Automated Assessment Tools

Barton P. Miller
Computer Sciences Department
University of Wisconsin

bart@cs.wisc.edu

Elisa Heymann

Computer Sciences Department
University of Wisconsin
Universitat Autònoma de Barcelona

elisa@cs.wisc.edu

CENTER FOR TRUSTWORTHY
SCIENTIFIC CYBERINFRASTRUCTURE
The NSF Cybersecurity Center of Excellence

CWE 78: OS Command Injection

```c
void CWE78_OS_Command_Injection__char_console_execl_41_bad(){
    char *data; char dataBuffer[100] = "";
    data = dataBuffer;
    /* Read input from the console */
    size_t dataLen = strlen(data);
    /* If there is room in data, read into it from the cons */
    if (100-dataLen > 1)  {
    /* POTENTIAL FLAW: Read data from the console */
      if (fgets(data+dataLen,(int)(100-dataLen),stdin)!= NULL)
      {
            /* Remove the carriage return from the string */
            dataLen = strlen(data);
            if (dataLen > 0 && data[dataLen-1] == '\n')
                 data[dataLen-1] = '\0';
            else {
                 printf("fgets() failed\n");
                 data[dataLen] = '\0';
            }
            /* POTENTIAL FLAW: Execute command without
               validating */
             system (data);
      }
}
```

How to Describe a Weakness

Descriptive name of weakness (CWE XX)

An intuitive summary of the weakness.

- **Attack point:** How does the attacker affect the program.
- **Impact point:** Where in the program does the bad thing actually happen.
- **Mitigation:** A version of the program that does not contain the weakness.

(CWEXX_Long_Detailed_File_Name_Containg_The_Code_yy.cpp)

OS Command Injection (CWE 78)

User supplied data is used to create a string that will be interpreted by a command shell.

- **Attack Point:** Input read from the console.
- **Impact Point:** Executing command with `system()`.
- **Mitigation:** Don't execute user provided input; instead use a fixed string.

CWE78_OS_Command_Injection__char_console_execl_41.c
(Highly modified to compensate for errors.)

Coverity Analyze

Coverity

- Commercial tool. Available at http://www.coverity.com/
- Starting Point: Accurate Compilation.
- Depth and Accuracy of Analysis
  - Interprocedural Dataflow Analysis.
  - False Path Pruning.
  - Design Pattern Intelligence.
  - Enterprise Framework Analyzer.
  - White Box Fuzzer.
- Scalable.

Coverity

1. **Download the license and the software:** `https://coverity.secure.force.com/apex/LicenseManagement2`
2. **Run the installation script:** `cov-analysis-linux64-7.6.0.sh`
3. **Include in PATH the location of** `~elisa/cov-analysis-linux64-7.6.0/bin`
4. **Command line and graphic interface.**

Coverity

Steps:

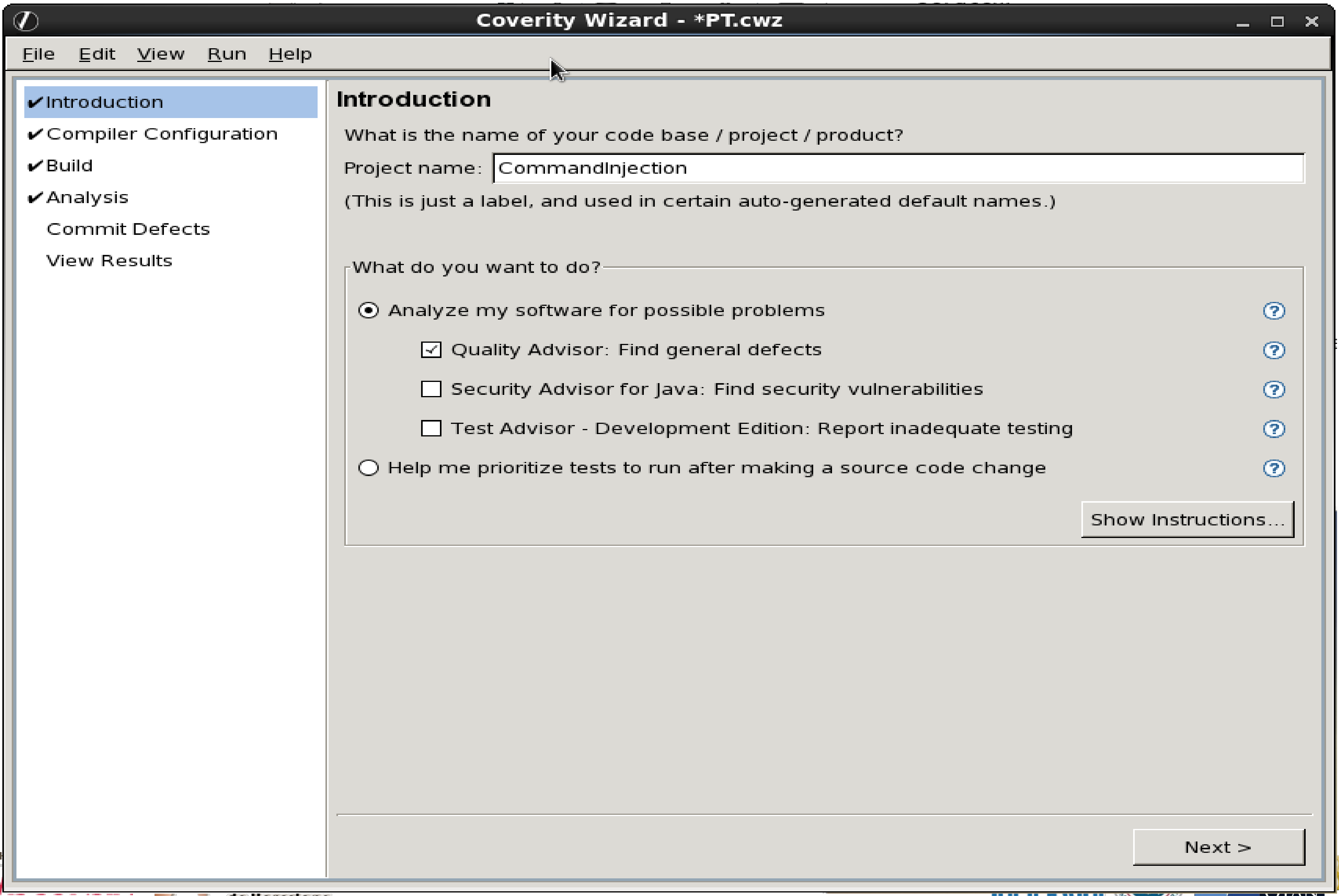
- Generate a configuration for the compiler:
  `cov-configure --gcc`
- Build the intermediate representation of the source code:
  `cov-build --dir <intermediate-dir> make`
- `cov-analyze --dir <intermediate-dir>`
- Check the checkers included by `cov-analize`:
  `cov-analyze --list-checkers`
- Read and interact with the analysis results.
- Graphic mode: `cov-wizard`

