International Journal of Engineering Research & Management Technology

ISSN: 2348-4039

January-February-2023 Volume 10, Issue-1

Email: editor@ijermt.org www.ijermt.org

HONEYPOTS AND CYBER SNIFFING IN NETWORK SECURITY FOR SMALL SCALE INDUSTRIES

Rahul Singh Chowhan **Dr. Poonam Keshwani**

Ph.D. Student, Shyam University Assoc. Prof., Shyam University

ABSTRACT

The combined power of the world's computer networks and the Internet continues to expand. Computer networks make it possible to communicate much more quickly than any previous facility. The user is able to access both local and distant databases thanks to these networks. It is not feasible to secure each individual system that is connected to the network. Because a breach in the system may create serious difficulties, the network and the security of the network are essential concerns in the industrial sector. An intrusion detection system, often known as an IDS, is a kind of software that may be installed on a computer or network to keep an eye on its various operations, analyze potential dangers, and provide warnings to system administrators. And since IDS can only provide a solution for large-scale enterprises and there is no such thing as a solution for small-scale industries, a model for honeypots that combines Snort, Nmap, Xprobe2, and P0f has been presented as a way to handle the issue facing small-scale industries. The actions of attackers are logged and captured by this model, which also keeps track of all of these activities. A virtual machine is required in order to complete the process of virtualization. The prevention of assaults, both from the outside and from inside, as well as the upkeep of log files by employing honeypots in conjunction with virtual machines, are the primary focuses of this paper.

Keywords*: Intrusion detection system, honeypots, attacker, security.*

INTRODUCTION

The scale of the technology behind the Internet is quite huge, and it is continually expanding each day. The establishment of network security is essential for the development of sectors that are reliant on the internet for the expansion of their businesses and the provision of services through the network. Therefore, the safety of the networks is the key concern of the enterprises in order to protect the sensitive information. These sorts of businesses have been the target of a significant number of assaults in recent years. An intrusion detection system, often known as an IDS, is a tool that monitors the processes running on a computer or network in order to identify potential security risks and notify the system administrator. IDS and firewalls are used to defend the system and network from assaults; nevertheless, despite the numerous efforts that have been put into network security, the network is still not completely safe, and as a result, several sorts of solutions have been provided by the experts.

Since the small scale companies that use LAN are responsible for handling the database, the server, and the clients on their own, it is imperative that they maintain a high degree of internal security. A solution is necessary for small scale networks in order to safeguard their internal networks since the danger that comes from inside their own networks is always the biggest difficulty that the administrators face. Honeypot is the

proposed remedy in this paper, which addresses the same issue.

Research honeypots:

These are the honeypots that are exploited and used in order to gain information and knowledge pertaining to the hacker culture. The information obtained by the specialists is used for the purpose of providing early warnings, making judgements about assaults, improving detection systems for intrusions, and building more effective security measures.

Production honeypots:

These are the honeypots that have been developed by various businesses as a component of the backbone of network security. As early warning systems, these honeypots are quite effective. The elimination of potential dangers in many businesses is the purpose of these honeypots. It supplies the administrator with the knowledge before the real assault takes place.

Honeypots may also be categorized according on the amount of engagement or interaction they need, as follows:

Low level interaction:

Honeypots function as an emulator of the operating system and the services it provides since they only supply a small number of bogus services. These honeypots are not only straightforward to create, but also straightforward to uncover. An attacker just has to execute a simple command to recognise it, which is something a honeypot with a minimal engagement level cannot achieve. Honeyd is a good illustration of this kind of honeypot.

High level interaction:

Honeypots with a high degree of engagement give genuine-like operating systems and some actual services, along with some real uncertainties. These enable the information about the attacker to be captured, as well as the recording of their activities and acts. This is the actual machine, which consists of one system and has just one network interface connected to the network. Honeynet is a good illustration of this kind of honeypot.

Honeynets:

Two or more honeypots on a network create a honeynet. A honeynet is a collection of honeypots that work together to monitor a bigger and/or more diversified network than can be adequately monitored by a single honeypot. Honeynets and honeypots are frequently deployed as elements of larger network intrusion detection systems. A honeypot and its associated analytic tools may be centralized in what is known as a honey farm.

