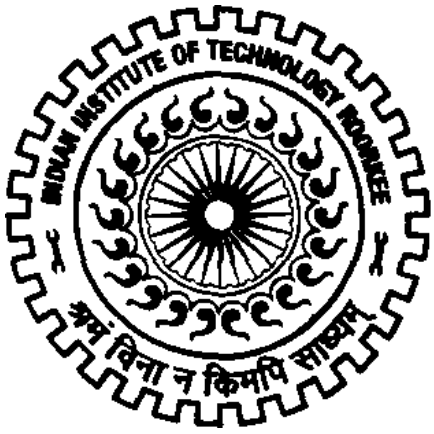
INDIAN INSTITUTE OF TECHNOLOGY

ROORKEE



SPRING SESSION 2014-15

Project Report

On

ANOMALY BASED INTRUSION DETECTION SYSTEM FOR CLOUD

Submitted to-

Dr. Sateesh Kumar Peddoju

Assistant Professor (CSD)

Submitted by-

Amala Thampi (14535003)

Geeta (14535016)

Nikita Jain (14535031)

Nishtha Behal (14535032)

Yashika Jain (14535054)

ABSTRACT

With the advent of Cloud Computing technology, sharing of resources among users over the Internet has increased. Along with it, various new security issues have also cropped up. In order to provide better security to the user of cloud environment against attacks it is necessary to provide an Intrusion Detection System (IDS) to detect the security threats. An IDS is a type of security management system for system and network.

In our project, we have implemented a kind of IDS that monitors change in system behaviour for possible attack(s). Such kind of IDS comes under the category of anomaly based intrusion detection system. We also incorporated the logic for detecting network based attacks. For our IDS to be adaptable to the cloud environment our main focus was to propose a light-weight system.

Table of Contents

Outline of the Report

In this report the work done by us for the formulation of an anomaly based Intrusion Detection System (IDS) for cloud. We introduce the basic idea about IDSs in section 1. In section 2 we discuss the reasons related to why anomaly based IDS is a better choice than signature based IDS and why is a IDS needed for cloud environments. In section 3 we describe the aim of our project. In section 4 we discuss the basic concepts involved in the formulation of IDS. In section 5 we discuss the implementation of the system. We discuss the results in section 6. Finally we conclude the report in section 7 and propose the future work in section 8.

1. Introduction

Cloud computing is a new IT delivery model accessed over the network, both Internet and Intranet. It enables the provisioning of on-demand computing resources ranging from applications to data centres. The computing facilities provided by the cloud can be accessed from anywhere and at anytime on pay-per-use basis.

As cloud computing is gaining popularity day by day, concerns are being raised about the security issues introduced through the adoption of this new model. Clients' data and applications must be protected from insider and outsider attacks. For providing better security to the users it is unavoidable to use intrusion detection systems for cloud environment.

IDSs can be broadly classified into two categories-

- Signature-based IDS
- Anomaly-based IDS

Signature based IDS works by searching for known identity or signature for each specific intrusion event. A key advantage of this detection method is that signatures are easy to develop and it is efficient in sniffing known attacks. They are more or less similar to virus scanners.

On the other hand, anomaly detection is concerned with identifying events that appear to be anomalous with respect to the normal behaviour of the system. Anomaly detection techniques can be applied to cloud to detect unknown attacks at different levels.

In a cloud, there are mainly four types of IDSs-

a) *Host-based IDS:* It monitors and analyzes the information collected from a specific host machine. Information like file system used, network usage, system calls etc. can be used.

b) *Network-based IDS:* It monitors network traffic to detect malicious activity such as DoS attacks, port scans or attempts to crack into the system. The information collected is compared with known attacks for intrusion detection.

c) *Distributed IDS:* It consists of various kinds of IDS like HIDS and/or NIDS, over a large network, that communicate with each other or with a central server. Central analyzer is a machine that collects the information and analyzes the same.

d) *Hypervisor-based IDS:* It runs on hypervisor layer. It allows user to monitor and analyze the communication between VMs, between hypervisor and VMs and within the hypervisor based virtual network.

In our project we have implemented an IDS that is a combination of anomaly based and signature based IDS. Anomaly based system can detect zero day attacks as they continuously monitor the system for abnormal activities.

