System Integration F2020

Exercises

Henrik Lund Kramshoej
hlk@zencurity.com

February 3, 2020

Contents

Preface

This material is prepared for use in *System Integration F2020* and was prepared by Henrik Lund Kramshoej, Zencurity Aps. It describes the setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is available as PDF from kramse@Github
Look for system-integration-exercisesin the repo security-courses.

These exercises are expected to be performed in a training setting with network connected systems. The exercises use a number of tools which can be copied and reused after training. A lot is described about setting up your workstation in the repo

https://github.com/kramse/kramse-labs

Prerequisites

This material expect that participants have a working knowledge of internet from a user perspective. Basic concepts such as web site addresses, IP-addresses and email should be known as well.

Have fun and learn

Exercise content

Most exercises follow the same procedure and has the following content:

- **Objective:** What is the exercise about, the objective
- **Purpose:** What is to be the expected outcome and goal of doing this exercise
- **Suggested method:** suggest a way to get started
- **Hints:** one or more hints and tips or even description how to do the actual exercises
- **Solution:** one possible solution is specified
- **Discussion:** Further things to note about the exercises, things to remember and discuss

Please note that the method and contents are similar to real life scenarios and does not detail every step of doing the exercises. Entering commands directly from a book only teaches typing, while the exercises are designed to help you become able to learn and actually research solutions.

Exercise 1

Date Formats 15 min

Objective:
See an example of time parsing, and realize how difficult time can be in system integration.

Purpose:
System integration often works with different representations of the same data. Time and dates are one aspect we often meet. Realize how complex it is.

Suggested method:
Visit the web pages of an existing tool, Logstash we will use througout the course and a standard for time and dates.

Write down todays date on a piece of paper, each one does their own.

Then lookup ISO 8601
`https://en.wikipedia.org/wiki/ISO_8601`

I recommend looking at a specific system, used for processing computer logs: Logstash
`https://www.elastic.co/guide/en/logstash/current/plugins-filters-date.html`

Hints:
When you receive a date there are so many formats, that you need to be very specific how to interpret it.

Parsing dates is a complex task, best left for existing frameworks and functions.

If you decide to parse dates using your own code, then centralize it - so you can update it when you find bugs.

Solution:
When you have a

Discussion:
Make sure to visit the web page:
`https://infiniteundo.com/post/25326999628/falsehoods-programmers-believe-about-time`

Did you realize how complex time and computers are?

Then consider this software bug:

”No, you’re not crazy. Open Office can’t print on Tuesdays.”
https://bugs.launchpad.net/ubuntu/+source/file/+bug/248619

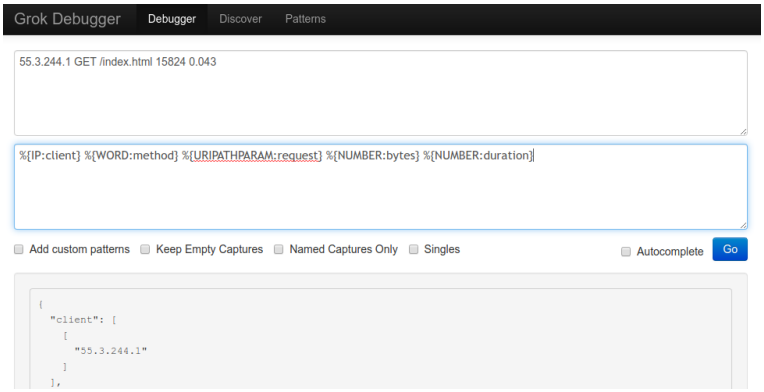
Linked from
https://www.reddit.com/r/linux/comments/9hdam/no_youre_not_crazy_open_office_cant_print_on/

Because a command `file` has an error in parsing data, files with PostScript data - print jobs with the text Tue - are interpreted as being Erlang files instead. This breaks the printing, on Tue(sdays).

We will go through this bug in detail together.

Exercise 2

Grok Debugger 15 min



Objective:
Try parsing dates using an existing system.

Purpose:
See how existing systems can support advanced parsing, without programming.

Suggested method:
Go to the web application Grok Debugger:
`https://grokdebug.herokuapp.com/`

Try entering data into the input field, and a parsing expression in the Pattern field.

