HoneySpider Network 2.0

detecting client-side attacks the easy way

Paweł Pawliński

CERT Polska / NASK

24th Annual FIRST Conference
21 June 2012

Outline

Outline

Origins of HSN 2.0

- Joint project
  - CERT Polska
  - NCSC-NL (GOVCERT.NL)
- Started in 2011
- Successor to HoneySpider Network version 1.x
  - used in production by CERTs
  - we gained experience in scanning web pages automatically

Project goals

- Detect attacks on client applications
  - web pages
  - files
- Apply multiple analyses
  - PDF, SWF, JavaScript, . . .
  - low and high interaction honeypots
- Configurable (processing details)
- Scalable (crawling)
- Open architecture

Project goals

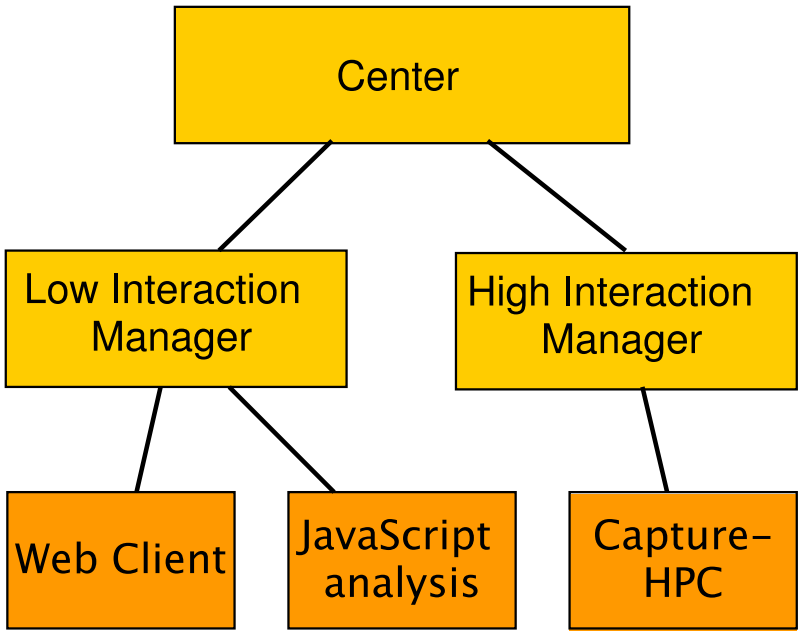
- Detect attacks on client applications
  - web pages
  - files
- Apply multiple analyses
  - PDF, SWF, JavaScript, . . .
  - low and high interaction honeypots
- Configurable (processing details)
- Scalable (crawling)
- Open architecture