2. Motivation

Despite the simplicity of signature-based IDSs, there are various flaws with these systems. Because they only detect known attacks, a signature must be created for every attack and novel attacks cannot be detected. While signatures work well against attacks with a fixed behavioural pattern, they do not work well against the multitude of attack patterns created by a human or a worm with self-modifying behavioural characteristics.

Anomaly based IDS takes a more generalized approach when looking for and detecting threats in the system. This technique centres on the concept of creating a baseline for system behaviour. This baseline is a description of accepted behaviour, which is learned or specified in advance. Events in an anomaly detection engine are caused by any behaviour that falls outside the predefined or accepted model. It is capable of detecting zero day attacks.

Usually the IDSs are comparatively more computationally complex. This motivated us to develop a light weight and less complex anomaly-based IDS that have the ability to detect unknown attacks and would be suitable for the cloud environment. In cloud environments users don't have much control over the location of their data and applications and so it is very important that the service provider takes care of security and employs an IDS.

3. Aim & Objective

To design an anomaly based IDS with the following characteristics:

- Low computational complexity,
- uses minimum resources for monitoring and analyzing the collected data for attack detection
- and therefore suitable for cloud environment.

4. Concepts and Background

The IDS designed by us is a combination of HIDS and NIDS.

- The first module which is HIDS, continuously monitors the resource usage. As soon as any misalignment from normal behaviour is observed, it notifies the user. This module is based on the concept of anomaly based IDS.
- The other module is based on the NIDS which is primarily signature based. This module keeps a check on the websites accessed by the user and prevents the user from accessing any malicious website. On providing a URL (Uniform Resource Locator) by the user, NIDS module first checks whether it is safe to access it and allows loading the page only when it is sure that it's safe to do that.

We implement a lightweight Intrusion Detection System to monitor the system activities and detect any abnormal pattern in resource usage. For the purpose of building up the IDS we studied CloudSim which is a set of java classes that can be used to simulate cloud environments. We studied how a basic

cloud environment can be simulated by studying about its various packages and understanding the examples.

The proposed HIDS works on the concept of self-similarity. Every system has its own similarity pattern as the system's internal events depend on the usage pattern of resources by the processes and the user applications. So, any deviation from normal usage behaviour would be detected by the violation of its self-similarity pattern.

The model is lightweight in the sense that data required for monitoring the system behaviour is self generated by the system. Since no additional data needs to be generated separately for this purpose, there is no possible performance degradation. This also leads to a low computational complexity. Another merit of this model is that it reduces other unnecessary outside variables which results in increased accuracy in detection of threats.

