Lecture 4
Denial-of-Service Attacks

Print version of the lecture in *CPS633 Computer Security*

presented on Week of September 30, 2019

by Ali Miri ©2019from Department of Computer Science at Ryerson University

4.1

Learning Objectives (page 225)

- Explain the basic concept of a denial-of-service attack.
- Understand the nature of flooding, application-based, and other types of attacks.
- Describe distributed denial-of-service attacks.
- Summarize some of the common defences against denial-of-service attacks.

This set of slides closely follows Chapter 7 of the textbook.

4.2

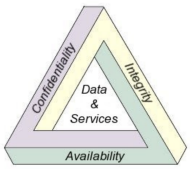
Contents

4.3

1 Preliminaries

1.1 Definitions

Definitions



- Availability ← A loss of availability is the disruption of access to or use of information or an information system
- **Denial-of-Service (DoS)** is attack on availability. It can target end hosts, critical servers such web, DNS, file, authentication, etc, or network (-based) infrastructure.
- Loss of availability can arise from malicious or benign causes.

Observation: Some differences between Confidentiality and integrity, and availablity

- Confidentiality and integrity focus on *preventing* unauthorized access/modification, whereas availability focuses on *preserving* authorized access
- Confidentiality and integrity tend to be binary, whereas availability tends to be more nuanced

4.4

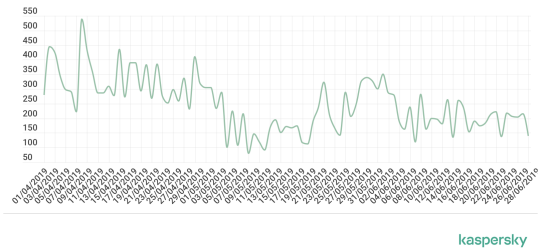
1.2 Motivations

We will consider a few questions:

- Should we be concerned with these types of attacks?
- What are typical venues and tools these attacks use?
- What are the countermeasures, and how effective are they?

4.5

- Normally one of the top security concerns for organizations
- Number of occurrences are steadily increasing



source: https://securelist.com/ddos-report-q2-2019/91934/

4.6

1.3 Examples

Morris worm

Morris worm:(Nov 1988) A classical case of worms and DoS

- First DDoS attack on network infrastructure. Was estimated that over 6000 servers were impacted.
- Was Self-replicating, self-propagating, and used common software flaws
- Cost millions of dollars in damages, and it took several days to respond to.

4.7

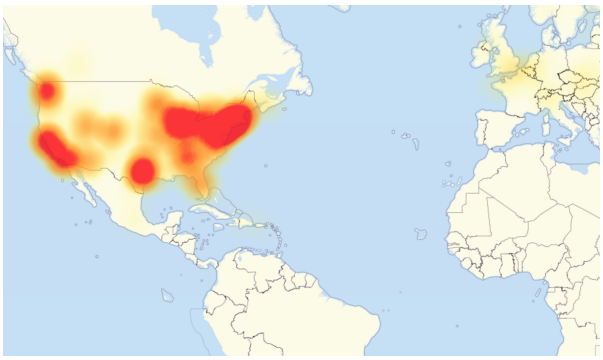
Estonia Cyberattack

- A series of DDoS attacks on governments, banks, news organizations, $\cdots$, starting on April 27, 2007
- It started by relocation of the Bronze Soldier of Tallinn
- It included over 100 different types of DDoS, such as ICMP floods, TCP SYN floods, etc.
- 10 largest attacks measured at 90 Mbps, lasting upwards of 10 hours. (source: Hun-CERT)

4.8

Dyn Cyberattack

- The attack targeted the DNS provider *Dyn* on Oct 21, 2016, with the attack lasting most of the day.
- It effected many sites, including Twitter, Netflix, Reddit, CNN, $\cdots$.
- It used *Mirai botnet*, largely made of IoT devices (over 100,000), and able to generate a very large volume of traffic ($\geq$ 1.2TBps).

