CS 556 – Computer Security
Fall 2013

Dr. Indrajit Ray
Email: indrajit@cs.colostate.edu

Department of Computer Science
Colorado State University
Fort Collins, CO 80523, USA

DENIAL OF SERVICE ATTACKS

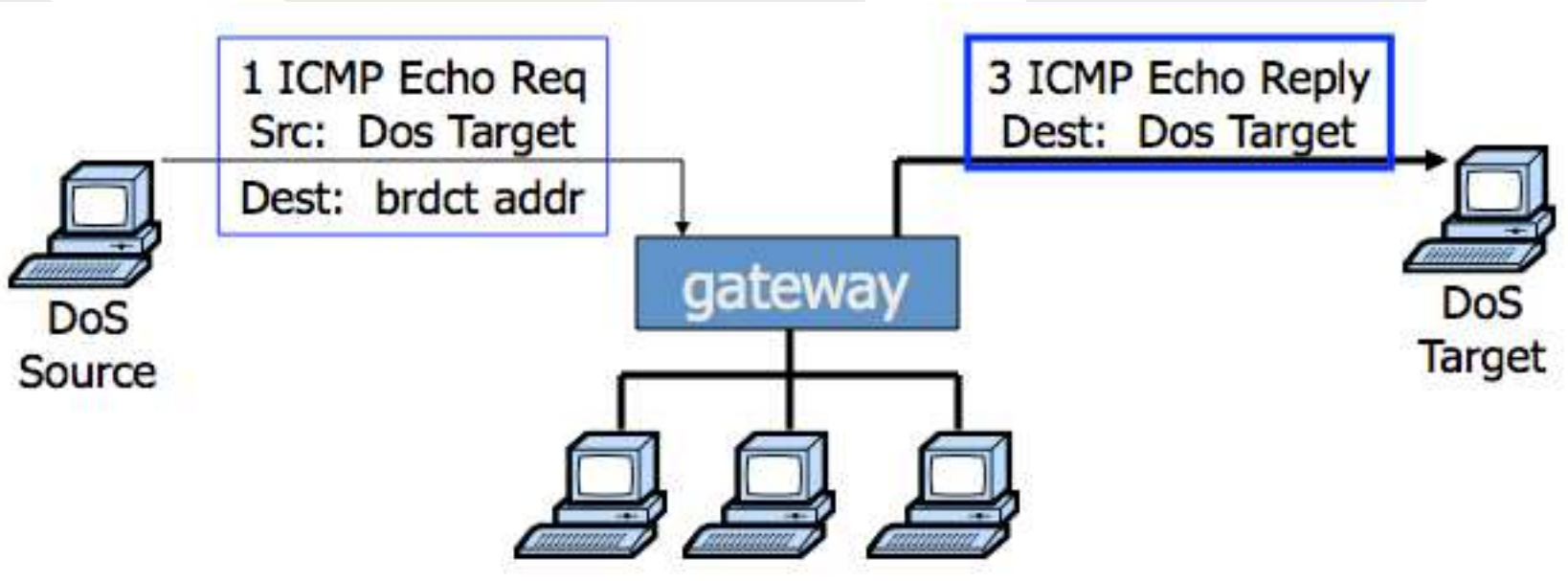
Denial of Service

- DoS attacks can be launched via
  - Consumption of network connectivity and/or bandwidth
  - Consumption of other resources. e.g. buffer, CPU
  - Destruction or alteration of configuration information
  - Physical destruction or alteration of network components
- DoS attacks can target
  - Critical servers such as databse, web, file, authentication, DNS
  - Infrastructure such as routers on a path or within a domain
  - End hosts
- Defining DoS is not easy