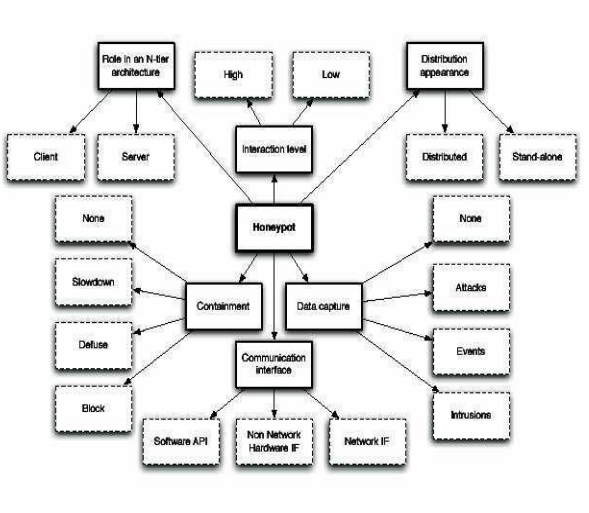

Figure 1: Taxonomy of Honeypot

HONEYPOT FOR SMALL SCALE INDUSTRIES

Honeypots created for the small-scale industry are tasked with maintaining information pertaining to the whole networking infrastructure as well as the records of all network log files. All of the information pertaining to the attacker is collected, and their actions are meticulously documented. Configuring the two or three tools together results in the successful implementation of the honeypot for small size enterprises. These tools are used by the attackers in order to collect information about the target. With the use of these instruments, sniffing may be avoided. It is possible to keep a record of packets that are traversing our network. It is possible to use it for port scanning in order to determine which ports are open and which are closed. A virtual computer might be used to trick an adversary into believing false information that it has obtained.

On the network, a series of simulated services are created in order to provide the impression to an attacker that they are interacting with a genuine system. These services consist of:

- HTTP
- POP3
- FTP
- TELNET

Therefore, these are the primary services for which the honeypot may function in order to offer protection for the network from cybercriminals. In order to make the actual system secure and make the proposed architecture more secure, I have employed a variety of technologies for the detection of intrusions and for keeping track of the actions of attackers. The honeypot will record the attacker's actions and behavior while they carry out their assault on the virtual computer.

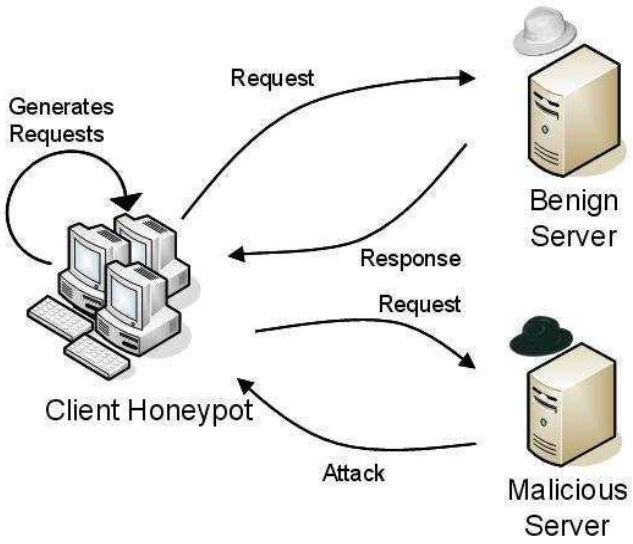

Figure 2: - Honeypot Architecture

Network decoys:

Honeypots are helpful tools for keeping an eye on networks. Honeypots are inserted into portions of a network that are not utilized for production and are then used for monitoring purposes. Honeypots are intended to catch a portion of an attacker's network traffic in the case that the attacker is probing the network. Because honeypots are not supposed to be visited by regular traffic, the warnings are quite trustworthy. Honeypots, on the other hand, are pointless if the attacker is aware that they exist. They are unable to determine whether or not there have been any attacks. Honeypots' primary use is to monitor networks, but they may also be used to trick potential attackers by posing as other systems. It's possible that the attacker won't be able to distinguish between the systems that are really valuable and those that aren't. Because of this, the adversary may have to exert more effort and spend more time in order to successfully target the system. This makes it much simpler to detect. However, putting up realistic decoys may be a very laborious process, and doing so also comes with an element of danger. The use of a network decoy to safeguard the system from an intruder or an unauthorized user (also known as a hacker) is thus recommended.

