University of Connecticut
Computer Science and Engineering
CSE 4402/5095: Network Security

‘Knowledge is Power’: Reconnaissance and Scanning

Last updated: Sunday, 08 December 2024

Reconnaissance and Scans: Agenda

- Introduction
- TCP scans
- UDP scans
- DNS scans

Penetration testing: ethical hacking ?

- Goal of pen-testing:
  - Evaluate security, find and fix vulnerabilities
  - By `playing’ attacker interacting with the system
  - Ethically: with permission of system owners (and users?)
- Should Pen-testers know network, organization, source?
  - Three approaches – often combined
  - Black-box: no info – ‘most realistic’
    - find, minimize ‘public’ exposure of network
  - White-box: Kerchkoffs’ principle’ – system should be secure even if details known [all but keys, secrets]
  - Grey-box: provide information and access like provided to users

Pen-Testing : risks, social engineering

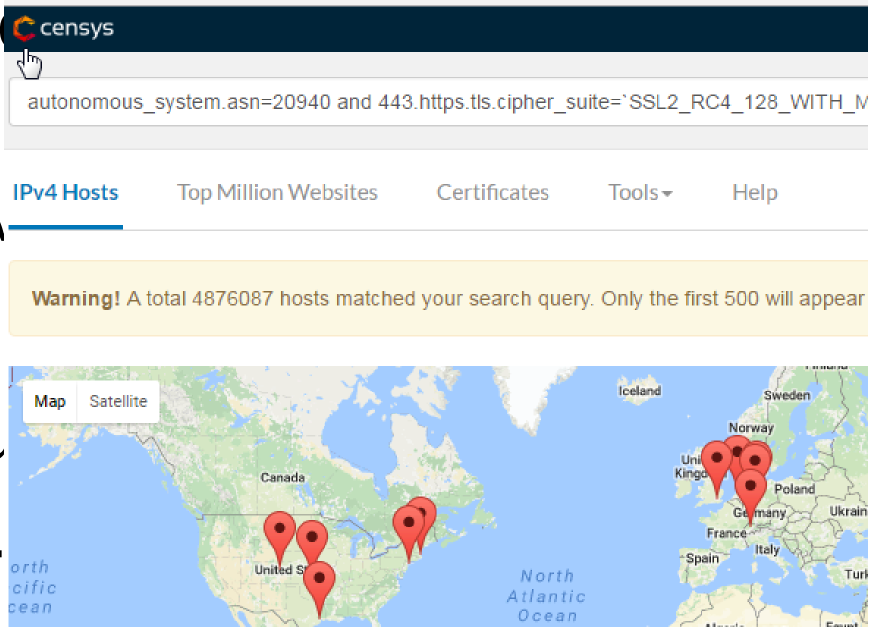
- Possible damage to operational systems
  - By mistake – or by ‘rogue tester’
  - As side-effect, e.g., annoying spam/phishing messages
- Include social engineering attacks in pen testing?
    - Social engineering attacks exploit users psychology and social behaviour to circumvent defences
    - Include (spear) phishing, social network scams, cracking of weak/multi-use passwords, …
  - Often most effective attacks
  - But most `costly’ to pen-test
  - Annoys legit users and operators

Reconnaissance - 'Knowledge is Power'

- First step of black-box hacking
  - And of many real attacks
- **Active reconnaissance: network scans**
  - Tools: NMAP (classic), ZMAP (efficient), …
  - We'll study this in a later lecture
- **Passive/public reconnaissance**
  - Google, WhoIs, Finger, social networks…
  - Reasonable queries in victim's site
  - Paid/Free Search Engines of Daily Internet-Scans
    - Shodan.IO: 'first search engine for internet-connected devices'
    - Censys.IO

Example: Censys Scanning Engine (1)

- Search in daily-ZMAP scans :
  - Hosts on public IPv4 spac
  - X.509 certificates
  - Websites in Alexa's top 1N
- Akamai webservers…
- using insecure cipher-su
  - SSL2 and RC4 and MD5…
    autonomous_system.asn=20940 and 443.https.tls.cipher_suite= `SSL2_RC4_128_WITH_MD5`



