Network Security
Attack and Defense Techniques
Anna Sperotto, Ramin Sadre

Design and Analysis of Communication Networks (DACS)
University of Twente
The Netherlands
Attack Taxonomy

- Many different kind of attacks
- Possible classifications:
  - Attack type (scan, denial of service,...)
  - Attack target (a service, a network, a user,...)
  - Attack goal (crash the target, steal information, modify information,...)
  - ...

UNIVERSITY OF TWENTE.
Port Scans

- Scans are *information gathering attacks*:
  - Find vulnerable services/hosts
  - Discover network topology (used IP addresses,...)
  - System fingerprinting
  - ...

- Can be combined with a “real” attack, e.g., a buffer overflow (Ping Of Death, 1997)
- Tool for scanning: `nmap`
TCP port scan: regular connection

**Attacker**
- SYN
- SYN/ACK
- ACK
- RST

**Target**
- accept connection
- deny connection
- ignore, block

+ Easy to implement
– Slow
TCP port scan: SYN scan

Attacker

SYN

Target

SYN/ACK

accept connection

RST
deny connection

RST

ignore, block

+ Fast
– Do-it-your-self

UNIVERSITY OF TWENTE.
UDP port scan

- UDP is connectionless
- Two approaches:
  1. Wait for negative answer (ICMP message “port unreachable”)
  2. Wait for positive answer
     Example: send DNS query to port 53 and wait for DNS response
SSH attacker
How to hide

- The target system knows your IP address
  - *Slow scan*
  - *Distributed scan*: multiple, coordinated scanners
  - *Indirect scan*: idle scan (1998),...
  - ...

UNIVERSITY OF TWENTE.
Idle scan

- How to ask the zombie?
- Fragment ID field in IP header
Idle scan

Attacker

SYN

Spoofed IP address

Zombie

SYN/ACK

RST ID=12345

Target

SYN/ACK

RST ID=12346
Denial-Of-Service (DoS)

- **Goal:** overload or crash the server to make the service unavailable
- **Types**
  - Brute-force:
    - Send a lot of data (overload network), a lot of queries (overload server CPU),…
  - Semantic:
    - Exploit vulnerability (buffer overflow,…)
    - Send heavy requests (triggering complex operations)
DoS against DNS server

- Overload DNS server with queries

- Problems:
  - Attacker may be too slow (CPU, network bandwidth,...)
  - Defense: blocking the attacker's IP address is easy
Distributed DoS (DDoS)

Coordinated attack from multiple hosts

Attacker 1

Attacker 2

Attacker 3

...
DDoS against IRC Server

• ~375 Million SYN packets in 800s
DDoS against IRC Server

- Attacks can have side effects on your monitoring/defense infrastructure
- Here: data loss at mirror port and at collector
On Large Scale: 
Backscatter Analysis with a Network Telescope

(SOURCE: Inferring Internet Denial-of-Service Activity, Moore et al., 2001)

UNIVERSITY OF TWENTE.
Backscatter Analysis for DoS attacks

- In Moore, 2001, a /8 network was monitored
- ~24.5 DoS attacks per hour

- Assuming uniformly distributed spoofed source addresses, this would correspond to

\[ 24.5 \cdot \frac{2^{32}}{2^{24}} = 6272 \text{ attacks/h} \]
UNIVERSITY OF TWENTE.

DNS
Simple DNS Query

query: www.utwente.nl
response: 130.89.1.50 (A record)

DNS server (UDP port 53)
Recursive DNS Query

- **host**
  - www.utwente.nl?
  - response: 130.89.1.50

- DNS server X
  - www.utwente.nl?
  - response: 130.89.1.50
  - DNS server ns1.utwente.nl (authoritative)
DNS Response Cache

host: www.utwente.nl?
response: 130.89.1.50

DNS server X

Cache: www.utwente.nl = 130.89.1.50

DNS server UT (authoritative)
Cache Poisoning (Variant 1)

- Goal: compromise the DNS information
- Based on:
  1. Feature: DNS clients and servers cache responses
  2. Feature: DNS responses can contain additional entries
  3. Bug: some DNS server implementations don't validate the authority of a responder
Cache Poisoning (Variant 1)

Attacker
www.hacker.com?
response: 1.2.3.4

DNS server X
www.hacker.com?
response: 1.2.3.4
additional section: www.utwente.nl=5.6.7.8

Cache:
www.utwente.nl = 5.6.7.8
Cache Poisoning (Variant 1)

- also possible for entire domains: modify the cache entry for the nameserver of another domain
Cache Poisoning (Variant 2)

- Host queries DNS server X for www.utwente.nl.
- DNS server X returns a response of 5.6.7.8.
- The attacker responds with 5.6.7.8 with a spoofed IP address.
- Host queries DNS server UT for www.utwente.nl.
- DNS server UT responds with 130.89.1.50.
- Cache records www.utwente.nl = 5.6.7.8.
Cache Poisoning (Variant 2)

- Not so easy! DNS uses query IDs:
  - queries and responses carry a random ID
  - response ID must match query ID
- Attacker has to guess query ID
  - brute-force: send thousands of responses with different IDs
  - predict ID: some DNS servers use(d) flawed RNG to generate next ID
Cache Poisoning (Variant 2)

Brute-force attacks work!

- Some DNS servers always use the same source port to query other servers
- Solution: randomize the source port, too (July 2008)
- Attacker has to guess ID and source port
Reflected DoS Attack

- usually as distributed attack: multiple attackers, multiple DNS servers (DDoS)
Amplification

- Initial DNS definition:
  60 bytes query → 512 bytes answer (8.5x)
- EDNS (RFC 2671) allows larger answers
- Combining different response types:
  answers larger than 4000 bytes possible (>60x)
- In 2006, Vaughn&Garon studied DDoS attacks with up to 140,000 DNS servers, resulting in 10Gbps
DNS tunneling

- You sit at the airport
- WLAN provided, but any access to a Web server, FTP, P2P,… is chargeable
- Is there a way to avoid the fee?
- Would it be an *attack*?
  - You are bypassing the billing/security policy of your ISP
  - Data exfiltration for cyber-espionage
DNS tunneling: upstream

host \( \rightarrow \) DNS server of ISP

\( helloworld.mydomain.nl? \)

DNS server mydomain.nl

\( helloworld.mydomain.nl? \)
DNS tunneling: upstream

- Query can contain up to 252 characters
- Character set restricted: not case-sensitive,...
- ~5 bit/character, ~110 bytes
DNS tunneling: downstream

host

DNS server X

helloworld.mydomain.nl?

response: hi!

DNS server mydomain.nl

helloworld.mydomain.nl?

response: hi!
DNS tunneling: downstream

Main limitation:
- Response < 512 bytes to prevent fragmentation

Server responds with TXT-record:
- Character set restricted: 7 bit ASCII
- ~6 bit/character, ~220 bytes

As for amplification, EDNS (RFC 2671) allows larger answers

Using MX-records and A-records is possible, too, but more complicated (data may be reordered)
Example of DNS Tunneling (Iodine)


1329812676.524541 IP 192.168.1.1.3128 > 192.168.1.2.60531: Flags [S] 1329812676.525743 IP 2.2.2.2.53 > 1.1.1.1.51823: 15184 1/0/1 NULL
