## Admin Teleport Command can be used by non admin users.

Anyone can use the admTp command which allows players to teleport anywhere they want effectively beating the game instantly.

\* \*\*Problem Type\*\*: Identification/Authentication failure.

\* \*\*Severity\*\*: Medium – massive advantage for players.

\* \*\* CWSS Score\*\*: 40.3

\*\*CWSS vector\*\*: TI:M,0.6/AP:A,1/AL:D,0.9/IC:L,0.9/FC:T,1/RP:RU,0.7/RL:A,1/AV:L,0.5/AS:N,1/IN:NA,1/SC:A,1/BI:L,0.3/DI:H,1/EX:H,1/EC:N,1/P:NA,1/

### Impact

Allows players to teleport anywhere complete beating the games point.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Allows players to have access to a command they shouldn’t have.

\* \*\*Availability\*\*: N/A

Could be much worse if it’s in multiplayer/network setting and players are able to teleport other players.

### Proof of concept

```

Year 2: Bad Adventure Game

> new

New Session Created

> admTp castle

You teleport to castle

> links

You are in castle

gate, dungeon

> go dungeon

You are in dungeon

A winner is you!

Look, I have no idea why going to the dungeon makes you win either.

I'm a programmer, not a designer...```

### Suggested Fix

A check to see if the player is an Admin before teleporting the player.

```c++

if (!is\_active(state) || state.m\_level != UserLevel::ADMIN) {

out << "You need to be Admin in to do that..." << std::endl;

return;

}

```

## Issue 2 - Admin Password is vulnerable.

The Admin Password which is "MagicPassword42" is not encrypted.

\* \*\*Problem Type\*\*: Sensitive data exposure.

\* \*\*Severity\*\*: High (Allows unauthorised users Admin Access) can lead to [3]

### Impact

Allows unauthorised users to access and use all the admin commands effective which can lead to sever crashes which will affect other players if in a server setting.

\* \*\*Confidentiality\*\*: Possibility to access sensitive data.

\* \*\*Integrity\*\*: Allows the user to access commands they shouldn't be able to.

\* \*\*Availability\*\*: Admin access might allow shutdown of the server.

It would have a lot more consequences if the game was multiplayer/network instead of single player.

### Proof Of Concept

```C++

if (password != "MagicPassword42") {

AdminTries -= 1;

out << "Incorrect password" << std::endl;

return;

}

```

### Possible Fix

Suggest Encrypting the password.

```c++

std::hash<std::string> Hashing;

if (Hashing(password) != 5029200682126204187) {

AdminTries -= 1;

out << "Incorrect password" << std::endl;

return;

}

```

Problem 2

## Admin Password can be Brute Forced

The password can be guessed repeatedly until users find the password.

\*\*\*Problem Type\*\* : Credential-based Attack

\*\*\*Severity\*\*: Medium – It takes time to do this with a modern-day computer estimated at 6 Hundred Million years by Security.org [1]

### Impact

Users can promote themselves to an admin if they guess the correct password.

\*\*\*Confidentiality\*\*: Possibility to access sensitive data.

\* \*\*Integrity\*\*: Allows the user to access commands they shouldn't be able to.

\* \*\*Availability\*\*: Admin access might allow shutdown of the server.

Likely hood of this happening is low but its still a chance and with computer becoming more powerful overtime the chance of this becomes higher and higher.

### Proof Of Concept

‘’’

COMP280 Security Assignment

Year 2: Bad Adventure Game

> new

New Session Created

> promote MagicPassword40

Incorrect password

> promote MagicPassword41

Incorrect password

> promote MagicPassword42

You are now admin.

>

‘’’

###Suggested Fix

Add a certain amount of tries e.g. 3 before stopping user from being allowed to guess

‘’’c++

if (AdminTries <= 0) {

out << "You have entered too many Inccorect passwords please wait:" << std::endl;

return;

}

if (password != "MagicPassword42") {

AdminTries -= 1;

out << "Incorrect password" << std::endl;

return;

}

‘’’

## Crashes when Invalid Argument are given

When you use a command but give it an incorrect number of arguments

\* \*\*Problem Type\*\*: Software Bug

\* \*\*Severity\*\*: Medium – (closes the player game could much more severe if it crashed the sever/network if setup)

### Impact

Will crash and close the player’s game. If network/sever was set up could crash that instead/Aswell.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Causes crash which will immediately stop the player or all players, this can cause data to corrupt.

\* \*\*Availability\*\*:Will stop other users from accessing the server if it has crashed.

### Proof of concept

