Identifying IP Address Spoofing

Joseph Faulls (1447637) - MSci Computer Science

Introduction

Creating Internet Protocol packets with false source addresses, or IP address spoofing, is trivial to do and very hard to identify on the final host machine. This technique is used to exploit IP address based authentication systems and in several applications of Denial of Service (DoS) attacks, either to mask the true source of the attack, or to trigger a response to the spoofed victim source. By analysing IP packet header information, namely TTL, I want to try and identify whether the packet has an illegitimate source. If the header information differs greatly to what is expected, the packet will be dropped. This will prevent large streams of requests from fake sources being processed, a characteristic analogous to a DoS attack.

Problems to Solve

To create what an expected packet may look like for a given source IP, large quantities of data must be processed and a database will be generated, effectively 'mapping' the internet:

Problem 1 – Training Data

To learn about similarity between IP packets and to use supervised machine learning techniques to get more accurate and reliable results, large amounts of data is required. This will also be an everchanging system, learning as more and more data comes through. The problem of acquiring training data is easy to overlook. Sometimes, this is not a problem and the data can be readily available through the internet or through local sources (i.e. the university). However, if it proves incredibly challenging to get this, I may have to design and implement my own system to create this data. Perhaps by sending requests to many addresses and analysing the response.

Problem 2 – Analysing header information

The next step is to calculate how predictable the header information such as TTL, protocol, source address and packet length is, given a source address. I could calculate the conditional entropy as a measure of predictability. Then I can compare this to packets that I know are spoofed, to see if there is a significant difference.

Problem 3 – Clustering IP Addresses

Because a simple one-to-on mapping of IP addresses to expected packet is too specific and would not work for new IP addresses, a mechanism to cluster IP addresses to similar packet properties must be implemented. An initial thought is to cluster based on physical location or similarity of IP address.

Problem 4 – Classifying

Machine learning classification techniques, namely a naïve Bayes classifier, will be used to differentiate legitimate sources from illegitimate ones. Statistical evaluation metrics such as K-fold cross validation will be the primary way of testing the accuracy of my classifier. This problem is heavily established, so I should not run into too many problems.

Timeline

Giving dates to phases is unrealistic at this current time. Instead, I have outlined a rough plan:

Goals by Christmas:

- Have enough training data available
- Measure the predictability of information in the IP header (mainly TTL)

- Have started on a system or have a working theory as to how to cluster IP addresses without loss of too much information.

To be done in the Spring Term:

- Build a working Bayes classifier
- Build working database that is constantly updated
- Test and evaluate the classifier
- Integrate the classifier to drop spoofed IP packet.

Fall-backs

Although I plan on using as much useful data as I can from the IP header, it may turn out this will increase the complexity and decrease the accuracy of my model. Therefore, if it gets out of hand, I will only be choosing to consider one piece of information, the Time To Live (TTL).

Integrating the classifier to demonstrate its practical use would be ideal, however, if I do not have enough time left at the end, simply demonstrating that a classifier is able to identify spoofed headers is enough. The practical implementation can be implied from this.

Clustering IP addresses may be left to the end, as this could prove an ugly task. If the database simply stored information for each IP address encountered, rather than clustering, the basic principle would work and I could have a working system without clustering.

Another feature that is currently planned but not detrimental to the project is concurrent learning as the system is running. Therefore, if time does not allow if – this shall be omitted.

Resources Required

No specific resources are required for this project. Internet traffic data from the university will be very helpful is available, but not required.