Universidade Estadual do Ceará
Centro de Ciências e Tecnologia
Curso de Graduação em Ciência da Computação
Disciplina: CC111 - Tópicos Especiais em Eng. de Software

Relatório de Análise Estática

BleachBit

Francisco Alessandro Carvalho Evaristo - 1595378
Igo Florentino Venâncio - 1538786
Jaime Silva de Abreu - 1590590
Matheus Vieira de Araújo - 1357315
Rhuan Mateus Matias Filgueira - 1605351

Histórico de versões

Versão	Data	Autor	Descrição
1.0	15/09/2023	Jaime	Preparação do Ambiente
1.1	18/09/2023	Igo	Inserção dos dados gerais do resultado da análise tópico 3 e resultados gerais tópico 2
1.2	18/09/2023	Alessandro	Descrição da introdução
1.3	18/09/2023	Matheus	Ajuste na descrição do erro 11 e 12
1.4	19/09/2023	Igo	Preenchimento dos erros 10, 13, 14, 15
1.5	20/09/2023	Alessandro	Preenchimento e edição dos tópicos 1, 2 e 5
1.6	20/09/2023	Jaime	Preenchimento dos erros 7 e 8
1.7	23/09/2029	Matheus	Preenchimento da conclusão
1.8	25/09/2023	Jaime	Revisão documento e melhoria nas descrições das seções

Sumário

1. Introdução

Esse documento apresenta o processo de análise estática do software BleachBit. Mostrando as ferramentas utilizadas para a análise estática, os resultados dessa análise e possíveis soluções ou melhorias para o código.

1.1. Aplicação e código fonte

A aplicação a ser analisada será o BleachBit, desenvolvido em Python, que é um software open source que auxilia a limpar o sistema e o disco de arquivos desnecessários. Ele também exclui arquivos permanentemente e seus vestígios. Além disso, é possível limpar a maioria dos aplicativos de computador.

1.2. Descrição da(s) ferramenta(s) de Análise Estática

A principal ferramenta utilizada de análise estática é o Pylint, que é um analisador estático de códigos em python 2 ou 3. O Pylint analisa um código sem executá-lo. Ele busca erros, incentiva boas práticas de programação, identifica **code smells** e faz recomendações de como melhorar o código. Adicionalmente iremos utilizar o bandit que é um analisador estático de código-fonte para python no qual irá procurar potenciais vulnerabilidades de segurança e problemas de código. Para fazer isso, o Bandit processa cada arquivo, constrói uma AST - árvore de análise sintática e a partir dela executa plug-ins apropriados nos nós AST. Assim que o Bandit terminar de escanear todos os arquivos, ele gerará um relatório. Essa ferramenta foi escolhida para dar mais robustez a análise estática já que o pylint não cobria a categoria segurança.

2. Resultados gerais

Serão relatados 15 problemas observados durante a análise estática através das ferramentas Pylint e bandit. O Pylint divide suas mensagens em fatal, error, warning, refactor e information, já o Bandit acusa issues, que informam o tipo de falha de segurança observada.

2.1. Lista de problemas analisados

Problema 1, arquivo 'bleachbit/_init_'

Identificador	Pylint(E1101:no-member).
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Erro de Sintaxe.
Localização	Linha 45, coluna 31.

Mensagem	Module 'sys' has no 'frozen' member.
Trecho do código	`if hasattr(sys, 'frozen') and sys.frozen == 'windows_exe':` `    stdout_encoding = 'utf-8'`
Proposta de solução	Retirar o método ou utilizar outro com utilidade semelhante.
Comentários	A ferramenta pylint indicou que o método .frozen do módulo sys não existe e não indicou uma possível solução para a resolução do problema.

Problema 2, arquivo ‘bleachbit/_init_’

Identificador	Pylint(E0401:import-error).
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Erro de Sintaxe.
Localização	Linha 50, coluna 09.
Mensagem	Unable to import 'win_unicode_console'.

Trecho do código	`if 'win32' == sys.platform:`
	`        import win_unicode_console`
	`        win_unicode_console.enable()`

Proposta de solução	Apenas retirar o módulo win_unicode_console ou encontrar outro módulo para realizar a tarefa que o win_unicode_console tenta.
Comentários	A ferramenta pylint indicou que não é capaz de importar o módulo win_unicode_console. Isso acaba por inutilizar o comando seguinte que utiliza esse módulo.