System Model

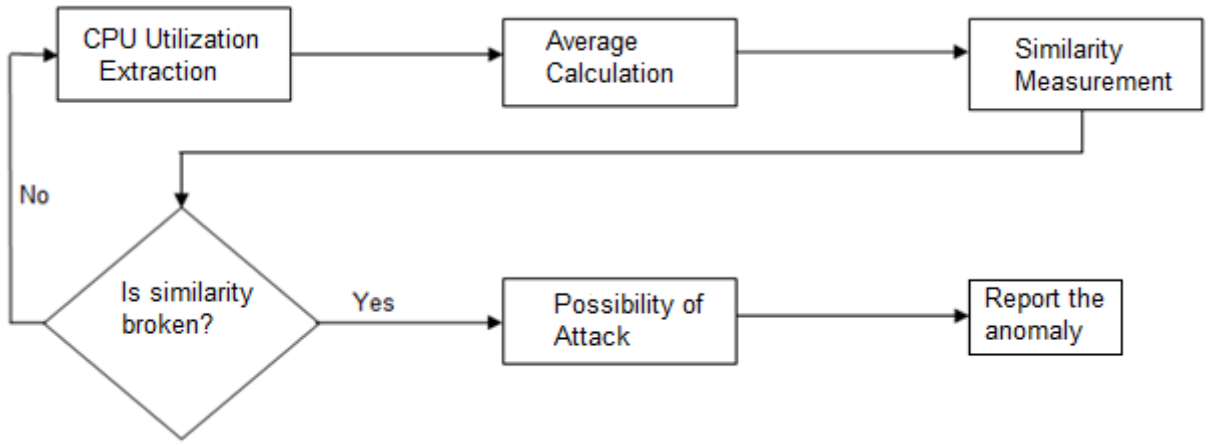


Figure 1 Flow chart of the host based intrusion detection system

In the event log pre-processing module, the system logs required for monitoring the system behaviour is collected by the IDS. Since anomaly-based detection works on continuous monitoring of the activities, the system log containing activities logged over a fixed range of time must be periodically collected. The next step is to select the required attributes for analysis of the log. Some of the significant attributes that must be stored are the date and time of the event, Security ID, Event ID etc. Based on the values of the chosen attributes, the IDS measures the self-similarity value for the given period of time. This value is compared with the self-similarity value obtained over the previous set of events. If the difference between two values exceeds a minimum threshold, then self-similarity is broken, indicating a possible attack. System administrator is alerted regarding the indication of an attack, and IDS takes preventive actions based on the level of vulnerability of attack on the system. In case self-similarity is not broken, this entire cycle is repeated for the next set of event log collected.

In the NIDS module a check is performed on every URL accessed by the user and the user is informed if URL is a malicious one. For this purpose predetermined information is used. A list of malicious URLs is needed.

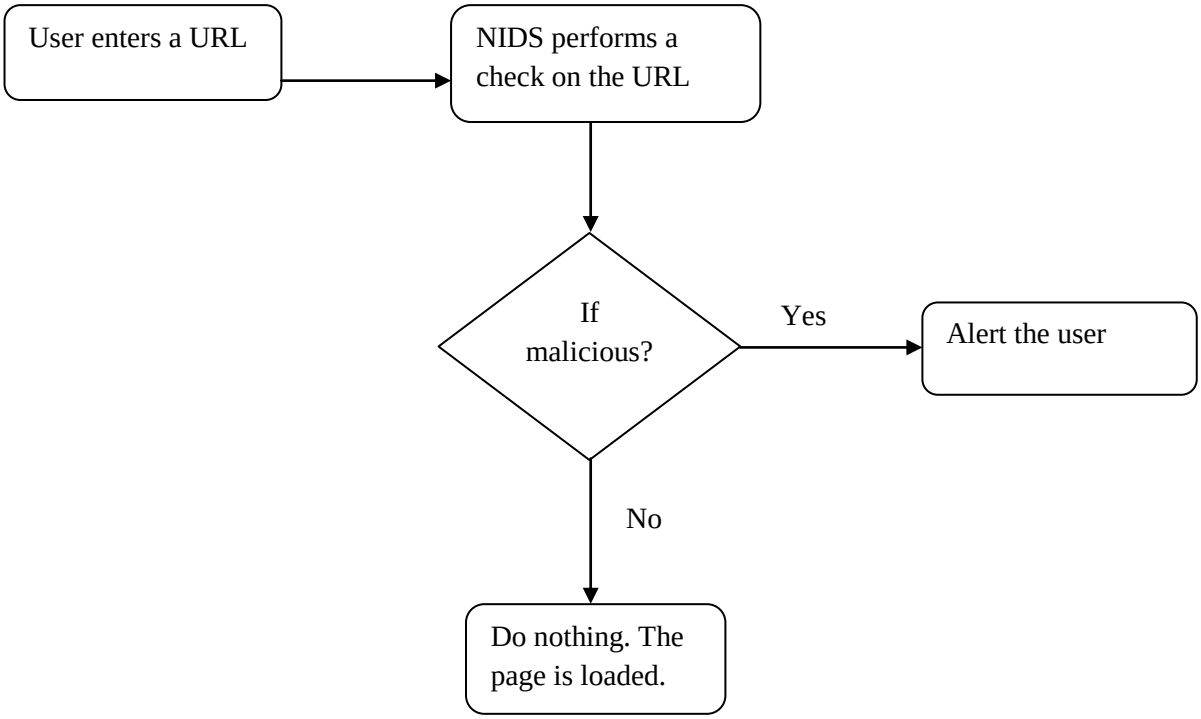


Figure 2 Flowchart for NIDS module.