General Classes of DoS Attacks

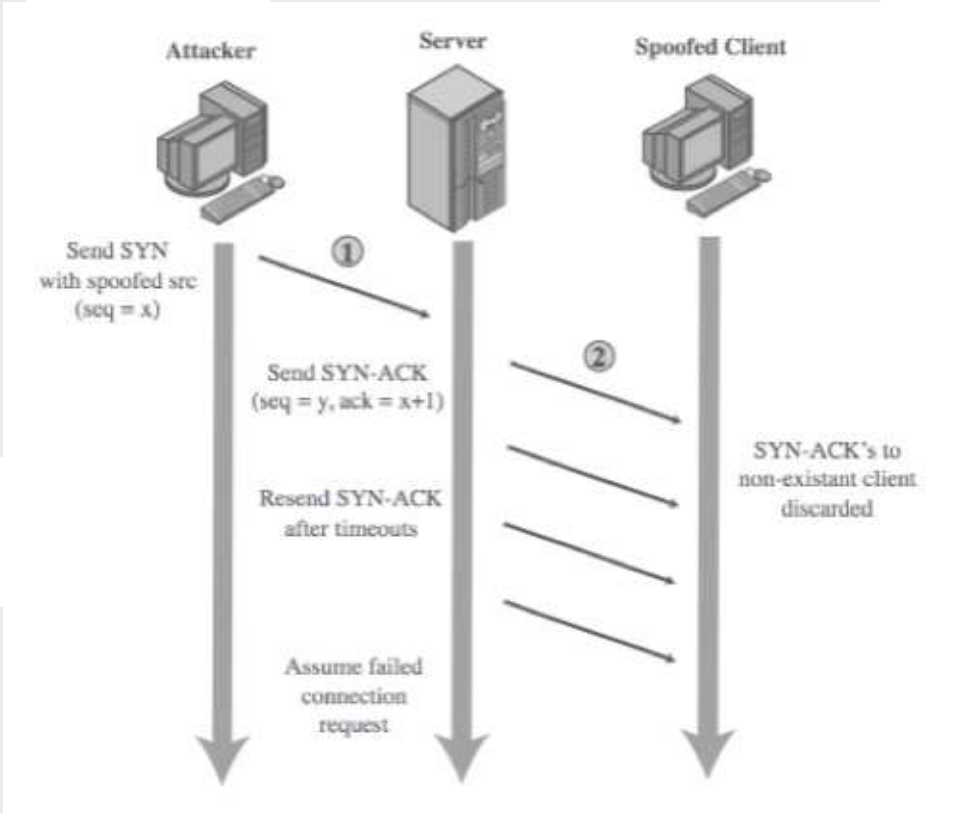
- Flooding Attacks
  - Point-to-point attacks: TCP/UDP/ICMP flooding, Smurf attacks
  - Distributed attacks: multiple point-to-point attacks in a hierarchical structure
- Corruption attack - application / service specific
  - DNS cache poisoning attack
  - Polluting P2P systems

Smurf DoS Attack



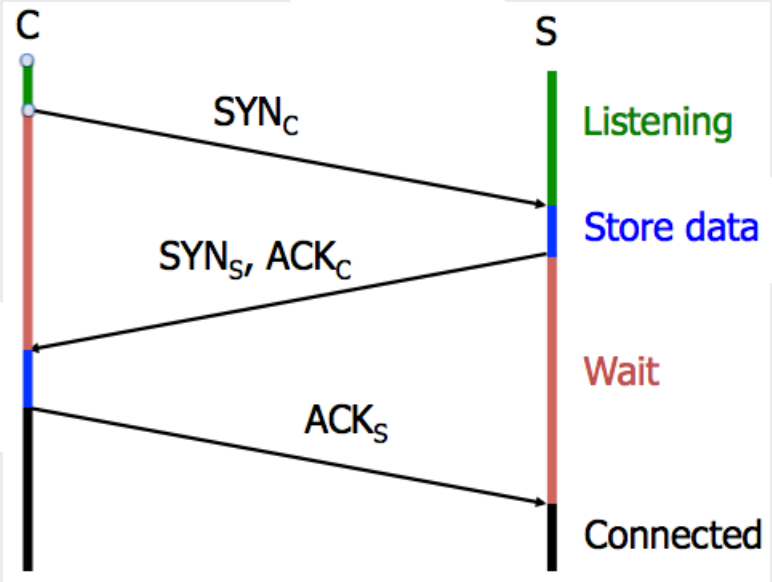
- Send ping request to broadcast address (ICMP Echo Req)
- Every host on target network generates a ping reply (ICMP Echo Reply) to victim
- Ping reply stream can overload victim