Problema 3, arquivo ‘bleachbit/_init_’

Identificador	Pylit(W4904:deprecated-class).
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Recomendações de melhorias.
Localização	Linha 33, coluna 1.
Mensagem	Using deprecated class SafeConfigParser of module configparser

	Pylint(w4904:deprecated-class) [Ln 33, Col 1]
Trecho do código	`from configparser import RawConfigParser, NoOptionError, SafeConfigParser`
Proposta de solução	Utilizar uma classe atualizada.
Comentários	A classe está marcada como obsoleta e será removida no futuro.

Problema 4, arquivo ‘bleachbit/_init_’

Identificador	Pylint(C0103:invalid-name).
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Recomendações de melhorias.
Localização	Linha 39, coluna 01.
Mensagem	Constant name "socket_timeout" doesn't conform to UPPER_CASE naming style.
Trecho do código	`socket_timeout = 10`
Proposta de solução	Colocar o nome da constante em maiúsculo.
Comentários	A ferramenta pylint indicou que o nome da constante socket_timeout não está escrito conforme as regras associadas ao tipo de dado constante. Essa recomendação de nome inválido aparece para muitas outras variáveis desse arquivo.

Problema 5, arquivo ‘bleachbit/_platform’

Identificador	Pylint(W0622:redefined-builtin).
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Aviso de prevenção.
Localização	Linha 121, coluna 41.
Mensagem	Redefining built-in 'id'.
Trecho do código	`def _dist_try_harder(distname, version, id):`

Proposta de solução	Ajustar o nome do parâmetro id para algum mais específico, como: id_value. Devido a possível confusão que o nome do parâmetro possa criar com o valor integrado id.
Comentários	A ferramenta pylint avisou que o parâmetro id pode sobrepor o valor integrado conhecido por id.

Problema 6, arquivo ‘bleachbit/_platform’

Identificador	Pylint(W1514:unspecified-encoding)
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Aviso de prevenção.
Localização	Linha 133, coluna 14.
Mensagem	Using open without explicitly specifying an encoding
Trecho do código	`with open('/var/adm/inst-log/info') as f:`
Proposta de solução	Inserir o argumento encoding = “utf-8” no método open.
Comentários	A ferramenta pylint avisou que utilizar o método open sem especificar a codificação do arquivo aberto pode gerar problemas operacionais do sistema.

Problema 7, arquivo ‘bleachbit/_testCLI’

Identificador	Pylint
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Aviso de prevenção
Localização	Linha 47, coluna 14.
Mensagem	Using open without explicitly specifying an encoding
Trecho do código	`with open(os.devnull, 'w') as stdout:`
Proposta de solução	Inserir o argumento encoding = “utf-8” no método open.
Comentários	Recomenda-se definir explicitamente uma codificação ao abrir

	documentos, pois depender do padrão do sistema de forma implícita pode causar complicações em diferentes sistemas operacionais.

Problema 8, arquivo ‘bleachbit/_testCLI’

Identificador	Pylint
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Ordem de importação errada
Localização	Linha 30, coluna 1.
Mensagem	standard import "import copy" should be placed before "from bleachbit.CLI import *"
Trecho do código	`import copy`
Proposta de solução	Primeiro, devem ser declaradas as importações padrões, depois as bibliotecas e, em seguida, as importações locais.
Comentários	Deve-se respeitar a ordem de importação

Problema 9, arquivo ‘bleachbit/Command’

Identificador	Pylint(E1101:no-member)
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Erro de sintaxe
Localização	Linha 97, coluna 26.
Mensagem	Instance of 'WindowsError' has no 'winerror' member
Trecho do código	`if 32 != e.winerror and 5 != e.winerror:`
Proposta de solução	Retirar o método ou substituir por outro semelhante.
Comentários	A ferramenta pylint informou um erro indicando que não existe o método winerror no módulo WindowsError.

Problema 10, arquivo ‘bleachbit/Command’

Identificador	Pylint(W0706:try-except-raise)
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Recomendações de melhorias.
Localização	Linha 101, coluna 17.
Mensagem	The except handler raises immediately.
Trecho do código	`except:` `raise`
Proposta de solução	Retirar o bloco de código do except
Comentários	A ferramenta pylint avisou que o bloco da condição except é inútil e pode ser removido, já que utilizou o método raise logo após entrar na condição.

