기능 분석 보고 발표 보고서

1조 : 황치신, 김서연, 주문도, 박상혁, 한효정

-목차-

1. PyT 개요
   1. PyT 란?
   2. 추상구문 트리란?
   3. 제어 흐름 그래프란?
   4. 데이터 흐름 분석이란?
2. PyT 기능분석

2-1. def main(command\_line\_args = sys.argv[1 : ] )

2-2. def parse\_args(args)

2-3. def generate\_ast(path)

2-4. def make\_cfg( )

2-5. class FrameworkAdaptor( )

2-6. def analyse(cfg\_list)

2-7. find\_vulnerabilities( )

2-8. get\_vulnerabilities\_not\_in\_baseline( )

2-9. args.formatter.report( )

1. 향후 방향 및 개선점
2. 참고자료

1. PyT 개요

1.1 PyT 란?

PyT는 제어흐름 그래프와 추상구문트리, 데이터 흐름분석에 이론적 토대를 둔 파이썬 웹 애플리케이션 정적 분석 도구로서 명령 인젝션, SSRF, SQL인젝션, XSS, directory traverse 등을 탐지하며 사용자의 요구에 따른 다양한 옵션을 제공한다.

* 1. 추상구문 트리(AST)란?

PyT내에서 추상구문 트리는 중요한 기능을 수행하는데 우선 추상구문 트리에 대하여 설명하자면 추상구문 트리는 프로그래밍 언어로 작성된, 소스코드로 이루어진 추상구문 구조의 트리이다. 추상구문 트리의 각 노드는 소스코드에서 발생되는 구조체를 나타내며 이때 추상적이란 의미는 실제 구문에서 나타나는 모든 세세한 정보를 나타내지는 않는다는 것을 의미한다. PyT내에서는 ast를 import 하여 ast.parse(f.read()) 등의 방식으로 쓰인다.

* 1. 제어 흐름 그래프(CFG)란?

제어흐름 그래프는 프로그램이 실행 중에 횡단할 수 있는 모든 경로를 그래프 표기법을 사용하여 표현한 것으로 이 그래프에서는 모든 노드가 각각의 명령어를 표현한다. PyT는 추상구문 트리를 이용하여 정리한 데이터를 제어흐름 그래프를 통하여 통제하며 CFG를 프레임 웨크 어댑터에 전달하여 특정기능의 인수를 감염된 소스로 표시하는데 이용된다.

* 1. 데이터 흐름 분석(Data-flow analysis)이란?

데이터 흐름 분석이란 프로그램에서 다양한 지점에서 계산된 분석 요구에 부함하는 값들의 집합에 대한 정보를 모으는 기법이다. 데이터 흐름 분석에서 CFG는 프로그램의 이러한 값들에서 변수에 저장되기 위해서는 어떤 값이 전파되야 하는지 결정하는데 사용된다. PyT 내에서는 CFG의 각 노드에 필요한 방정식을 세우고 전체 시스템이 안정될 때 까지 각 노드의 입력에서 결과물을 반복적으로 계산하는 수행방식을 이용한다.

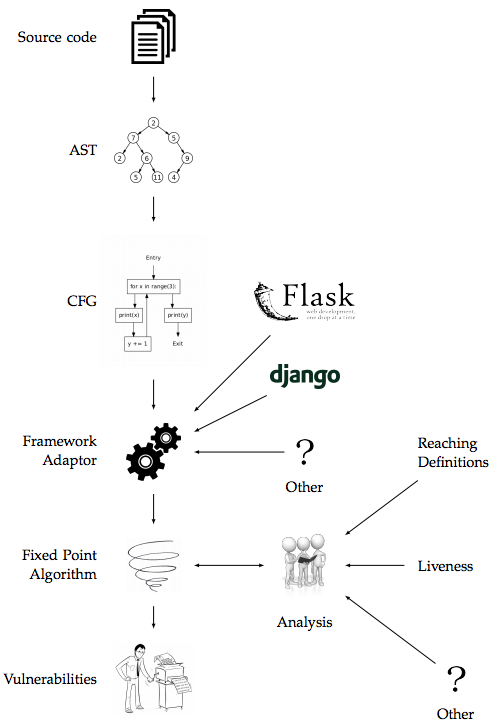


그림 1. 오픈소스 PyT의 전체적인 기능도는 다음과 같다.

1. PyT 기능분석

2.1 def main(command\_line\_args = sys.argv[1 : ] )

def main의 파라메터(command\_line\_args)에 프롬프트의 명령어가 저장된다.

2.2 def parse\_args(args)

Main에서 명령어를 해석하는 함수를 실행, usage.py parse\_args 함수 의 인자로 command\_line\_args 를 넘겨준다. 이를 받은 parse\_args함수에서 command\_line\_args를 해석하여 반환한다.