Coverity. OS Command Injection

```
$ cov-build --dir cov-comm-injection make
$ cov-analyze --dir cov-comm-injection
              --security
```

- 1 defect found.
- 1 true positive: **It detects the command injection.**

Coverity. OS Command Injection



Coverity Wizard - *PT.cwz

File Edit View Run Help

- ✓ Introduction
- ✓ Compiler Configuration
- ✓ Build
- ✓ Analysis
- Commit Defects
- View Results

Introduction

What is the name of your code base / project / product?

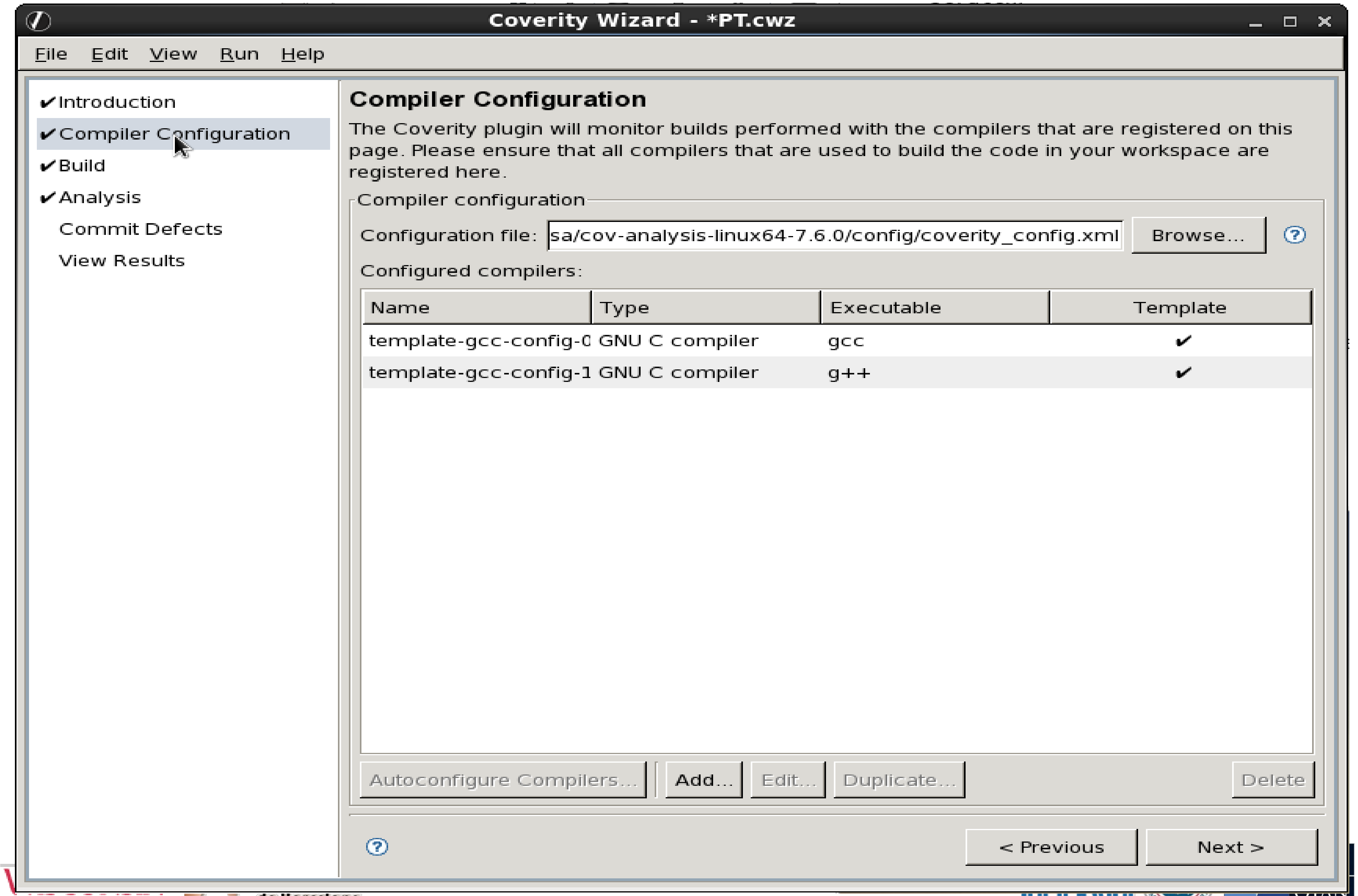
Project name: CommandInjection

(This is just a label, and used in certain auto-generated default names.)

What do you want to do?

- [✓] Analyze my software for possible problems
  - [✓] Quality Advisor: Find general defects
