


# Differential IP Clustering For Vulnerability Detection Using Applied Machine Learning



Sebastian Bocquier

# Introduction to the Problem

- Machine learning as a study has been widely used and adopted since 1959 where the term was first coined by Arthur Samuel.
- In the recent year's, machine learning's accuracy and efficiency has increased dramatically, causing the field to rapidly gain in popularity. Large corporation examples include Google's self driving car technology and LeNet image recognition system.
- This is mainly due to the increase in processing power. This leads to models being adopted in almost every field, where efficiency over humans can be improved.
- The Information Security industry has been innovating in machine learning models for almost as long as its existence, however, this is only true for the Blue team.
- The Red team has severely lacked in useful machine learning implementations, creating a gap and unbalance in research.

# The Solution – Project Aims

- To help restore the balance in innovation by creating grounds for more research on the red team side of Pentesting.
- The increased demand for penetration testers justifies the need for more effective and advanced tools to conduct their security assessments.
- The aim was to design and develop a useful tool to be used during penetration tests or CTF events, in other words, Red Teaming activities.
- The application created uses common scanning tool outputs to provide a high level view of a network and its hosts. Highlighting the system differences in the hosts and clustering them, allowing a user to prioritise attack vectors.
- The application recommends hosts to be prioritised during a manual security assessment. This is based on hundreds of factors per host but prioritises vulnerability information with a fluid dynamic ratio.
- At its core, the application is a IP clusterer.

# The Application

- Command line application with optional GUI, portable and writing in Python 2.7.
- Takes input from Nmap, Nessus or both
- The primary, most beneficial, mode is using both, or “Dual” mode
- This will recommend attack vectors as the ‘most vulnerable’ machine on the network
- Within those modes there are three primary sub modes; **Manual**, **Assisted** and **Automatic**. These are explained in the *next slide*.
- The application outputs in Text form with several levels of verbosity, The optional graphing interface GUI changes depending on the selected modes.
- Designed with versatility and reliability in mind, allowing for full customization of algorithms due to the modularity and readability.

# Application primary modes

## ■ Manual:

- *Allows full control of clustering parameters, uses Gap Statistic or Elbow method if number of clusters is not selected.*
- *Choose from the following clustering algorithms*

**K-means Clustering(Default)**

**Agglomerative**

**DBscan**

## ■ Assisted

- *Same options as manual however, the user makes the decisions of which host should go in which cluster if borders are contested.*