Problema 11, arquivo ‘bleachbit/Unix.py

Identificador	Bandit(B603:subprocess_without_shell_equals_true)
Ferramentas envolvidas	Bandit e Visual Studio Code.
Categoria	Segurança
Localização	Linha 591, coluna 13.
Mensagem	subprocess call - check for execution of untrusted input
Trecho do código	590 env = {'LC_ALL': 'C', 'PATH': os.getenv('PATH')} 591 output = subprocess.check_output([cmd] + args, stderr=subprocess.STDOUT, 592 universal_newlines=True, env=env) 593 freed_space = 0
Proposta de solução	Certificar-se de que os comandos passados para o subprocess sejam seguros e não contenham entradas não confiáveis que possam ser exploradas.
Comentários	O código está usando o subprocess sem especificar a flag ‘shell’ explicitamente. Isso pode ser perigoso, pois, por padrão, o ‘shell’ é definido como ‘False’, o que significa que o comando é executado diretamente sem passar por um shell.

Problema 12, arquivo ‘bleachbit/SystemInformation.py

Identificador	Bandit(B110:trye_except_pass)
Ferramentas envolvidas	Bandit e Visual Studio Code.
Categoria	Recomendações de melhorias.
Localização	Linha 62, coluna 4.
Mensagem	Try, Except, Pass detected
Trecho do código	61 s += '\nGTK prefer dark theme = %s' % Gtk.Settings.get_default().get_property('gtk-application-prefer-dark-theme') 62 except: 63 pass 64 import sqlite3
Proposta de solução	Tratar os cenários de except para que os erros que ocorrem durante a execução do programa sejam detectados e registrados.
Comentários	Se não tratar as exceções, o programa pode continuar executando mesmo quando algo dá errado.

Problema 13, arquivo ‘bleachbit/_testCLI’

Identificador	Pylint
Ferramentas envolvidas	Pylint e Visual Studio Code.
Categoria	Ordem de importação errada
Localização	Linha 30, coluna 1.
Mensagem	standard import "import copy" should be placed before "from bleachbit.CLI import *"
Trecho do código	`import copy`
Proposta de solução	Primeiro, devem ser declaradas as importações padrões, depois as bibliotecas e, em seguida, as importações locais.
Comentários	Deve-se respeitar a ordem de importação

Problema 14, arquivo ‘bleachbit/Action.py’

Identificador	Issue: [B404: blacklist]
Ferramentas envolvidas	Bandit e Visual Studio Code.
Categoria	Segurança
Localização	Linha 555, coluna 20.
Mensagem	Consider possible security implications associated with the subprocess module.

Trecho do código	`rc = 0  # unknown because we don't wait`
	`from subprocess import Popen`
	`Popen(self.cmd)`

Proposta de solução	Considerar possíveis implicações de segurança associadas a esses módulos.
Comentários	Uma lista negra contém vários módulos de Python conhecidos por terem possíveis implicações de segurança.

Problema 15, arquivo ‘bleachbit/Action.py’

Identificador	Issue: [B101: assert_used]
Ferramentas envolvidas	Bandit e Visual Studio Code.
Categoria	Segurança
Localização	Linha 138, coluna 8.
Mensagem	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.

Trecho do código	`self.nwholeregex = action_element.getAttribute('nwholeregex')`
	`assert(isinstance(self.nwholeregex, (str, type(None))))`
	`self.search = action_element.getAttribute('search')`