  - [ ] Security Advisor for Java: Find security vulnerabilities
  - [ ] Test Advisor - Development Edition: Report inadequate testing
- [ ] Help me prioritize tests to run after making a source code change

Coverity. OS Command Injection



Coverity Wizard - *PT.cwz

File Edit View Run Help

✔ Introduction
✔ Compiler Configuration
✔ Build
✔ Analysis
Commit Defects
View Results

Compiler Configuration

The Coverity plugin will monitor builds performed with the compilers that are registered on this page. Please ensure that all compilers that are used to build the code in your workspace are registered here.

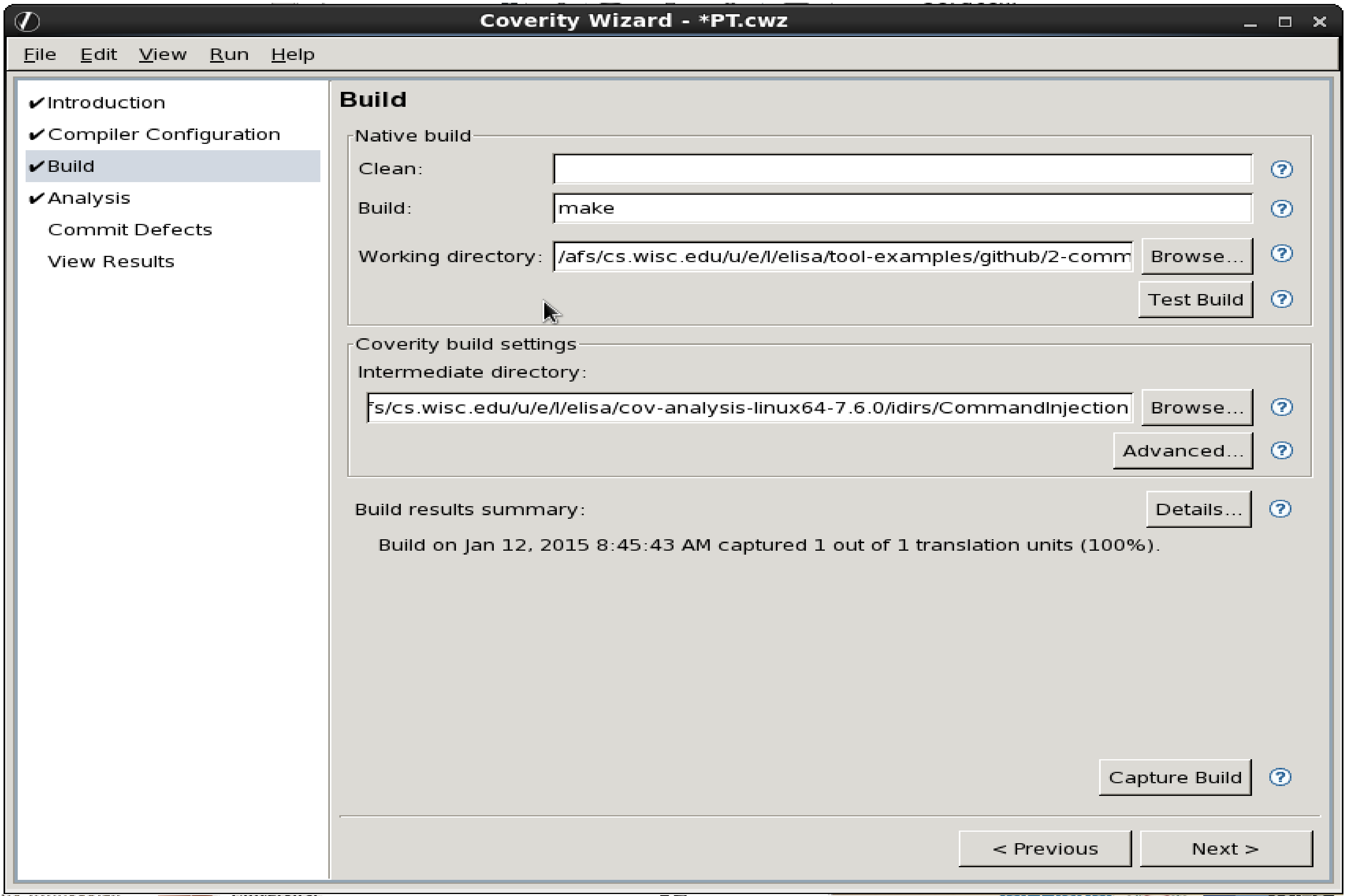
Compiler configuration

Configuration file: sa/cov-analysis-linux64-7.6.0/config/coverity_config.xml

Configured compilers:

Name	Type	Executable	Template
template-gcc-config-0	GNU C compiler	gcc	✔
template-gcc-config-1	GNU C compiler	g++	✔

Coverity. OS Command Injection



Coverity Wizard - *PT.cwz

File Edit View Run Help

✔ Introduction
✔ Compiler Configuration
✔ Build
✔ Analysis
Commit Defects
View Results

Build

Native build

Clean:

Build: make

Working directory: /afs/cs.wisc.edu/u/e/l/elisa/tool-examples/github/2-comm

Coverity build settings

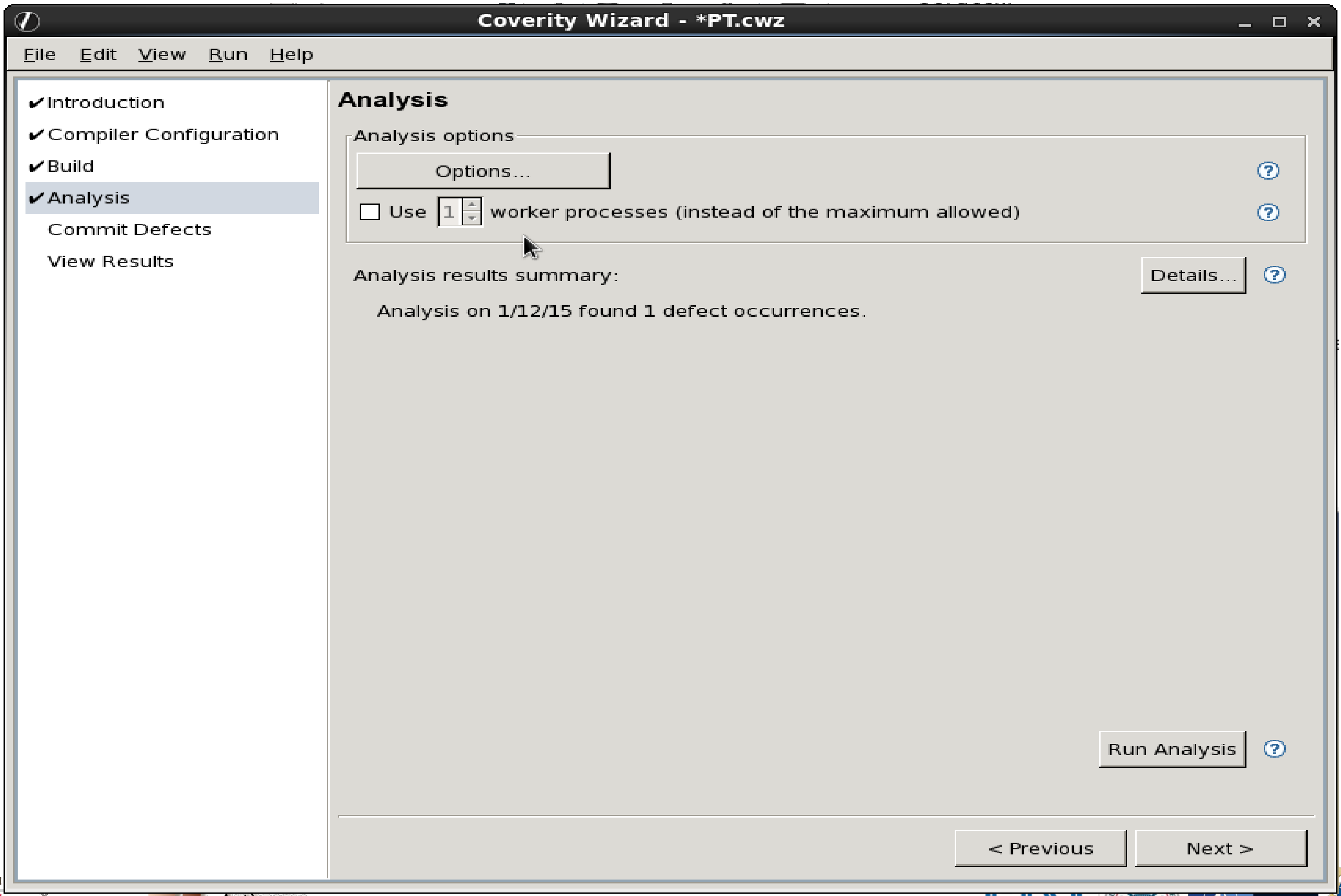
Intermediate directory:

s/cs.wisc.edu/u/e/l/elisa/cov-analysis-linux64-7.6.0/idirs/CommandInjection

Build results summary:

Build on Jan 12, 2015 8:45:43 AM captured 1 out of 1 translation units (100%).