```

(Proof goes here)

```

### Suggested Fix

(Summary of a fix or possible fix)

```c++

(C++ code goes here)

```

## Player Use of potions buggy

When the player uses a potion whilst having none in inventory allows use of a potion.

\* \*\*Problem Type\*\*: authorization issue.

\* \*\*Severity\*\*: Low – it only gives the player a slight advantage in the game.

### Impact

This allows the player to use potions regardless of if the player has a potion or not. (Some case it can even give the player max number of potions)

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Displays incorrect information and allows access to and unauthorised action.

\* \*\*Availability\*\*: N/A

Even if network/multiplayer was active I doubt it would have any sever effects.

### Proof of concept

```

Year 2: Bad Adventure Game

> new

New Session Created

> inv

You have:

no items

you have 5000 gold

> use 0

you drink a potion

```

### Suggested Fix

Add a If statement that checks if the player actually has potion in their inventory.

```c++

if (!state.m\_gameState->hasItem(Item::POTION)) {

out << "you don't have that item..." << std::endl;

return;

}

```

## Player Item Inventory capped at 255

If the player gets to over 255 of one single type of item, it resets the players items to 0.

\* \*\*Problem Type\*\*: Unsigned Integer overflow

\* \*\*Severity\*\*: Low – it’s unlikely a player will ever reach 255 of a single item since the shop by default is capped to 101 items.

### Impact

It will completely remove all of one type of item from the players inventory.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Modifies players inventory in an unexpected/unwanted way.

\* \*\*Availability\*\*: N/A

Its very unlikely this will ever happen. Will become a major issue as chances increase in network/multiplayer setting if player to player trading is available. There was a case on the NES version of super Mario bros when the player would get anymore than 127 lives it would play Game over the next time the player died [2].

### Proof of concept

```

> inv

You have:

0: Potion (x255)

you have 5000 gold

> buy 0 1

You purchase 1 of 0: Potion

> inv

You have:

no items

you have 5000 gold

>

```

### Suggested Fix

Possible Fixes are increasing the array potential size fix below allows up to 65535 items.

```c++

std::array<uint16\_t, ITEMS.size()> items = { 0 };

```

Or you can set a limit and stop the player from receiving anymore items this will stop the players inventory from looping back to 0.

```c++

if (state.m\_gameState->items.at((int)itemToBuy) + amountToBuy > 255) {

out << "you don't have enouph inventory space" << std::endl;

return;

}

```

## Selling negative amount of items buys items instead

You are allowed to sell a negative number of items which instead buys item from the store instead.

\* \*\*Problem Type\*\*: Authorisation failure.

\* \*\*Severity\*\*: Low – it is reversible as the player can just sell it back but might be a bit confusing.

### Impact

This might make the players accidentally buy items they do not want to which they can just resell back to the shop.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Modifies player inventory in unexpected way.

\* \*\*Availability\*\*: N/A

### Proof of concept

```

> inv

You have:

no items

you have 5000 gold

> sell 0 -10

You sell -10 of 0: Potion

> inv

You have:

0: Potion (x10)

you have 4500 gold

>

```

### Suggested Fix

A simple check to see if the player is trying to sell anything less than 1.

```c++

if (amountToSell <= 0) {

out << "you need to sell atleast one item" << std::endl;

return;

}

```

## Admin Commands displayed to non admin users

Even if the user is not an admin it will display the admin commands to them.

\* \*\*Problem Type\*\*: Information disclosure vulnerability.

\* \*\*Severity\*\*: Low – even if users can see the commands, they should not be able to use them.

### Impact

Players might get confused and try to use the admin commands which won’t work.

\* \*\*Confidentiality\*\*: Displaying extra possibly sensitive data.

\* \*\*Integrity\*\*: Displays unusable actions to players.

\* \*\*Availability\*\*: N/A

This could lead to players trying to find ways and bugs that allow them to use the admin commands just from knowing what commands exist.

### Proof of concept

```

Year 2: Bad Adventure Game

> new

New Session Created

> s

unknown command

commands are: admPrice, admQty, admTp, buy, exit, go, inv, links, load, new, ping, promote, quit, save, sell, shop, use

```

### Suggested Fix

Adding a check to see if the user is a admin or not and then displaying the correct commands.

