

**Information Security-IV**  
**Prof. M J Shankar Raman**  
**Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

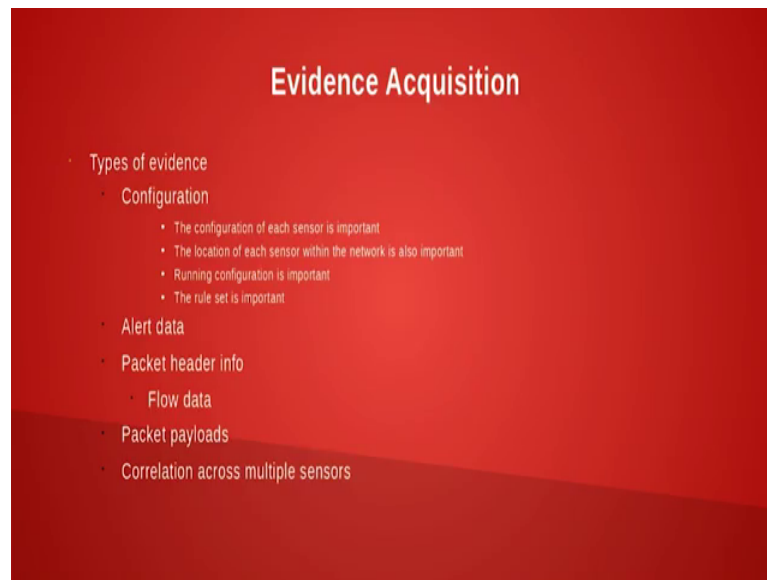
**Lecture – 48**  
**IDS Evidence Acquisition and SNORT**

Hi, welcome to this session on network security and forensics. In the last session we were discussing a lot about the different ideas both commercial and non commercial versions. Essentially the intrusion detection system and prevention system actually tries to acquire evidence ok. So, the question is how are you going to acquire evidence, first one is how are you going to acquire evidence. We will also see what are the types of configuration logs, these systems have. We will also review something about snort before we actually look at a demo of snort. So, hopefully we will see whether we can give the installation instructions also otherwise we will just show you them off snort and installation you could actually do it by yourself.

Now, with respect to evidence acquisition, the intrusion detection system and the prevention system must be configured to ensure that, you are able to capture any anomaly in or any scanning kind of activity a reconnaissance, or anomaly, or scanning kind of activity that is activities that are not normal in the network. The idea is that; so, in this case a person you has to be expert in configuring the network see it is in computer science, it is told that garbage in garbage out.

So, similarly if you take any intrusion detection it has some kind of a language ok, in which you have to specify to the intrusion detection software, how to capture packets and how to find out patterns and, once you find out patterns how do you react and after that what do you do.

(Refer Slide Time: 01:58)



So, these four issues have to be tackled by any intrusion detection software. Now, the configuration of sensors for example, suppose I want to see whether someone is doing a port mapping ok, then I should configure my intrusion detection system to scan for any port knock, I mean port knocking that someone is trying to attempt. So, let us say is tries to knock usually people knock ports like DNS, telnet, FTP all these ports because, that is what you actually leave it open in general in many of the times ok.

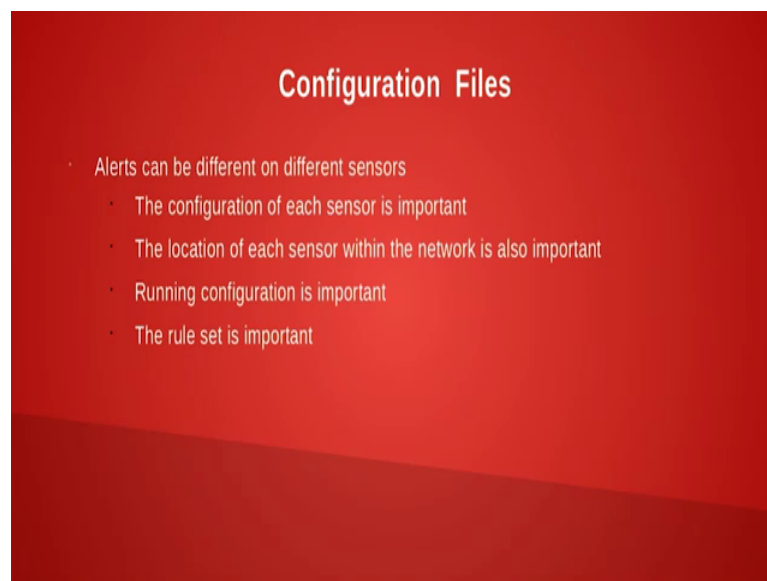
So, the configuration of each of the sensor is very important in a evidence acquisition. The second one is the location of the sensor is also important. So, you should find out where the attack or the malicious user is trying to enter and, then put the device in the right place. One more important thing is your idea should not be hackable see that is one of the reasons we actually try to suppress the IP address of the ideas, when you want to acquire evidence ok.