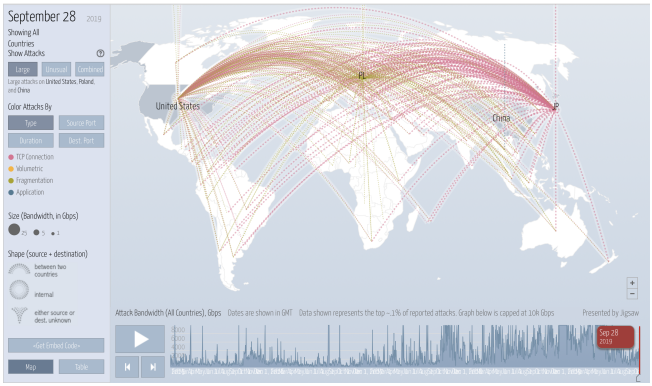


DownDetector Level 3 Outage Map

4.9

Digital Attack Map

Arbor Network's Digital Attack Map



source: https://www.digitalattackmap.com/

4.10

1.4 Categories

Categories

Three typical categories of attacks:

- **Network-based**: the vast majority of traffic directed at the target server is malicious, generated either directly or indirectly by the attacker. This traffic overwhelms any legitimate traffic, effectively denying legitimate users access to the server (example: Ping flooding attack).
- **Resource-based:** typically aims to overload or crash its network handling software (example SYN spoofing attack).
- **Application-specific:** target application resources (example: *Slowloris*)

4.11

Characteristics

Some of the attacker's possible (design) requirements:

- Attacks be as effective as possible, and hard to detect.
- Attacker uses as little resources as possible.
- Attacker is not impacted by the attack, and it be difficult to identify the attacker.

4.12

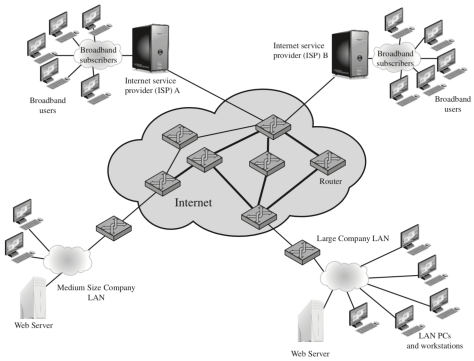
2 Network-Based

Flooding

Basic idea: Direct as much traffic as possible toward the target.

4.13

Flooding



Q. How would you as the attacker achieve this, and how does this meet the requirements listed earlier? What type of traffic would you use?

4.14

Flooding

Different types of traffic that can be used include

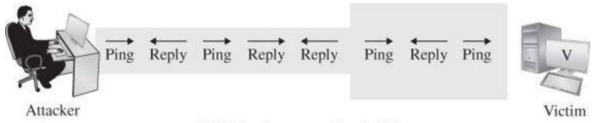
- ICMP traffic. ⇒ ICMP flooding attacks
  - Example: *Ping flood* using ICMP echo request packets
- UDP traffic ⇒ UDP flooding attacks
  - Example: *UDP Echo* attack, sending UDP packets to the diagnostic echo service
- TCP traffic ⇒ TCP flooding attacks
  - Example: Use TCP packets with as large as payload as possible.

4.15

2.1 Ping Flood

Ping Flood

Aim: Send an abnormally large number of Ping at target, in order to overwhelm its services

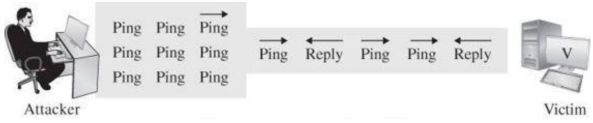


source: 'Security in Computing' by PFleeger, PFleeger, and Margulies, 5th edition

Q. Does this approach achieve the attacker's requirements we listed earlier?

4.16

Ping Flood



source: 'Security in Computing' by PFleeger, PFleeger, and Margulies, 5th edition

- The attacker must have at least as large as bandwidth capacity as the target. But
- The attack would impact attacker's resources as well.
- Attacker can be easily identified.
- The target may limit the number of ping request from one source

4.17

Ping Flood

How to increase the 'effectiveness' of this attack?