The flowchart in figure 2 describes the NIDS module. Whenever the user enters a URL the NIDS checks whether it is safe to access the URL or not. It performs this by checking the URL against the data about malicious sites provided by GSB (Google Safe Browsing API). If it finds that the URL is safe to access then the NIDS doesn't notify the user and the page gets loaded. But if the NIDS determines that the URL belongs to a malicious site then it doesn't load the page and notifies the user.

5. Implementation

5.1 Experimental setup- In order to use the IDS developed, the system should be configured with any version higher than or same as XP of Microsoft Windows operating system. In addition to it Java Runtime Environment (JRE) is required. The software "Compute" which is the HIDS module should also be installed.

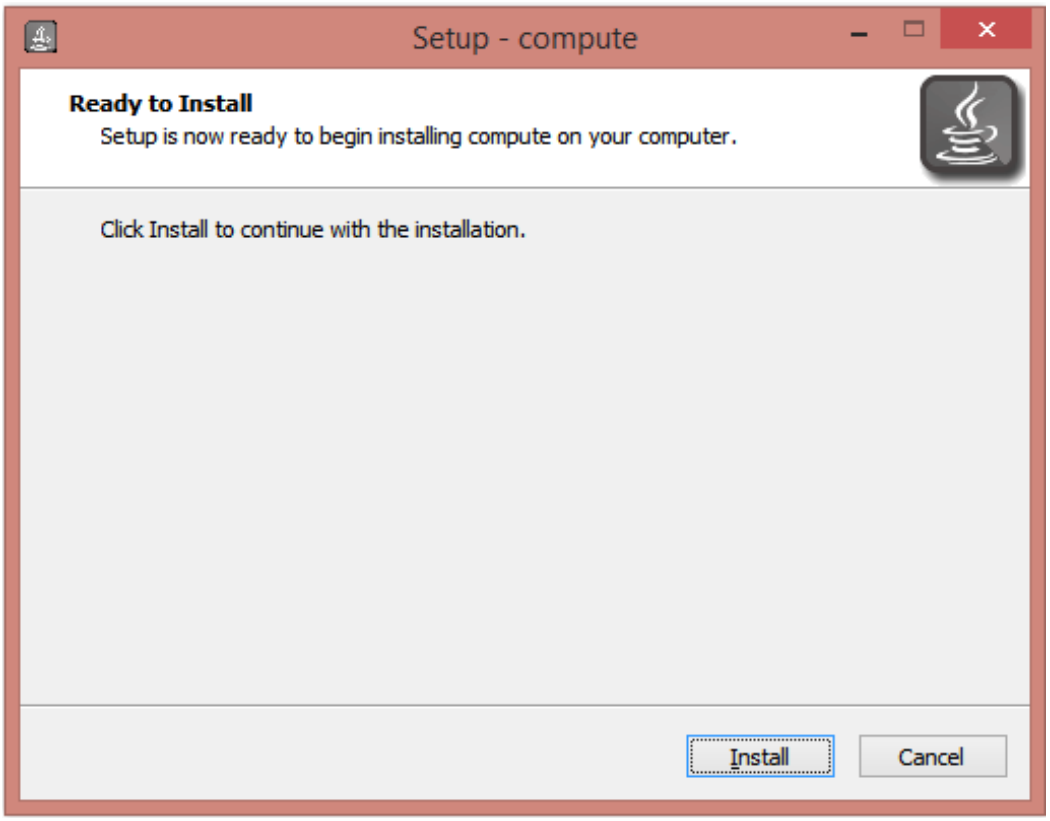


Figure 3 Install the software for HIDS.

5.2 Tools used- The various tools used by us are:

- NetBeans which is a software development platform written in Java. The CloudSim can be integrated in NetBeans and then can be used. We wrote the programs for our HIDS and NIDS in NetBeans environment.
- Fiddler which is an HTTP debugging proxy server application. It captures all the HTTP and HTTPs traffic and thus allows monitoring it. Fiddlercore API which is available in visual basic, C++ and C# was used for capturing the request data.
- Google Safe Browsing (GSB) is a service provided by Google that contains a list of URLs that are malicious.
- Visual Studio 2013 Ultimate which is an IDE from Microsoft, was used to program the NIDS in C#.