So, when you see any malicious activity you have to send alerts ok. This alerts you could actually send it to a particular machine through SNMP like some kind of trap kind of messages, or you could also contact a server and send these alert messages. You should also look at the protocol information and, you should look at the flow information, you should look at the packet information. So, many of the intrusion detection devices provide facility to actually do, or actually snoop into the packets and, identify this kind of patterns see you can only look into the packets and identify the patterns. So,

sometimes you might have to read the metadata, or sometimes you have to do the deep packet inspection in some many cases. And after this you have to correlate across multiple sensors see, what you could do is you might have to correlate against the operating system logs. So, because operating systems also you could configure if for example, someone tries to do an unauthorized entry by logging in with some username and trying deliver and trying to attack using some dictionary based attack on passwords ok.

So, if this happens then you should be able to correlate against the operating system logs, as well as the logs that are generated by your intrusion detection system. So, that is very important. And each of these intrusion detection systems work, or especially I mean be I am talking more with respect to snort ok. So, the snort and most of the intrusion detection systems are similar to snort. Snort it is not there could be, but each of these systems will have something known as a configuration file ok.

(Refer Slide Time: 04:54)

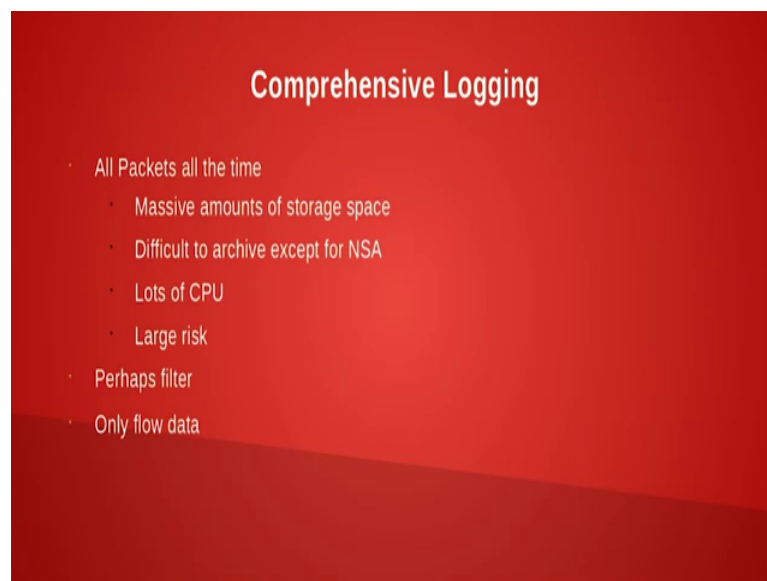


The configuration file actually what it does is it you actually had spread sent something known as a rule ok. So, what is this rule, you actually when the rule sets tells you that, if you see a particular pattern emerging in the network traffic, what is the action you have to take ok. It could be as simple as just printing a log message by the ideas itself, or it could be very high end like blocking certain ports, which is actually the work of a firewall ok. So, your activity or the action can be anywhere between just logging to some

serious action being taken. So, this how to be configured and set in a set of rules, this is known as the rule set. And then the rule set has a rule header the rule body etcetera. So, we will see all those things later.

Many times you might have to do some sort of comprehensive logging and, if you are going to do comprehensive logging, you fairly know that all packets, when you want log all the time you might have huge data. And this also needs lot of CPU time and there is a heavy risk involved because of corruption of data and things like preserving see as the data size increases, then storing it and preserving becomes an issue, because security is an issue in with data anyway ok. And. so, usually what you do is you collect all the data, but then filter it all these filter rules can also be done with ideas. So, I collect the data and then filter it and then based on the filtered data, I take some kind of action this can be mentioned in the rule set.

