UBS

Yasca (Yet Another Source Code Analyzer)

Introduction

January 2008

Not for Public Use

Introduction

This presentation accompanies the Yasca User & Developer Guide and describes at a high level the purpose and benefit of Yasca.

Contents:

- What is Yasca?
- How does it work?
- How can it provide real value?
- Where to go from here?

What is Yasca?

Yasca is both an <u>engine</u> and a <u>framework</u> for conducting file analysis.

It was first designed to be a very basic source code analysis tool, but was extended using a "plugin" concept to allow for analyses for arbitrary file types.

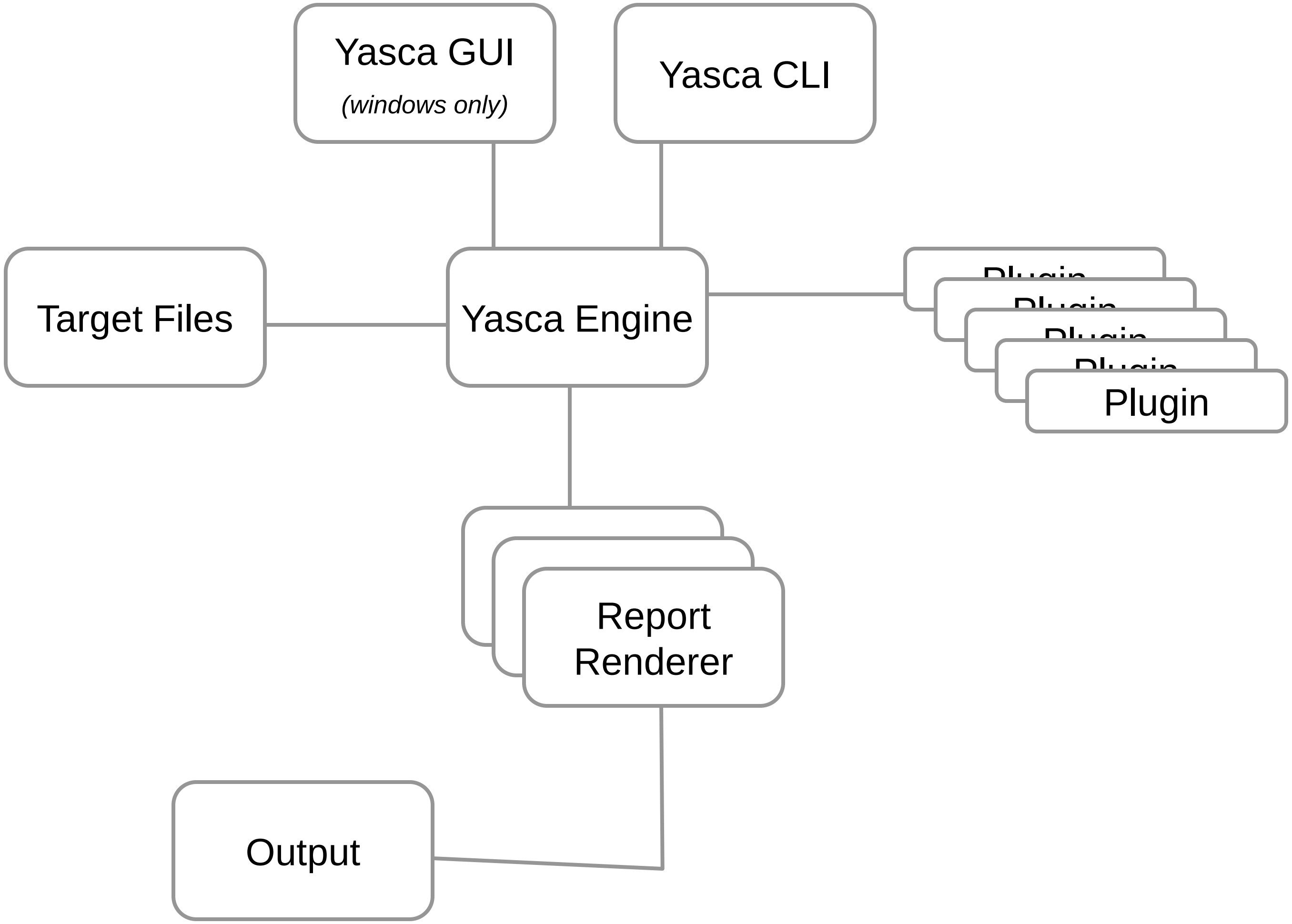
Yacsa can be thought of as a glorified grep script, plus the ability to call out to external programs for separate analysis.

