TED (10)-4079/5059 Reg. No...........................

(REVISION-2010) Signature.........................

FIFTH/SIXTH SEMESTER DIPLOMA EXAMINATION IN ENGINEERING/
TECHNOLOGY-MARCH, 2014

INFORMATION SECURITY

(For Vth semester CT and for VIth semester CM and IF)

[*Time:* 3 hours

(Maximum marks: 100)

Marks

PART –A

I Answer the following questions in one or two sentences. Each question carries 2 marks

1. **Write different types of security mechanism.**

   Peer-Entity Authentication
   Data-origin Authentication
   Connectionless confidentiality
   Connectionless Integrity

2. **In general terms ,list he means of authenticating a user identity**

   There are four general means of authenticating a user's identity, which can be used alone or in combination:

   • Something the individual knows: Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions.

   • Something the individual possesses: Examples include electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a *token*.

   • Something the individual is (static biometrics): Examples include recognition by fingerprint, retina, and face.

   • Something the individual does (dynamic biometrics): Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

3. **State the purpose of salt in Unix password scheme**

   In older implementations, the salt is related to the time the password is assigned to the user. Newer implementations use a pseudorandom or random number

4. **Define CPU emulator**

CPU emulator: A software-based virtual computer that interprets instructions in an executable file rather than executing them on the underlying processor.

5. **State intrusion prevention system.**

   Intrusion Prevention System (IPS): Reports violations and prevents attacks from occurring

   - Does inline processing, similar to a Firewall: drop packets, reset connections, route suspicious traffic for analysis
   - Problems: Delays in processing; bottleneck
   - Since IDS/IPS have high rate of False Positives, they require extensive optimization

PART—B

II Answer *any five* of the following. Each question carries 6 marks.

1. **Discuss security functional requirements**

   Technical measures

   Access control; identification & authentication; system & communication protection; system & information integrity

   Management controls and procedures

   Awareness & training; audit & accountability; certification, accreditation, & security assessments; contingency planning; maintenance; physical & environmental protection; planning; personnel security; risk assessment; systems & services acquisition

   Overlapping technical and management

   Configuration management; incident response; media protection

2. **Illustrate message authentication using MAC**

   One authentication technique involves the use of a secret key to generate a small block of data, known as a message authentication code, that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K_{AB}. When A has a message to send to B, it calculates the message authentication code as a function of the message and the key: $\text{MAC}_M = \text{F}(K_{AB}, M)$. The message plus code are transmitted to the intended recipient. The recipient performs the same calculation on the received message, using the same secret key, to generate a new message authentication code. The received code is compared to the calculated code

3. **Describe the categories of access control policies**

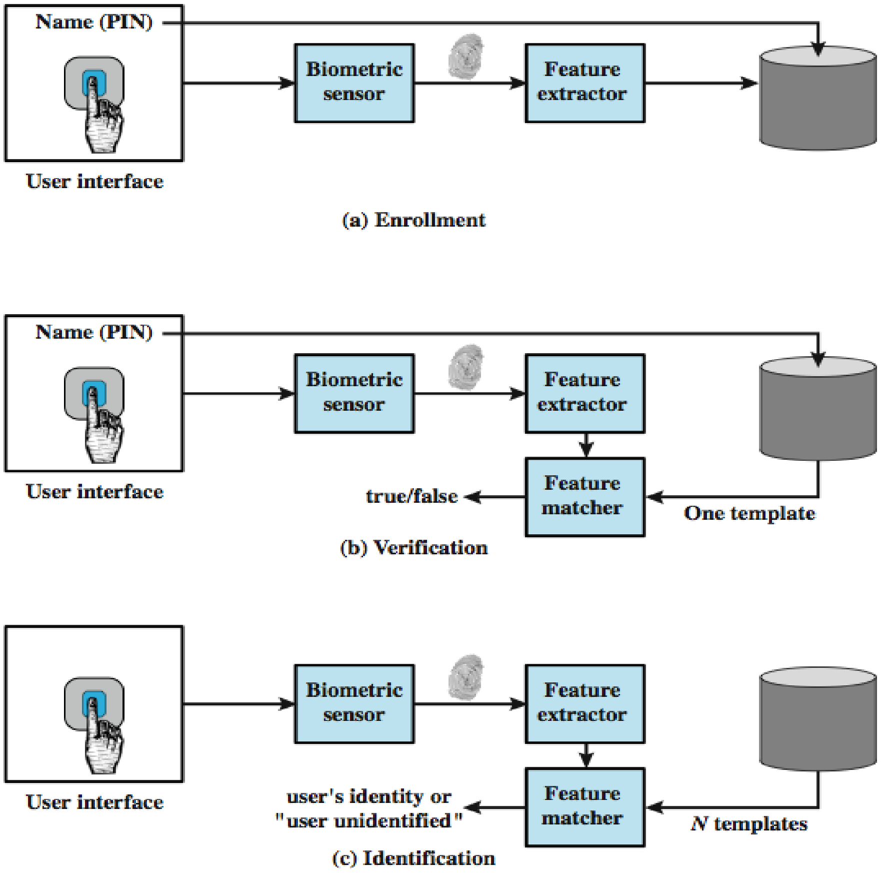
   An access control policy, which is embodied in an authorization database, dictates what types of access are permitted, under what circumstances, and by whom. Access control policies are generally grouped into the following access control categories:

• Discretionary access control (DAC): based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do. This policy is termed *discretionary* because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.

• Mandatory access control (MAC): based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed *mandatory* because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource.

• Role-based access control (RBAC): based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.

4. With neat diagram explain the operation of biometric authentication system



(a) Enrollment

(b) Verification

(c) Identification

.For a biometric system, the user presents a name and, typically, some type of password or PIN to the system. At the same time the system senses some biometric characteristic of this user (e.g. fingerprint of right index finger). The system digitizes the input and then extracts a set of features that can be stored as a number or set of numbers representing this unique biometric characteristic ;this set of numbers is referred to as the user’s template. The user is now enrolled in the system, which maintains for the user a name

(ID), perhaps a PIN or password, and the biometric value. Depending on application, user authentication on a biometric system involves either verification or identification. Verification is analogous to a user logging on to a system by using a memory card or smart card coupled with a password or PIN. For biometric verification, the user enters a PIN and also uses a biometric sensor.

5. **List the requirements of intrusion detection system**

- run continually with minimal human supervision.
- be fault tolerant in the sense that it must be able to recover from system crashes and re initializations.
- resist subversion. The IDS must be able to monitor itself and detect if it has been modified by an attacker.
- impose a minimal overhead on the system where it is running.
- be able to be configured according to the security policies of the system that is being monitored.
- be able to adapt to changes in system and user behavior over time.
- be able to scale to monitor a large number of hosts.
- provide graceful degradation of service in the sense that if some components of the IDS stop working for any reason, the rest of them should be affected as little as possible.
- allow dynamic reconfiguration; that is, the ability to reconfigure the IDS without having to restart it.

6. **State the uses of BOTs**

A bot (robot), also known as a zombie or drone, is a program that secretly takes over hundreds or thousands of Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the bot's creator. The collection of bots often is capable of acting in a coordinated manner; referred to as a botnet. A botnet exhibits three characteristics: the bot functionality, a remote control facility, and a spreading mechanism to propagate the bots and construct the botnet. Some uses of bots include: distributed denial-of-service attacks, spamming, sniffing traffic, keylogging, spreading new malware, installing advertisement add-ons and browser helper objects (bhos), attacking irc chat networks, manipulating online polls/games.

The remote control facility is what distinguishes a bot from a worm. A typical means of implementing the remote control facility is on an IRC (Internet relay chat) server. More recent botnets tend to avoid IRC mechanisms and use covert communication channels via protocols such as HTTP

7. **List any six characteristics of bastion host**

- executes a secure version of its operating system, making it a trusted system.
- only essential services are installed on the bastion host. These include proxy applications such as Telnet, DNS, FTP, SMTP, and user authentication.
- may require additional authentication before a user is allowed access to the proxy services, and may require its own authentication before granting user access.
- each proxy is configured to support only a subset of the application’s command set.
- each proxy is configured to allow access only to specific host systems.