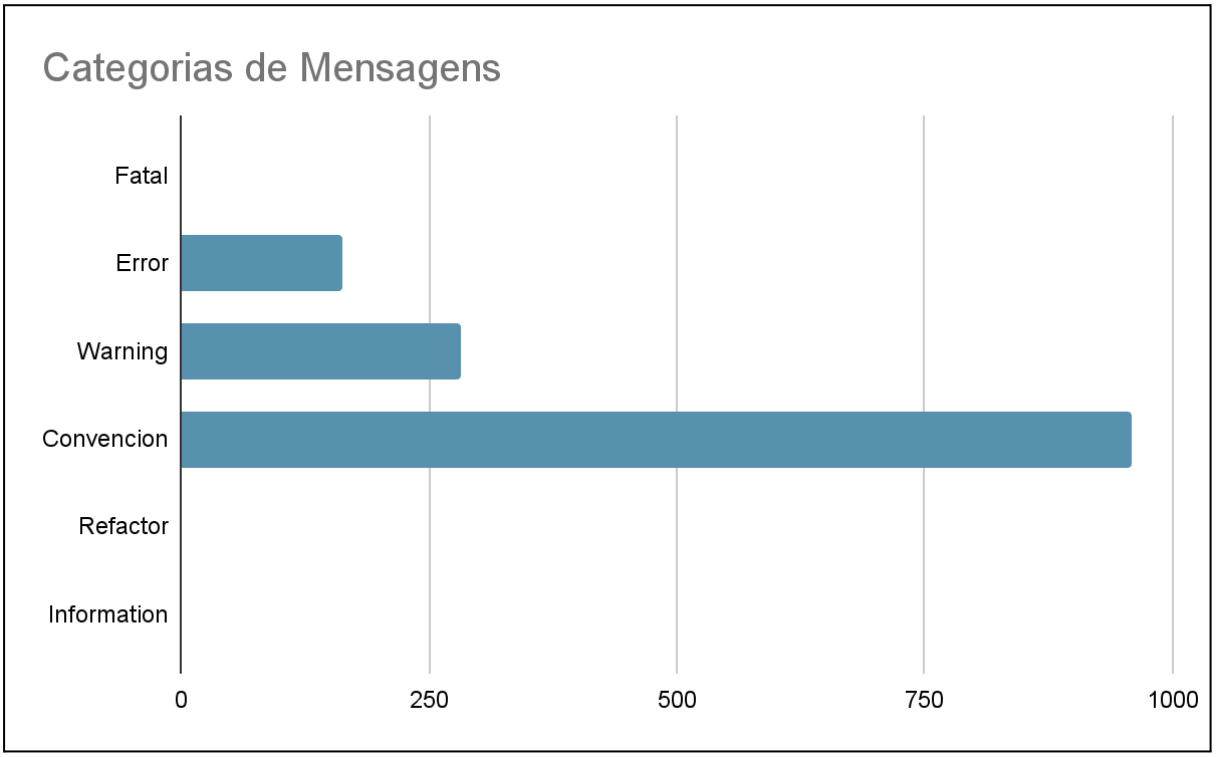
Proposta de solução	Configurar arquivos que ignoram esta verificação.
Comentários	O assert é removido com a compilação e isso faz com que várias

	proteções sejam removidas.

3. Discussão

Durante a execução do Pylint foi observado que a maioria dos erros apresentados são relacionados a convenção da linguagem Python, ficando em segundo lugar as mensagens de warning e terceiro as de error. Não foram observadas mensagens apresentando erros fatais e nem de informação. Já a ferramenta Bandit apresentou 89 issues, que representam possíveis erros de segurança, sendo analisadas 8998 linhas de código, com as métricas de severidade apresentando 67 pontos para o médio e confiança em 85, que se referem a todas as issues (problemas) acumuladas.

Pylint



Bandit

Issues	89 security problems
Code scanned	Total lines of code: 8998
	Total lines skipped: 0

Run metrics	Total issues (by severity): Undefined: 0, Low: 67 Medium: 15, High: 0. Total issues (by confidence): Undefined: 0, Low: 3 Medium: 4, High: 75.
Files skipped	0

4. Conclusão

O relatório de análise estática do BleachBit apresentou uma análise detalhada do código fonte da aplicação, utilizando ferramentas específicas para identificar possíveis problemas e melhorias. Foram apresentados os resultados gerais da análise, incluindo o número da linha do erro no código, os problemas identificados por severidade, além disso, foram listados as possíveis soluções para cada um deles. Por fim, o documento apresentou a bibliografia utilizada na elaboração do relatório.

5. Referências

Site do pylint sobre os códigos das mensagens geradas:
https://pylint.readthedocs.io/en/latest/user_guide/messages/messages_overview.html

Bandit. Disponível em:
https://bandit.readthedocs.io/en/latest/

Glossário

Termo	**Definição**
Code Smells	Entendido como qualquer característica no código-fonte de um programa que possivelmente indica um problema mais profundo.
Plugins	Ferramenta que adiciona recursos a um determinado programa.
Issues	Problemas
Files Skipped	Arquivos que não foram analisados
Code Scanned	Código analisado
Run metrics	Dados referentes às informações obtidas após análise.
Security problems	Problemas de segurança