(Refer Slide Time: 06:28)



**Comprehensive Logging**

- All Packets all the time
  - Massive amounts of storage space
  - Difficult to archive except for NSA
  - Lots of CPU
  - Large risk
- Perhaps filter
- Only flow data

So, what we will do is now we have been generally talking about logging and things like that.

(Refer Slide Time: 06:36)



So, now, what we will try to do is we will try to take one specific example like; we will look at us snort ok. And we will see what are the functionalities of snort, we will look at some examples configuration of snort. Snort and, after this we will have a case study on snort ok. So, we will just take some simple examples because snort is really powerful and so, we will not look into all the features of snort, we will just look at some simple example one or two example for as a case study and, then see how is snort actually identifies intrusion intrusions ok.

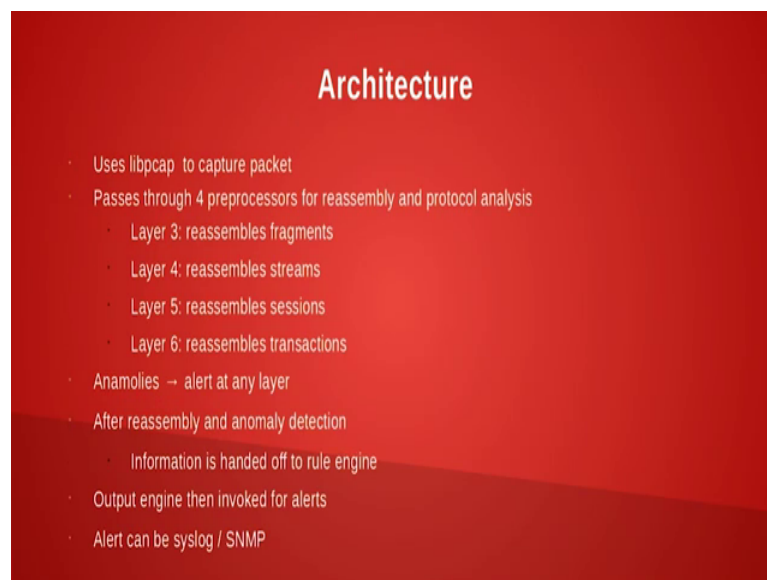
Snort is a widely used IDS it was purchased by Cisco, it actually uses the libpcap utility ok. So, because it uses the libpcap it is functionality is similar to TCP dump, as we have seen but the best part of it is that it can do analysis from layer 2 to layer 4 ok. And it also has the ability to do deep packet inspection of course, I mean with wireshark I mean we were actually doing it, we were not doing actually deep packet inspection, but at least we are looking into the packets ok.

So, and as I told you it has a open rule language ok, it has got a configuration file and, in the configuration file you can declare the variables and all that and you can also specify the set of rules ok. And these rules are extremely flexible; you can actually change the rules. In fact, there are places where once they identify a pattern they actually do some machine learning and then I mean there are some kind of learnable intrusion detection systems ok.

So, they actually learn what is that the mentality of the attack, or the way the attack is trying to happen and, then they actually do I mean they actually go and prevent that acts much better ok. This because not is extremely versatile ok. So, it rule additions and rule additions, and the rule language is very easy, not only that it does one of the things you should be very careful about using of network forensics tools is the support commercial support, sometimes some tools are just distributed with no support. So, you might have a problem in using such kind of tools, but snort I think almost many of the community people accept that it is a good software intuition directions software ok.

And this also has a community with the commercial business model therefore, life's becomes much more easy as far as support is concerned ok.

(Refer Slide Time: 09:05)



As long as the architecture of snort is concerned it uses libpcap to capture the packets, then it passes through 4 pre processors and these P processors essentially is for reassembly and protocol analysis ok. So, layer 3 it actually it can reassemble fragments and, layer 4 also it can reassemble streams ok, layer 5 it can reassemble sessions and layer 6 it can reassemble transactions and this is very important. Because if you remember the tool that we used network miner life was made very easy, when we were able to capture the document as a whole, rather than we searching for the signature of the fingerprint of the particular document inside the email message ok.

So, if a software can do this kind of activity, then it makes life pretty simple. So, now, as I told you it has an architecture where if there is an anomaly, then the alert can be sent to any layer ok. So, that that is what we discussed it can either be just logging or it can do activity of closing the connection also. So, after the reassembly and anomaly of detection ok, information is handed out to the rule engine ok. So, rule engine is the one which specifies what action to be taken.