<u>Architecture:</u>

- PHP v4 (via command line)
  - Compiled for Win32 so no dependency on PHP being installed.
  - WinBinder DLLs for Win32 GUI
- No installation necessary, but one is available
- Can integrate into the Explorer context menu (on Folders)

<u>Yasca is a tool for developers.</u>

How does it work?



Included Plugins

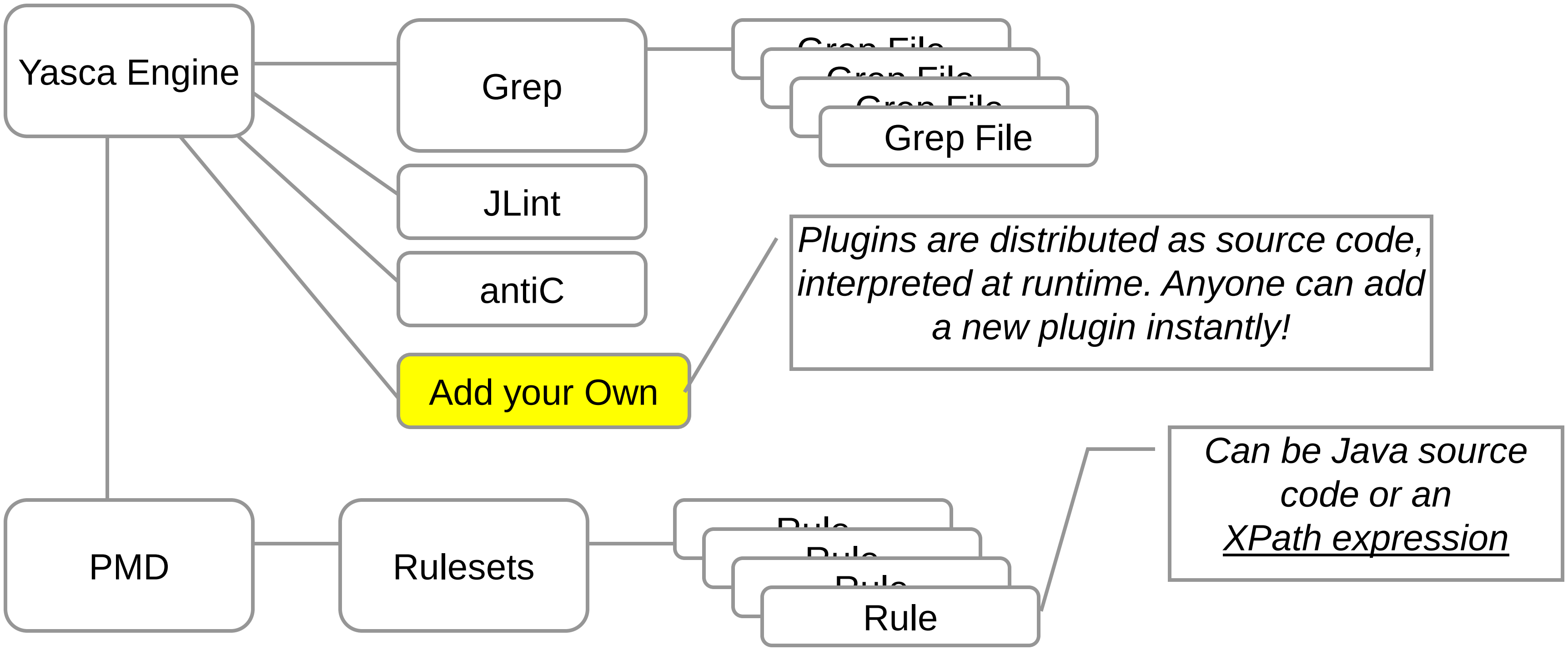
Yasca includes open-source components for some scanning:

- **JLint** (Java bytecode)
  - Scans for bugs, inconsistencies, and synchronization problems
- **antiC** (Java, C, C++, Objective C source code)
  - Scans for places where confusing language grammar can lead to bugs
- **PMD** (Java source code)
  - Scans for bugs, dead code, suboptimal code, and overly complicated expressions

Additional plugins were written to find additional things:

- Cross-Site Scripting
- Race Conditions
- Temporary Files
- Licensing Restrictions on code snippets found on the Internet
- Weak Authentication (and more…)

Plugin Methodology



Plugins are distributed as source code, interpreted at runtime. Anyone can add a new plugin instantly!

Can be Java source code or an *XPath expression*

Reporting

Canned Reports

- Rich HTML Output
- XML
- CSV

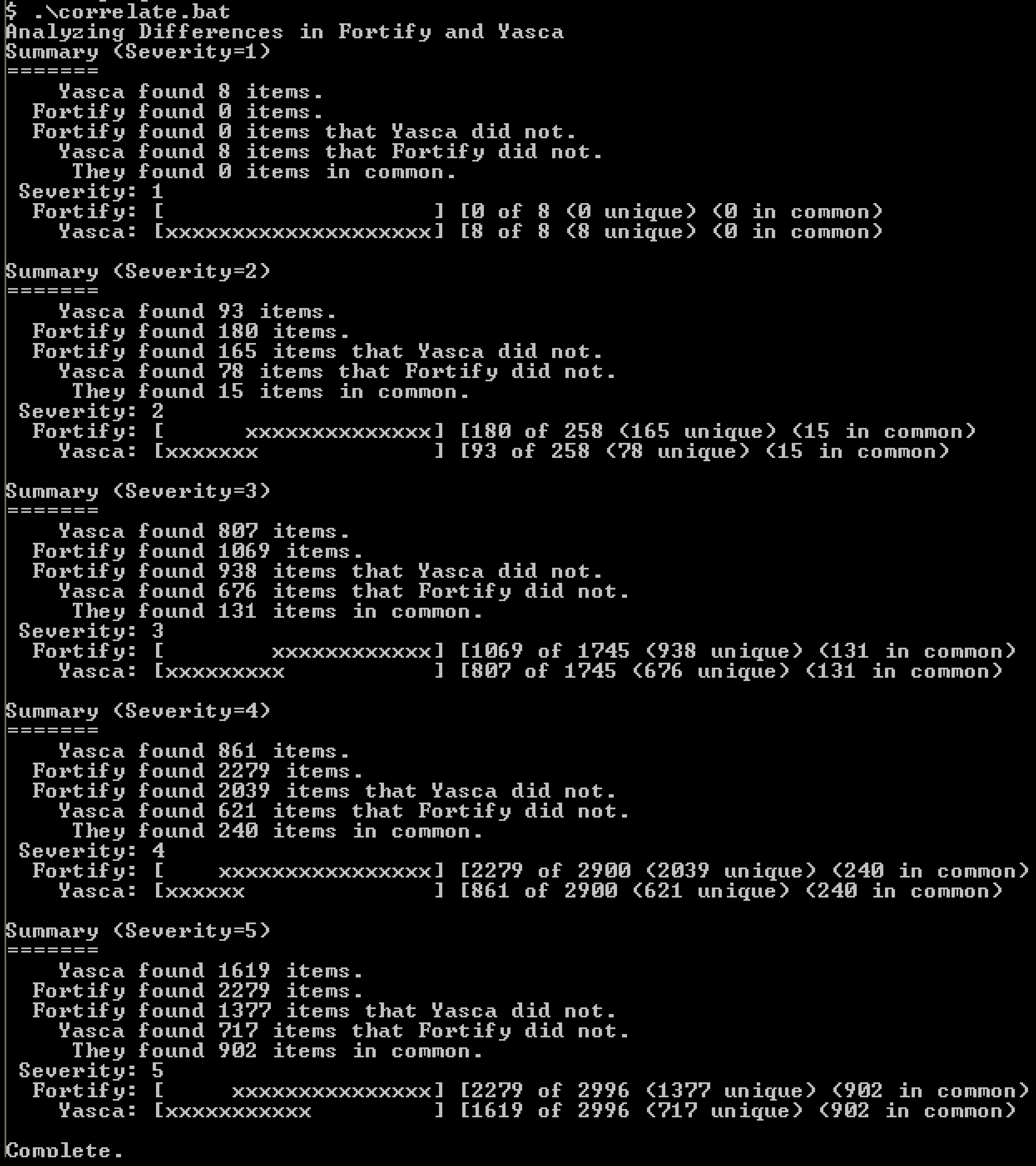
Yasca could be modified to allow for additional report generators to be included at runtime.

