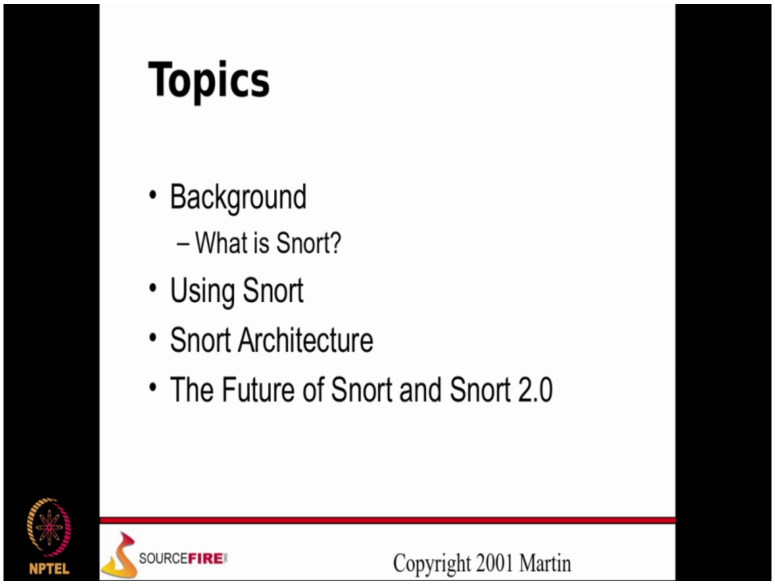
Information Security 3
Sri M J Shankar Raman,
Consultant Department of Computer Science and Engineering,
Indian Institute of Technology Madras
Module 58
Network Intrusion Detection Tool

So in this module we are basically going to be seeing another tool that is actually used for detecting any kind of intrusions on the network right. So this tool is basically called as snort and this tool is also very powerful and it has been actually in use for a long time as part of any kind of a network security auditing mechanism or network security prevention of

(Refer Slide Time: 00:38)



Topics

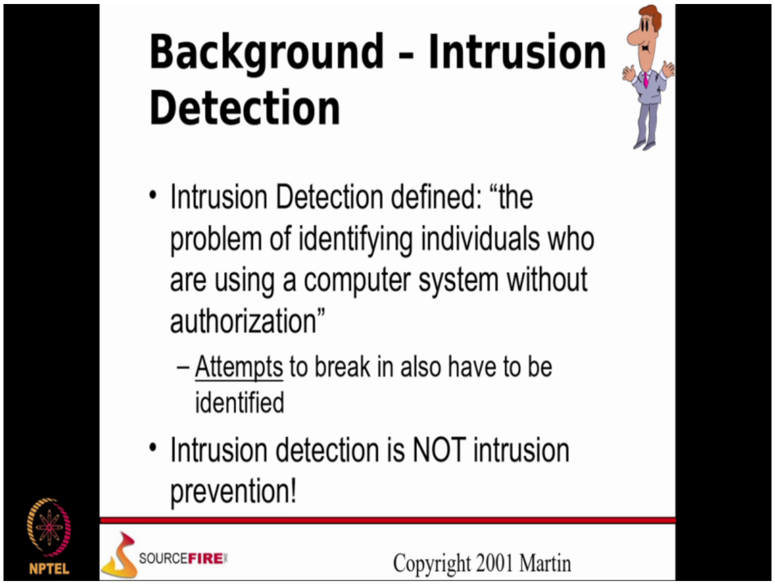
- Background
  - What is Snort?
- Using Snort
- Snort Architecture
- The Future of Snort and Snort 2.0

NPTEL
SOURCEFIRE

Copyright 2001 Martin

Any kind of attacks as well.

(Refer Slide Time: 00:41)



Background - Intrusion Detection

- Intrusion Detection defined: “the problem of identifying individuals who are using a computer system without authorization”
  - Attempts to break in also have to be identified
- Intrusion detection is NOT intrusion prevention!

NPTEL
SOURCEFIRE

Copyright 2001 Martin

So we will take a a quick understanding on what exactly is intrusion detection initially. So intrusion detection is basically define as a problem of identifying individuals who are using a computer system without authorization right,

So we will just try to explain this definition of an intrusion detection from the without authorization any system actually has an authorization to validate the user who is trying to make use of the systems, so if it is basically an OS we have a login mechanism typically first where we will have to login right.

Only after the credentials is successfully be given the system, the OS is actually going to recognized this particular user to be a a valid authentic user right, similarly if it is a network device like a router or a switch, I might have to give a password to validate myself saying that ok I am actually a valid user right, now when somebody says is the user without authorization we actually have a name called as a hacker or a attacker for him right.

Now when it comes to hacking there are two categories here, one who is a ethical hacker and one who is a a attacker who basically has a sole intention of trying to bring down a network server or a network service or further matter even the entire network right, so ethical hacker is somebody who is actually performing an audit of how secure a network is and at the end of it he will basically be giving a report so that the administrator can basically take some preventive actions.

For the network to be attacked by any unauthorized persons right, so that is basically what we are referring to here as a person without authorization right, now basic the purpose of this is we want to identify any kind of a person who is actually acting as an attacker and trying to use any kind of computer resource like maybe a system or trying to attack a network device like a router or whatever it is without authorization right.