And then after the rule engine specifies it is sent to another. So, it you could you could think of it as a sort of pipeline because, I mean performance is a must in this. So, this architecture works in a sort of a pipeline take the package do some sort of reassembly, I mean and then reassembling the fragments and all that give it to the next layer, that guy will do certain operations you will give to the next layer and so on. And in this way you are able to achieve good performance ok. So, one of the things is you can even integrate SNMP based alerts ok. So, this is if you are using network management platform, which many organizations usually do and, then integrating with ideas can I mean provide a nice kind of GUI based integration ok.

(Refer Slide Time: 11:14)



The most important aspect of snort is it has some configuration files, we see and I am talking with respect to Linux ok. So, you can see some of these configuration files, we will actually show it in a case study session, but for now we will let us understand that we will get a overall picture So, that when we do the case study we will understand the

case study much better ok. So, what we will do is we will look at the this snort dot conf has global values of snorts that are declared ok.

So, for example, I could monitor a set of HTTP servers and a set of IP address connections. So, for this I will declare a variable and then for that variable I will assign bunch of IP addresses ok. So, in this way I can have global values of variables ok, I can have I can define network whatever network is internal to my organization or external to my organization. So, it is its very flexible I mean if you had done a course on networking you would you would say you know that, a router interconnects 2 networks ok. We were talking about firewalls and demilitarized zones etcetera in our previous lectures.

So, in and so that we always configured it as an internal network, where the IP addresses were different and then we had external network where the IP addresses were different. So, with snort you would be able to identify, or clearly specify about internal networks and external networks, then you can also configure the pre processors ok, pre processors are you can I mean as I told you could look up look at specific ports activities etcetera this can be configured.

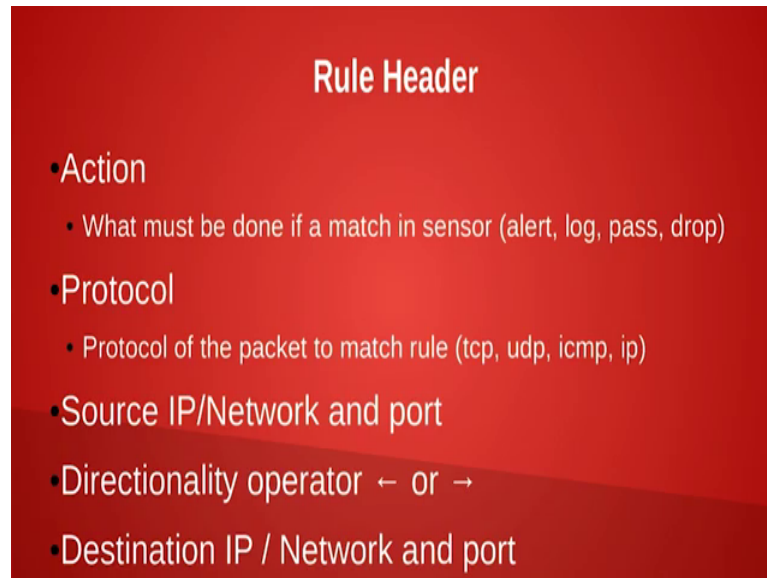
Then you can look at an output processor configuration, output processor is what is after the action is taken what are you going to do ok. And then you take usually call it as the rule chunks ok. So, so all these things are available within this snort dot com ok, you can include certain files and, then you can specify the rules in a separate location and, then point to those rules etcetera ok. And if I have 100 rules files, I need not implement all the 100 rule files, I can even we can only specify 50 rules are enough for me for the time being, I will incorporate the other rules maybe later, or I will disable the other rules etcetera. So, you have a option of enabling and disabling certain rules.

For example I want to set up a honey pot ok. In that case I will how to disable all the rules. So, that when the attacker comes I welcome the attacker to catch him ok. So, in that case I have to disable some of the rules. So, this can be done with snort and the snort also puts the log see just a slight deviation, I just want to tell you that when you are looking at OS forensics operating system forensics ok. This where log is very very important, I mean almost all the software that you put on Linux ok, stores the login where log syslog, kernel log and almost all the logs. So, this log files are very important and snort also follows the same procedure ok, this is we will see that this is this



capability obtained by Linux using something known as a syslog feature, anyway we will look at more details about the header ok.

(Refer Slide Time: 14:45)



So, the rule header actually tells you what is the action. So, what must be done if a match in sensor so, the actions could be something like alert, or log, or pass the message, or drop the message. So, this is so, the action could be any one of these activities ok. The next one the rule writer will specify is protocol ok. So, you can actually specify that the packets if it is TCP or UDP or ICMP, I mean this if you remember this is similar to what we did in wireshark, I mean we applied something known as filtering rules at the top if you had remember.