- Attacker can use spoofed addresses.
- `'Attack by force or by choice!'`

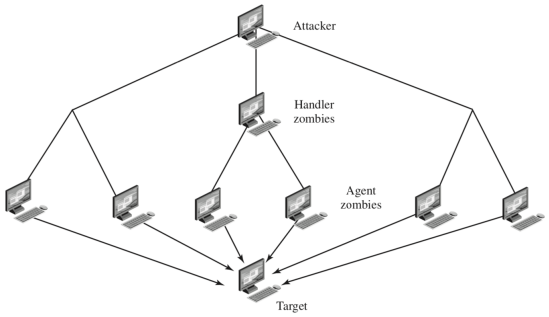
  Mount an indirect attack, using multiple (collaborating?) intermediaries → **Distributed Denial-of-Service (DDoS)** attack
- *Reflector* attack
- *Amplifier* attack

4.18

2.2 DDoS

DDoS

A typical Distributed Denial-of-Service (DDoS) attack architecture



4.19

2.3 Reflector

Reflection Attacks

If attacker sends too much traffic to the victim itself, it can lead to

- its identification
- consumption of (major part of) the attacker's resources
- possible blockage of traffic originated from the attacker

To get around that, the attacker may use "normal" systems as intermediaries

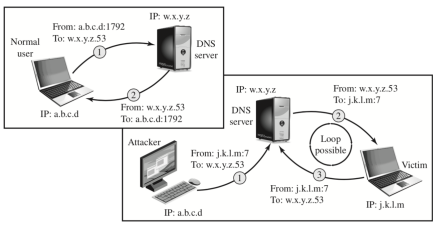
- The malicious traffic can blend in the normal one
- May hide the identity of the attacker
- Resources of intermediaries, and not the attacker are used.
- Lack of backscatter traffic

4.20

DNS reflector attack

Domain Name Service (DNS)

- DNS is used to translate between names and IP addresses.
- The Simple Network Management Protocol (SNMP) is used to manage network devices by sending queries to which they can respond with large volumes of detailed management information.
- Typically runs on port 53.



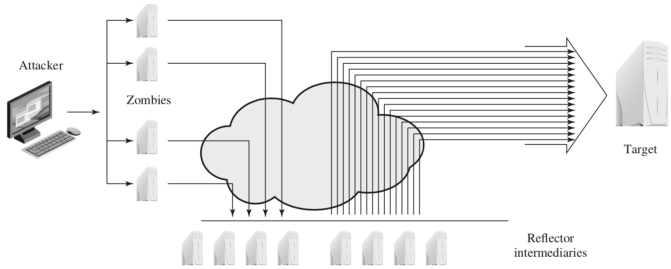
• Any UDP services: DNS, SNMP, ISAKMP, chargen, etc. with sufficiently large response packet can be used.

4.21

2.4 Amplifier

Smurf

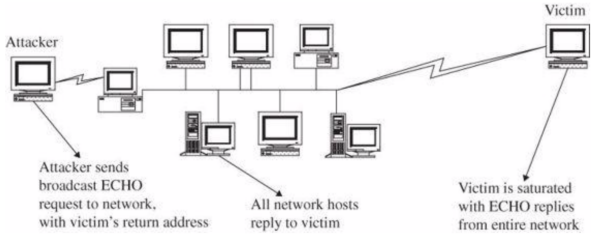
Amplification attacks: a variant of reflector attacks that employs large number of intermediaries.



4.22

Smurf

Smurf attack - a variant of Ping flood attack



source: 'Security in Computing' by PFleeger, PFleeger, and Margulies, 5th edition

- Attacks are reflected from intermediaries (and not the attacker) → a *reflector + amplified attack*

4.23

Fraggle

- very similar to a Smurf Attack, but uses spoofed UDP traffic rather ICMP to achieve the same goal
- Since late 90's, most routers no longer forward packets directed at their broadcast addresses.

4.24

DNS Amplification Attacks

- a 60-byte UDP DNS request → up to 512-4000 bytes response
- Attacker creates a creates a series of DNS requests containing the spoofed source address of the target system, and sends to a selection of DNS servers
- This could also result in degradation of services for intermediate systems
- The attack may use 'recursive DNS name servers'

4.25

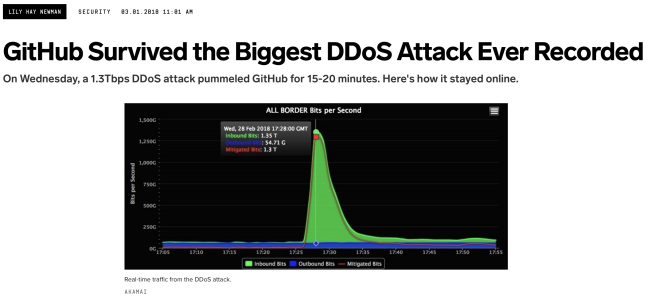
DNS Amplification Attacks Countermeasures

