

ISA 656, Assignment 2 Solutions

THEORY/WRITTEN QUESTIONS (50 points)

1) [15 points] Firewalls:

- a) What is a proxy firewall and how is it different from a network (or transparent) firewall?

Proxy level firewalls and application level firewalls are used interchangeably, application/proxy level firewalls work on application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or ftp traffic), processing access requests on behalf of the network on which it is located. This protects the individual computers on the network, because they never interact directly with incoming client requests. Proxy level firewalls provide more granular control, but tend to be slower to process requests than network firewall.

- b) What does NAT stand for, and how does the mechanism work? Describe what, if any, security NAT provides (or fails to provide).

NAT stands for Network Address Translators. NAT, which was originally intended for short-term use to solve the globally depleting IP address problem, has become the standard method of connecting private networks with internet resources.

NATs work by translating the source IP and port numbers. When a NAT device receives a packet from the internal network, it modifies the packet header and replaces the source IP address with its own IP address. The packet is then sent to the outside. The NAT device stores the original internal IP address, the destination IP address and the port number in a table. When the request is returned to the same port, the NAT device searches its table and matches the IP address and port number. It modifies the IP header to match the internal address and then forwards it to the internal address.

NATs provide security by allowing multiple hosts to hide behind one or more public IP addresses. These hosts are generally not accessible directly from the outside, unless directly specified (i.e., port forwarding). On the other hand, NAT complicates tunneling protocols such as IPsec as well as VOIP, videoconferencing, gaming, and others applications requiring peer-to-peer connectivity.

c) Where would you place a web server in an organization assuming that you can use a network firewall and why?

All servers that are exposed to the public network (internet), including web servers and DNS servers should be placed in the DMZ. The DMZ is a separate subnet located outside of the organization's network, but still protected by a firewall. Traffic originating from the organization's network is allowed into the DMZ, but traffic from the DMZ is not allowed into the organization's network. This is important because should a server become compromised, an attacker cannot access the organization's internal network.

2) [20points] Assume you have the following firewall policy:

	Action	prot	Source	Destination	Source Port	Destination Port
1	ACCEPT	udp	0.0.0.0/0	129.174.17.180	*	53
2	ACCEPT	Tcp	55.66.77.0/24	129.174.17/180	*	22
3	REJECT	Tcp	55.66.77.12	129.174.17.180	4500	22
4	ACCEPT	Tcp	127.0.0.1	129.174.17.180	*	6000
5	DENY	Tcp	0.0.0.0/0	129.174.17.180	*	6000
6	REJECT	udp	0.0.0.0/0	129.174.17.180	*	32768
7	REJECT	tcp	0.0.0.0/0	129.174.17.180	*	32769
8	REJECT	tcp	0.0.0.0/0	129.174.17.180	*	32768
9	ACCEPT	tcp	0.0.0.0/0	129.174.17.180	*	80
10	ACCEPT	udp	129.174.16.20	0.0.0.0/0	53	1025:65535
11	ACCEPT	udp	129.174.20.100	0.0.0.0/0	53	1025:65535
12	ACCEPT	udp	129.174.18.100	0.0.0.0/0	53	1025:65535
13	ACCEPT	All	0.0.0.0/0	0.0.0.0/0	*	*
14	REJECT	tcp	0.0.0.0/0	0.0.0.0/0	*	*
15	REJECT	udp	0.0.0.0/0	0.0.0.0/0	*	*
16	DENY	tcp	0.0.0.0/0	129.57.17.180	*	6000:6010
17	DENY	tcp	0.0.0.0/0	129.174.17.180	*	0:1024
18	DENY	All	0.0.0.0/0	129.174.17.180	*	*

a) Identify any policy conflicts or redundancies.