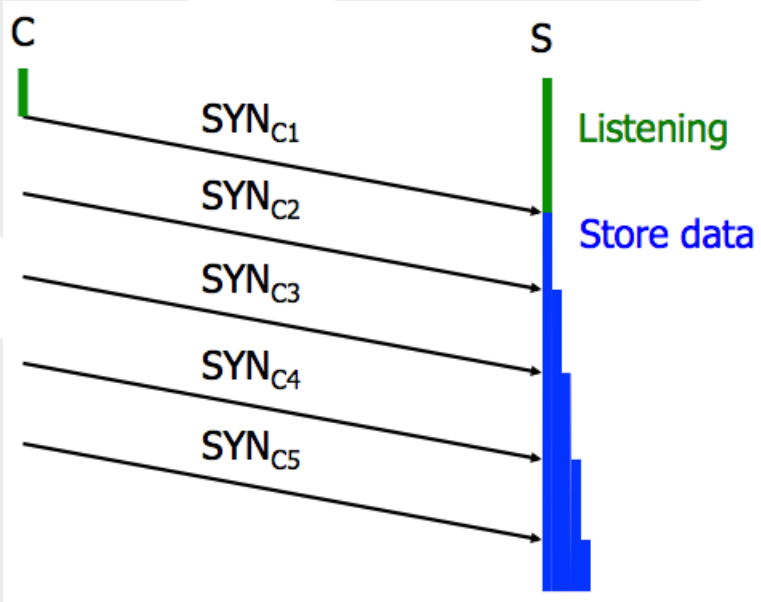
TCP SYN Spoofing Attack



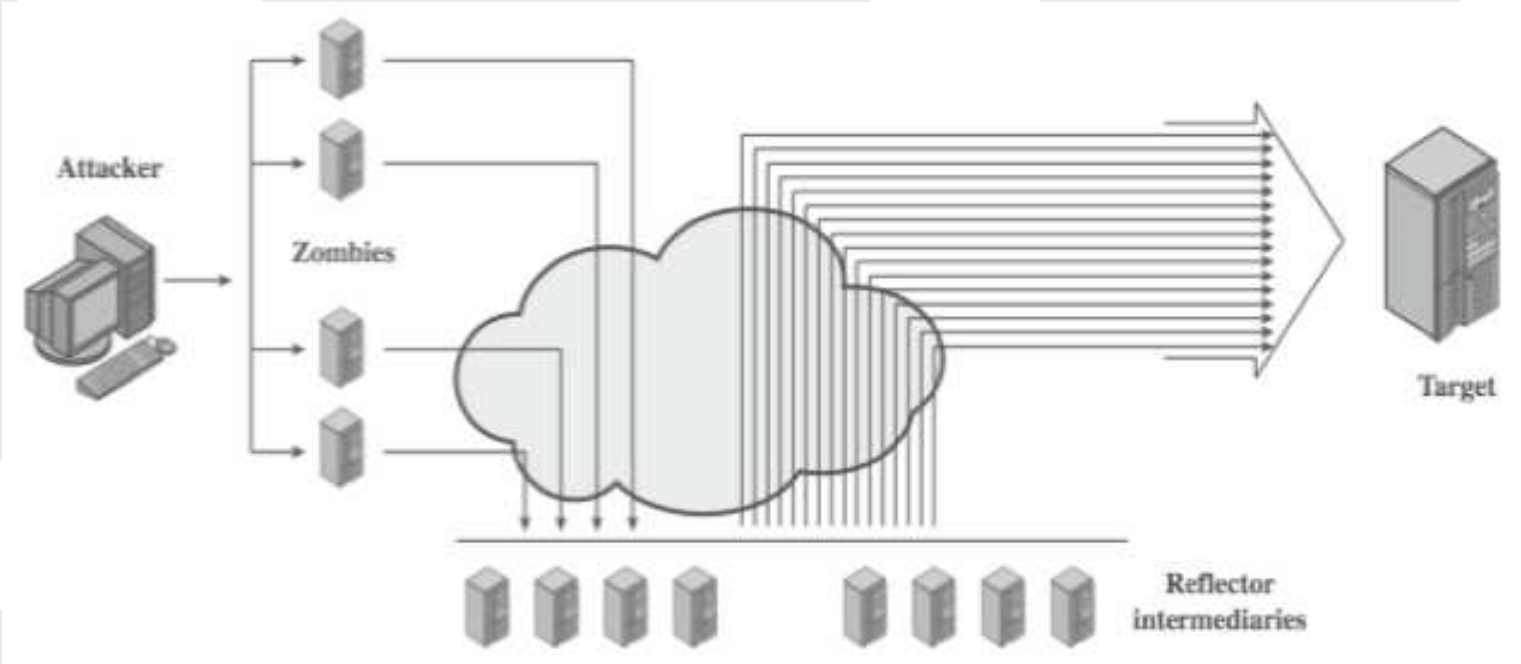
- Attacker can use random source address, or that of an overloaded server in order to block return on TCP reset packets
- Attack succeeds even if attacker is on a much lower bandwidth link