5.3 Evaluation Procedure-

HIDS module of the project was implemented by writing batch script and VB script. This module extracts the features of task manager like process status, CPU usage, memory information and network utilization. The information extraction happens continuously in the background. Over a time period the extracted values are logged in a file and these values are used to calculate average utilization over the fixed time interval. This process goes on for every other interval. If the calculated average value deviates over a certain threshold from previously averaged value then the self similarity will be broken and possibility of an attack is communicated to the logged in user. This mechanism is used for monitoring CPU utilization, memory utilization and network traffic. The concept behind this approach is that

resource utilization patterns usually do not change drastically in a short period of time. It is perceived that this might only happen when there is some malicious activity going on in the system. For instance, when certain process is excessively using the memory resource, it can be brought into user's notice. Because such a process might be malicious as it is occupying more than its fair share and consequently reducing the system performance.

Efficient use of valued resources is necessary as in cloud environment, users are served on pay-per-use basis, hence if the resources are being consumed by insignificant processes, time for which the cloud services are required by the genuine user will increase thus causing monetary loss to him and loss of trust in service provider.

In addition to CPU and memory resource monitoring, status of the active processes and network traffic are also considered for detecting anomalies.

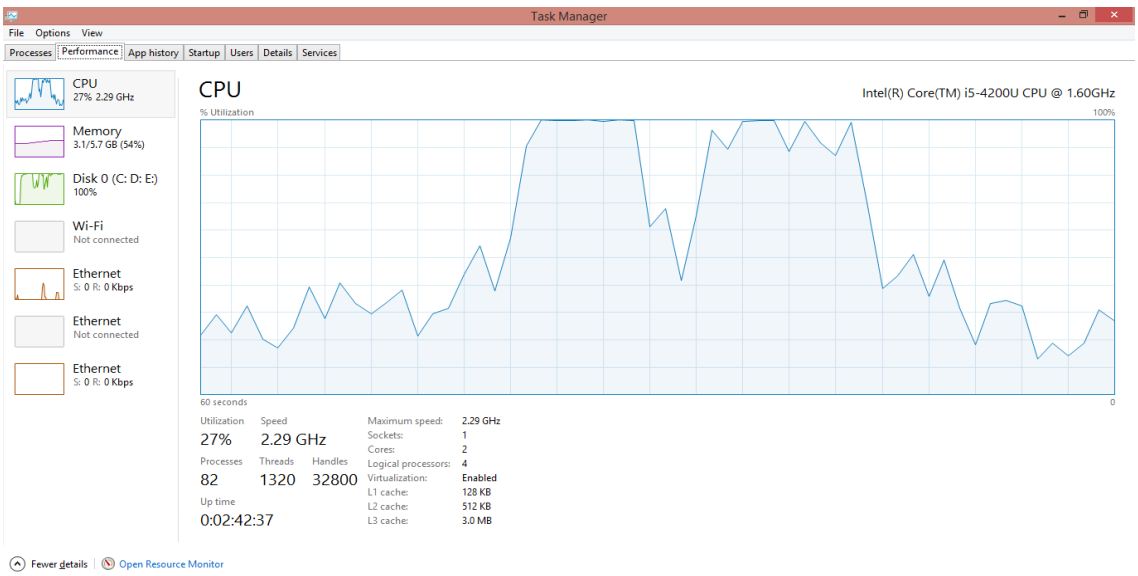
For evaluating HIDS and deciding on the values of threshold we ran our implementation at different times under different loads. For better performance, some good heuristics are required. This can be extended as future work.

The NIDS module works as follows- As soon as the user enters a URL the URL gets logged into a text file using the fiddlercore API. These URLs are read by our NIDS program line by line and are checked against the URLs stored in the database of malicious URLs populated using the data from GSB. This data is updated every time the program starts.

