Ultraviolet Audit

By: Cygaar

UVCollectable.sol:

1. Should this be renamed to UVCollectible.sol?
2. Import on line 20 can be removed since it should be part of the ERC721BurnableUpgradeable import
3. Line 58: a uint256 is cheaper to use than a uint16 because every variable will take up one storage slot which is 32 bytes.
4. Line 110: I'm pretty certain that doing contractURI = __contractURI directly will be slightly cheaper than calling updateContractURI. The amount of saved gas is negligible though.
5. Line 129: I'm pretty sure the return type is incorrect here, my IDE is giving me an error. The type of ERC2771Recipient._msgData is bytes calldata not bytes memory. The fix should be to change the return signatures to bytes calldata on line 133.
6. Additionally, I would double check that OperatorFiltererUpgradeable is using the overwritten _msgSender() value. I think it should be, but this is something that I would personally test.
7. Line 174: right now admins can remove other admins (or themselves). Do you want this behavior? Might be worth having admin adding/removing restricted to onlyOwner.
8. updateName and updateSymbol input parameters can be changed to calldata.
9. You should not need to mark any of the functions in this contract as virtual unless you're planning on having another contract inherit UVCollectible for some reason. Virtual means that inheriting contracts can override the given function.
10. Just so I understand your metadata setup correctly, all metadata info will be stored at the contractURI location. At the root of this directory you will have the storefront metadata (for OpenSea), and the token-specific metadata will be queryable using the collection id and token id?
11. updateContractURI doesn't need to exist if you use suggestion #4.
12. Line 366, is there a reason why you're using the "this" keyword? "this" can be used to call an external function within the same contract, but I don't think you really need it here.
13. Line 386, lockToken can only be called by the owner of the token right now. Should this be onlyOwnerOrAdmin?
14. Line 497, do you want to return true if the tokenId doesn't exist?

15. Line 523: the comment should be changed from “to the owner’s wallet” to “to the recipient’s wallet”.
16. Line 559, this increment can be put in an unchecked block to save gas. If you want to keep the overflow check, you can do ++i to save a bit of gas.
17. For the minting logic, it doesn’t look like you will ever return false. If an operation fails, you will get an error. I would remove the boolean return since it doesn’t do anything.
18. Nit: I would probably rename _mintTokenV2 to be _mintToken and change the original _mintToken to be _mintTokenWithDuration or something similar.
19. Just so you’re aware, the first tokenId in your collection will be 1. I know some collections want to start at id 0 so I wanted to highlight this.
20. In the _mintToken function, you can save gas by in-lining the logic for setting the expiration date of the subscription. Since you know the starting expiration is 0, you can circumvent the if-check in _extendSubscription and use setExpiration directly (similar to what you have in mintUserToManyCollections.
21. Seems like _setDefaultRoyalty can be made external and updateDefaultRoyalty can be removed. Same for _setCollectionRoyalty.
22. Line 940 typo: this should be called setTrustedForwarder

Some general questions:

- Are payments completely handled outside of the contract? None of the current functions are payable so I’m assuming you have a payment layer on top of the contract.