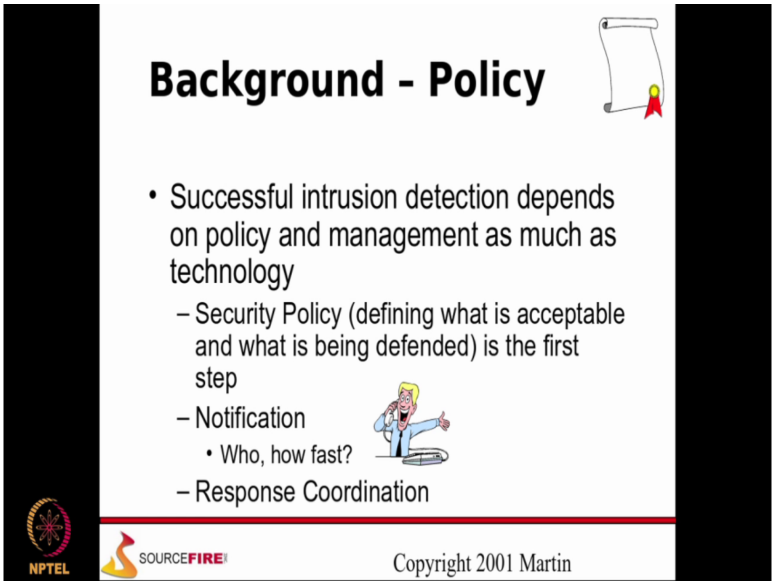
So without trying to authenticate himself and certifying that he is actually a valid authentic user anybody who is trying to basically bring down a particular server or a system or a network is basically has to be identify and the entire process of identification of these kind of attacks is what we were referring to here as intrusion detection right, now we will have to identify and sort of differentiate intrusion detection with intrusion prevention right.

Now intrusion prevention is a preventive mechanism that I actually try to employ after I start doing an intrusion detection right, over a period of time so over a period of time I sort of build my knowledge database in sort of detecting intrusions and based on this knowledge database that I have I put in mechanisms in place so that in if future intrusion that could happen can be prevented itself in a first place, right?

So if I have to basically give a very simple representation of intrusion detection vs intrusion prevention, intrusion detection is a sort of a post mortem and analysis after the intrusion has happened right, I take some corrective mechanisms so that I try to find out first maybe what kind of data has been compromised and secondly apply that understanding for me to effectively stop any kind of intrusions later on right, whereas intrusion prevention is a preventive mechanism before the problem is actually trying to happen right.

Before the problem of intrusion is really trying to happen, so when somebody talks of intrusion detection we will have to be very careful in understanding that we are not going to try to prevent an intrusion when we are talking of an intrusion detection right, so these are actually two different aspects and there is a very high possibility that we might be getting confused between the two of them which we will have to be very careful of when we are talking of intrusion detection verses intrusion prevention right,

(Refer Slide Time: 04:51)



Background - Policy

- Successful intrusion detection depends on policy and management as much as technology
  - Security Policy (defining what is acceptable and what is being defended) is the first step
  - Notification
    - Who, how fast?
  - Response Coordination

NPTEL
SOURCEFIRE

Copyright 2001 Martin

So successful intrusion detection will basically depend on what is the kind of a security policy that I really have right, and also what kind of management of my network as well as security that I am employing right, so on both these factors it is going to be very much dependent on as well as what is a kind of technologies that I am using right, so essentially if I basically have let say my management.

Tells that there is not enough money for budgeting on deployment for network security purposes, I am basically going to have a very weak intrusion detection mechanism and place because of which there is a possibility that I could be attacked very easily by an attacker outside my network right, secondly it also depends on the policy where as part of an business requirement s if I can for example say that I want only very minimal access to be allowed from externally.

I could actually blocked lot of things because of which the possibility of an attack itself is very limited, but on other hand if because of my specific business requirements I need to allow lot of traffic coming into my network from outside then I also need to put in place lot of restrictions as part of my security policy because with the security policy in turn is governed by my business needs right, so I need to put a lot of restrictions in my security policy.

Because of which my tools deployment and the kind of performance that I might be getting out of the network traffic might become extremely limited because you will have to understand the more strange and security policy that I try to deploy for for my network it is actually mutually contradictory with whatever amount of performance that I'll be getting right, so for example if I I part of security put lot of rules to be checked in right.

And I specify that I don't want any packet which is violating those rules to be allowed inside or allow outside of my network then for each and every packet that is going through my device right, my security device all those rules have to be checked in right, so more the security policy strange and security policy more number of rules that is there, more number of rules that is there, that many more checks that I'll have to do for every packet.

