17: Network Management and Monitoring

Last Modified:
4/21/2003 2:46:25 PM

Network Management Tasks

- Protecting the network (e.g. intrusion detection)
- Detecting failed components (interfaces, links, hosts, routers)
- Monitoring traffic patterns (recommend needed upgrades, cap certain types of traffic)
- Detect abnormal traffic (rapid changes in routing tables, huge spikes in BW usage)

Snort

- Detection/logging of packets matching filters/rule sets similar to Ethereal capture/display filters
- Three primary uses
  - Packet sniffer
  - Packet logger
  - Intrusion Detection System

Snort IDS

- Snort consists of three subsystems:
  - packet decoder (libpcap-based)
  - detection engine
  - logging and alerting subsystem
- Detection engine:
  - Rules form signatures
  - Modular detection elements are combined to form these signatures
  - Anomalous activity detection is possible: stealth scans, OS fingerprinting, invalid ICMP codes, etc.
  - Rules system is very flexible, and creation of new rules is relatively simple

Snort Rules

- Snort rules consist of two parts
  - Rule header
    - Specifies src/dst host and port
    - Alert tcp !128.119.0.0/16 any -> 128.119.166.5 any
    - Notice: negation, any in network 128.119.0.0
  - Rule options
    - Specifies flags, content, output message
    - (flags: SFAPR; msg: “Xmas tree scan”)

Writing Snort Rules

- Snort uses a simple rules language
- http://www.snort.org/writing_snort_rules.htm
- Rule header consists of
  - Rule Actions
    - Alert, Log, Pass Dynamic, activate, etc…
  - Protocol
    - Tcp, udp, icmp, etc…
  - IP Addresses
    - Source, dest, CIDR mask
  - Port numbers
    - Source, dest, range
  - Direction
  - Negation

Simple examples

- log tcp any any -> $SMTP 23 (msg: “telnet to the mail server!”;)
- alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg: “TELNET login incorrect”; content: “Login incorrect”; flags: A+;)
- alert icmp any any -> any any (msg:”ICMP Source Quench”; itype: 4; icode: 0;)

Prewritten Rulesets

- Snort comes packaged with a number of prewritten rulesets
  - include bad-traffic.rules
  - include exploit.rules
  - include scan.rules
  - include finger.rules
  - include ftp.rules
  - include telnet.rules
  - include smtp.rules
  - include rpc.rules
  - include rservicesrules
  - include dos.rules
  - include ddos.rules
  - include dns.rules
  - include tftp.rules
  - include web-cgi.rules
  - include web-coldfusion.rules
  - include web-frontpage.rules
  - .....

Vulnerability databases

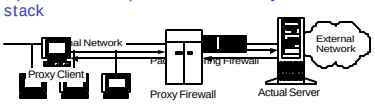
- Rules correlated to common databases
- Bugtraq
  - http://www.securityfocus.com/cgi-bin/vulns.pl
  - Ex. Bugtraq id 2283: 23-01-2001: Lotus Domino Mail Server 'Policy' Buffer Overflow Vulnerability
- ArachNIDS
  - http://www.whitehats.com/ids/index.html
- Common Vulnerabilities and Exposures
  - http://cve.mitre.org

Firewalls

- Gateway machines through which all traffic passes
- Can *stop* rather than simply log traffic that matches rules/filters

Types of firewalls

- Packet Filtering firewall
  - Operate on transport and network layers of the TCP/IP stack



- Application Gateways/Proxies
  - Operate on the application protocol level

Packet Filtering Firewall

- Operate on transport and network layers of the TCP/IP stack
- Decides what to do with a packet depending upon the following criteria:
  - Transport protocol (TCP,UDP,ICMP),
  - Source and destination IP address
  - The source and destination ports
  - ICMP message type/code
  - Various TCP options such as packet size, fragmentation etc
- A lot like Ethereal capture/display filters

Packet Filtering

- Example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23.
  - All incoming and outgoing UDP flows and telnet connections are blocked.
- Example 2: Block inbound TCP segments with ACK=0 or with SYN bit set and ACK bit unset.
  - Prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