Coverity. OS Command Injection



Analysis

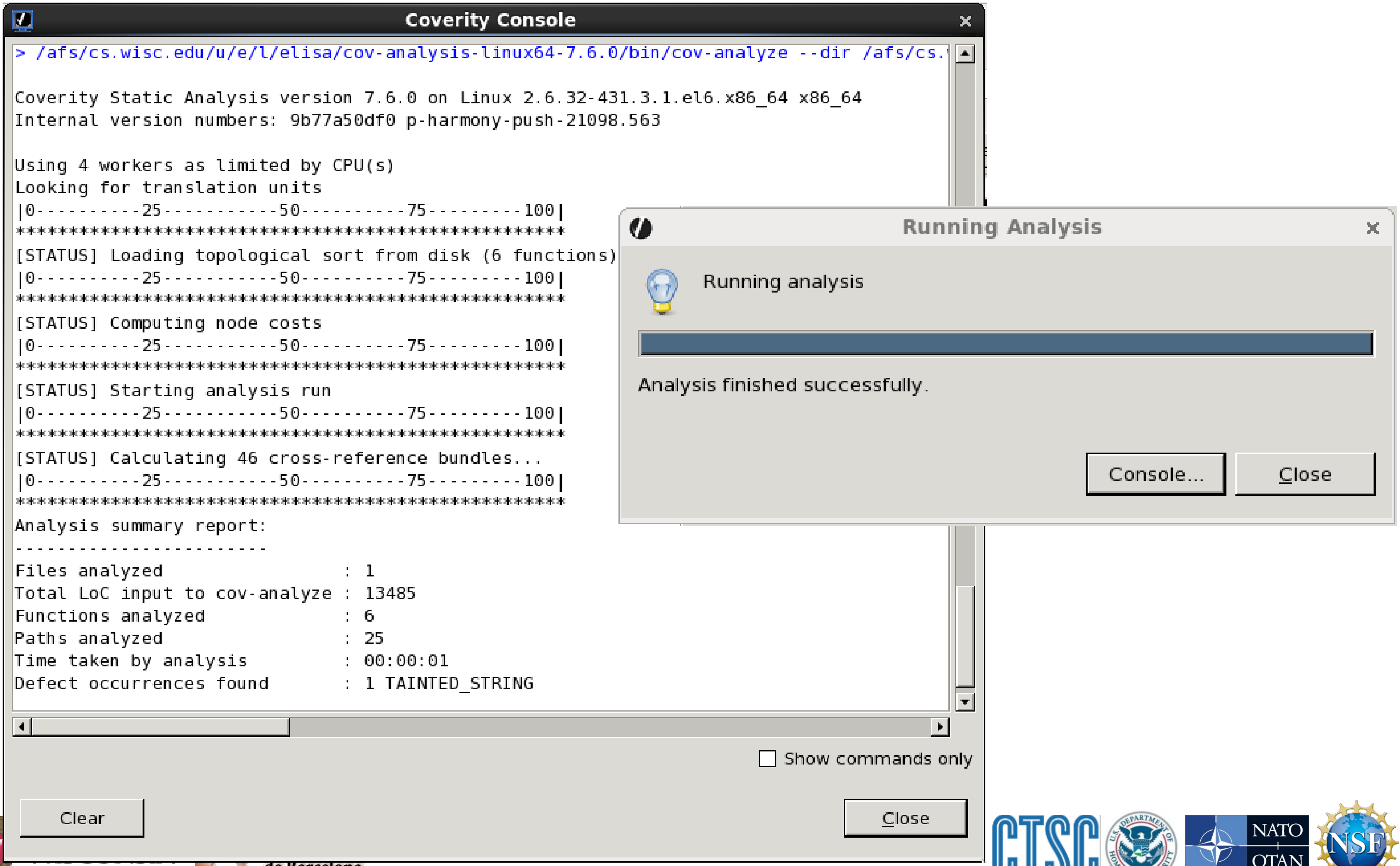
Analysis options

☐ Use 1 worker processes (instead of the maximum allowed)

Analysis results summary:

Analysis on 1/12/15 found 1 defect occurrences.