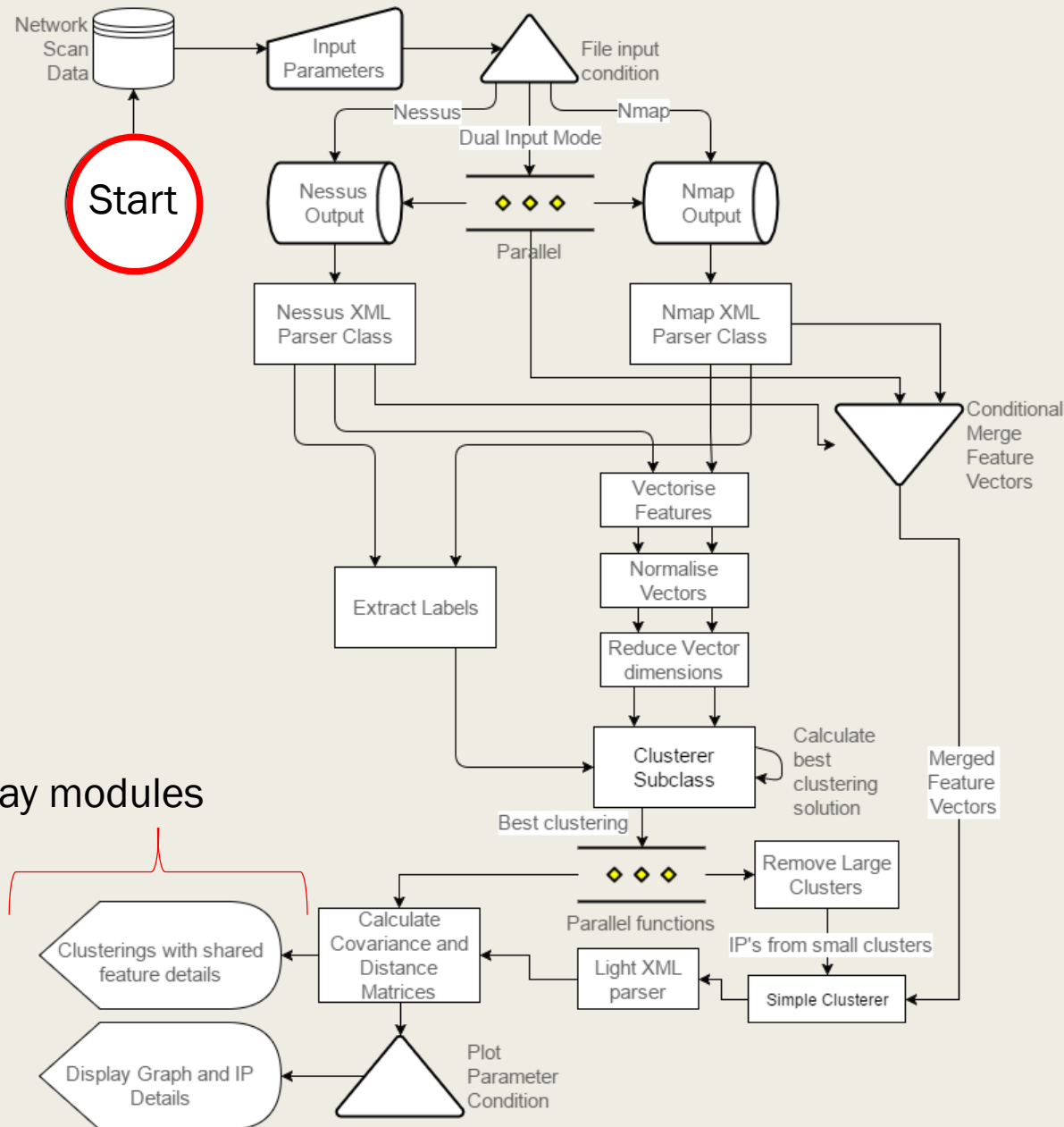
## ■ Automatic

- *Recommended for best results.*
- *Uses all clustering algorithms and amount of clusters. Detects best clustering using many methods such as gap statistic and hierarchical clustering.*