Packet Filtering Firewall: Terminology

- Stateless Firewall: The firewall makes a decision on a packet by packet basis.
- Stateful Firewall : The firewall keeps state information about transactions (connections).
- NAT - Network Address translation
  - Translates public IP address(es) to private IP address(es) on a private LAN.
  - We looked at this already (must be stateful)

Packet Filtering Firewall: Functions

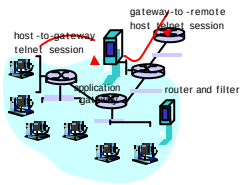
- Forward the packet(s) on to the intended destination
- Reject the packet(s) and notify the sender (ICMP dest unreach/admin prohibited)
- Drop the packet(s) without notifying the sender.
- Log accepted and/or denied packet information
- NAT - Network Address Translation

Application Gateway (Proxy Server)

- Operate at the application protocol level. (Telnet, FTP, HTTP)
- Filters packets on application data as well as on IP/TCP/UDP fields
- Application Gateways "Understand" the protocol and can be configured to allow or deny specific protocol operations.
- Typically, proxy servers sit between the client and actual service. Both the client and server talk to the proxy rather than directly with each other.

Application gateways

- Example: allow select internal users to telnet outside.



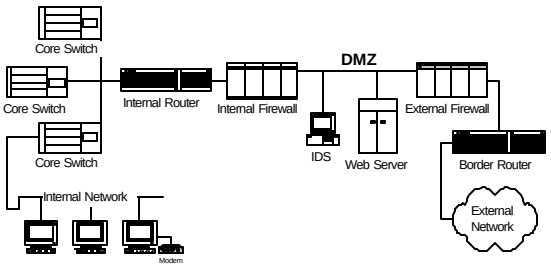
1. Require all telnet users to telnet through gateway.
2. For authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. Firewall filter blocks all telnet connections not originating from gateway.

Firewall Hardware/Software

- Dedicated hardware/software application such as Cisco PIX Firewall which filters traffic passing through the multiple network interfaces.
- A Unix or Windows based host with multiple network interfaces, running a firewall software package which filters incoming and outgoing traffic across the interfaces.
- A Unix or Windows based host with a single network interface, running a firewall software package which filters the incoming and outgoing traffic to the individual interface.

Firewall Architecture

In the real world, designs are far more complex



Limitations of firewalls and gateways

- IP spoofing: router can't know if data "really" comes from claimed source
- If multiple app's. need special treatment, each has own app. gateway.
- Client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- Filters often use all or nothing policy for UDP.
- Tradeoff: degree of communication with outside world, level of security
- Many highly protected sites still suffer from attacks.

Managing the network?

- autonomous systems (network under a single administrative control): 100s or 1000s of interacting hw/sw components
  - Many complex pieces…that can break
    - Hardware (end hosts, routers, hubs, cabling)
    - Software
  - Something is broken – where?
    - What is normal? What is abnormal?
  - Planning for the future – where is the bottleneck?
- Need information stream from remote components

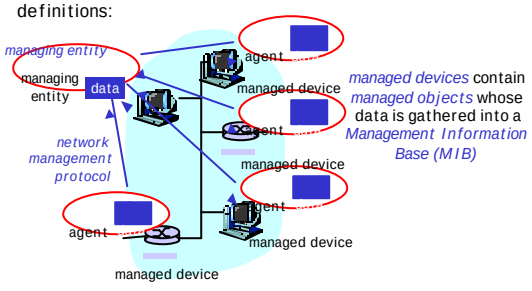
Network Management Architecture

- (1) a network manager
- (2) a set of managed remote devices
- (3) management information bases (MIBs)
- (4) remote agents that report MIB information and take action under the control of the network manager
- (5) a protocol for communicating between the network manager and the remote devices

Network Operations Center (NOC) = control center

Infrastructure for network management

definitions:



Network Management standards

OSI CMIP

- Common Management Information Protocol
- designed 1980's: *the* unifying net management standard
- too slowly standardized

SNMP: Simple Network Management Protocol

- Internet roots (Simple Gateway Monitoring Protocol, SGMP)
- started simple
- deployed, adopted rapidly
- growth: size, complexity
- *de facto* network management standard