censys

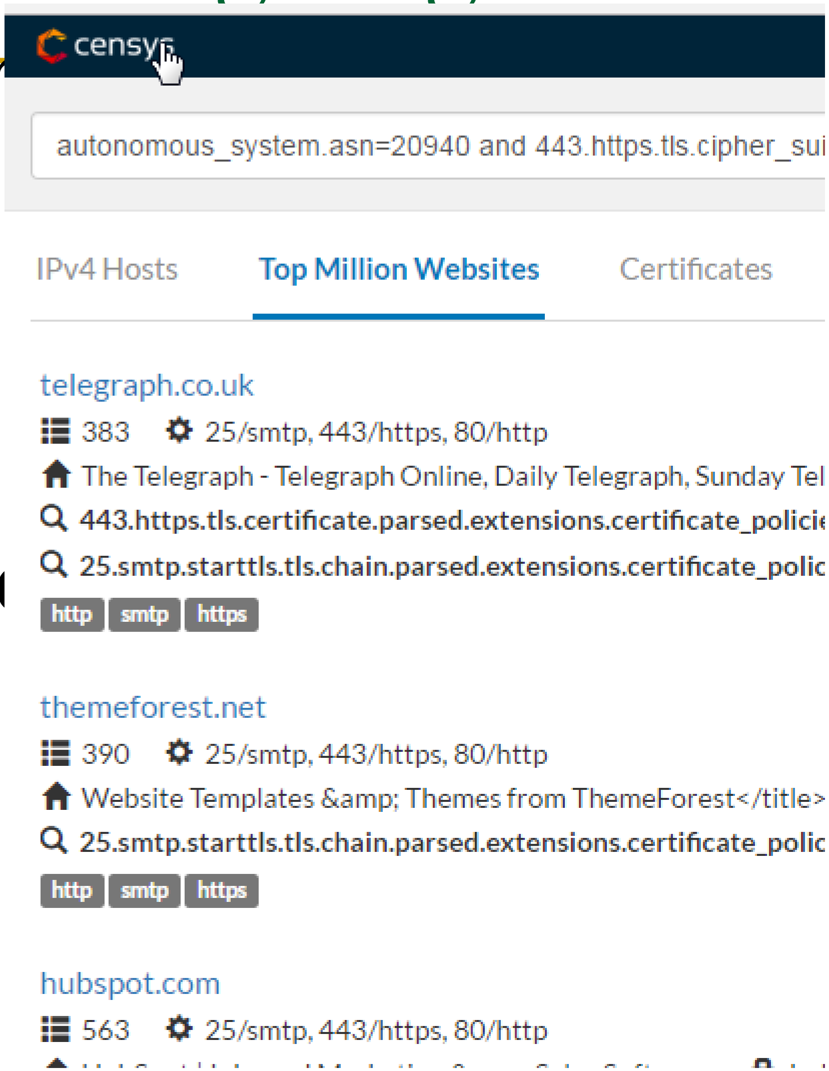
autonomous_system.asn=20940 and 443.https.tls.cipher_suite=`SSL2_RC4_128_WITH_M

IPv4 Hosts | Top Million Websites | Certificates | Tools▾ | Help

Warning! A total 4876087 hosts matched your search query. Only the first 500 will appear

Example: Censys Scanning Engine (2)

- Search in daily-ZMAP sca
  - Hosts on public IPv4 space
  - X.509 certificates
  - Websites in Alexa’s top 1M
- Akamai webservers…
- using insecure cipher-suit
  - SSL2 and RC4 and MD5… autonomous_system.asn=20940 and 443.https.tls.cipher_suite= `SSL2_RC4_128_WITH_MD5`
  - Same, ranked (in Alexa 1M list)…



autonomous_system.asn=20940 and 443.https.tls.cipher_sui

telegraph.co.uk

383 25/smtp, 443/https, 80/http

The Telegraph - Telegraph Online, Daily Telegraph, Sunday Tel

443.https.tls.certificate.parsed.extensions.certificate_policie

25.smtp.starttls.tls.chain.parsed.extensions.certificate_polic

themeforest.net

390 25/smtp, 443/https, 80/http

Website Templates &amp; Themes from ThemeForest</title>

25.smtp.starttls.tls.chain.parsed.extensions.certificate_polic

hubspot.com

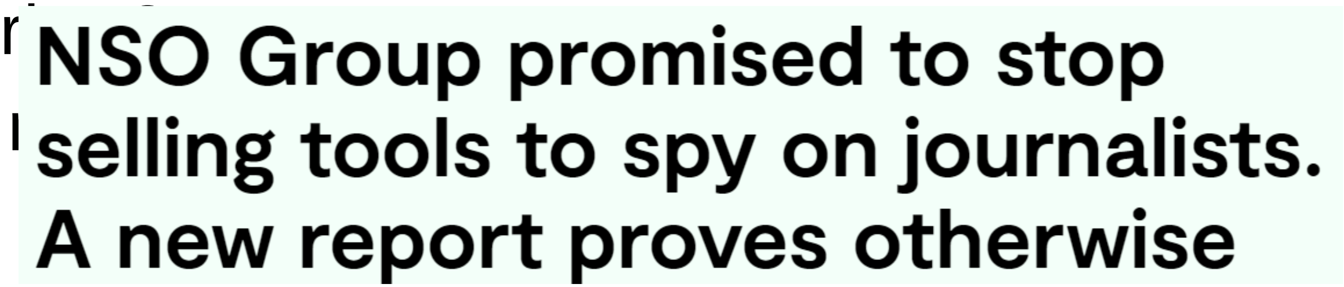
563 25/smtp, 443/https, 80/http

Cybersecurity Ethics

- Basic cyber-sec ethics:
  - Do no harm
    - Intentional – or by negligence (e.g., experiment `in wild')
    - Don't attack, don't provide attack tools,…
- But there are dilemmas…
  - Ok to provide 'dual-use' tools, e.g., Shodan?
    - Can be (and was) abused by black-hat hackers
    - Many 'awesome' (exploitable) queries
    - Unlike Censys, does not follow ethical guidelines
    - So, some consider it unethical
    - Wiki: named after SHODAN (Sentient Hyper-Optimized Data Access Network), an AI antagonist of the cyberpunk-horror themed game System Shock

