve engagement, similar to the approach taken with *Cerebrex Ultimate* (Enriquez, Lemus, Ajin, 2015), a serious educational entertainment software.

References:

Denning, T., Lerner, A., Shostack, A., & Kohno, T. (2013). ControlAlt-Hack: The Design and Evaluation of a Card Game for Computer Security Awareness and Education. Seattle, WA, USA.

Enriquez, H., Lemus, A., & Ajin, B. Galileo University (2015). Testing Engagement Improvement in the Serious Educational Videogame: Cerebrex Ultimate GLS11. Madison, WI, USA.

FIRST Forum of Incident Response and Security Teams. (2017). Common Vulnerability Scoring System SIG. [online] Available at: https://www.first.org/cvss/ [Accessed 15 Nov. 2017].

Hendrix, M., AlSherbaz, A. & Victoria, B. (2016). Game based cyber security training: are serious games suitable for cyber security training? International Journal of Serious Games. 3(1), pp. 5361. 23848766.

Koch, S., Schneider, J., & Nordholz, J. Technische Universitaet Berlin (2012). Disturbed playing: Another kind of educational security games 5th Workshop on Cyber Security Experimentation and Test. Bellevue, WA, USA.

Le Compte, A., Elizondo, D., & Watson, T. (2013). A renewed Approach to Serious Games for cyber Security. 2015 7th International Conference on Cyber Conict: Architectures in Cyberspace M.Maybaum, A.-M.Osula, L.Lindstrm (Eds.) 2015 NATO CCD COE Publications, Tallinn.

Michael, F., Thompson & Cynthia E., Irvine (2014). CyberCIEGE Scenario Design and Implementation USENIX Summit of Gaming, Games, and Gamification in Security Education. Sand Diego, CA, USA.

Olano et al. University of Maryland, Baltimore County (UMBC)(2014). SecurityEmpire: Development and Evaluation of a Digital Game to Promote Cybersecurity Education USENIX Summit of Gaming, Games, and Gamification in Security Education. Sand Diego, CA, USA.

Schreuders, Z., & Butterfield E. Leeds Beckett University (2016). Gamification for Teaching and Learning Computer Security in Higher Education USENIX Summit of Gaming, Games, and Gamification in Security Education. Austin, TX, USA.