SNMP overview: 4 key parts

- SNMP protocol
  - convey manager<->managed object info, commands
- Structure of Management Information (SMI):
  - data definition language for MIB objects, format of data to be exchanged
  - Protocol independent type language
- Management information base (MIB):
  - distributed information store of network management data, collection of MIB objects
- security, administration capabilities
  - major addition in SNMPv3

SMI: data definition language

Purpose: syntax, semantics of management data well-defined, unambiguous

- base data types:
  - straightforward, boring
- Higher level structs
  - OBJECT-TYPE
  - MODULE_IDENTITY

SMI Basic Data Types

INTEGER
Integer32
Unsigned32
OCTET STRING
OBJECT IDENTIFIED
IPaddress
Counter32
Counter64
Guage32
Tie Ticks
Opaque

OBJECT-TYPE

- SYNTAX = basic type of this object
- MAX-ACCESS = operations allowed on the object (read, write, create, notify)
- STATUS = current/valid, obsolete (should not be implemented), deprecated (implemented for backwards compatibility)
- DESCRIPTION = comment, human readable description

```
ipInDelivers OBJECT-TYPE
   SYNTAX Counter32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION "The
       total number of
       input datagrams
       successfully
       delivered to IP
       user-protocols
       (including
       ICMP)."
::= { ip 9 }
```


MODULE-IDENTITY

- MODULE-IDENTITY construct allows related objects to be grouped together within a "module."
- Contains the OBKECT-TYPE constructs for each object in the module
- Plus contact and description information

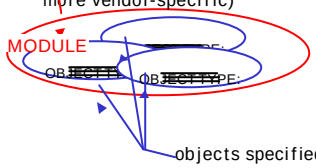
```
ipMIB MODULE-IDENTITY

    LAST-UPDATED "941101000Z"
    ORGANZATION "IETF SNPv2
        Working Group"
    CONTACT-INFO
        " Keith McCloghrie
        ....."
    DESCRIPTION
        "The MIB module for
        managing IP and ICMP
        implementations, but
        excluding their
        management of
        IP routes."
    REVISION "019331000Z"
    .........

::= {mib-2 48}
```


SNMP MIB

MIB module specified via SMI
MODULE-IDENTITY
(100+ standards-based MIBs written by IETF, more vendor-specific)



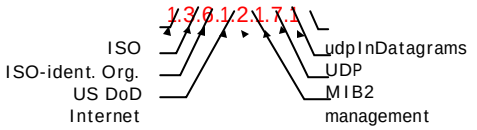
objects specified via SMI
OBJECT-TYPE construct

SNMP Naming

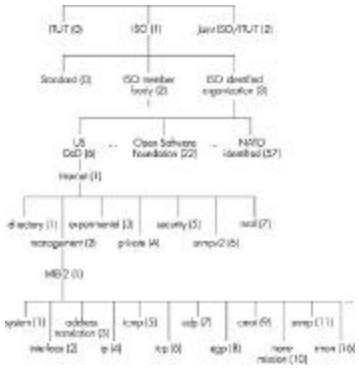
question: how do we keep track of/name every possible standard object (protocol, data, more..) in every possible network standard??

answer: ISO Object Identifier tree:

- hierarchical naming of all objects
- each branchpoint has name, number



OSI Object Identifier Tree



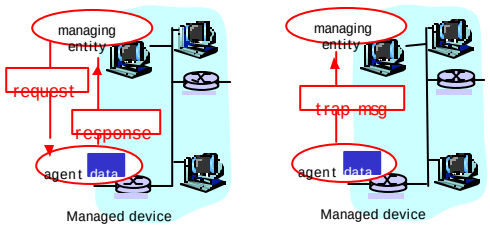
Check out www.alvestrand.no/harald/objectid/top.html

MIB example: UDP module

Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	total # datagrams delivered at this node
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	# underliverable datagrams no app at portl
1.3.6.1.2.1.7.3	UDInErrors	Counter32	# undeliverable datagrams all other reasons
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	# datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	one entry for each port in use by app, gives port # and IP address

SNMP protocol

Two ways to convey MIB info, commands:



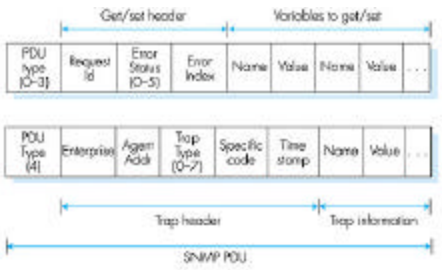
request/response mode: Give me your regular report

trap mode: Better hear about this now!