Prevention of spam:

Spammers obscure their identities by sending messages over open mail relays and open proxies. An open mail relay allows any sender to send mail without requiring them to authenticate themselves beforehand. Any client that is connected to the network is allowed to establish connections via an open proxy. Honeypots that pretend to be open mail relays or open proxies may be used to detect spam and disclose the origins of the unwanted messages. The improvement of spam filtering is made possible by the capture of spam. If the source of the spam can be identified, the network administrator may be able to disable the spammer. A honeypot is another option for gathering the sender's address when an attempted delivery of mail is made. The addresses are temporarily added to the blacklist that is maintained by the real mail server. This aids in the process of filtering out sources that are probably likely attempting to distribute spam. Since spammers have developed mechanisms to identify phony open proxies, honeypots seem to have been successful, albeit to a certain level. A straightforward test would be to make an attempt to send mail back to itself via the proxy. If the proxy reports that it was successful when, in fact, the message did not come back, then the proxy is almost certainly a honeypot. However, there is a rather straightforward way to circumvent the test. The honeypot needs just

to check if the source address and the destination address are the same in order to determine whether or not to allow the connection to proceed. A test with a higher degree of difficulty would separate the sender and the recipient onto separate hosts. Because the honeypot shouldn't be a genuine open proxy, it is far more difficult to circumvent this challenge in a general environment without being discovered. Honeypots, on the other hand, are likely to be less successful against spam transmitted via botnets than they are against spam sent using open mail relays and open proxies. It is safe to assume that the controller of a botnet is well concealed and that they cannot be identified based on spam delivery efforts. In addition, efforts to blacklist senders are not particularly beneficial either since there are so many possible senders.

Collecting malware:

A honeypot that is designed properly may automatically gather samples of malicious software that propagates on its own. This enables the capturing of malware at a big scale that is actively operating. This, in turn, makes it possible to conduct research on live data and continually improve software for both the detection of intrusions and the prevention of viruses. The manual collection of malware would just take too much time. The purpose of a honeypot designed to capture malware is, in essence, to download the malicious software and make a record of the particulars of the occurrence. The honeypot is designed to collect the payload of a network connection whenever it has the potential to lead to an exploit. The payload is next examined to see if it includes code that is executable by machines or network addresses. In the event that sufficient information is discovered, the honeypot will download the potential infection. Emulation is what low-interaction honeypots do, therefore these honeypots have the ability, at least in theory, to only catch malicious software that targets known vulnerabilities. A honeypot with a high level of interactivity that uses a genuine operating system is required for more complete collection.

Detection of malicious Web content:

Web browsers may include security flaws that make it possible for rogue websites to secretly install malware on users' computers. Exploited sites are rather widespread in the modern day, and because of this, it is impractical to manually find and analyze them. Client honeypots are able to automate at least some of the detection process and contribute to the study. HoneyMonkey is an exploit detection honeypot that has a high level of client interactivity. The system is made up of a collection of Windows XP installations that are carried out on virtual machines and include varying degrees of patching. The system is provided with a list of URLs, which a customized version of a web browser running inside of a virtual machine then views one at a time. In the time between URL visits, the system, files, and registry are inspected for their current condition. If there were any alterations that occurred outside of the browser's working area, the URL would be flagged as an exploit and sent through more scrutiny if it was reported. When this occurs, the instance of the virtual machine that was abused is thrown away, and a new one that is uncontaminated is launched. Therefore, this should be the primary goal that the honeypot serves for the intrusion detection system. Therefore, in order to safeguard the system from the invader or the attacker, we have conducted an in-depth study of all of the goals.

OBJECTIVES

1. The main objective of the honeypot is to find the attacker by the connection tracking
2. To study cyber security

RESEARCH METHODOLOGY

i. **Data Capture / Traffic logging Components: -** Honeyd and Tcpdump are included in this section for the purpose of data collecting.

ii. **Data analysis / analysis and extraction components: -** This section of the mechanism provides the data analysis component of the signature extraction process, which is used to extract accurate attack signatures.

iii. **Signature Extraction: -** a step-by-step guide to obtaining our high-quality attack signatures. The extraction of the signature was also utilized to explain the different attack signatures.