TCP SYN Flooding Attack





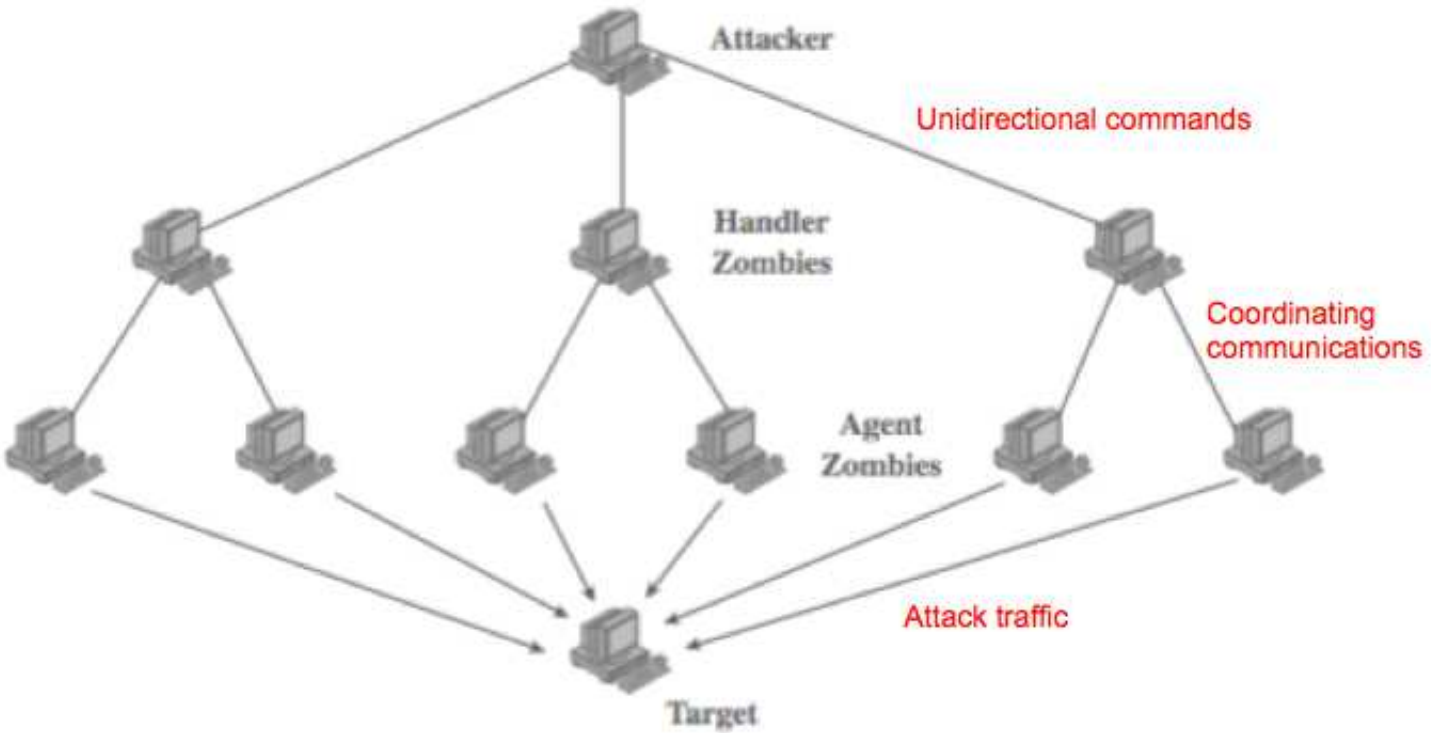
Amplification Attack



DNS Amplification Attack

- Use DNS requests with spoofed source address being the target
- Exploit DNS behavior to convert a small request to a much larger response: (60 byte request to 512–4000 byte response)
- Attacker sends requests to multiple well connected servers, which flood target
  - Need only moderate flow of request packets
  - DNS servers will also be loaded

Distributed DoS Attack



Buffer Overflow Attacks

Buffer Overflow Basics