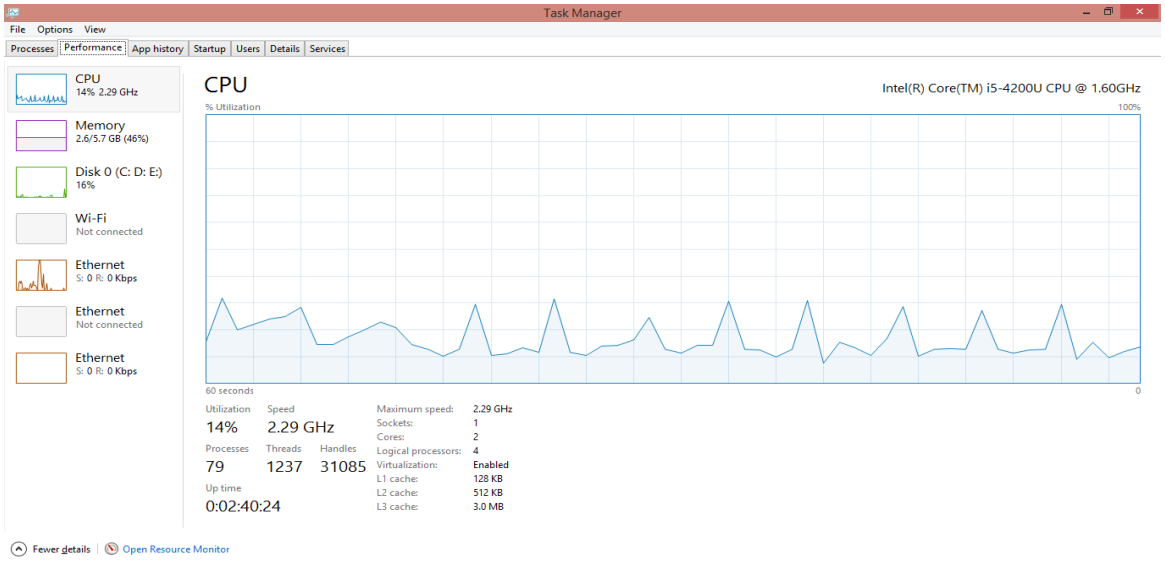
For the NIDS module for evaluating the NIDS we checked its working for a set of URLs including both malicious and non malicious URLs. We evaluated the module against two main parameters, firstly whether the NIDS gives the correct result and secondly how much time it takes to generate this result.

6. Results

- For HIDS.



(a)



(b)

Figure 4 This behaviour of the CPU is being used by our HIDS to determine risk.

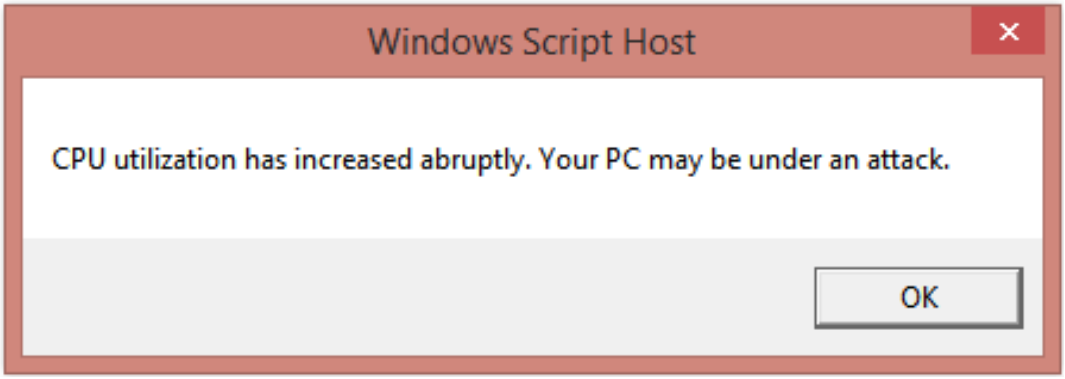


Figure 5 When load increases this will be the alert.