SNMP protocol: message types

Message type	Function
GetRequest GetNextRequest GetBulkRequest	Mgr-to-agent: "get me data" (instance,next in list, block)
InformRequest	Mgr-to-Mgr: here's MIB value
SetRequest	Mgr-to-agent: set MIB value
Response	Agent-to-mgr: value, response to Request
Trap	Agent-to-mgr: inform manager of exceptional event

SNMP protocol: message formats

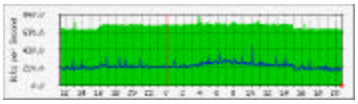


SNMP security and administration

- encryption: DES-encrypt SNMP message
- authentication: compute, send Message Integrity Code (MIC) MIC(m,k): compute hash (MIC) over message (m), secret shared key (k)
- protection against playback: use nonce
- view-based access control
  - SNMP entity maintains database of access rights, policies for various users
  - database itself accessible as managed object!

Multi Router Traffic Grapher (MRTG)

- SNMP client
- Will gather data from remote machines via SNMP
- Graphs changes in info over time



Outtakes

Packet Filtering Firewall: Disadvantages

- Filters can be difficult to configure. It's not always easy to anticipate traffic patterns and create filtering rules to fit.
- Filter rules are sometimes difficult to test
- Packet filtering can degrade router performance
- Attackers can "tunnel" malicious traffic through allowed ports on the filter.

Application Gateway (Proxy Server): Disadvantages

- Requires modification to client software application
- Some client software applications don't accommodate the use of a proxy
- Some protocols aren't supported by proxy servers
- Some proxy servers may be difficult to configure and may not provide all the protection you need.

Snort: Sample IDS output

- Apr 12 01:56:21 ids snort: EXPLOIT sparc setuid 0: 218.19.15.17:544 → xxx.yyy.zzz.41:37987
- Apr 12 01:56:21 ids snort: EXPLOIT x86 NOOP: 23.91.17.7:544 → xxx.yyy.zzz.41:37987
- Apr 12 07:31:03 ids snort: ICMP Nmap2.36BETA or HPING2 Echo : 63.26.255.221 → xxx.yyy.zzz.34
- Apr 12 09:59:38 ids snort: RPC portmap request rstatd: 28.11.67.132:1033 → xxx.yyy.zzz.29:111
- Apr 12 13:20:05 ids snort: ICMP Nmap2.36BETA or HPING2 Echo : 12.13.1.67 → xxx.yyy.zzz.126
- Apr 12 14:13:22 ids snort: RPC portmap request rstatd: 134.1.5.12:3649 → xxx.yyy.zzz.29:111
- Apr 12 20:19:34 ids snort: BACKDOOR back orrifice attempt: 209.255.213.130:1304 → xxx.yyy.zzz.241:31337
- Apr 12 22:53:52 ids snort: DNS named iquery attempt: 209.126.168.231:4410 → xxx.yyy.zzz.23:53

Example: smtp.rules

- alert tcp $EXTERNAL_NET any -> $SMTP 25 (msg:"SMTP RCPT TO overflow"; flags:A+; content:"rcpt to|3a|"; dsize:>800; reference:cve,CAN-2001-0260; reference:bugtraq,2283; classtype:attempted-admin; sid:654; rev:1;)
- alert tcp $EXTERNAL_NET 113 -> $SMTP 25 (msg:"SMTP sendmail 8.6.9 exploit"; flags: A+; content:"|0a|D/"; reference:arachnids,140; reference:cve,CVE-1999-0204; classtype:attempted-admin; sid:655; rev:1;)
- alert tcp $EXTERNAL_NET any -> $SMTP 25 (msg:"SMTP expn root"; flags: A+; content:"expn root"; nocase; reference:arachnids,31; classtype:attempted-recon; sid:660; rev:2;)