Try the data from `https://www.elastic.co/guide/en/kibana/current/xpack-grokdebugger.html`

Hints:
The expression with greedy data is nice for matching a lot of text:
`%{GREEDYDATA:message}`

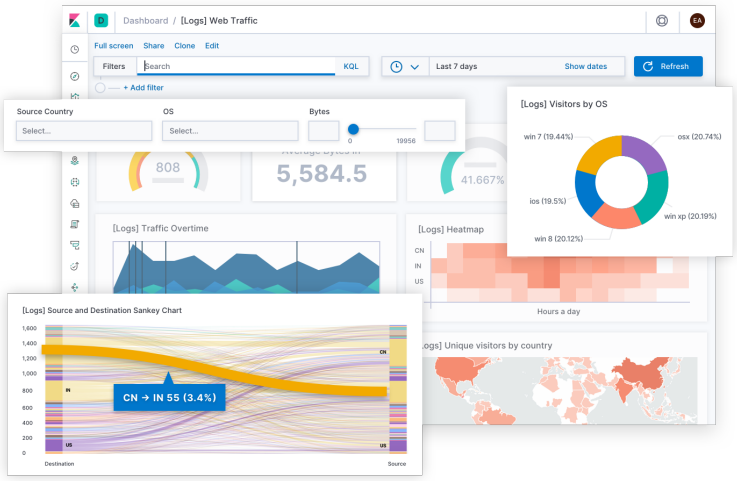
Try adding some text at the end of the input, and another part of the parsing with this.

Solution:
When you have parsed a line and seen it you are done.

Discussion:
The functionality Grok debugging is included in the tool Kibana from Elastic:
`https://www.elastic.co/guide/en/kibana/current/xpack-grokdebugger.html`

Exercise 3

Getting started with the Elastic Stack 15 min



Screenshot from `https://www.elastic.co/kibana`

Objective:
Get ready to start using Elasticsearch, read - but dont install.

Purpose:
We need some tools to demonstrate integration. Elasticsearch is a search engine and ocument store used in a lot of different systems, allowing cross application integration.

Suggested method:
Visit the web page for *Getting started with the Elastic Stack* :
`https://www.elastic.co/guide/en/elastic-stack-get-started/current/get-started-elastic-stack.html`

Read about the tools, and the steps needed for manual installation.

You dont need to install the tools currently, I recommend using Debian and Ansible for bringing up Elasticsearch. You are of course welcome to install, or try the Docker method.

Hints:
Elasticsearch is the name of the search engine and document store. Today Elastic Stack contains lots of different parts.

We will focus on these parts:

- Elasticsearch - the core engine
- Logstash - a tool for parsing logs and other data.
  https://www.elastic.co/logstash

  > "Logstash dynamically ingests, transforms, and ships your data regardless of format or complexity. Derive structure from unstructured data with grok, decipher geo coordinates from IP addresses, anonymize or exclude sensitive fields, and ease overall processing."

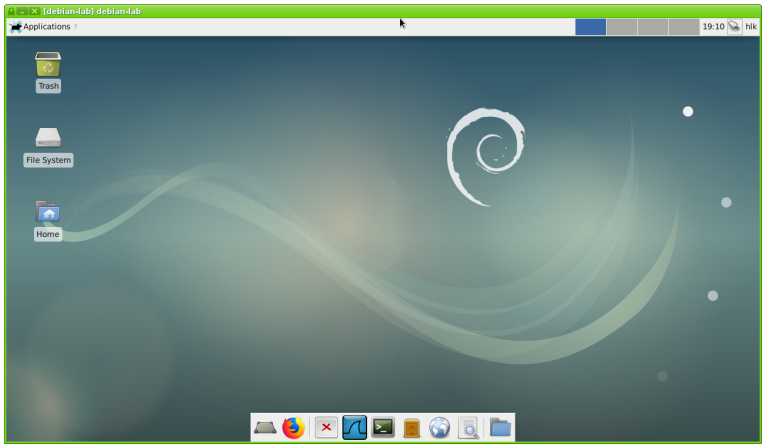
- Kibana - a web application for accessing and working with data in Elasticsearch
  https://www.elastic.co/kibana

Solution:
When you have browsed the page you are done.

Discussion:
You can read more about Elasticsearch at the wikipedia page:
https://en.wikipedia.org/wiki/Elasticsearch

Exercise 4

Bonus: Check your Debian VM 10 min



Objective:
Make sure your virtual Debian machine is in working order. We need a Debian 10 Linux for running a few extra tools during the course.

This is a bonus exercise - only one Debian is needed per team.

Purpose:
If your VM is not installed and updated we will run into trouble later.

Suggested method:
Go to https://github.com/kramse/kramse-labs/ Read the instructions for the setup of a Debian VM.

Solution:
When you have a updated virtualisation software and Debian Linux, then we are good. Create a snapshot of the server, so you can return to this, in case it breaks later.

Discussion:
Linux is free and everywhere. The tools we will run in this course are made for Unix, so they run great on Linux.

Even Microsoft has made their cloud Linux friendly, and post articles about creating Linux applications:
`https://docs.microsoft.com/en-us/azure/security/develop/`