So, this similar filtering rules is applicable for snort also, the source IP bar network and the port ok. So, probably in filtering we did not look at IP bar network ok, we are looking only at IP addresses and ports when using wireshark. So, in snort we will see that not only IP addresses, we could also look at network as such network using the network mask and the port number.

And then you can also the rule letter will also specify the direction and operator; I mean which is the source, which is the destination ok. Usually left hand side of the rule is the source, on the right hand side is the destination, but you change the arrow I mean, even though we call source destination it could be the other way around ok. And then it also talks about a destination IP network and the destination port.

(Refer Slide Time: 16:16)



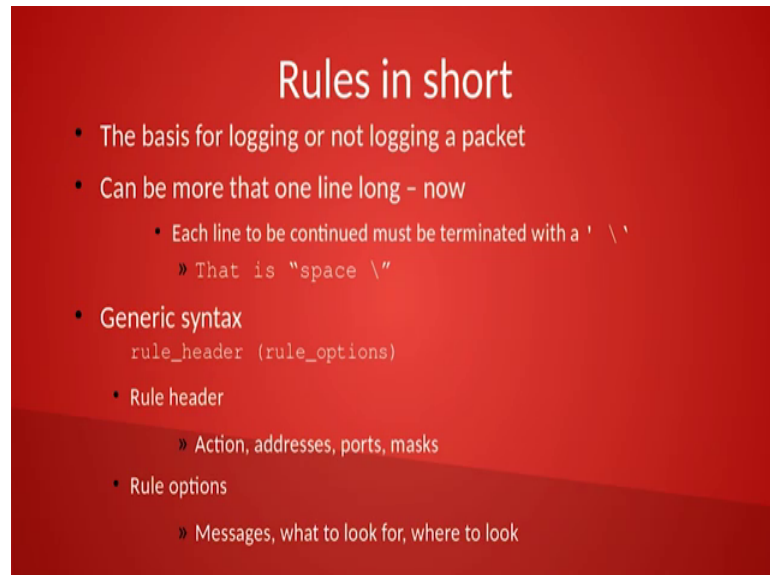
So, let us look at an example of a rule header. So, here is an example of a rule header ok. So, it says alert space tcp space any space any space and then it has a right side pointed arrow, then 192 dot 168 dot 2 dot 180 and then something inside ok. So, what it tells you is that, I can receive packet a tcp packet from any connection any network and any port, but when it is directed towards 192 dot 168 or 2 dot 180, 80 as we know is http port, then create an alert that is the first line. So, you see that this rule is very flexible and easy I mean you do not have a lot of English words, but at least this conveys what you should do.

Similarly, the next line so, it tells you log all udp packets, that comes from 192 dot 168 dot 1 dot 1 and port number 53, but not directed to the network 192 dot 168 dot 1 dot 0 bar 24. So, this is when you put slash 24, then it becomes a network ok. And the port is any port so; this tells you that if someone a DNS server is trying to respond to any of this just put a log message ok. The next one as I told you, I told you that you can use external net and HTTP servers now, these are variables ok. So, external net is defined can be any network that is external to your network and HTTP servers could be any HTTP server and, you could have a list of ports also http port also.

So, in this way now you see if you look at the way these three examples are first one is a very simple example, then we are addressing a network and the third one you are addressing networks and machines together. And this could be bunch of machines and

bunch of networks ok. So, that is the advantage of snort. So, the last line is actually very powerful it drops the packets ok.

(Refer Slide Time: 18:31)



### Rules in short

- The basis for logging or not logging a packet
- Can be more than one line long - now
  - Each line to be continued must be terminated with a ' \ ' » That is "space \"
- Generic syntax

```
rule_header (rule_options)
```

  - Rule header
    - » Action, addresses, ports, masks
  - Rule options
    - » Messages, what to look for, where to look

So, the rules in short ok, the basis for logging or not logging a packet and, it can be more than one line in length ok. So, if you want just similar to C program, I mean you just put a backslash and then, you are good to go you can go on extending ok. So, you have a rule header and rule options ok. The rule header can have action addresses ports and mask and rule options can have messages, what to look, where to look etcetera. What we will do is that, in the next session we will look into some more details of snort.

Thank you.