

Possible attack

Can overwrite other player achievements

GameSummary.Sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/GameSummary.sol#L303>)

```
function safeTransferFrom(address _from, address _to, uint256 _id, uint256 _am
    require(playerGameData[_from][_id].tokenId != 0, "Token doesn't exists"
    require(!playerGameData[_from][_id].soulbounded, "You can't transfer th
    PlayerGameData storage playerData = playerGameData[_from][_id];
    uint256 transferachievements = playerData.achievementsMinted;
    playerGameData[_from][_id] = PlayerGameData(0, 0, false);
    playerGameData[_to][_id] = PlayerGameData(_id, transferachievements, fa
    super.safeTransferFrom(_from, _to, _id, _amount, _data);
}
```

Doesnt check the existance of

```
playerGameData[_to][_id].tokenId == 0
```

A player can send their achievements to another one, and overwrite their progress.

Invalid check

Old level check is incorrect

LevelsBound.Sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/LevelsBound.sol#L53>)

```

function levelUp(address account, uint256 newLevel) public onlyOwner {
    require(newLevel > 0, "New level must be greater than 0");
    // check if the user has the previous lvl token
    require(balanceOf(account, newLevel) == 0, "Player already has this level t

    if (newLevel == 1) {
        mintLevel(account, newLevel);
        return;
    }

    uint oldLevel = newLevel - 1;

    // check if the user has the previous lvl token
    require(balanceOf(account, oldLevel) == 1, "Player does not have the previo

    // check if the "lvl up" actually is a "lvl down"
    require(balanceOf(account, oldLevel) < newLevel, "Is not possible to do lvl

    // Burn the old token
    burnLevel(account, oldLevel);
    mintLevel(account, newLevel);
}

```

```

require(balanceOf(account, oldLevel) < newLevel, "Is not possible to do lvl
down");

```

This is checking the balance of a level token (should be 0 or 1) against newLevel height.

- It should be something like:

```

require(playerLevel[account] < newLevel, "Is not possible to do lvl down");

```

Inconsistency 1

Burning tokens behaviour

LevelsBound.Sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/LevelsBound.sol#L84>)

```

function burnLevel(address account, uint256 tokenId) public onlyOwner {
    _burn(account, tokenId, 1);
}

```

If the Admin burns the token it doesn't reset the player level, but if the player is the one burning it, it reset it.

- Solution:
Add the missing the lines the the admin burning function to behave like the other burnLevel methods

```
// Missing
playerLevel[account] = 0;
```

Inconsistency 2

Soulbound and SoulboundBatch behave differently

ERCSoulbound.Sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/ERCSoulbound.sol#L149>)

```
function _soulbound(address to, uint256 tokenId, uint256 amount) internal virtu
    _soulbounds[to][tokenId] += amount;
    emit Soulbound(to, tokenId, amount);
}
```

ERCSoulbound.Sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/ERCSoulbound.sol#L162>)

```
function _soulboundBatch(address to, uint256[] memory tokenIds, uint256[] memo
    require(tokenIds.length == amounts.length, "ERCSoulbound: tokenIds and
    for (uint256 i = 0; i < tokenIds.length; i++) {
        _soulbounds[to][tokenIds[i]] = amounts[i];
    }
    emit SoulboundBatch(to, tokenIds, amounts);
}
```

- First case adds to amount
 _soulbounds[to][tokenId] += amount;
- Second case overwrites
 _soulbounds[to][tokenIds[i]] = amounts[i];

Incorrect check

Can not complete 100% of the achievements

GameSummary.Sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/GameSummary.sol#L178>)

```

function addPlayerAchievements(address player, uint256 tokenId, uint256 newA
    require(tokenId > 0, "TokenId must be greater than 0");
    require(playerGameData[player][tokenId].tokenId != 0, "Token doesn't exi
    PlayerGameData storage playerData = playerGameData[player][tokenId];
    if (playerData.achievementsMinted + newAchievements >= commonGameSummari
        revert("total achievements exceeded");
    }
    playerData.achievementsMinted += newAchievements;
    emit PlayerGameSummaryUpdated(player, tokenId, playerData.achievementsMi
}

```

- Total achievements exceeded check should be greater than (>) instead of greater or equals (>=).

```

    if (playerData.achievementsMinted + newAchievements > commonGameSummari
        revert("total achievements exceeded");
    }

```

Events

Wrong event emitted

AvatarBound.sol (<https://gitfront.io/r/ogarciarevett/2DuWU5ScT8f4/achievo-contracts/blob/contracts/upgradeables/AvatarBoundV1.sol#L365>)

```

function setMintNftWithoutGatingEnabled(bool _mintNftWithoutGatingEnabled) publ
    require(_mintNftWithoutGatingEnabled != mintNftWithoutGatingEnabled, "N
    mintNftWithoutGatingEnabled = _mintNftWithoutGatingEnabled;
    emit MintNftGatingEnabledChanged(_mintNftWithoutGatingEnabled);
}

```

- Should be

```

emit MintNftWithoutGatingEnabledChanged(_mintNftWithoutGatingEnabled);

```