RESULT

Data Capture: - The gathering of all of an attacker's actions in a log is the goal of the data capture process. The Honeypot performs precisely this function, which is that of information collection. The HoneyAnalyzer System Gathers Information from Two Different Sources, Namely the Honeypot Log and the Network Traffic Log Obtained through the Tcpdump Program. The Honeyd framework provides support for a number of different logging methods for network activities. It is able to build connection logs that indicate successful and unsuccessful attempts to connect for all protocols. However, in order for the system to conduct a comprehensive analysis of the attack scenario, it requires the whole payload of each packet that enters and exits the honeypot. Tcpdump, the second component, is responsible for carrying out this operation. It does so by capturing the whole payload of each packet. Tcpdump is one of the most well-known sniffers for Linux and is an utility that is used for monitoring networks. After that, it writes the header information of each packet to the log file.

Data Analysis: - A data analyzer has been constructed in the manner that is presented above in order to extract the more exact attack signature: -

The web interface provides a graphical display, making it simple for the security administrator to identify the port that is being targeted the most. The IP address provided may then be used to identify the geographical location of the hacker or attacker. The following is a description of the suggested approach for the implementation of the HoneyAnalyzer, which is intended to extract a more specific attack signature: -

i. Set honeyd up such that it can imitate a network.

ii. Conduct a traffic analysis using the Tcpdump programme.

iii. Invoke the auto run shell script, which will run at a specific time period, and execute the parser tool, which will parse the data from the honeyd log file and enter it into the database. The functionality of the parser tool may be implemented in any language, such as Java, that is capable of string tokenization to a significant degree.

iv. Run the auto-run shell script to import the data from the honeyd logs into the database. This will be executed when the cron job runs.

v. Authenticate yourself using the web interface in order to observe the attack patterns and analyze the data in order to extract a signature of high quality.

The web-based graphical user interface offers the following capabilities, which together make it possible for

the Security Administrator to choose potentially malicious data: -

i. The capacity to show packet information retrieved from the database.

ii. The capacity to show historical traffic statistics in addition to current and real-time network traffic based on information stored in a database.

iii. Show the ports that have been attacked within a certain time frame.

iv. Now, here's the major scenario: which remote IP addresses Honeypot "visited" over a certain time span. In this section, you have the option of entering a port number to see activity on a particular port.

v. A numerical representation of the number of times a piece of text has been accessed during a certain period of time. It is possible to zero in on certain happenings if a specific IP address or port number is entered.

iii. Signature Extraction: - While the current system applies the LCS method to the complete data set, the graphical interface provides capability for applying the LCS algorithm just to the data that are of interest. The process of locating attack signatures is not entirely automated; rather, it relies on the knowledge and expertise of the security administrator (SA). The SA has the ability to choose the types of traffic for which the LCS algorithm will be used. The exact signature that is produced as a result will have a lower rate of both false positive and false negative results. The following procedures were carried out in order to locate an attack signature of sufficient quality:

a. Using the web-based graphical user interface, locate the data from the database that is of interest to you. This is an explanation of all there is to know about the signature extraction method, which identifies hackers by looking at graphic website content.

b. Perform an analysis on the combined data from the various data sources, which are Honeypot and Tcpdump. Start the following sequence of actions with each packet that is successfully received:

Using the web-based graphical user interface, you may search the database for data that is of interest (or significant) to you.

Perform an analysis on the data obtained from various sources such as honeypot and Tcpdump.