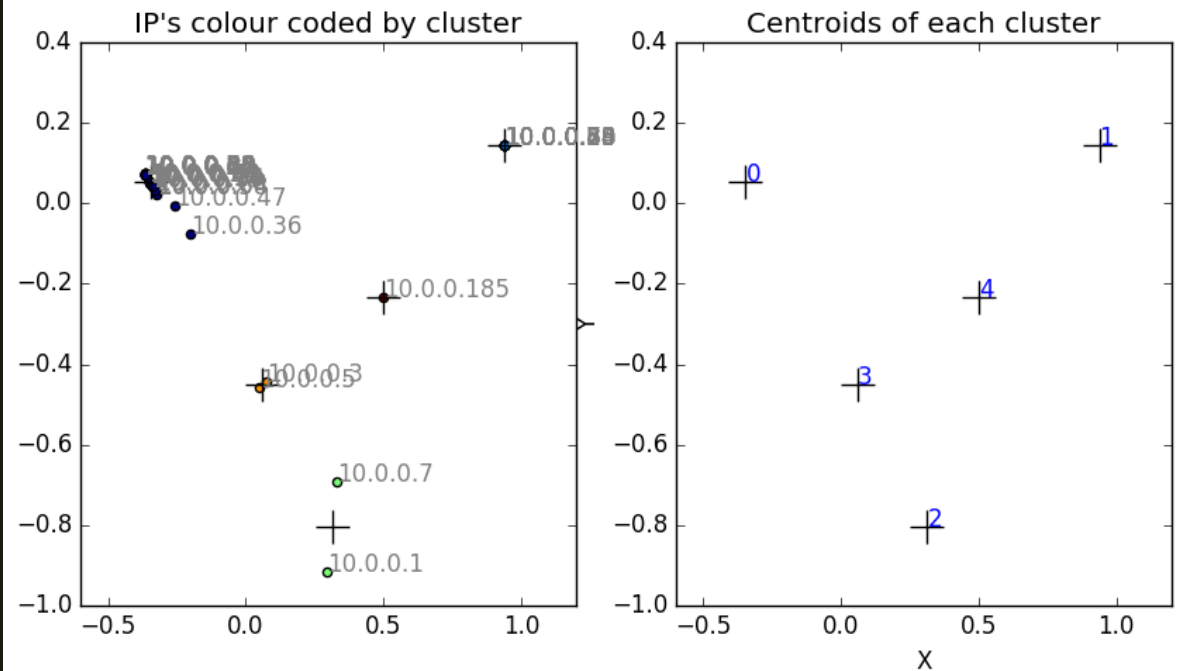
# Infrastructure

- This is the application dataflow infrastructure
- It does not include the complicated clustering and display modules.