Conflicts:
Rules that fully or partly overlap and have different ACTIONS,
examples:

2	ACCEPT	TCP	55.66.77.0/24	129.174.17.180	*	22
3	REJECT	TCP	55.66.77.12	129.174.17.180	4500	22

13	ACCEPT	ALL	0.0.0.0/0	0.0.0.0/0	*	*
14	REJECT	TCP	0.0.0.0/0	0.0.0.0/0	*	*
15	REJECT	UDP	0.0.0.0/0	0.0.0.0/0	*	*

Redundancies:

Rules that fully or partly overlap and have the same ACTION, examples:

16	DENY	TCP	0.0.0.0/0	129.57.17.180	*	6000:6010
17	DENY	TCP	0.0.0.0/0	129.174.17.180	*	0:1024
18	DENY	ALL	0.0.0.0/0	129.174.17.180	*	*

The FIRST match strategy applies the first (top to bottom) rule that matches the packet.

The BEST match strategy applies the rule that is more specific in terms of network.

The LAST match strategy applies the rule that is first (bottom to top).

- b) Show what would happen to the following packet under the three tie-breaking strategies (first match, best match, last match).

```
+-----+-----+-----+-----+
| TCP | s:55.66.77.12:4500 | d:129.174.17.180:22 | data... |
+-----+-----+-----+-----+
```

First Match: **Accept**, based on firewall rule (row #2)

Best Match: **Reject**, based on firewall rule (row #3)

Last Match: **Deny**, based on last firewall rule (row #18)

- c) Show what would happen to the following packet under the three tie-breaking strategies (first match, best match, last match).

```
+-----+-----+-----+-----+
| UDP | s:55.66.77.12:4500 | d:129.174.17.180:22 | data... |
+-----+-----+-----+-----+
```

First Match: **Accept**, based on firewall rule (row #13)

Best Match: **Reject**, based on last firewall rule (row #15)

Last Match: **Deny**, based on last firewall rule (row #18)

3) [15 Points] Assume a cryptographic algorithm in which the performance for the good guys (the ones that know the key) grows linearly with the length of the key, and for which the only way to break it is a brute-force attack of trying all possible keys. Suppose the performance for the good guys is adequate (e.g., it can encrypt and decrypt as fast as the bits can be transmitted over the wire) at a certain size key. Then suppose advances in computer technology make computers twice as fast. Given that both the good guys and the bad guys get faster computers, does this advance in computer speed work to the advantage of the good guys, the bad guys, or does it not make any difference?

The advance in computer speed works to the advantage of the good guys. If the performance of the good guys grows linearly with the length of the key, then doubling the computer speed would allow for doubling the length of the key without any performance penalty. Doubling the length of the key would have a significant impact on the bad guys, since the number of keys that must be checked grows exponentially with the length of the key.

For example, if the key were originally 8 bits, the bad guys would need to check 256 keys (2^8). If both good and bad guys get computers that run twice as fast, then, for the same amount of processing time, the good guys can use a 16 bit key and the bad guys can check 512 keys. However, because the number of keys grows exponentially, there will now be (2^{16}) or 65,536 keys. Checking this many keys will take the bad guys 128 times longer than the original 8-bit key ($65,536 / 512 = 128$).

EXTRA CREDIT [15points]

Can you create a multi-threaded version of the previous program that can encrypt and decrypt multiple files at the same time? When can we benefit from using a multi-threaded approach for encryption and decryption? Are we going to have a performance gain for all the modes of AES encryption?

In general, we benefit from a multi-threaded approach in decryption and encryption when we perform multiple independent operations: having multiple resources allows us to allocate them in different tasks using independent threads. For example, while a thread is decrypting, using most of the CPU resource, another thread can be reading the next file to decrypt, mostly using the I/O resources. In the case of a single-threaded approach, the process must first complete decryption before the next file can be read; I/O operations are waiting on the CPU.

Multi-threaded approach is especially useful in multiprocessor systems, as that is the case with most computers these days. In a single-threaded approach, only 1 CPU is utilized. On the other hand, in a multi-threaded approach, each thread can utilize a CPU.

In a multi-threaded approach, we will see a performance gain irrelevant of the Mode of encryption **only when** multiple encryption processes are running. This means that multiple files are being encrypted simultaneously into different encrypted files – not a single file. In the case of encrypting a single file, or multiple files into a single encrypted file, then the mode of operation does make a difference. Specifically, some of the block cipher modes of operations require the previous packets to be encrypted before used as an IV. In this case, a multi-threaded approach doesn't offer a noticeable performance gain since each operation must be done in sequence.