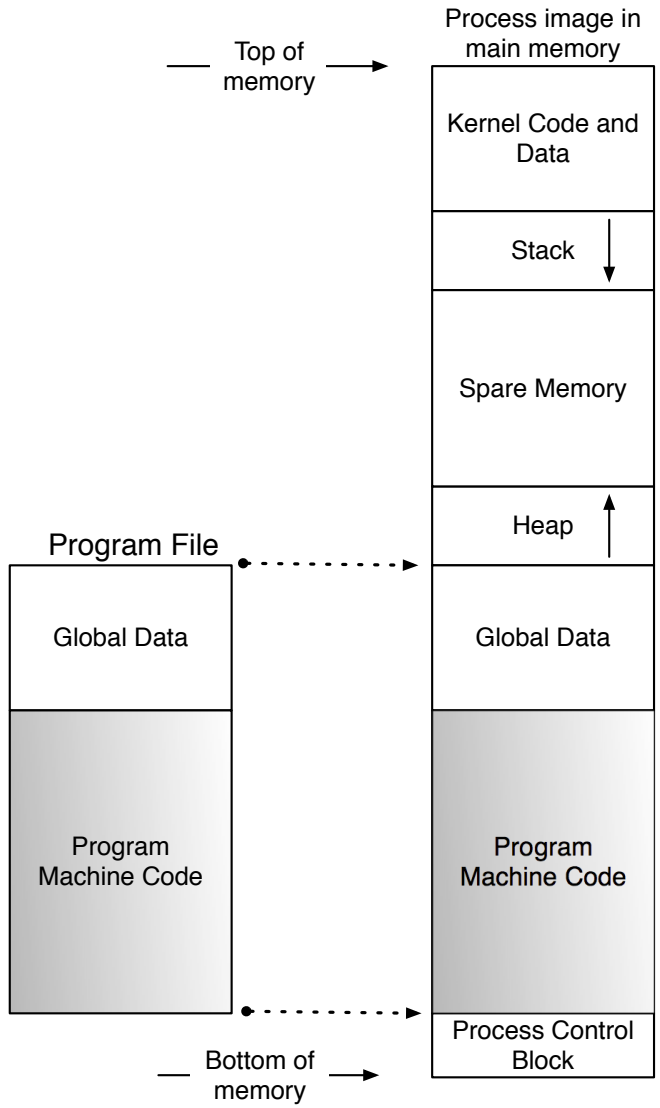
- Programmer error causes more data to be stored than capacity available in fixed size buffer, thereby overwriting adjacent memory locations
  - Buffer can be on system stack, heap or global data area
- Overwriting can result in
  - Corruption of program data
  - Unexpected transfer of control
  - Memory access violation
  - Execution of code chosen by attacker
- Prevention techniques know

Simple Buffer Overflow


```
int main( ) {
        int buffer[10] ;
        buffer[20] = 37;
}
```