Coverity. OS Command Injection



Coverity Console

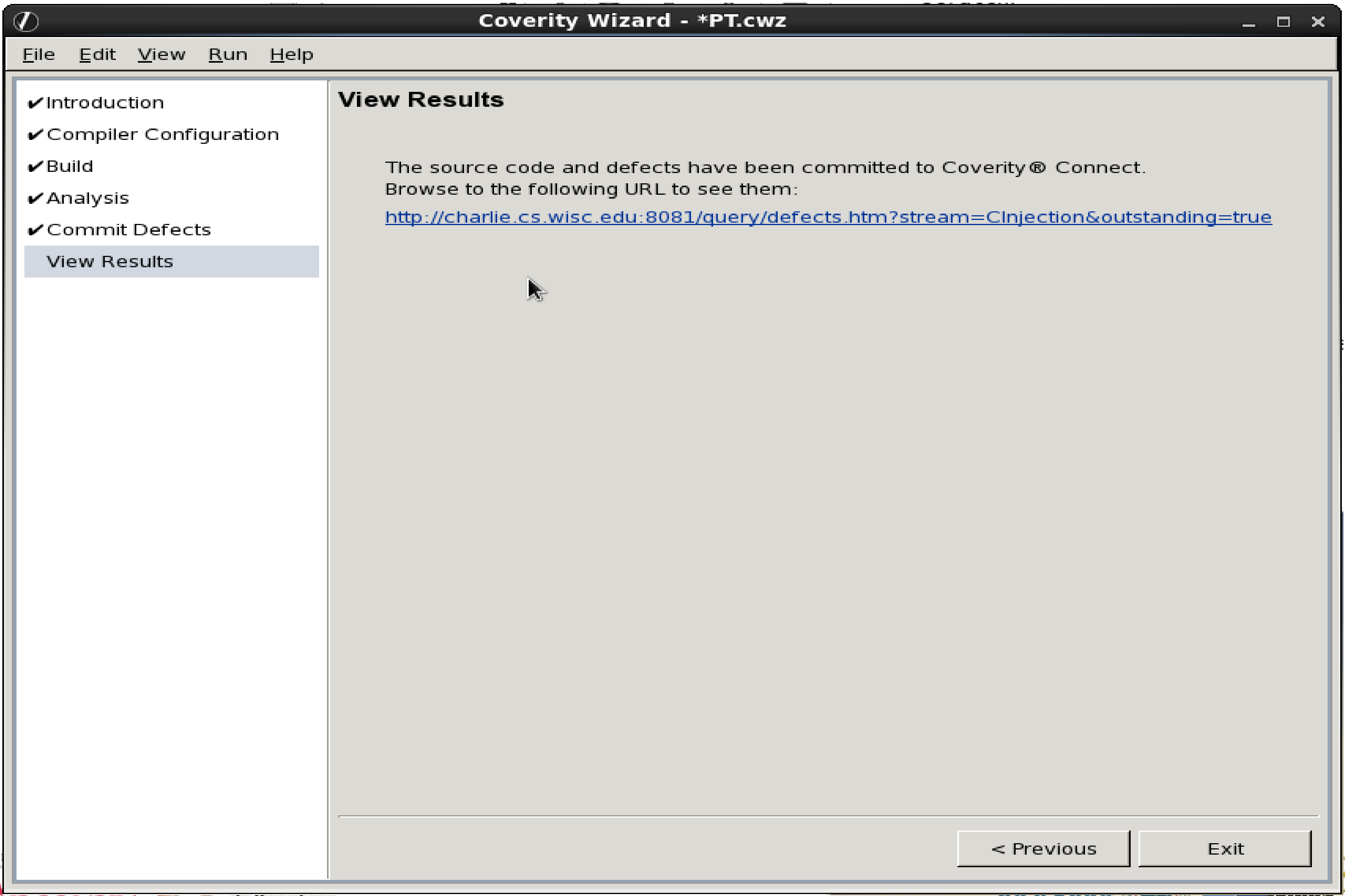
```
> /afs/cs.wisc.edu/u/e/l/elisa/cov-analysis-linux64-7.6.0/bin/cov-analyze --dir /afs/cs.

Coverity Static Analysis version 7.6.0 on Linux 2.6.32-431.3.1.el6.x86_64 x86_64
Internal version numbers: 9b77a50df0 p-harmony-push-21098.563

Using 4 workers as limited by CPU(s)
Looking for translation units
|0----------25-----------50----------75---------100|
****************************************************
[STATUS] Loading topological sort from disk (6 functions)
|0----------25-----------50----------75---------100|
****************************************************
[STATUS] Computing node costs
|0----------25-----------50----------75---------100|
****************************************************
[STATUS] Starting analysis run
|0----------25-----------50----------75---------100|
****************************************************
[STATUS] Calculating 46 cross-reference bundles...
|0----------25-----------50----------75---------100|
****************************************************
Analysis summary report:
------------------------
Files analyzed                 : 1
Total LoC input to cov-analyze : 13485
Functions analyzed             : 6
Paths analyzed                 : 25
Time taken by analysis         : 00:00:01
Defect occurrences found       : 1 TAINTED_STRING
```

Running Analysis

Running analysis

Analysis finished successfully.