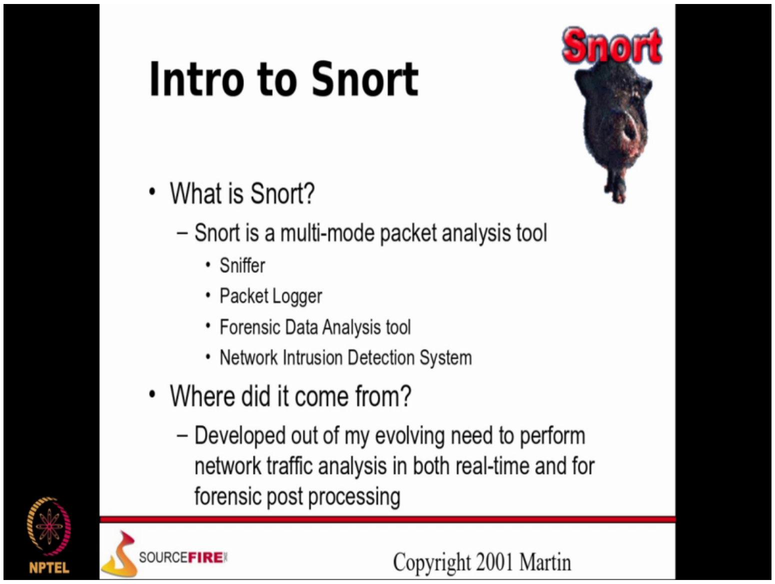
More the amount of checks that I'll have to do for every packet, the packet is going to take that much more time for it to either get out or get inside my network and that is going to have a direct bearing on the amount of latency that my application at the end user level is going to be facing right, so security policy depends on a whole plethora of factors and this has to be actually be sort of analyze and address for me to come to a conclusion on what kind of detection mechanisms.

That I'll have to deploy and then the next factor is notification right, so how quickly I am going to recognize and and detection and who is it that I am going to be monitoring and who is it that I am going to reporting to, so that again is basically part of my policy definition which I will have to basically define as part of my rules and then once I have detected that there has been an intrusion how is the entire response to that deduction going to be coordinated right.

So because if I detect that there has been an intrusion I will have to take some actions, now for each of those actions who is going to give an approval in my in my organization, in my network security organization how quickly that person is expected to respond for that approvals all those comes under as part of a response coordination right, so all these is is is basically parts of my security policy that I will define right from the point of view of what kind of traffic I am going to allow.

Or deny based on the business requirements that I my organization is currently in and once I detect an intrusion how am I going to notify, whom I going to notify, how quickly am I going to notify, thirdly once I have notified how is the entire response to that detection intrusion detection going to be coordinated.

(Refer Slide Time: 09:04)



Intro to Snort

- What is Snort?
  - Snort is a multi-mode packet analysis tool
    - Sniffer
    - Packet Logger
    - Forensic Data Analysis tool
    - Network Intrusion Detection System
- Where did it come from?
  - Developed out of my evolving need to perform network traffic analysis in both real-time and for forensic post processing

Snort
NPTEL
SOURCEFIRE

Copyright 2001 Martin

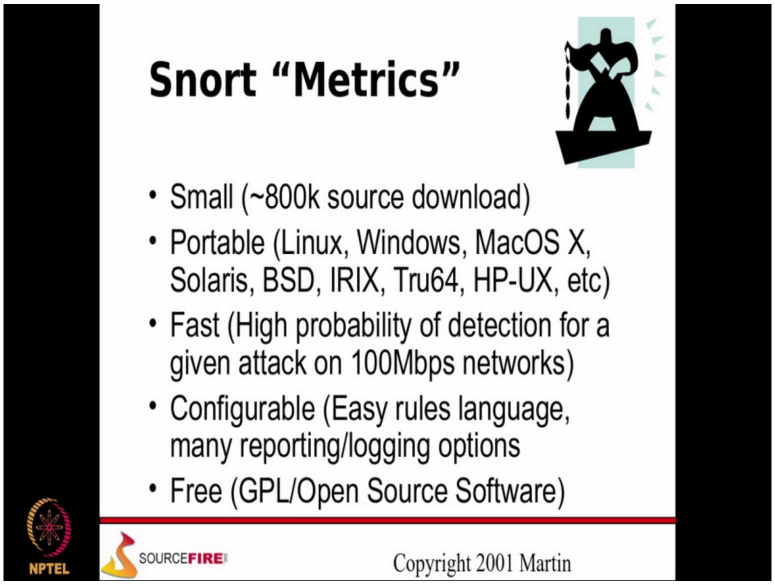
So what is snort so snort is basically a multi mode packet analysis tool which could actually act as a sniffer right, so it could actually sniffed the packets, then once it is sniffed that packets it could also do a packet logging mechanism, once it is logged the packet, it could also do a packet data analysis tool on what kind of data is actually getting transported on those packets, once it is done the data analysis and the logging it could then try to apply the network intrusion detection system principles and sort of identify packets that is basically creating possible problem if it is actually allowed inside the network right.

So we will have to understand sometimes that could also be a false positive or a false negative because all these governed by certain rules and configurations that are applied and has been predefined right, so sometimes those rules or configurations could also be resulting in flagging of false positive case wherein the packet cannot be really considered as an intrusion but because of it matching certain rules that has been specified it is reporting it as a positive case to be flagged as an intrusion right.

Os there is also certain possibilities that is there but the attempt on the maturity of my intrusion detection mechanism will always be to ensure that the amount of false positive that the tool is reporting that my specific tool insulation is reporting is kept to the bare minimum possible extent right, so it has been developed basically for a need to perform network traffic analysis and both real time as well as doing a a post mortem analysis for any kind of forensic analysis right.