• each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection.

PART—C

(Answer one full question from each unit. Each question carries 15 marks.)

UNIT--I

III (a) briefly explain security services in OSI security architecture. **8**

ITU-T Recommendation X.800, *Security Architecture for OSI*, defines a systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. The OSI security architecture is useful to managers as a way of organizing the task of providing security. The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as:
• Security attack: Any action that compromises the security of information owned by an organization. cf. network security attacks slide earlier
• Security mechanism: A mechanism that is designed to detect, prevent, or recover from a security attack. cf. functional requirements from previous slide or Table 1.6 in text.
• Security service: A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

- ➢ Authentication services
    - ♦ Peer entity authentication
    - ♦ Data origin Authentication
- ➢ Access Control
- ➢ Data Confidentiality
    - ♦ Connection confidentiality
    - ♦ Connectionless confidentiality
    - ♦ Selective field confidentiality
    - ♦ Traffic flow confidentiality
- ➢ Availability
- ➢ Data Integrity
- ➢ Non Repudiation

(b) List the key elements in computer and network security taxonomy. **7**

The key elements are:
• Action: A step taken by a user or process in order to achieve a result
• Target: A computer or network logical entity or physical entity
• Event: An action directed at a target that is intended to result in a change of state, or status, of the target
• Tool: A means of exploiting a computer or network vulnerability
• Vulnerability: A weakness in a system allowing unauthorized action

• Unauthorized result: An unauthorized consequence of an event
• Attack: A series of steps taken by an attacker to achieve an unauthorized result
• Attacker: An individual who attempts one or more attacks in order to achieve an objective
• Objectives: The purpose or end goal of an incident
• Incident: a group of attacks that can be distinguished from other attacks because of the distinctiveness of the attackers, attacks, objectives, sites, and timing

OR

IV (a) Discuss the requirements of hash functions. **8**

- ➢ applied to any size data
- ➢ H produces a fixed-length output.
- ➢ H(*x*) is relatively easy to compute for any given *x*
- ➢ one-way property
  - ● computationally infeasible to find *x* such that H(*x*) = *h*
- ➢ weak collision resistance
  - ● computationally infeasible to find *y* ≠ *x* such that H(*y*) = H(*x*)
- ➢ strong collision resistance
  - ● computationally infeasible to find any pair (*x*, *y*) such that H(*x*) = H(*y*)

(b) Illustrate stream cipher and block cipher encryption. **7**

- A block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size for each plaintext block. The algorithm processes longer plaintext amounts as a series of fixed-size blocks. Typically, symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block. Plaintext sources must be broken up into a series of fixed-length block for encryption by a symmetric block cipher. The simplest approach to multiple-block encryption is known as electronic codebook (ECB) mode, in which plaintext is handled b bits at a time and each block of plaintext is encrypted using the same key. Typically b=64 or b=128. Figure 2.3a here shows the ECB mode. A plaintext of length nb is divided into n b-bit blocks. Each block is encrypted using the same algorithm and the same encryption key, to produce a sequence of n b-bit blocks of ciphertext
- A stream cipher processes the input elements continuously, producing output one element at a time. Although block ciphers are far more common, there are certain applications in which a stream cipher is more appropriate. . The output of a pseudorandom number generator (the keystream), is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation. With a properly designed pseudorandom number generator,

UNIT II

V (a) Compare memory card and smart card. **6**

- **Memory Card**
  - store but do not process data
  - magnetic stripe card, e.g. bank card
  - electronic memory card
  - used alone for physical access
  - with password/PIN for computer use
  - drawbacks of memory cards include:
    - need special reader
    - loss of token issues
    - user dissatisfaction
- **Smartcard**
  - credit-card like
  - has own processor, memory, I/O ports
    - wired or wireless access by reader
    - may have crypto co-processor
    - ROM, EEPROM, RAM memory
  - executes protocol to authenticate with reader/computer