`RFC 5358 – Preventing Use of Recursive Nameservers in Reflector Attacks`
`https://www.ietf.org/rfc/rfc5358.txt`

- IP address based authorization
- Incoming interface based selection
- Use signed queries to authenticate the clients
- Use a local caching nameserver or use a VPN to a trusted server
- Do not offer recursive service to external networks
- Prevent address spoofing ← would need network operators' cooperation

4.26

Largest DDoS Flooding attacks


LILY HAY NEWMAN SECURITY 03.01.2018 11:01 AM

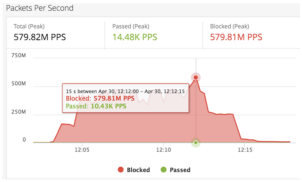
GitHub Survived the Biggest DDoS Attack Ever Recorded

On Wednesday, a 1.3Tbps DDoS attack pummeled GitHub for 15-20 minutes. Here's how it stayed online.

Real-time traffic from the DDoS attack.

AKAMAI

source: https://www.wired.com/story/github-ddos-memcached/



Impreva's attack on April 30, 2019 -

Source: https://www.imperva.com/blog/this-ddos-attack-unleashed-the-most-packets-per-second-ever-heres-why-thats-important/

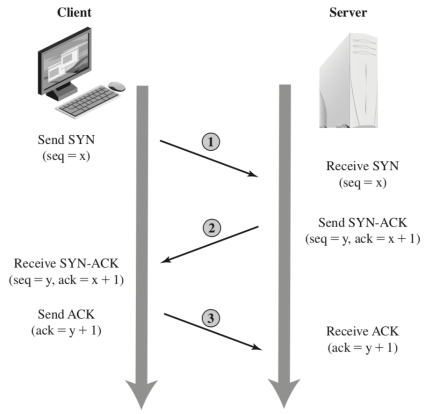
4.27

3 Resource-Based

3.1 SYN Flood

SYN Flood Attacks

Analogy: 'shutting down a restaurant by making lots of no-show bookings'



4.28

SYN Flood Attacks

Aim to keep 'the known TCP connections table filled' taking advantage of 3-way handshake

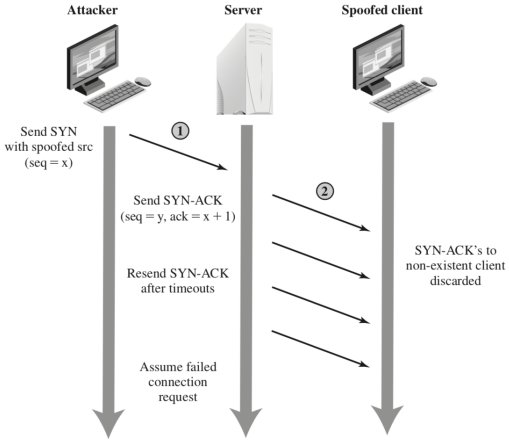
- Typically starts with the target received streams of spoofed TCP SYNs

- Target starts “half-open” connections, as per requests
- Attacker never completes the connection
- The attack uses comparatively low traffic, as compared to flooding attacks
- Do not offer recursive service to external networks

Asymmetry: take very little resource for attacker to initiate, where the target has to spawn a new thread for each request!

4.29

SYN Flood Attacks



4.30

3.2 SYN Cookies

SYN Cookies

Countermeasures: Use

SYN cookies

4.31

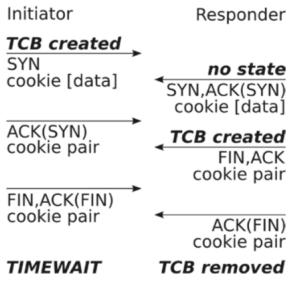
SYN Cookies

- Any solution has to work with TCP and its extensions
- What if the server does ‘not’ store the state, UNTIL we heard ‘enough/more’ from the sender?
- **Cookies:** Phil Karn (1994) - also see RFC 2522 `https://www.rfc-editor.org/rfc/rfc2522`.