Accuracy

This will be answered during the beta phase.

But…for a sample application (KDD):

	= Critical	>= High	>= Medium	>= Low	>= Informational
Yasca (total)	8	93	807	861	1619
Fortify (total)	0	180	1069	2279	2279
Fortify (only)	0	165	938	2039	1377
Yasca (only)	8	78	676	621	717
Both	0	15	131	240	902



```
$ .\correlate.bat
Analyzing Differences in Fortify and Yasca
Summary (Severity=1)
=======
    Yasca found 8 items.
  Fortify found 0 items.
  Fortify found 0 items that Yasca did not.
    Yasca found 8 items that Fortify did not.
     They found 0 items in common.
 Severity: 1
  Fortify: [                    ] [0 of 8 (0 unique) (0 in common)
    Yasca: [xxxxxxxxxxxxxxxxxxxx] [8 of 8 (8 unique) (0 in common)

Summary (Severity=2)
=======
    Yasca found 93 items.
  Fortify found 180 items.
  Fortify found 165 items that Yasca did not.
    Yasca found 78 items that Fortify did not.
     They found 15 items in common.
 Severity: 2
  Fortify: [      xxxxxxxxxxxxxx] [180 of 258 (165 unique) (15 in common)
    Yasca: [xxxxxxx             ] [93 of 258 (78 unique) (15 in common)

Summary (Severity=3)
=======
    Yasca found 807 items.
  Fortify found 1069 items.
  Fortify found 938 items that Yasca did not.
    Yasca found 676 items that Fortify did not.
     They found 131 items in common.
 Severity: 3
  Fortify: [        xxxxxxxxxxxx] [1069 of 1745 (938 unique) (131 in common)
    Yasca: [xxxxxxxxx           ] [807 of 1745 (676 unique) (131 in common)

Summary (Severity=4)
=======
    Yasca found 861 items.
  Fortify found 2279 items.
  Fortify found 2039 items that Yasca did not.
    Yasca found 621 items that Fortify did not.
     They found 240 items in common.
 Severity: 4
  Fortify: [    xxxxxxxxxxxxxxxx] [2279 of 2900 (2039 unique) (240 in common)
    Yasca: [xxxxxx              ] [861 of 2900 (621 unique) (240 in common)

Summary (Severity=5)
=======
    Yasca found 1619 items.
  Fortify found 2279 items.
  Fortify found 1377 items that Yasca did not.
    Yasca found 717 items that Fortify did not.
     They found 902 items in common.
 Severity: 5
  Fortify: [     xxxxxxxxxxxxxxx] [2279 of 2996 (1377 unique) (902 in common)
    Yasca: [xxxxxxxxxxx         ] [1619 of 2996 (717 unique) (902 in common)

Complete.
```

Accuracy

So what does Fortify do that Yasca cannot (yet)?

- Null Pointer References:

```
Foo x = getFoo();
x = doSomething(x);

x.quux();      ← No check against null
```

- Lack of Validation
  - Yasca can do some of it
    - Some flavors of XSS
    - Some types of SQL Injection
    - Intra-method taint propogation (in progress)
    - Inter-method taint propogation (maybe in Q2)

- Plus all of the nice management stuff.

Future Improvements

User Interface:

- Prettier interface
- RAD plugin?

Plugins:

- Additional plugins written as things are discovered or requested
  - Involve ISD in the creation of these new plugins (message board!)
- Integration with other open source tools
  - Dozens of candidates – **focus on security**
- Data Flow Analysis