So I want to do a a post mortem analysis of the network after let say it has been attacked right, so I need to do a a very quick analysis of the kind of different packets that is actually come in just before the attack and among those packets I want to quickly identify which of those packets could have been could have been the culprits which could have been the the result by which this attacker actually happened right, so I could use this basically for doing that analysis also right.

(Refer Slide Time: 11:16)



Snort "Metrics"

- Small (~800k source download)
- Portable (Linux, Windows, MacOS X, Solaris, BSD, IRIX, Tru64, HP-UX, etc)
- Fast (High probability of detection for a given attack on 100Mbps networks)
- Configurable (Easy rules language, many reporting/logging options
- Free (GPL/Open Source Software)

SOURCEFIRE Copyright 2001 Martin

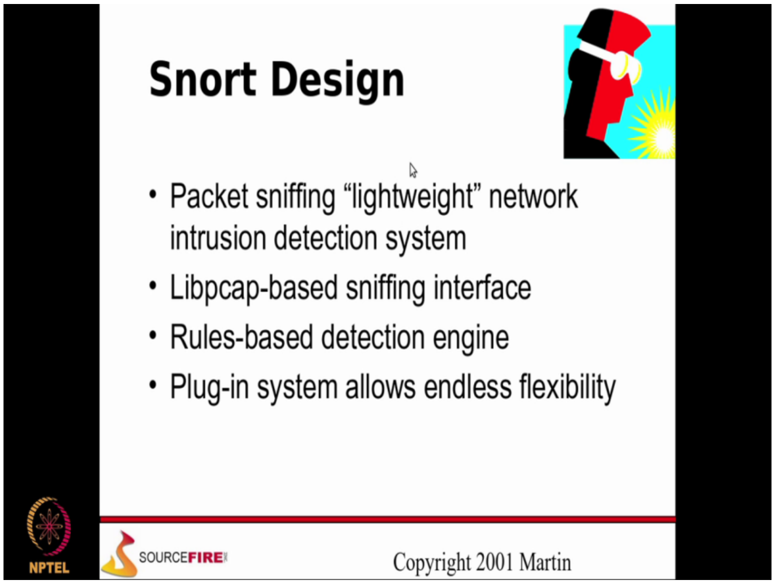
So the with respect to the different metrics on the snort implementation we'll have to understand that the code is actually not very huge, it is actually customizable depending on the kind of features that we require, we could also limit the amount of code that is actually used for building the product it's extremely portable because any kind of common platform that you named today or it is in very popular use any kind of operating system that is in common use, I have this software available on that right.

So in that way it is extremely portable it's very fast because it could actually operate on a 100 mbps networks and it could also operate on a gigi networks as close as possible to the lan speed right, so I wouldn't say that it will operate at the lan speed but it's extremely fast from the time for for with respect to the time that it is actually taking for doing the processing on a per packet basis right, so it is extremely configurable again so as I was telling.

I could essentially say what kind of packets I want to be really looking at what kind of rules I want those packets to be satisfied before it is marked as a a possible intrusion packet, undesirable intrusion, I could actually have configuration done for different kinds of reporting and logging mechanisms for the captured packets and most importantly it is free software that I could actually take, download, customize it, use it.

And then also very importantly if I have the capability and the technical skills I could also go ahead modify and then enhance it to suit my own requirements and subsequently submit my enhancements also back to the open those community, so this is basically governed by the GPL license.

(Refer Slide Time: 13:03)



Snort Design

- Packet sniffing "lightweight" network intrusion detection system
- Libpcap-based sniffing interface
- Rules-based detection engine
- Plug-in system allows endless flexibility

NPTEL
SOURCEFIRE

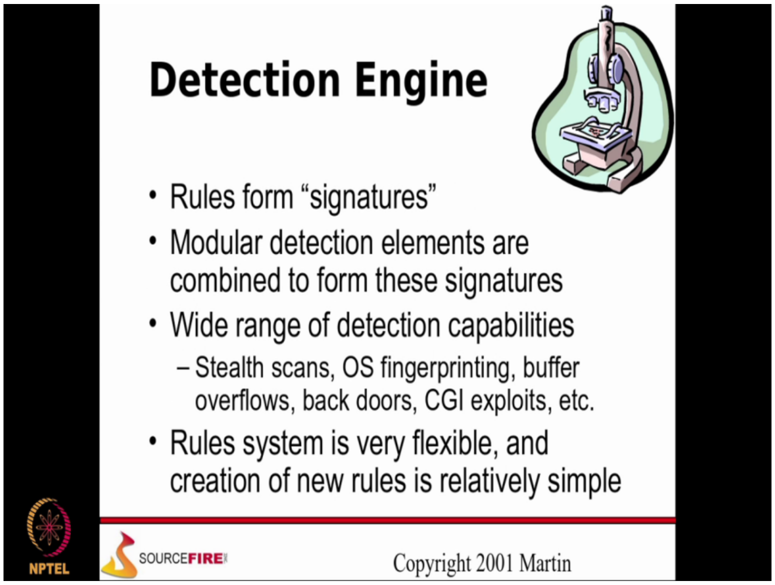
Copyright 2001 Martin

So essentially it's a packet sniffing light weight network intrusion the system that we were discussing right now, it is based on the lib p cap sniffing interface.