version 1

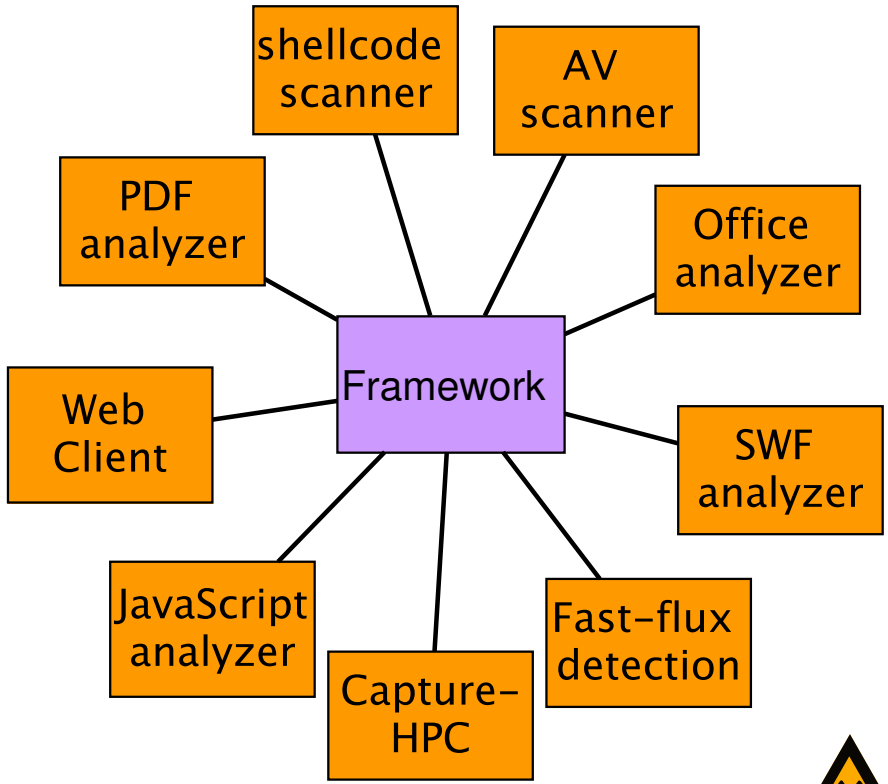
Outline

HSN: 1.x vs 2.0

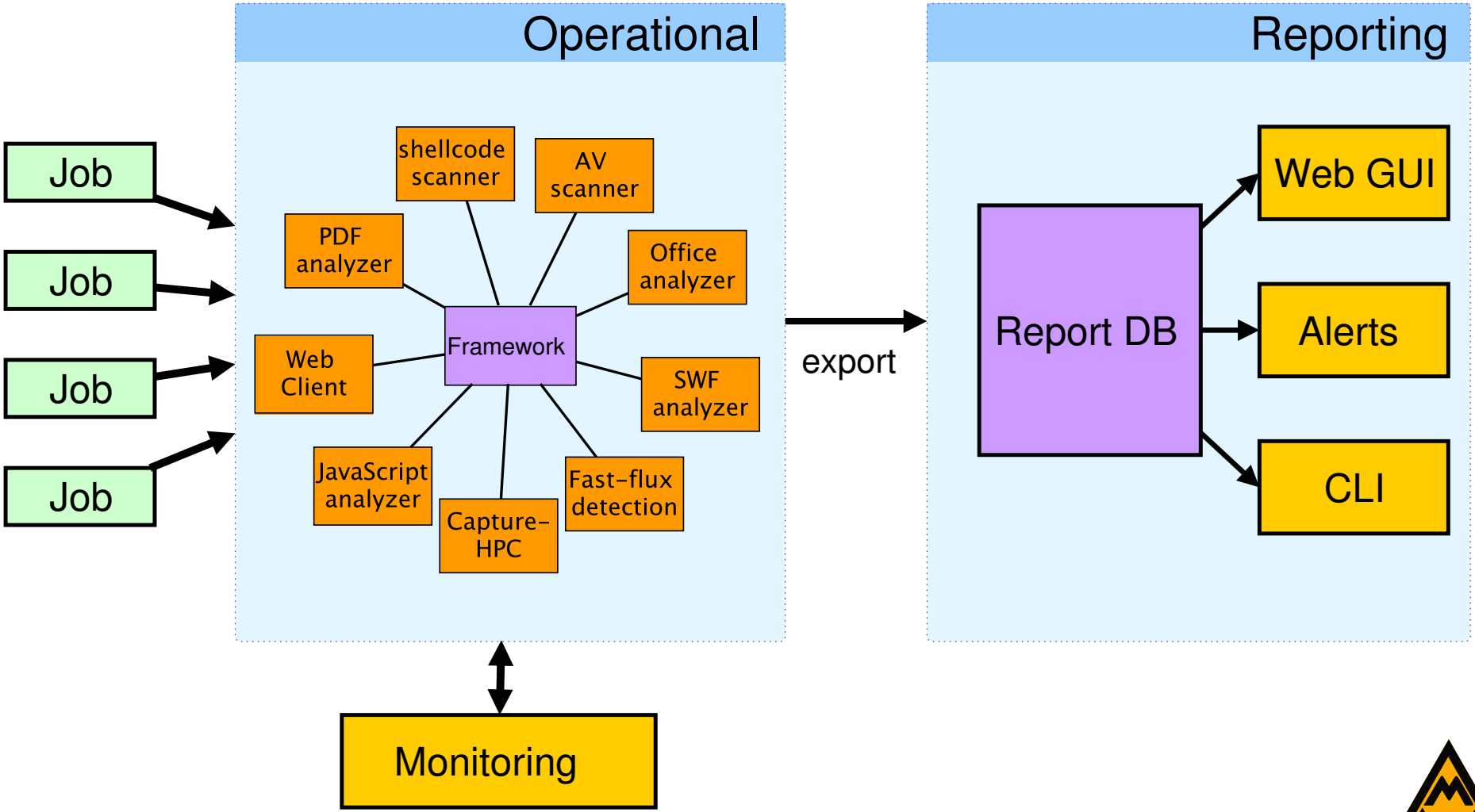
1.x



2.0