(b) Discuss various security issues for user authentication **9**

- **Authentication Security Issues**

- **Client attacks** are those in which an adversary attempts to achieve user authentication without access to the remote host or to the intervening communications path. The adversary attempts to masquerade as a legitimate user. e.g. in a password-based system, the adversary may attempt to guess the likely user password.
- **Host attacks** are directed at the user file at the host where passwords,token passcodes, or biometric templates are stored.
- **Eavesdropping** refers to an adversary's attempt to learn the password by observing the user, finding a written copy of the password, keystroke logging, etc.
- **Replay** attacks involve an adversary repeating a previously captured user response. The most common countermeasure to such attacks is the challenge-response protocol.
- **Trojan horse attack**, an application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric. The adversary can then use the captured information to masquerade as a legitimate user**.**
- **A denial-of-service attack** attempts to disable a user authentication service by flooding the service with numerous authentication attempts.

OR

VI (a) Explain about access control matrix **5**

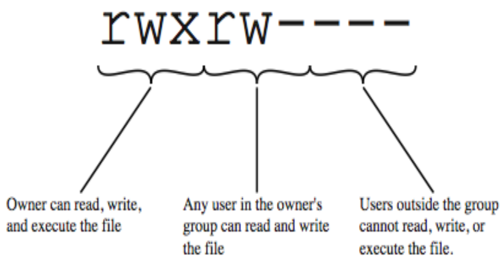
For discretionary access control, a general approach to access control as exercised by an operating system or a database management system is that of an access matrix. One dimension of the matrix consists of identified subjects that may attempt data access.. Each entry in the matrix indicates the access rights of that subject for that object. In practice, an access matrix is usually sparse and is implemented by decomposition in one of two ways

The matrix may be decomposed by columns, yielding **access control lists** (ACLs

- . The ACL may contain a default, or public, entry. This allows users that are not explicitly listed as having special rights to have a default set of rights. Elements of the list may include individual users as well as groups of users. When it is desired to determine which subjects have which access rights to a particular resource, ACLs are convenient, because each ACL provides the information for a given resource. However, this data structure is not convenient for determining the access rights available to a specific user.

Decomposition by rows yields **capability tickets** A capability ticket specifies authorized objects and operations for a user. Each user has a number of tickets and may be authorized to loan or give them to others. Because tickets may be dispersed around the system, they present a greater security problem than access control lists.. These tickets would have to be held in a region of memory inaccessible to users.

(b) Discuss Unix file access control **10**


rwxrw----
Owner can read, write, and execute the file
Any user in the owner's group can read and write the file
Users outside the group cannot read, write, or execute the file.

If the “set user ID”(SetUID) and “set group ID”(SetGID) are set on an executable file, the operating system functions as follows. When a user (with execute privileges for this file) executes the file,the system temporarily allocates the rights of the user’s ID of the file creator, or the file’s group, respectively, to those of the user executing the file. These are known as the “effective user ID”and “effective group ID” and are used in addition to the “real user ID” and “real group ID” of the executing user when making access control decisions for this program.

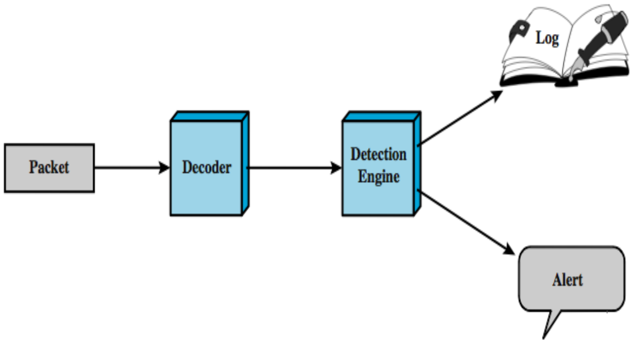
Alternatively, when applied to a directory, the SetGID permission indicates that newly created files will inherit the group of this directory. The SetUID permission is ignored.

The final permission bit is the “Sticky”bit. When set on a file, this originally indicated that the system should retain the file contents in memory following execution.