Very similar to the wire shark that we were actually discussing in our previous module, it's a rule based deduction engine so as I was telling, I will be specifying a certain set of rules that every packet that is coming into my network will be subjected to and as long as any packet is matching any of those rules that has been specified that packet will be actually flagged as a a detection intrusion detection packet, that needs to be analyzed separately right.

So I could also have different plug in modules on top of the existing base implementation which could be very much customized to my specific detection requirements or my own customization on the basic functionality or whatever so this is basically the the one of the main reason that is why this is actually a a very very customizable and a pluggable software and because of this is actually become a very very common intrusion detections software's, even in lot of commercially available security products in the market today.

(Refer Slide Time: 14:14)



Detection Engine

- Rules form "signatures"
- Modular detection elements are combined to form these signatures
- Wide range of detection capabilities
  - Stealth scans, OS fingerprinting, buffer overflows, back doors, CGI exploits, etc.
- Rules system is very flexible, and creation of new rules is relatively simple

Copyright 2001 Martin

NPTEL
SOURCEFIRE

So far as a detection engine goes I actually have rules based detection engines so what we called as a signature, so signature is nothing but a particular pattern that I expect to be seeing in the incoming packet and the moment I see the same signature pattern on the incoming packet, that packet will actually be flagged for an analysis to be done with a tax saying that this packet specifically.

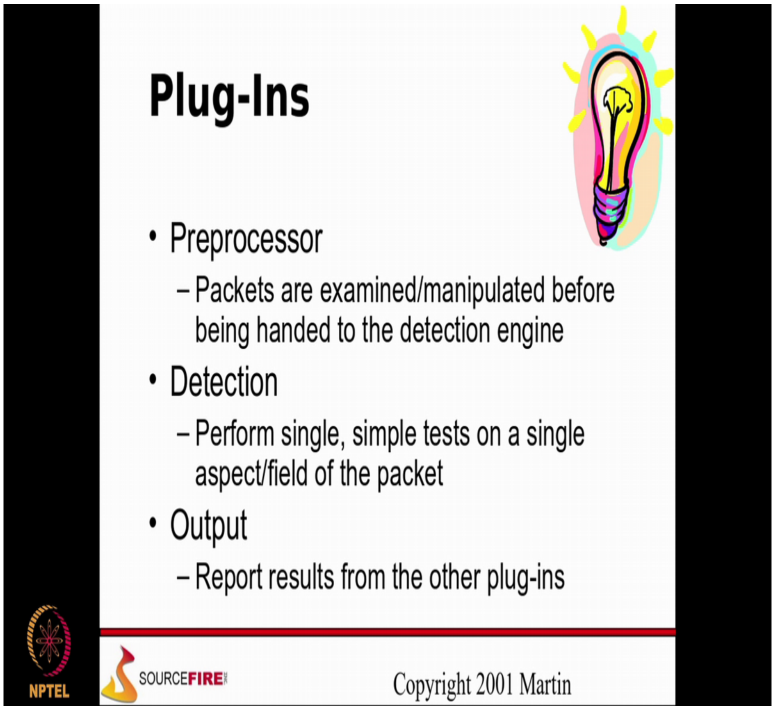
Looks to be like an intrusion packet from unauthorized user right, so I could really do very wide range of detection capabilities, I could find actually a stealth scans, I could do a voice fingerprinting, a a packet that is being attempted I could do an analysis detection of the packet of the buffer overflow, any kind of most of the back door entries that are possible and so on and so forth right, so these are the different typical kind of initial attacks that attacker would try to employ to get access.

Into the network, number one and number two once get access into the network to find out more details about each of the devices in the network for example the os fingerprinting as we were discussing in the earlier modules or network security it basically tries to find out what is the os that is running, not specifically what is a OS alone but what version of that os is running right, because with that kind of a detail I will be able to quickly come to a conclusion.

On what kind of possible loop holes or compromises or there in that particular os and in that particular OS version which the attacker will try to make use of and try to bring down the system right, so all these kind of things could be detected by putting in the the signatures for these different kinds of packets as part of the rules, so once it is put as a rules every packet that is being attempted to enter inside my network through the snort will be subjected to this comparison.

And if the comparison matches that packet will be silently flagged saying that this is possibly an intrusion packet that is actually coming in which needs further analysis right, so that's basically what the detection engine which is a core part of my snort ids is actually doing right.