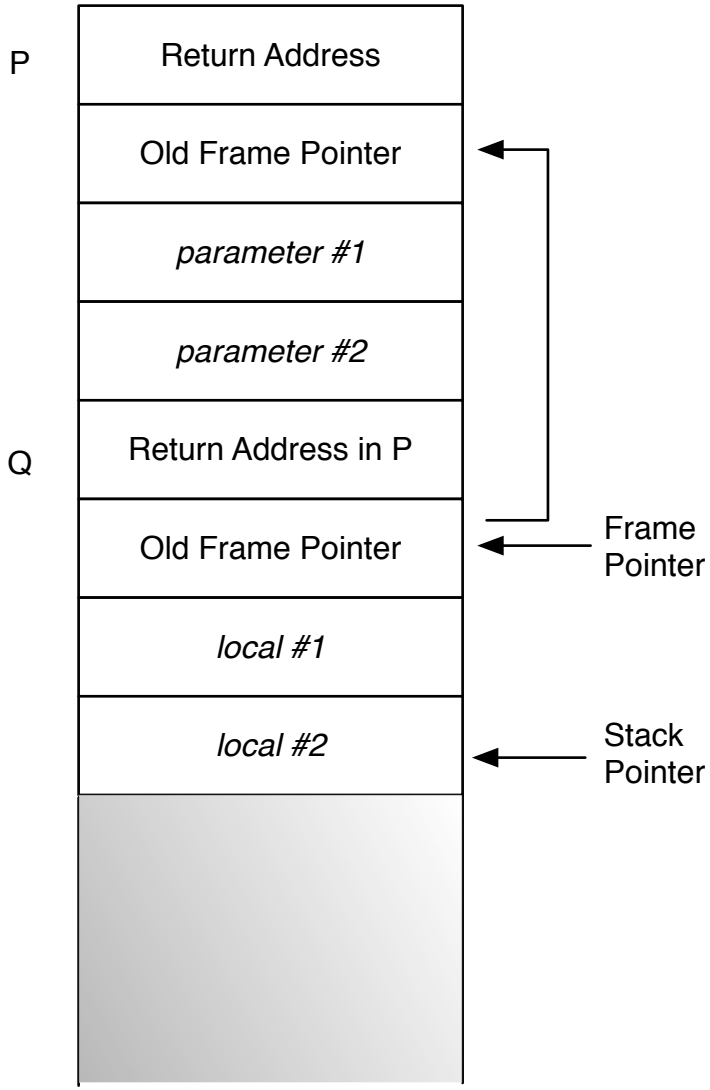
- Depending on what resides at memory location buffer[20]
  - Overwrite user data or code
  - Overwrite system data or code
  - Or program executes just fine

Process Memory Map



- Top of memory = high address
- Data = static variables
- Stack holds
  - Dynamic local variables
  - Parameters to functions
  - Return address
- Heap holds dynamic data

Stack Overflow



- Occurs when buffer is located on stack
- Local variables below saved frame pointer and return address
- Overflow of a local buffer can potentially overwrite key control items
- Attacker overwrites return address with address of desired code (**Code Injection**)
- Program returns to wrong location; executes code of the attacker's choice

Stack Overflow Variants

- Target program can be:
  - A trusted system utility
  - Network service daemon
  - Commonly used library code, e.g. image
- Shellcode functions
  - Spawn shell
  - Create listener to launch shell on connect
  - Create reverse connection to attacker
  - Flush firewall rules
  - Break out of choot environment

Other Malware Attacks

- Heap overflow, global data overflow, format string overflow, integer overflow
- Injection attacks: Program flaws related to invalid input handling that then influences program execution
  - Most common type occurs with scripting languages when input data is passed as a parameter to another helper program on the system, whose output is then processed and used by the original program.
- SQL Injection: User supplied invalid input is used to construct a SQL query against a database
- Code injection: User input include code that is then executed
- Cross Site Scripting (XSS): Input provided to a program by one user is later output to another user. Involves the inclusion of script code in the HTML content of a web page displayed by a user's browser.