One particular user ID is designated as “superuser.” The superuser is exempt from the usual file access control constraints and has systemwide access. Any program that is owned by, and SetUID to, the “superuser” potentially grants unrestricted access to the system to any user executing that program. Hence great care is needed when writing such programs.

UNIT III

VII (a) Explain SNORT architecture with diagram **6**



Snort is an open source, highly configurable and portable host-based or network-based IDS. Snort is referred to as a lightweight IDS. Snort can perform real-time packet capture, protocol analysis, and content searching and matching. Snort can detect a variety of attacks and probes, based on a set of rules configured by a system administrator. A Snort implementation can be configured as a passive sensor, which monitors traffic but is not in the main transmission path of the traffic, or an inline sensor. In the latter case, Snort can perform intrusion prevention as well as intrusion detection.
A Snort installation consists of four logical components

- Packet Decoder: efficiently processes each captured packet to identify and isolate protocol headers at the data link, network, transport, and application layers.
- Detection Engine: does actual work of intrusion detection, analyzing each packet using rules defined for this configuration of Snort by the security administrator.
- Logger: of each packet that matches a rule, if specified. The security administrator can then use the log file for later analysis.
- Alerter: can be sent for each detected packet to a file, a UNIX socket, or a database.

(b) Briefly explain SNORT rule format **9**

Snort uses a simple, flexible rule definition language that generates the rules used by the detection engine. Although the rules are simple and straightforward to write, they are powerful enough to detect a wide variety of hostile or suspicious traffic. Each rule consists of a fixed header and zero or more options. The header includes:

- Action: tells Snort what to do when it finds a packet that matches the rule criteria (alert, log, pass, activate, dynamic, drop, reject, sdrop)
- Protocol: if packet protocol matches this field then analysis proceeds
- Source IP address: source of packet
- Source port: for the specified protocol (e.g.,a TCP port).
- Direction: unidirectional (->) or bidirectional (<->).
- Destination IP address: destination of packet.
- Destination port

Following the rule header may be one or more rule options. Each option consists of an option keyword, which defines the option; followed by arguments, which specify the details of the option. Table 6.5 in the text gives examples of options in each category. The example shown of a Snort rule checks detects a type of attack at the TCP level known as a SYN-FIN attack. It has specified source and destination networks, and checks if the SYN and the FIN bits are set, ignoring reserved bit 1 and reserved bit 2 in the flags octet

OR

VIII (a) Discuss general approaches in defending worms. **8**

There is considerable overlap in techniques for dealing with viruses and worms. Once a worm is resident on a machine, antivirus software can be used to detect it. In addition, because worms propagation generates considerable network activity, the monitoring of that activity can lead form the basis of a worm defense. Have classes:

A. Signature-based worm scan filtering: generates a worm signature, which is then used to prevent worm scans from entering/leaving a network/host.
B. Filter-based worm containment: focuses on worm content rather than a scan signature. The filter checks a message to determine if it contains worm code.
C. Payload-classification-based worm containment: examine packets to see if they contain a worm using anomaly detection techniques
D. Threshold random walk (TRW) scan detection: exploits randomness in picking destinations to connect to as a way of detecting if a scanner is in operation
E. Rate limiting: limits the rate of scanlike traffic from an infected host.
F. Rate halting: immediately blocks outgoing traffic when a threshold is exceeded either in outgoing connection rate or diversity of connection attempts. Rate halting can integrate with a signature- or filter-based approach so that once a signature or filter is generated, every blocked host can be unblocked; as with rate limiting, rate halting techniques are not suitable for slow, stealthy worms.

(b) Explain different states of worm technology **7**

The state of the art in worm technology includes the following:

- Multiplatform: Newer worms are not limited to Windows machines but can attack a variety of platforms, especially the popular varieties of UNIX.