Coverity. OS Command Injection

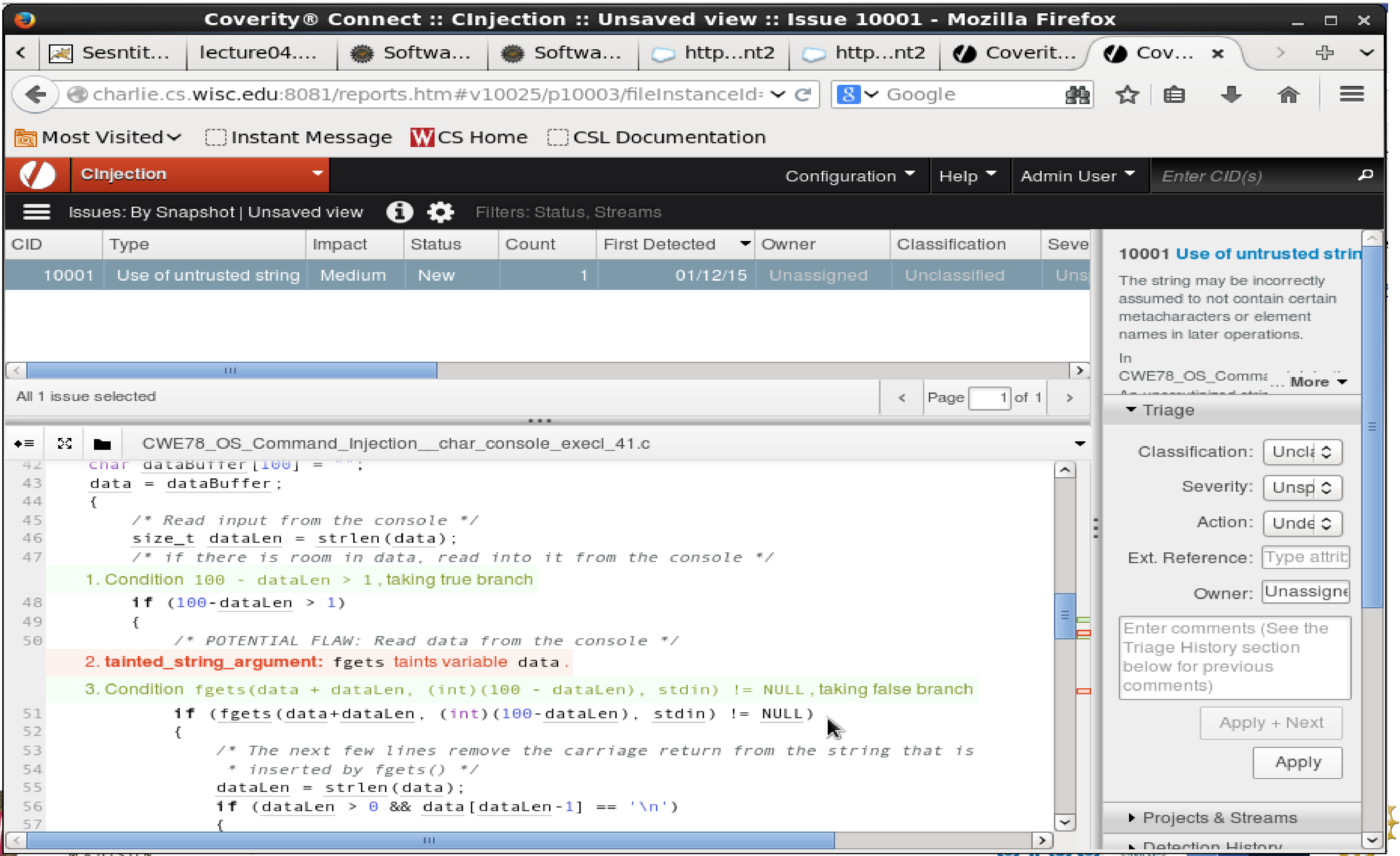


View Results

The source code and defects have been committed to Coverity® Connect.
Browse to the following URL to see them:

http://charlie.cs.wisc.edu:8081/query/defects.htm?stream=CInjection&outstanding=true

Coverity. OS Command Injection


Coverity® Connect :: CInjection :: Unsaved view :: Issue 10001 - Mozilla Firefox

charlie.cs.wisc.edu:8081/reports.htm#v10025/p10003/fileInstanceId=

Issues: By Snapshot | Unsaved view   Filters: Status, Streams

CID	Type	Impact	Status	Count	First Detected	Owner	Classification	Seve
10001	Use of untrusted string	Medium	New	1	01/12/15	Unassigned	Unclassified	Uns

All 1 issue selected

CWE78_OS_Command_Injection__char_console_execl_41.c

```
    char dataBuffer[100] = "";
    data = dataBuffer;
    {
        /* Read input from the console */
        size_t dataLen = strlen(data);
        /* if there is room in data, read into it from the console */
    1. Condition 100 - dataLen > 1, taking true branch
        if (100-dataLen > 1)
        {
            /* POTENTIAL FLAW: Read data from the console */
    2. tainted_string_argument: fgets taints variable data.
    3. Condition fgets(data + dataLen, (int)(100 - dataLen), stdin) != NULL, taking false branch
            if (fgets(data+dataLen, (int)(100-dataLen), stdin) != NULL)
            {
                /* The next few lines remove the carriage return from the string that is
                 * inserted by fgets() */
                dataLen = strlen(data);
                if (dataLen > 0 && data[dataLen-1] == '\n')
                {
```

10001 Use of untrusted strin

The string may be incorrectly assumed to not contain certain metacharacters or element names in later operations.