2.3 def generate\_ast(path)

메인함수에서 ast\_helper.py의 generate\_ast(path)를 호출한다. 호출된 generate\_ast함수는 전달받은 path가 파일이면 path를 읽기모드로 f참조변수를 통해 연다. 이때 ast.parse(f.read())를 실행하여 반환한다.

2.4 def make\_cfg( )

메인 함수에서 make\_cfg 함수를 호출할 때, AST를 받아 ExprVisitor를 생성하고 CFG를 반환한다. Visitor는 ExprVisitor의 인스턴스이고 ExprVisitor는 make\_cfg의 인자들을 넘겨받아 생성된다. 생성된 make\_cfg 함수를 호출하면 ExprVisitor 객체가 생성되고, CFG 클래스에 인스턴스인 visitor를 넘겨주고 이것을 반환한다.

2.5 class FrameworkAdaptor( )

특정 함수의 인자를 감염된 소스로 마크하게 되면 메인 함수에서 FrameworkAdaptor 클래스를 생성한다. FrameworkAdaptor 클래스는 framework\_route\_criteria를 인자로 갖는데, 이 인자는 선택된 웹 프레임워크의 경로를 가지는지 아닌지를 반환해주는 함수다.

2.6 def analyse(cfg\_list)

메인 함수에서 analyse 함수를 호출하고 인자로 CFG 리스트를 넣는다. analyse함수는 일련의 CFG를 주어진 분석타입으로 분석하여 cfg\_list의 원소를 FixedPointAnalysis 클래스에 넣고 fixpoint\_runner를 호출한다. fixpoint\_runner 함수는 fixpoint 알고리즘을 실행하여 CFG를 분석한다.

2.7 find\_vulnerabilities( )

메인 함수에서 find\_vulnerabilities 함수를 호출하고 인자로 CFG 리스트를 받아, 취약점 리스트를 반환한다. Find\_vulnerabilities 함수는 파일 내 모든 소스와 싱크를 찾고 소스코드가 싱크에 도달하는지 확인하는 함수이며 이를 반복하여 소스와 싱크 사이의 모든 경로를 발견한다.

이 때 취약점들을 얻게 되는데, 이것이 ‘어떻게’ 취약한지를 how\_vulnerable함수를 호출하여 알아본다. sink란, 다른 객체나 함수로부터 들어오는 이벤트를 받게 설계된 클래스 혹은 함수이다.

2.8 get\_vulnerabilities\_not\_in\_baseline( )

baseline파일이 존재할 때, get\_vulnerabilities\_not\_in\_baseline함수를 실행한다. baseline에 이미 기록된 취약점의 경우 바로 제거하고, 없는 경우 baseline에 기록한다.

2.9 args.formatter.report( )

결과를 text(표준출력) 혹은 JSON(출력파일) 형태로 출력하여 완전히 취약점이 제거 됐는지 확인한다.

1. 개선점

오픈소스 PyT의 정밀분석 강화를 위해서는

1) 보다 디테일한 소스 분석을 절차적으로 진행하여 committer들이 프로그래밍한 코드 이해도를 높여야 한다.

2) 제어흐름그래프, 추상구문트리, 도달 정의 분석, 고정점 알고리즘 등의 주 기능에 대한 조사 및 이해 강화가 필요하다.

3) 파이썬에 대한 지속적인 학습이 필요하다.

이상의 3가지 개선점을 적용할 시 본 프로젝트에 대한 더욱 긍정적인 결과를 도출할 것으로 기대된다.

1. 참고자료

추상 구문 트리[https://ko.wikipedia.org/wiki/%EC%B6%94%EC%83%81\_%EA%B5%AC%EB%AC%B8\_%ED%8A%B8%EB%A6%AC](https://ko.wikipedia.org/wiki/ì¶ì_êµ¬ë¬¸_í¸ë¦¬)[. 2018. 11. 14.](https://ko.wikipedia.org/wiki/ì¶ì_êµ¬ë¬¸_í¸ë¦¬. 2018. 11. 14.)

제어 흐름 그래프. [https://ko.wikipedia.org/wiki/%EC%A0%9C%EC%96%B4\_%ED%9D%90%EB%A6%84\_%EA%B7%B8%EB%9E%98%ED%94%84. 2018. 11. 14.](https://ko.wikipedia.org/wiki/ì ì´_íë¦_ê·¸ëí. 2018. 11. 14.)

데이터 흐름 분석. [https://ko.wikipedia.org/wiki/%EB%8D%B0%EC%9D%B4%ED%84%B0\_%ED%9D%90%EB%A6%84\_%EB%B6%84%EC%84%9D. 2018. 11. 14.](https://ko.wikipedia.org/wiki/ë°ì´í°_íë¦_ë¶ì. 2018. 11. 14.)