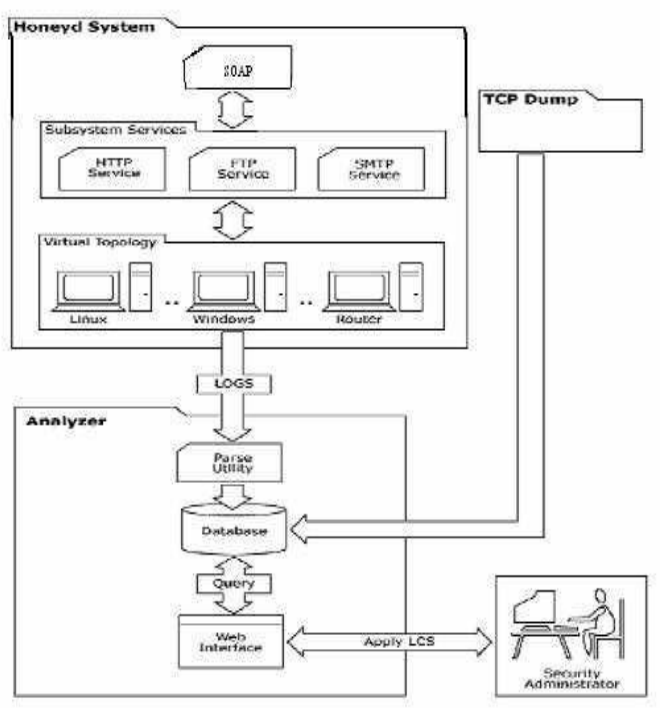

Figure 3: Architecture of the Honey Analyzer, showing honeyd when it is emulating a number of distinct computers, each of which is executing a number of pre-configured services.

a. If there is already a connection state for the new packet, it will be updated; otherwise, a new connection state will be generated.

b. Do not handle the packet if it is destined for transmission outside the network.

c. Carry out protocol analysis on both the network layer and the transport layer.

d. For each connection that has been saved, carry out header comparison in order to identify comparable IP networks, TCP sequence numbers, and so on.

e. Apply a content-based string matching algorithm to the payload of interest by carrying out the operations listed below:

f. Use the Longest Common Substring technique to conduct pattern recognition on the messages that have been transmitted if the connections have the same destination port. The provides an explanation of the string-based pattern identification method.

g. If during the procedure a new signature is generated, the existing signature pool should be updated with the new signature; otherwise, the process should be terminated.

COMPARISON OF HONEY ANALYZER /HONEYCOMB

Honeycomb's use of Pairwise LCS often results in duplicate signatures, which are signatures that are not similar to one another. This results in several alarms being generated for the same attack. Honey Analyzer, on the other hand, generalizes the method in such a way that a security administrator who is familiar with protocol semantics may groom the signature to make it far less susceptible to the creation of repeated

signatures.

Honeycomb's lack of semantic awareness results in signatures that are made up of innocuous substrings. Honeycomb is unable to build accurate signatures for protocols such as NetBIOS, MS-SQL, and HTTP assaults such as Nimda since these factors result in false positives. Nimda is an example of an attack in which the exploit material makes up just a tiny percentage of the overall attack string. In the case of HoneyAnalyzer, ensuring semantic awareness falls within the purview of the security administrator. He has a better understanding of the harmless substrings that the local network contains and is able to filter out redundant and pointless strings.

As a consequence, the signatures acquired using Honey Analyzer are of a high quality, which leads to more accurate detection of infiltration attempts. Honey Analyzer is also capable of performing the function of an intrusion indicator, which involves determining when, how, and from where various infiltration attempts are taking place. The graphical user interface may be used to display this information. Honeypots are increasingly being used in networks, but most of the time, administrators merely sit back and wait to see what occurs while the honeypot is running in the background. The suggested system will provide the security administrator with more control over the intrusion detection process, which will allow for the extraction of attack signatures of higher quality.

ADVANTAGES / DISADVANTAGES

There are a number of benefits, as well as some drawbacks, to using the honeypot in order to make the network system safe and shielded from an external attacker or hacker. Here are some of the benefits and drawbacks, in no particular order:

Advantages of honeypots:

There is a wide variety of protection options available on the market today. Through the use of the internet, anybody may explore the myriad of options available to them and get the answer that is most suited to meet their requirements. The following are the justifications for why I ought to go with honeypots: Honeypots have the ability to detect assaults and provide information on the kind of attack. Additionally, owing to the logs, it is possible to see additional details regarding the attack if they are required. When new threats are investigated, new security measures might be conceived as a direct result of the findings. Examining the many types of malicious activities enables one to gain a greater number of exams. It is helpful to have an understanding of further assaults that might occur. Capturing data. They are just focusing on dealing with the malicious traffic that is coming in. As a result, the information that was intercepted is a far smaller portion of the total transmission. When conducting an investigation, it is much simpler to zero in exclusively on the malicious communications. As a result of this, honeypots are quite beneficial. There is no need for a large amount of data storage if there is simply harmful traffic. There is no need for the implementation of new technology to maintain. Honeypot software may be installed on almost any machine. As a result, the creation of such a system does not need an extra budgetary allocation. They are straightforward to comprehend, as well as straightforward to set up and install. They do not use complicated algorithmic structures. Some items do not need modernization or adjustments at this time. Honeypots have the ability to pick up anything dangerous, which means that they may also pick up new tools for detecting assaults. It provides additional ideas and a deeper understanding of the topic while also demonstrating that it is feasible to learn about diverse points of view and incorporate them into our security solutions.