(Refer Slide Time: 16:27)



Plug-Ins

- Preprocessor
  - Packets are examined/manipulated before being handed to the detection engine
- Detection
  - Perform single, simple tests on a single aspect/field of the packet
- Output
  - Report results from the other plug-ins

NPTEL

SOURCEFIRE

Copyright 2001 Martin

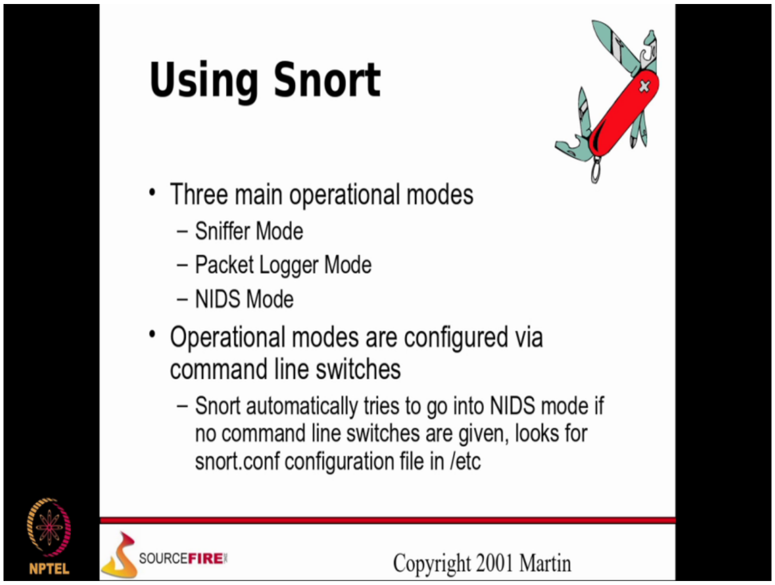
So there are lot of plug ins that are possible as we were discussing in the earlier slide, so I could really have a pre-processor plug in wherein I could actually have a newer packet which I would like to be examined or analyze.

Before being handed to the detection engine, so if I want to make any kind of modifications on the incoming packet before the direction engine picks it up I could do it as part of the pre processor, then I could do lot of customizations as part of the detection and once it is detected that it is matching any of the rules, I have actually could have the output the reported by my own customized plug in in in a particular format or in a particular specific output medium or whatever it is right.

So all these are different types of plug in at different levels of the packet capture and my detection capabilities that the snort is providing me.

So all these are basically customizations that I could really do, so in terms of the uses of snort I could effectively do a package sniffing a network intrusion detection, I could do a policy enforcement based on the security policy that the organization has defined like how we were discussing initially or the background requirement for the ids and I could also do a detection of the scan and any kind of an inbound trap that is actually coming in right,

(Refer Slide Time: 17:44)



Using Snort

- Three main operational modes
  - Sniffer Mode
  - Packet Logger Mode
  - NIDS Mode
- Operational modes are configured via command line switches
  - Snort automatically tries to go into NIDS mode if no command line switches are given, looks for snort.conf configuration file in /etc

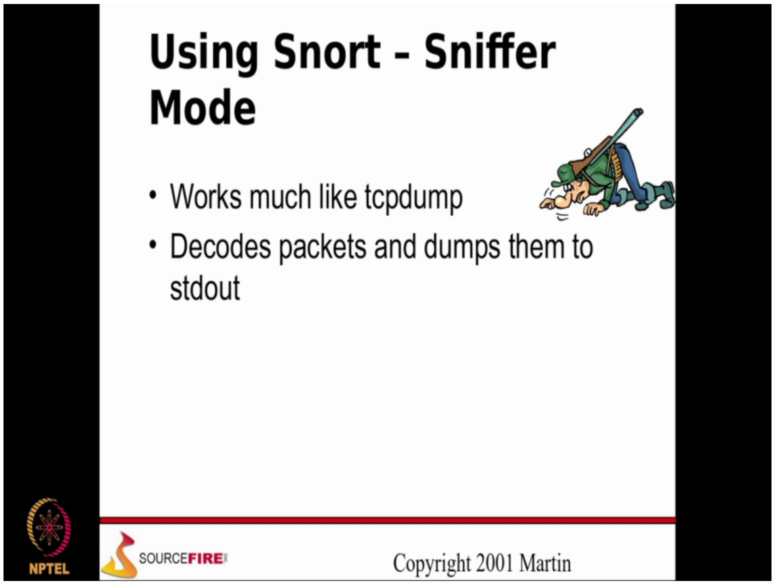
NPTEL
SOURCEFIRE

Copyright 2001 Martin

So three main operational modes, one is a plane sniffer mode, second one is a packet logger mode, when a sniffer mode I will just life the packets and then display it in a packet logger mode I would also sniffs the packet along with that I will go and log the packets in a particular format and in and in a particular medium and in a nids mode my detection engine also parallely runs wherein I will be able to do the detection.

Comparing it with a signatures that are there as part of my rules engine right, so all these operational modes whatever I want to run for the snort or typically configurable in my configuration file or via command line options when I try to bring up the application right,

(Refer Slide Time: 18:25)