```c++

if (state.m\_level == UserLevel::ADMIN) {

if (!isFirst) {

out << ", ";

}

out << cmd.first;

isFirst = false;

}

else

{

if (cmd.first.substr(0, 3) != "adm") {

if (!isFirst) {

out << ", ";

}

out << cmd.first;

isFirst = false;

}

}

```

## Displaying Unusable commands to guests

As soon as you launch the game and type any incorrect command it will display all commands even unusable ones.

\* \*\*Problem Type\*\*: Information disclosure vulnerability.

\* \*\*Severity\*\*: low – players will see wrong commands and take more time to find them.

### Impact

If players who are completely new to the game, try figuring out how to create a new instance of the game with the command “new” they might struggle to find it.

\* \*\*Confidentiality\*\*: Displaying extra possibly sensitive data.

\* \*\*Integrity\*\*: Displays unusable actions to players.

\* \*\*Availability\*\*: N/A

### Proof of concept

```

COMP280 Security Assignment

Year 2: Bad Adventure Game

> start

unknown command

commands are: buy, exit, go, inv, links, load, new, ping, promote, quit, save, sell, shop, use

>```

### Suggested Fix

Add a check to see what the user level is and if its guest only looks for the new and load commands.

```c++

if (state.m\_level == UserLevel::GUEST) {

if (cmd.first.substr(0, 3) == "new" || cmd.first.substr(0, 4) == "load") {

if (!isFirst) {

out << ", ";

}

out << cmd.first;

isFirst = false;

}

}

```

## Issue 10 – Integer overflow as arguments

When you put more than 255 arguments as a command or an action it will break the game and make it start repeatedly printing the options for unknown commands.

\* \*\*Problem Type\*\*: Integer overflow.

\* \*\*Severity\*\*: High – It will break the user’s game and during cases of sever/ networking could possibly make servers crash or disrupt the experience for other players.

### Impact

Stops players from being able to play if they attempt to enter a command with more than 255 characters, this could happen in cases where players attempt to enter.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: N/A

\* \*\*Availability\*\*: Stops the player or players from accessing the game.

(Additional notes for Issue)

### Proof of concept

```

COMP280 Security Assignment

Year 2: Bad Adventure Game

> 12121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212121212

unknown command

commands are: load, new

> unknown command

commands are: load, new

> unknown command

commands are: load, new

```

### Suggested Fix

In commands.cpp -> handle\_input

```c++

// Code from

// https://stackoverflow.com/questions/5864540/infinite-loop-with-cin-when-typing-string-while-a-number-is-expected

if (in.fail()) {

out << "To large" << std::endl;

in.ignore(std::numeric\_limits <std::streamsize>::max(), '\n');

in.clear();

}

```

## Issue 11 – the item shop state cannot be saved and loaded.

When the player loads up a previously saved game the shop will reset.

\* \*\*Problem Type\*\*: Logic Error

\* \*\*Severity\*\*: medium –

\*\*\*CWSS Score\*\*: 26.3 [2]

\*\*\*CWSS Vector\*\*: TI:M,0.6/AP:N,0.1/AL:NA,1/IC:N,1/FC:T,1/RP:RU,0.7/RL:A,1/AV:P,0.2/AS:N,1/IN:NA,1/SC:A,1/BI:M,0.6/DI:M,0.6/EX:N,0/EC:N,1/P:C,0.8/ [2]

### Impact

The player can repeatedly buy items that were meant to be out of stock by reloading the game.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Shop data is being modified/refreshed unexpectedly.

\* \*\*Availability\*\*: N/A

In a case of a server or multiplayer this could be more devastating depending on what the game contains. This also can lead on to another bug see Issue 12.

### Proof of concept

```

The shop has:

0: Potion ( x90) @ 50 gold each

> save

> quit

COMP280 Security Assignment

Year 2: Bad Adventure Game

> load

> shop

The shop has:

0: Potion ( x100) @ 50 gold each

1: Key ( x1) @ 100 gold each

> inv

You have:

0: Potion (x10)

1: Key (x1)

you have 4400 gold

>

```

### Suggested Fix

Include the Shop State in the save function. (Not Fixed)

```c++

```

## Issue 12 – Use multiple keys in gate

Player can use multiple keys which spawns multiple castles.