In CWE78_OS_Comma

Triage

Classification: Uncla

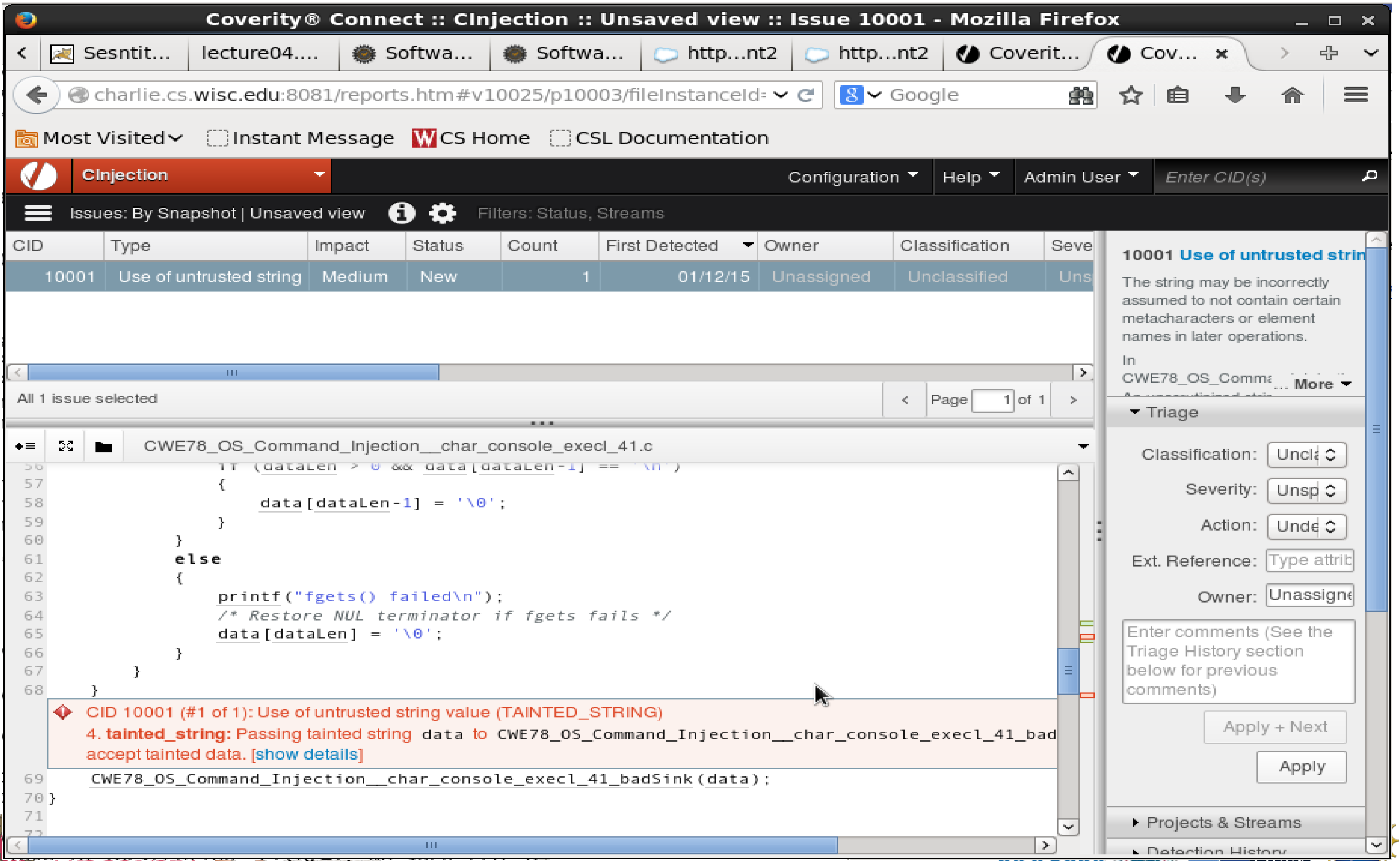
Severity: Unsp

Action: Unde

Ext. Reference: Type attrib

Owner: Unassigne

Coverity. OS Command Injection


Coverity® Connect :: CInjection :: Unsaved view :: Issue 10001 - Mozilla Firefox
Sesntit... lecture04.... Softwa... Softwa... http...nt2 http...nt2 Coverit... Cov...
charlie.cs.wisc.edu:8081/reports.htm#v10025/p10003/fileInstanceId=
Google
Most Visited
Instant Message
CS Home
CSL Documentation
CInjection
Configuration
Help
Admin User
Enter CID(s)
Issues: By Snapshot | Unsaved view
Filters: Status, Streams

CID	Type	Impact	Status	Count	First Detected	Owner	Classification	Seve
10001	Use of untrusted string	Medium	New	1	01/12/15	Unassigned	Unclassified	Uns

All 1 issue selected

Page 1 of 1

CWE78_OS_Command_Injection__char_console_execl_41.c

```
                {
                    data[dataLen-1] = '\0';
                }
            }
            else
            {
                printf("fgets() failed\n");
                /* Restore NUL terminator if fgets fails */
                data[dataLen] = '\0';
            }
        }
    }
```

CID 10001 (#1 of 1): Use of untrusted string value (TAINTED_STRING)
4. **tainted_string:** Passing tainted string `data` to `CWE78_OS_Command_Injection__char_console_execl_41_bad` accept tainted data. [show details]

```
    CWE78_OS_Command_Injection__char_console_execl_41_badSink(data);
}
```

10001 Use of untrusted strin

The string may be incorrectly assumed to not contain certain metacharacters or element names in later operations.

In CWE78_OS_Comma... More

Triage

Classification: Uncl

Severity: Unsp

Action: Unde

Ext. Reference: Type attrib

Owner: Unassigne

Enter comments (See the Triage History section below for previous comments)

Apply + Next

Apply

Projects & Streams

Detection History