Disadvantages of honeypots:

The use of honeypots comes with a number of significant benefits; nevertheless, there are also a few drawbacks associated with their use. The only time you will be able to collect data is when the hacker is actively targeting the system. It will not be feasible to get any information if he does not attempt to breach the system. The honeypot won't be able to detect an attack that's happening on another system since it's not connected to that system. Therefore, assaults that aren't directed at the honeypot system might potentially affect other systems and create significant issues. Honeypots have the drawback of being susceptible to fingerprinting. Honeypot systems are designed to make it difficult for hackers to determine whether or not they are assaulting a legitimate computer network. The use of fingerprinting makes it possible to differentiate between these two. The outcome of the experiment is not what was intended at all. It is possible for the honeypot to be turned into a zombie and used to access other systems in order to compromise them. This might put you in a very precarious position.

CONCLUSION

Honeypots are not a solution to network security; rather, they are a useful tool that complements and enhances existing security technologies to build an alternate active defensive strategy for network security. Honeypot is a novel approach to attack prevention, detection, and response that works in conjunction with intrusion detection systems (IDS) and firewalls. The capability of a honeypot to lure an attacker to a decoy system makes it an effective deception weapon that may be used for the purpose of preventing product system breaches. Honeypots, when used with intrusion detection systems, are more effective in reducing both false positives and false negatives. The intelligence routing control system allows for a more flexible response to threats. Honeypots come in a variety of shapes and sizes, but they always use the same data management and data gathering technology. The experts concentrate on these two aspects in order to build honeypots that are both simpler to instal and more challenging to detect. Based on the recent developments in honeypot research and manufacturing, I believe that the honeypot of the future will have the capabilities of integration, virtualization, and dispersion. All of the components are contained inside a single device thanks to integrated honeypots. A huge number of honeypot systems may be generated by a single computer using a virtual honeypot. A distributed honeypot is a honeypot that is spread over a real network and uses several honeypot systems to provide a high level of interaction between assaults and the system. All of these things will make future honeypots simpler to apply and less expensive to keep up.

REFERENCES

[1]. R.Baumann, C.Plattner “honeypots” Diploma Thesis in Computer Science, 2002.

[2]. Gurleen Singh., Sakshi Sharma, Prabhdeep Singh “Design and develop a Honeypot for small scale organization “in IJITEE. Vol 2, issue-3, Feb2013.

[3]. H.Artail, H.Safa, M.Sraj, I.Kuwalty, Z.Masri “A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks” Science Direct, 2006.

[4]. Deniz Akkaya – Fabien Thalgott, “Network Security Using Honeypot” IEEE, June 2010.

[5]. Y.K.Jain, S. Singh “Honeypot based Secure Network System” in IJCSE. Vol 3. No.2 Feb 2011.

[6]. S. Mrdovic, E. Zajko "Secured Intrusion Detection System Infrastructure", ICAT 2005.

[7]. Erwan Lemonnier, Defcom, "Protocol Anomaly Detection in Network-based IDSs", http://erwan.lemonnier.free.fr/.

[8]. Urjita Thakar, Sudarshan Varma, A.K. Ramani " HoneyAnalyzer – Analysis and Extraction of Intrusion Detection Patterns & Signatures Using Honeypot" in Second International Conference on Innovations in Information Technology (IIT'05) Dubai, UAE September 26-28, 2005.

[9]. Hyang-Ah Kim, Brad Karp, "Autograph: Toward Automated, Distributed Worm Signature Detection," In Proceedings of the 13th Usenix Security Symposium, San Diego, CA, August 2004. Pp. 271– 286.

[10]. Christian Kreibich, Jon Crowcroft, "Honeycomb- Creating Intrusion Detection Signatures" Using Honeypot, ACM SIGCOMM Computer Communication Review archive Volume 34,Issue1 January 2004, Pp. 51 – 56.

[11]. C K Shyamala, N Harini, Dr T R Padomanabhan – Cryptography and Security, May 2011.