\* \*\*Problem Type\*\*: Authorisation failure

\* \*\*Severity\*\*: Low – has a minor impact on gameplay

\* \*\*CWSS Score\*\*: 12.9

\* \*\*CWSS Vector\*\*: TI:L,0.3/AP:N,0.1/AL:NA,1/IC:N,1/FC:T,1/RP:RU,0.7/RL:A,1/AV:L,0.5/AS:N,1/IN:NA,1/SC:A,1/BI:L,0.3/DI:L,0.2/EX:N,0/EC:N,1/P:L,0.7/

### Impact

Will unnecessary information to player.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Allow player to modify the links and key quantity unexpectedly

\* \*\*Availability\*\*: N/A

If there are more places added that require key becomes bigger problem since player might accidentally use all required keys on one location effectively locking the game. Currently will not be issue since player is meant to have access to one key and one use of it.

### Proof of concept

```

> go gate

You are in gate

> use 1 4

invalid number of arguments

> use 1

you use the key on the gate and it opens...

> use 1

you use the key on the gate and it opens...

> use 1

you use the key on the gate and it opens...

> use 1

you use the key on the gate and it opens...

> use 1

you don't have that item...

> links

You are in gate

road, castle, castle, castle, castle

>

```

### Suggested Fix

Adding a check to see if Castle is already a link and stopping key usage if it is.(Not Fixed)

```c++

```

## Shop does not restock items you sell to it

If you sell a potion or a key to the shop it doesn’t go into the shops inventory.

\* \*\*Problem Type\*\*: Data Processing Errors [9]

\* \*\*Severity\*\*: Medium – can lock player progress

\* \*\*CWSS Score\*\*: 24.9

\* \*\*CWSS Link\*\*: https://www.cwss-score.info/#TI:M,0.6/AP:N,0.1/AL:NA,1/IC:N,1/FC:T,1/RP:RU,0.7/RL:A,1/AV:L,0.5/AS:N,1/IN:H,0.1/SC:A,1/BI:M,0.6/DI:M,0.6/EX:L,0.2/EC:N,1/P:L,0.7/

### Impact

Players could completely buy out shops and resell it deleting the item including keys.

\* \*\*Confidentiality\*\*: N/A

\* \*\*Integrity\*\*: Items are not being saved in shop state

\* \*\*Availability\*\*: Can lock the player progression

Multiplayer/Server setting could change the impact of this bug massively if the shop is shared across multiple players.

### Proof of concept

```

> sell 1 1

You sell 1 of 1: Key

> shop

The shop has:

0: Potion ( x100) @ 50 gold each

>

```

### Suggested Fix

Alter the shop item quantity to add the sold items.

Shop.cpp -> shopSell()

```c++

//returns item to the shop

state.m\_world->alterItemQty(itemToSell, amountToSell);

```

# References

[1]Security.org. (2023, November 25). *How Secure Is My Password*. Retrieved from How Secure Is My Password?: https://www.security.org/how-secure-is-my-password/

[3] R. Singh Verma and B. R. Chandavarkar, “Hard-coded credentials and web service in IOT: Issues and challenges,” SSRN, https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3358283 (accessed Dec. 2, 2023).

Issue 10

[1]

“CWE - CWE-190: Integer Overflow or Wraparound (4.5),” *cwe.mitre.org*. <https://cwe.mitre.org/data/definitions/190.html> (accessed Dec. 04, 2023).

Issue 4

[1]

“CWE - CWE-1284: Improper Validation of Specified Quantity in Input (4.13),” *cwe.mitre.org*. https://cwe.mitre.org/data/definitions/1284.html (accessed Dec. 04, 2023).

Issue 1

[1] “A07:2021 – identification and authentication failures,” A07 Identification and Authentication Failures - OWASP Top 10:2021, https://owasp.org/Top10/A07\_2021-Identification\_and\_Authentication\_Failures/ (accessed Dec. 5, 2023).

Cia triad

[1]

S. Samonas and D. Coss, “The Cia Strikes Back: Redefining Confidentiality, Integrity and Availability in Security,” 2014. Available: https://www.proso.com/dl/Samonas.pdf

‌

This is where I documented all the Issue’s I found. To identify the type of issue I used CWE which stands for Common Weakness Enumeration [1] this is a community-developed List of most known Issue.

For calculating and justifying the severity I used CWSS which stands for Common Weakness Scoring System I used a calculator [2] you can see what I inputted into the calculator by using the CWSS Vector in the URL.

For identifying and justifying the CIA triad I went off the definitions in “The Cia Strikes Back: Redefining Confidentiality, Integrity and Availability in Security” [3].

‌

‌