- For NIDS

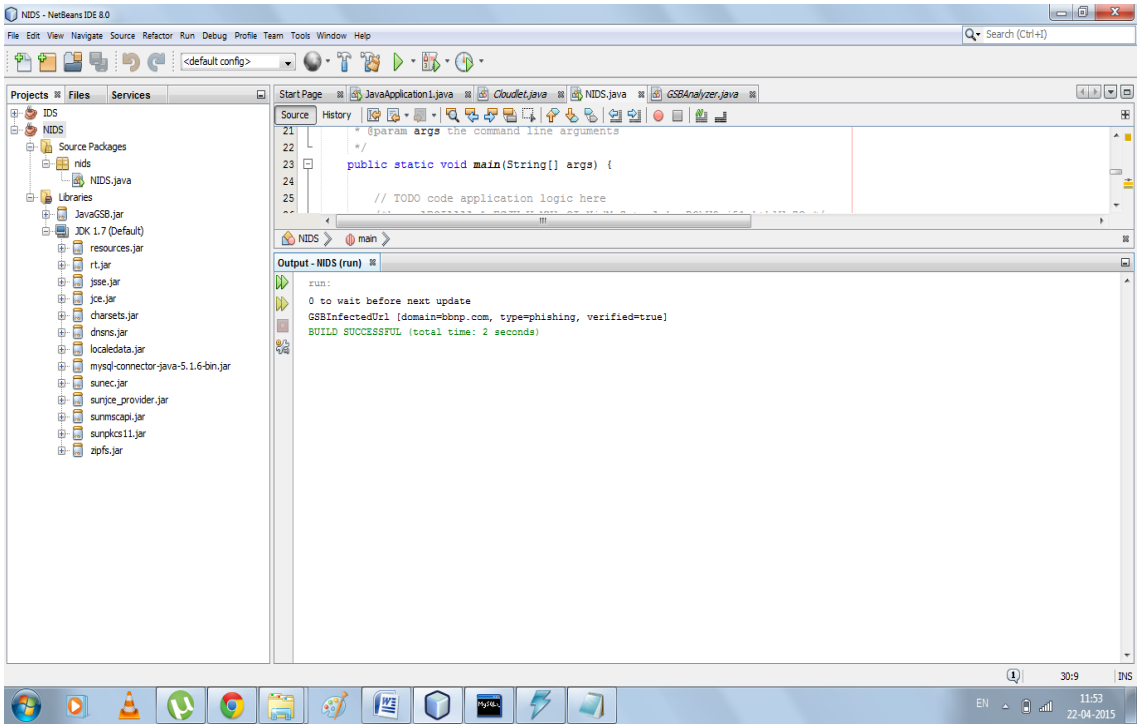


Figure 6 Malicious site detected by the NIDS.

7. Conclusion

The system implemented is partially anomaly based because the NIDS module uses the principles of signature based IDS as it checks phishing URLs from a pool of URLs provided by GSB. The HIDS module is anomaly based as it detects anomalies by learning the behaviour of the system over a predefined period of time. The system performs very well in Windows Operating System (OS); however, presently it is not OS independent.

8. Future Work

As the system is OS dependent, it can be extended for other operating systems so that it can serve as an OS independent IDS. There are several features using which system behaviour can be determined. More such features can be used for predicting the normal behavioural pattern of the system and the anomaly. Features that are used in developed IDS can also be utilized in weighted fashion in which weights can be assigned to different attributes on the basis of relevancy of their contribution in anomaly detection.

The developed IDS doesn't use any defined heuristic for determining the threshold value. An effective IDS must be flexible enough to work under extremely different environments. And it is cognizable the threshold will differ under dissimilar situations. Thus, for developing an influential anomaly based IDS, good heuristics must be used for deciding on the value of threshold.

In other way, the system can be enhanced by installing different IDSs on different VMs and thus sharing the reports hence generated. This system will enable the VMs to detect anomalies with more accuracy and in less time.

Presently, the implemented NIDS module is signature-based. However, it can be made anomaly based by analyzing the packet’s data in order to detect any anomalies during data exchange over the internet.

REFERENCES

[1] H. Kwon, T. Kim, S. J. Yu, H. K. Kim, "Self-similarity Based Lightweight Intrusion Detection Method for Cloud Computing," ACIIDS 2011, LNAI, Springer, pp. 353–362, 2011.

[2] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Softw. Pract. Exper. 2011.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, "A survey of intrusion detection techniques in Cloud," Journal of Network and Computer Applications 36, Elsevier, pp. 42-57, 2013.

[4] G. Nascimento, M. Correia, "Anomaly-based Intrusion Detection in Software as a Service," IEEE, 2011.