- Multi-exploit: New worms penetrate systems in a variety of ways, using exploits against Web servers, browsers, e-mail, file sharing, and other network-based applications.
- Ultrafast spreading: One technique to accelerate the spread of a worm is to conduct a prior Internet scan to accumulate Internet addresses of vulnerable machines.
- Polymorphic: To evade detection, skip past filters, and foil real-time analysis, worms adopt the virus polymorphic technique. Each copy of the worm has new code generated on the fly using functionally equivalent instructions and encryption techniques.
- Metamorphic: In addition to changing their appearance, metamorphic worms have a repertoire of behavior patterns that are unleashed at different stages of propagation.
- Transport vehicles: Because worms can rapidly compromise a large number of systems, they are ideal for spreading other distributed attack tools, such as distributed denial of service bots.
- Zero-day exploit: To achieve maximum surprise and distribution, a worm should exploit an unknown vulnerability that is only discovered by the general network community when the worm is launched.

UNIT IV

IX (a) Explain distributed denial of service attack. **8**

All of these flooding attack variants are limited in the total volume of traffic that can be generated if just a single system is used to launch the attack. By using multiple systems, the attacker can significantly scale up the volume of traffic that can be generated. Each of these systems need not be particularly powerful, or on a high capacity link. But what they don’t have individually, they more than compensate for in large numbers. These systems were typically compromised user workstations or PC’s. The attacker used some well-known flaw in the operating system or in some common application, to gain access to these systems, and to install their own programs on it. Such systems are known as “zombies”. Once suitable “backdoor” programs were installed on these systems, they were entirely under the attacker’s control.

- Large collections of such systems under the control of one attacker can be created, collectively forming a “botnet”. One of the earliest and best known DDoS tools is Tribe Flood Network (TFN), written by the hacker known as Mixter. The original variant from the 1990’s exploited Sun Solaris systems. It was later rewritten as Tribe Flood Network 2000 (TFN2K), and could run on UNIX, Solaris, and Windows NT systems. The agent was a Trojan program that was copied to, and run on compromised, zombie systems. It was capable of implementing ICMP flood, SYN flood, UDP flood, and ICMP amplification forms of denial of service attacks.

(b) Describe reflector attack **7**

- Reflection attack use network systems functioning normally. The attacker sends a network packet with a spoofed source address to a service running on some network server, which responds to this packet, sending it to the spoofed source address that belongs to the actual attack target. If the attacker sends a number of requests to a number of servers, all with the same spoofed source address, the resulting flood of responses can overwhelm the target's network link. The fact that normal server systems are being used as intermediaries, and that their handling of the packets is entirely conventional, means these attacks can be easier to deploy, and harder to trace back to the actual attacker. Ideally the attacker would like to use a service that created a larger response packet than the original request. This allows the attacker to convert a lower volume stream of packets from the originating system into a higher volume of packets from the intermediary directed at the target. Common UDP services are often used for this purpose. Another variant of reflection attack uses TCP SYN packets, and exploits the normal 3-way handshake used to establish a TCP connection. possible.

OR

X Write short notes: **($3\times 5 = 15$)**

(i) State full inspection firewall

- reviews packet header information but also keeps info on TCP connections
- typically have low, "known" port no for server
- and high, dynamically assigned client port no
- simple packet filter must allow all return high port numbered packets back in
- stateful inspection packet firewall tightens rules for TCP traffic using a directory of TCP connections
- only allow incoming traffic to high-numbered ports for packets matching an entry in this directory
- may also track TCP seq numbers as well

(ii) Circuit level proxy firewall

- Establishes a TCP connection with remote end before passing information through.
- Creates two sessions: one with sender & one with receiver
- Does not filter based on packet contents (other than state)
- Also known as Pass-Through Proxy or Generic Proxy
- Advantages: If firewall failure, no packets are forwarded through firewall
- Catches fragmentation errors

Problems:

- Does not detect invalid application data
- Moves security issues from service to firewall: e.g., DOS attacks
- Less able to handle high loads since each connection becomes two
- Requires much greater memory and processor at application level (Web page is > 1 connection)
- Slower interfaces can result in poor performance for streaming applications

(iii) Packet filtering firewall.

- applies rules to packets in/out of firewall
- based on information in packet header
    - src/dest IP addr & port, IP protocol, interface
- typically a list of rules of matches on fields
    - if match rule says if forward or discard packet
- two default policies:
    - discard - prohibit unless expressly permitted
    - forward - permit unless expressly prohibited