Using Snort - Sniffer Mode

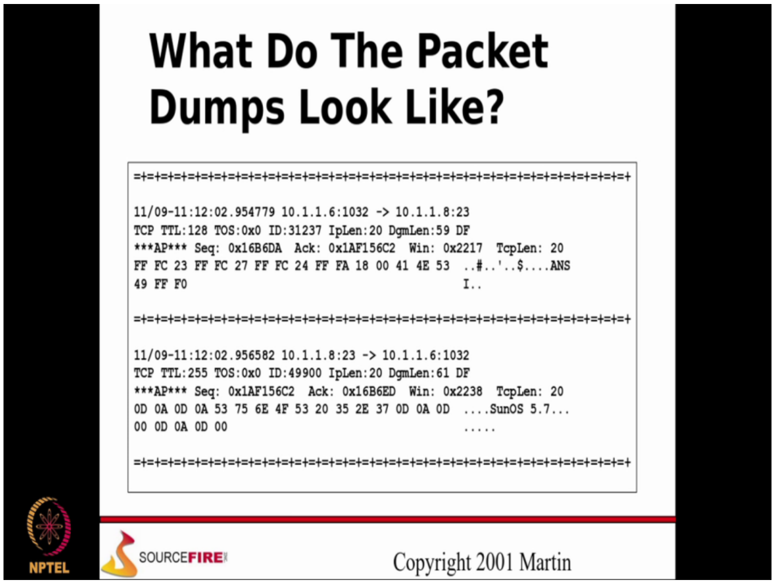
- Works much like tcpdump
- Decodes packets and dumps them to stdout

SOURCEFIRE

Copyright 2001 Martin

So sniffer mode it basically works like a tcp dump application where it just decode the software and dumps them, so it doesn't do anything else right,

(Refer Slide Time: 18:32)



What Do The Packet Dumps Look Like?

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/09-11:12:02.954779 10.1.1.6:1032 -> 10.1.1.8:23
TCP TTL:128 TOS:0x0 ID:31237 IpLen:20 DgmLen:59 DF
***AP*** Seq: 0x16B6DA  Ack: 0x1AF156C2  Win: 0x2217  TcpLen: 20
FF FC 23 FF FC 27 FF FC 24 FF FA 18 00 41 4E 53  ..#..'..$....ANS
49 FF F0                                         I..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/09-11:12:02.956582 10.1.1.8:23 -> 10.1.1.6:1032
TCP TTL:255 TOS:0x0 ID:49900 IpLen:20 DgmLen:61 DF
***AP*** Seq: 0x1AF156C2  Ack: 0x16B6ED  Win: 0x2238  TcpLen: 20
0D 0A 0D 0A 53 75 6E 4F 53 20 35 2E 37 0D 0A 0D  ....SunOS 5.7...
00 0D 0A 0D 00                                   .....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

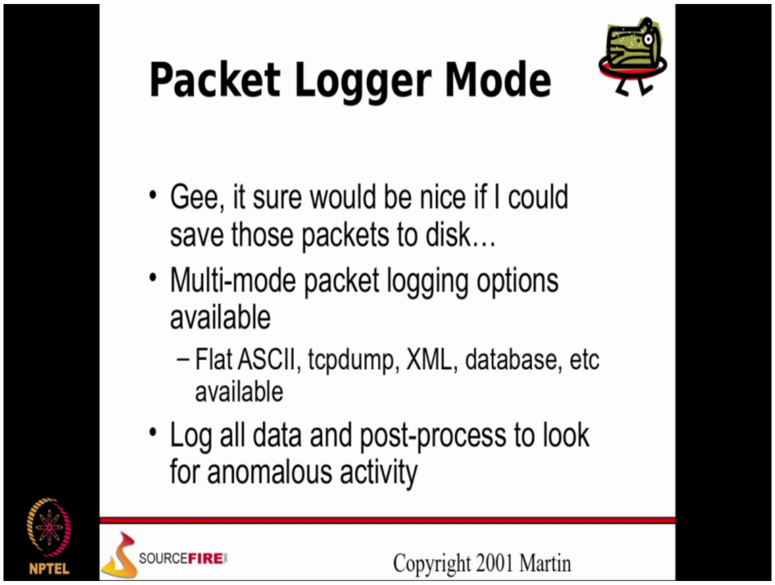
SOURCEFIRE

Copyright 2001 Martin

So the dump will look something like this it's just an example dump that has been given.

Where I could just get the headers because basically the tcp dump applications running in the back and forth is and the output from tcp dump is simply captured and that displayed into your standard output device which is typically by default your terminal window.

(Refer Slide Time: 18:52)



Packet Logger Mode

- Gee, it sure would be nice if I could save those packets to disk…
- Multi-mode packet logging options available
  - Flat ASCII, tcpdump, XML, database, etc available
- Log all data and post-process to look for anomalous activity