- -Cookies can replace the random server sequence number returned to client, and typically is a function of source addr, source port, dest addr, dest port, coarse time, and **server secret**
- Cookies have to unforgeable and tamper-proof
- The cookie has to returned to the server
- The server will recompute cookie, compare with the one received, establish connection only if they match.

4.32

SYN Cookies



source: https://www.usenix.org/system/files/login/articles/126-metzger.pdf

4.33

SYN Cookies

Some possible drawbacks:

- Requires server computational resource (to calculate the cookie)
- Method blocks the use of certain TCP extensions, such as larger windows
- Available in various Linux implementations and newer Windows OS, but it is not enabled by default

• This solution is also suggested as part of Domain Name System (DNS) Security (DNSSEC) efforts

• Other suggested options include a *selective drop* or *random drop* of incomplete connections from the TCP connections table when it overflows

4.34

3.3 Teardrop

Teardrop Attack

IPv4 Header Format

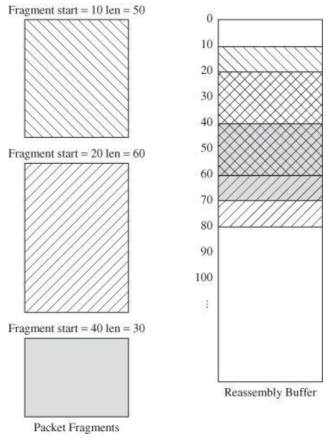
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

Fragment Offset indicates the starting position, or *offset*, of the data contained in a fragmented packet relative to the data in the original packet.

4.35

Teardrop Attack

- If the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap.
- In older Windows and Linux OS, this resulted in a crash!



source: 'Security in Computing' by PFleeger, PFleeger, and Margulies, 5th edition

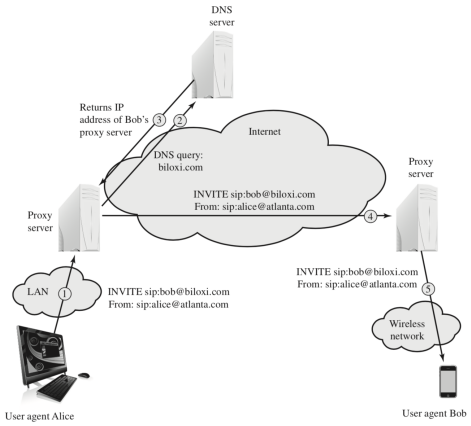
4 Application-Based

4.1 SIP Flood

Application-Based Attacks

Aim: force execution of application-related, resource-consuming operations

Example: Session Initiation Protocol (SIP) flood attack against Voice over IP (VoIP) calls

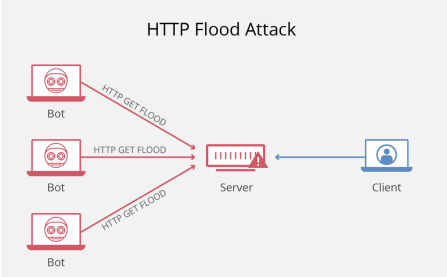


SIP Invite Attack

4.2 HTTP Flood

HTTP Flood Attacks

- Many of organizations' web servers are public, and respond to HTTP inquires such as HTTP GET or POST requests
- The attacker sends a large volume of seemingly legitimate session-based HTTP requests to victims' servers.



source: https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/

- Recursive HTTP flood: start with a given link, and requests all the links in it in a recursive way ⇒ *Spidering*

4.38

Countermeasures

HTTP Flood attacks are difficult to counter, as it may be difficult to identify legitimate traffic from a malicious one.

Example:

NET EFFECT

Michael Jackson is the new DDoS

As I am digging my way out of my RSS feeds after a summer hiatus, here comes an interesting story on how a surge of interest following Michael Jackson's death made some news sites suspect they are under a DDoS attack. Google, Wikipedia, BBC, CNN, Twitter and especially gossip site TMZ, which broke the news ...

BY EVGENY MOROZOV | JULY 2, 2009, 9:05 AM

source: https://foreignpolicy.com/2009/07/02/michael-jackson-is-the-new-ddos/

These kind of ocurrences are referred to as *slashdotted*, or *flash crowd/event*

4.39

Countermeasures

Here are some possible countermeasures:

- Use challenge request, like a captcha to identify requests from a bot
- Use rate-based, or other ML-based detection engines (will discuss more in the next chapter)

Check out the example at https://blog.sucuri.net/2014/02/layer-7-ddos-blocking-http-flood-attacks.html

4.40

4.3 Slowloris

Slowloris Attacks

"I am still here! I am just slow. Please wait for me!"