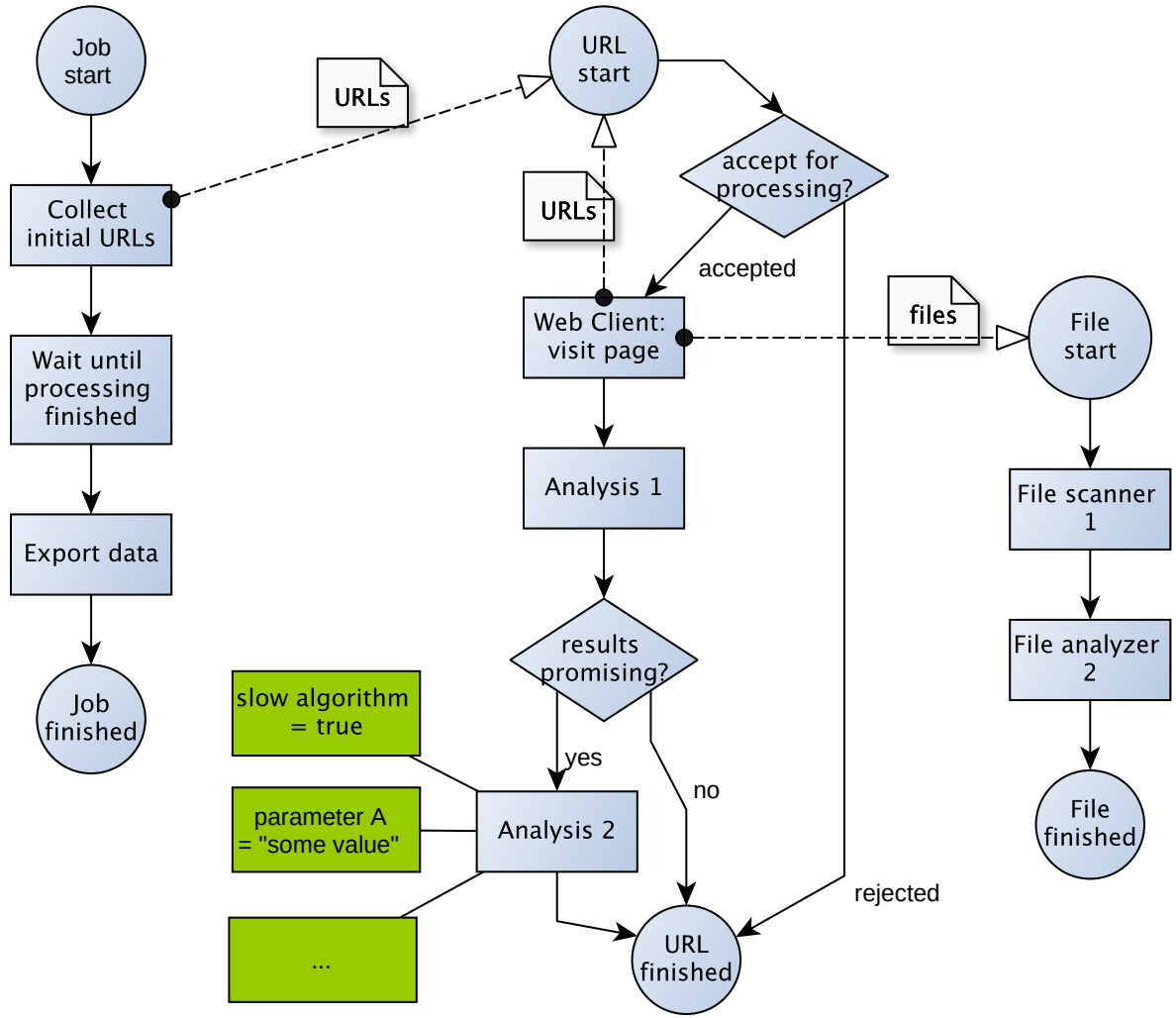
Architecture overview



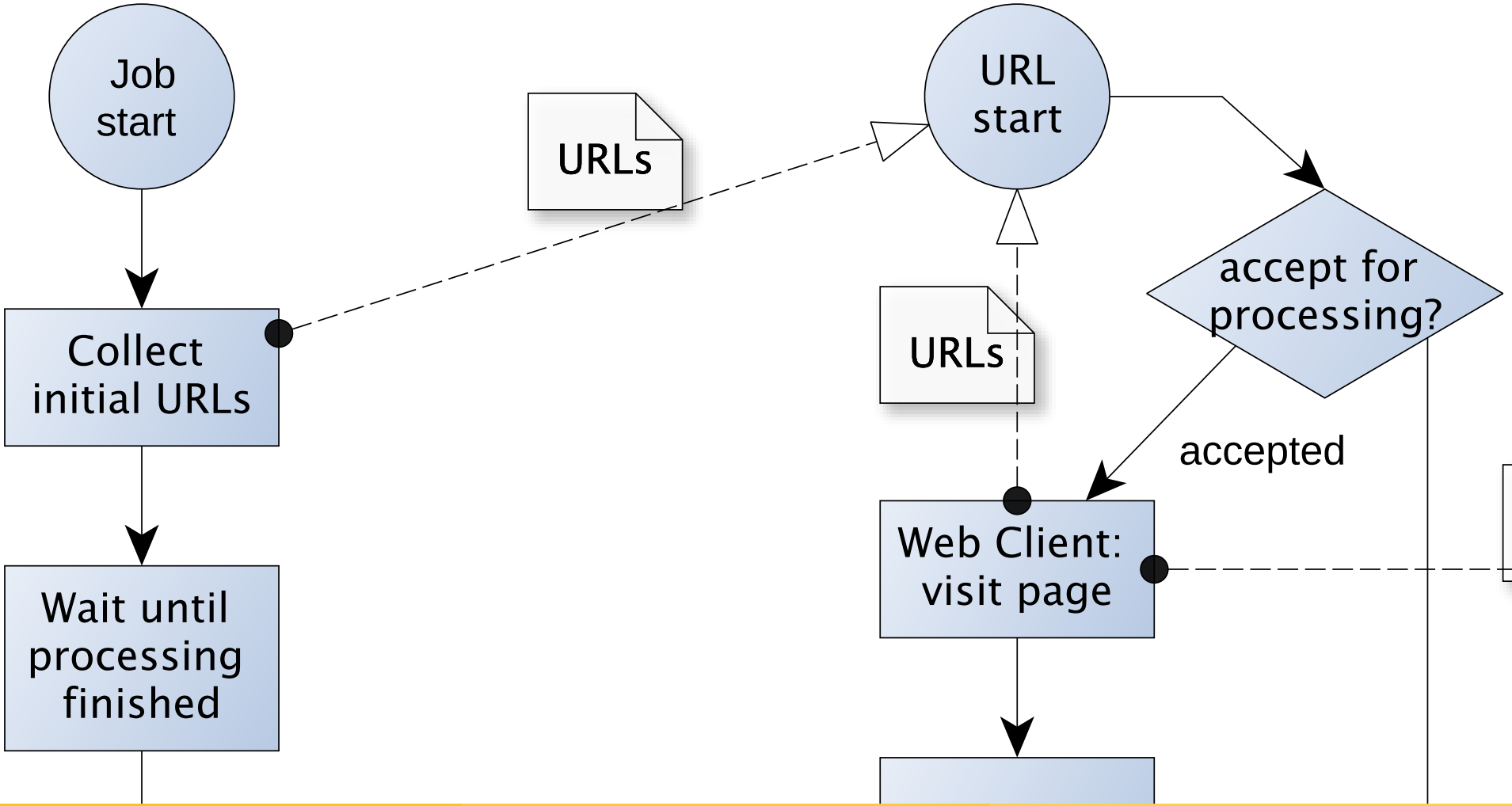
Technical foundations

- Network communication
  - Advanced Message Queueing Protocol
  - Google Protocol Buffers
- Storage
  - CouchDB
  - JSON documents
  - operational data + flexible mapping $\rightarrow$ persistent reports
- Programming languages
  - Java
  - Python
  - (C++)