## Display modules



# Results

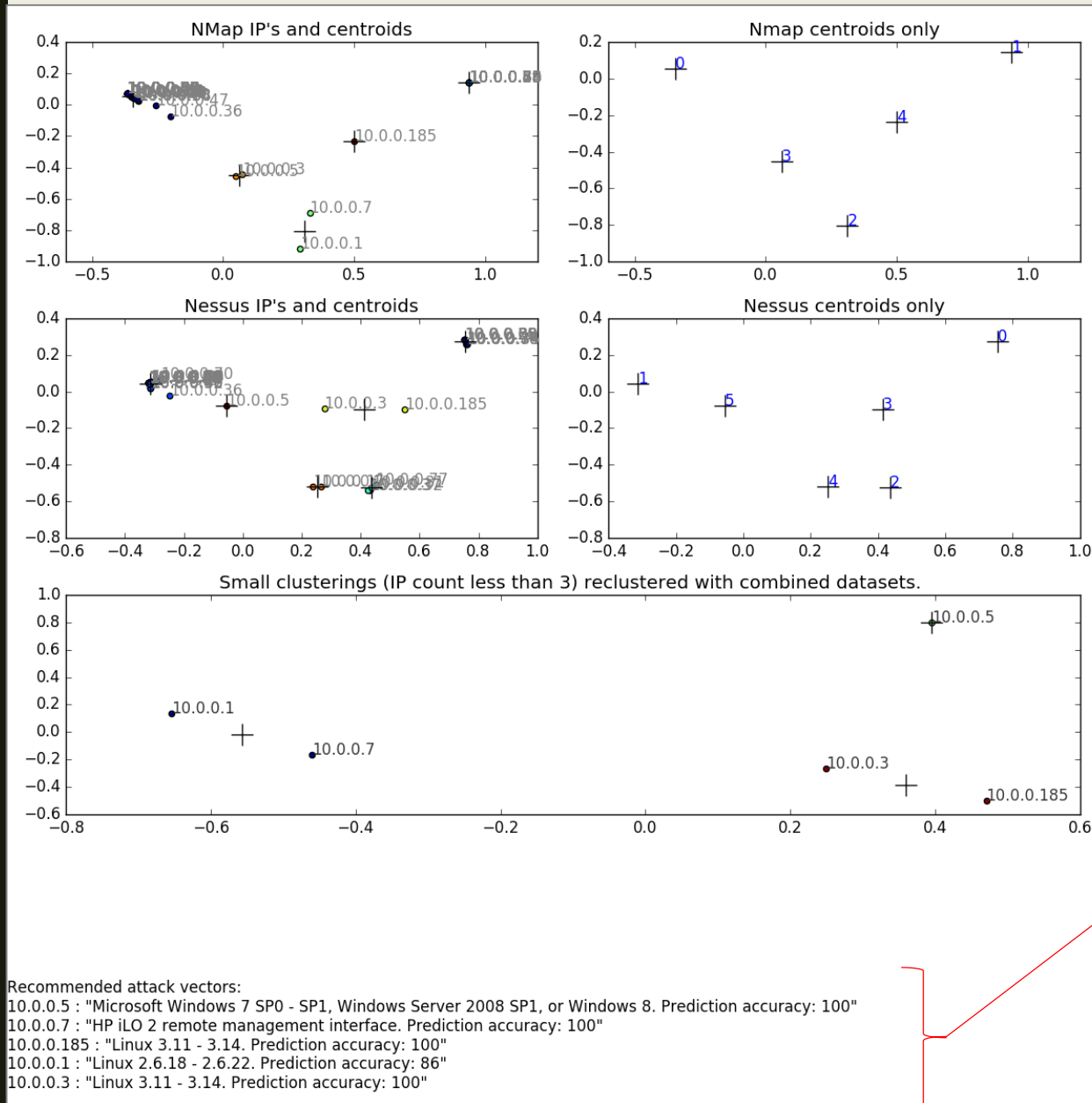


\*Using Hacklab 4511 dataset

- The full verbosity text output contains all the information the user could be interested in, such as:
  - *Individual cluster details including shared features*
  - *Average and per cluster Silhouette values, for accuracy measurement*
  - *Covariance and Distance matrices.*
- The optional graphing interface changes depending on the input mode.
- The **figure on the left** shows the graphing interface when using a single input mode with centroids enabled. In this case using Nessus input.
- The application upon completion, will write out two files. A .dot file of the primary clustering and a targets file containing selected recommended hosts.

# This is the Dual mode interface

- The dataset was created from the Hacklab computers in the room next door, 4511.
- The top four 'grid' of graphs are the single input type clusterings. With the right most consisting of solely the centroids.
- The bottom most graph shows a combined clustering as previously described.
- This incorporates both inputs and shows the most different hosts from clusters with less than 3 hosts (by default).
- The text information at the bottom of the interface displays the operating system information for these selected hosts along with the prediction accuracy.





# Future Development Recommendations

- **Reprogram the application with a lower level language such as C++**
  - *This will dramatically decrease execution time due to the heavy CPU requirements*
  - *Will lose code customisability*
- **Create an Executable GUI**
  - *The current GUI is executed via command line arguments and is limited to graphing.*
  - *Promotes ease of use for less technical users*
  - *Eliminates having to remember the many available parameters and options*
- **Implement dual mode with more algorithms other than K-means Clustering**
  - *Dual mode currently only implemented with K-means clustering due to limitations with Sklearn python library*
- **Eliminate problem of common vulnerabilities between the hosts**
  - *The issue: The application calculates the difference between hosts, therefore, is there is a wide spread vulnerability among all the hosts on the network, the algorithm will ignore it.*

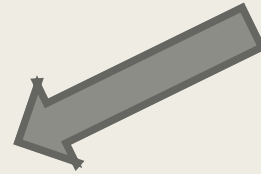
# Conclusion

- This thesis has created grounds for further research and development by proving the effectiveness of machine learning for penetration testing.
- This thesis shows that red teaming can benefit greatly from the use of tools which incorporate machine learning, and that there is a lack of these tools currently available.
- Further research into the use of machine learning for red teaming is required in order to bring the team up to the same level of innovation as found in the blue team.
- By clustering hosts on a network based off information retrieved from common security scanners, it is possible to identify and highlight the most differential hosts on the given network.



\*made with machine learning

funny dog meme



Just incase I dragged on for a bit too long

Thanks for listening!

*END*