NPTEL

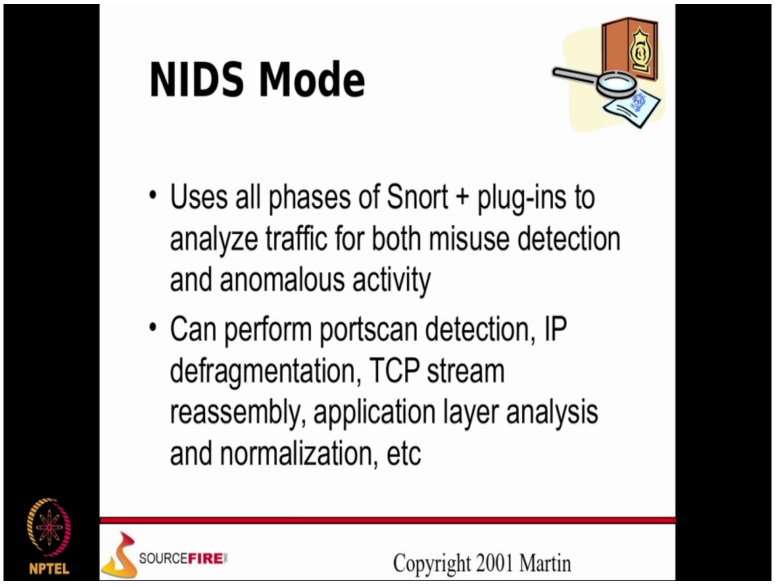
SOURCEFIRE

Copyright 2001 Martin

So in a packet logger mode I could actually take the take the packets and then store it onto disk and I have different types of logging options where I could just dump it into a normal ASCII file, normal ASCII text file I could dump it in a TCP dump format.

I could dump it in the xml, I could dump it in a database specific format, in specific data base and so on and so forth right, so once I dump it in this manner separately I might have other tools to do an analysis of it to detect any kind of unsuspicious activity of the packets through the packets that is actually coming in the network which has been captured in this packet logger mode.

(Refer Slide Time: 19:30)



NIDS Mode

- Uses all phases of Snort + plug-ins to analyze traffic for both misuse detection and anomalous activity
- Can perform portscan detection, IP defragmentation, TCP stream reassembly, application layer analysis and normalization, etc

NPTEL
SOURCEFIRE

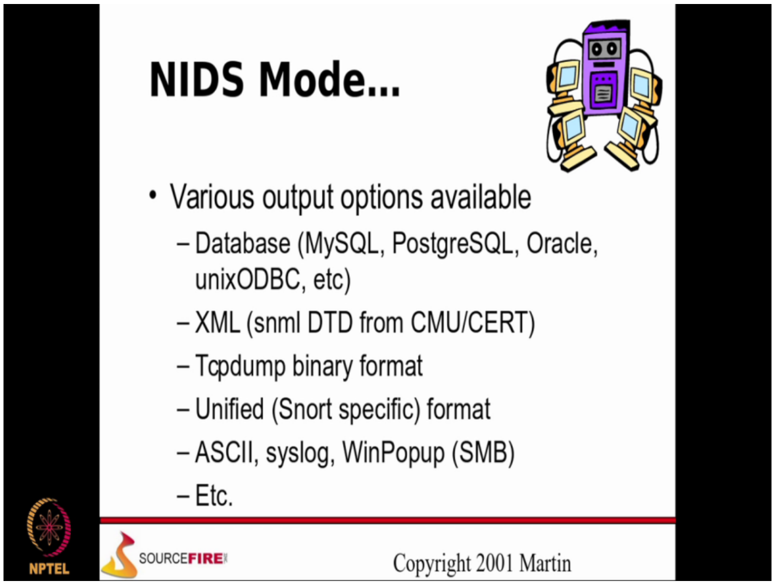
Copyright 2001 Martin

So in aNIDS mode apart from the the sniffer and the logging part I also do the analysis of the traffic then and there and try.

To do the detection of any kind of an anomalous activity straight right, so in that way I will get some real time inputs and as long as I have an effective reporting mechanism of reporting any kind of a suspicious activity, I will be able to quickly take some corrective action hopefully and then sort of prevent from extreme damage happening right, so whereas in a packet logger mode I could only do a post mortem reactive analysis and sort of prevent any kind of attacks in the future.

In NIDS mode as long as I have a effective logging and reporting mechanism including who and how the action should be taken on a the the NIDS reporting any kind of possible suspicious traffic, I will be able to stop from the attack getting to be in a severe damage mode right, so even if the attack happens I will be able to react it quickly and then sort of prevent any kind of very severe action right,

(Refer Slide Time: 20:40)



NIDS Mode...

- Various output options available
  - Database (MySQL, PostgreSQL, Oracle, unixODBC, etc)
  - XML (snml DTD from CMU/CERT)
  - Tcpdump binary format
  - Unified (Snort specific) format
  - ASCII, syslog, WinPopup (SMB)
  - Etc.

NPTEL
SOURCEFIRE

Copyright 2001 Martin

So again in the NIDS mode there are lot of different options that are actually available for dumping the output.

Once it has actually identified specific traffic which could actually be used for the final analysis and then the input from this analysis as we were discussing could go in as part of my updated rules engine so that any kind of a subsequent attack, a similar subsequent attack in the future could be prevented then and there immediately right, so this also helps me into sort of getting from a network intrusion detection mode into a intrusion prevention mode over a period of time in the future right.

So we've actually seen a couple of sample network security products like wire shark and snort just to give us an idea on what kind of things are possible likewise there are so many tools that are actually available with which we will be able to deploy these tools in different parts of my network to effectively try to ensure that the network is protected from different types of attack and then secondly even if the network is attacked there is a possibility that I will be able to quickly recover from the attack.

Without a a major chunk of my network getting affected right, so we we we just saw a couple of examples of the tools that are very commonly used but likewise there are so many other tools that are possible to be used and this was just a a sort of appetizer and to sort of give you a basic idea on what kind of tools are very commonly used as part of network security right.

Thank you.