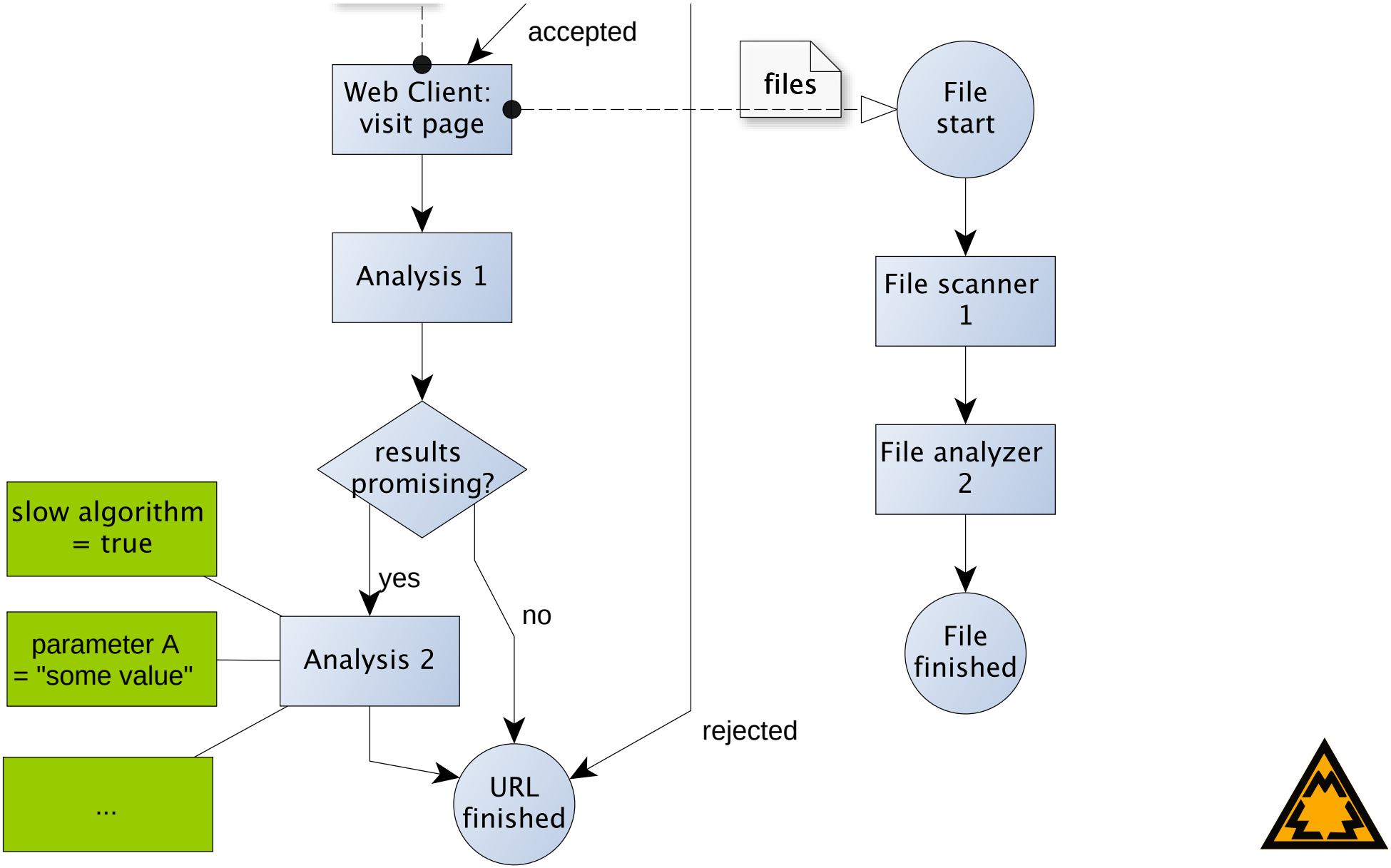
Sample workflow



Sample workflow



Sample workflow



Outline

Web client emulators

- HtmlUnit-based custom browser emulator
  - implemented in Java
  - uses Rhino engine
  - complete control over all behaviors (requests, redirects, frames)
  - link extraction
- Thug (low interaction honeypot)
  - implemented in Python
  - uses V8 engine
  - less control
  - detects common attacks
- These are not crawlers!

Analyzers

- Static JavaScript analyzer
  - port from version 1
  - n-grams + Bayes classifier
- SWF analyzer (NASK)
- Shellcode detection (scdbg)
- Cuckoo Sandbox
- Capture-HPC
  - high-interaction honeypot
  - used in HSN 1.x
  - new features and stability fixes

Utilities

- Feeder
  - file with URLs
  - search engine results
  - ...
- URL normalizer
- Reporter (persistent data)

Razorback: short introduction

- Modular IDS
- Data acquisition decoupled from offline analyses
- Dispatcher: routes data
- Nuggets (services)
  - collection (Snort, SMTP, . . .)
  - analyzers
  - enrichment (DNS, . . .)
- SQL database
- GUI

Razorback: short introduction

- Modular IDS
- Data acquisition decoupled from offline analyses
- Dispatcher: routes data
- Nuggets (services)
  - collection (Snort, SMTP, . . .)
  - analyzers
  - enrichment (DNS, . . .)
- SQL database
- GUI

Razorback analyzers

- Universal Razorback-to-HSN 2.0 adapter
- Only recompilation required, no changes to source code
- Tested nuggets:
  - swfScanner
  - pdfFox
  - clamavNugget
  - officeCat
  - virusTotal
  - archiveInflate

Extensibility

- Open communication protocol
- Well-defined data contract for each service
- Open technologies: AMQP, protobuf, REST, JSON
- Libraries provided for Java and Python

Outline

Demonstration

Outline

Current state of HSN 2.0

- All essential components implemented
  - framework
  - storage
  - web client
- Growing set of analyzers
- Functional web interface
- More tests and stabilization needed

Future plans

- Release as open source (soon!)
- Improve management of the whole system
- More analyzers
  - integrate existing tools
  - analysis of sandbox data
  - alternative web clients (high-interactive?)
  - looking for more ideas!

Thank you for your attention.

Questions?