- Attacker opens multiple connections, and send multiple partial HTTP requests to the victim
- The victim's server will wait (up to timeout period) for the attacker to complete the HTTP request header.
- The attacker keep the requested server's threads by periodically sending it additional header lines, without indicating the end line has been sent.

The resulting DoS does require much less resource-allocation for attacker than that of HTTP Flood. 4.41

Countermeasures

It is difficult to detect this attack using IDS/IPS, since it uses (seemingly) legitimate, 'good' HTTP traffic

Possible countermeasures include:

- Provisioning more resource to the HTTP server
- Using web proxies
- If attacks are coming for a single IP, limiting slow transfer time, and the number of incomplete threads that machine request can spawn.

4.42

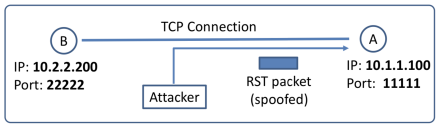
5 Countermeasures

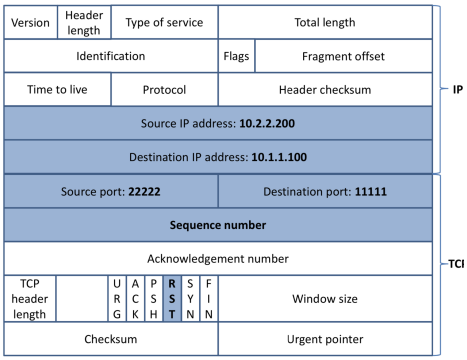
Different lines of defence against DDoS

- **Attack prevention and preemption (before the attack)**
  - Resource consumption policies
  - Provisioning adequate resource backups
  - Turn off directed broadcasts
  - Deploy source address anti-spoof filters (*very important!*)
  - Employ IPS mechanisms to modify systems/protocols, when under attack
- **Attack detection and filtering (during the attack)**
  - Strategies and solutions as part of Intrusion Detection Systems (IDS) - discussed in next chapter
- **Attack source traceback and identification (during and after the attack)**
- **Attack reaction (after the attack)** - see Chapter 17.

4.43

TCP Reset Attack

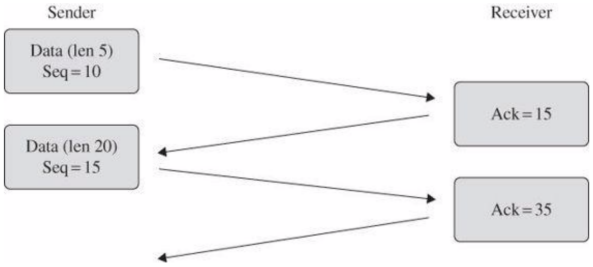




Source: Chapter 16 of Wenliang Du, 'Computer and Network Security' book

TCP session hijacking attack

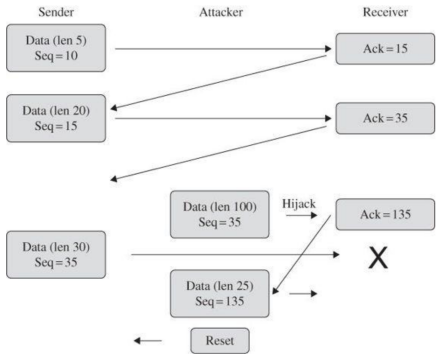
- TCP is made to work with unreliable network communication ← it uses *sequence numbers* (together with source and destination addresses) to identify error and reconstruct data



Simplified view of a normal TPC exchange (only client's perspective is shown) Source: Chapter 6 of 'Security in Computing' by PFleeger, PFleeger, and Margulies, 5th edition

TCP session hijacking attack

- TCP protocol is self-healing: it allows the two sides to determine the last successful exchange, and retransmit from that point forward.
- The attacker inserts a packet to synchronize with the receiver, and typically sends a RST to the sender to let it know that the connection is dropped.



Source: Chapter 6 of 'Security in Computing' by PFleeger, PFleeger, and Margulies, 5th edition