Cybersecurity Ethics

- Basic cyber-sec ethics:
  - Do no harm
    - Intentional – or by negligence (e.g., experiment `in wild’)
    - Don’t attack, don’t provide attack tools,…
- But there are dilemmas…
  - Ok to provide ‘dual-use’ tools, e.g., Shodan?
  - Ok to help law enforcement, e.g., against terror
  - One



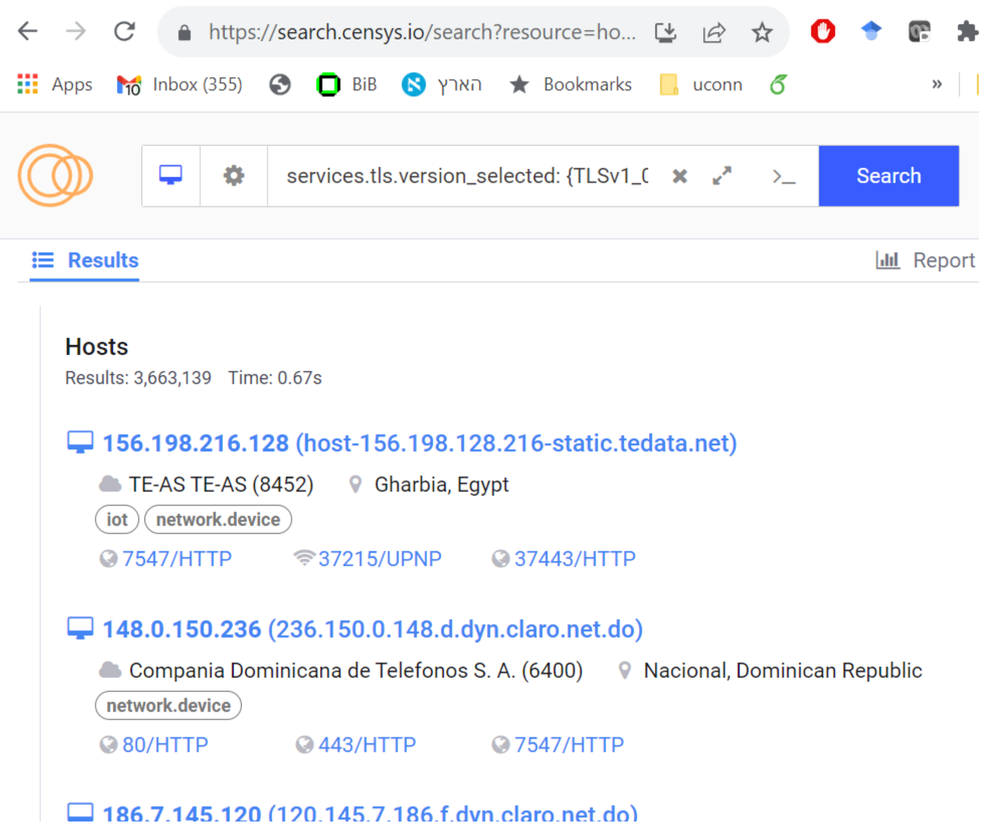
NSO Group promised to stop selling tools to spy on journalists. A new report proves otherwise

Reconnaissance - 'Knowledge is Power'

- First step of attack and of black-box pen-testing
  - Also: for research (academic, industry) and identify customers
- **Passive/public reconnaissance**
  - General-info: search engines, social networks…
  - Specific information (free/pay):
    - DNS, WhoIs, Caida, …
    - Internet-wide network scan engines: Shodan.IO, **Censys.IO**

Example: Censys Scanning Engine (1)

- Search in daily-ZMAP scans :
  - Hosts on public IPv4 space
  - X.509 certificates
- Some simple examples…
- Servers running insecure TLS (1.0, 1.1):
  - services.tls.version_selected: {TLSv1_0, TLSv1_1}



https://search.censys.io/search?resource=ho...

services.tls.version_selected: {TLSv1_(

Hosts

Results: 3,663,139 Time: 0.67s

156.198.216.128 (host-156.198.128.216-static.tedata.net)

TE-AS TE-AS (8452) Gharbia, Egypt

iot network.device

7547/HTTP 37215/UPNP 37443/HTTP

148.0.150.236 (236.150.0.148.d.dyn.claro.net.do)

Compania Dominicana de Telefonos S. A. (6400) Nacional, Dominican Republic

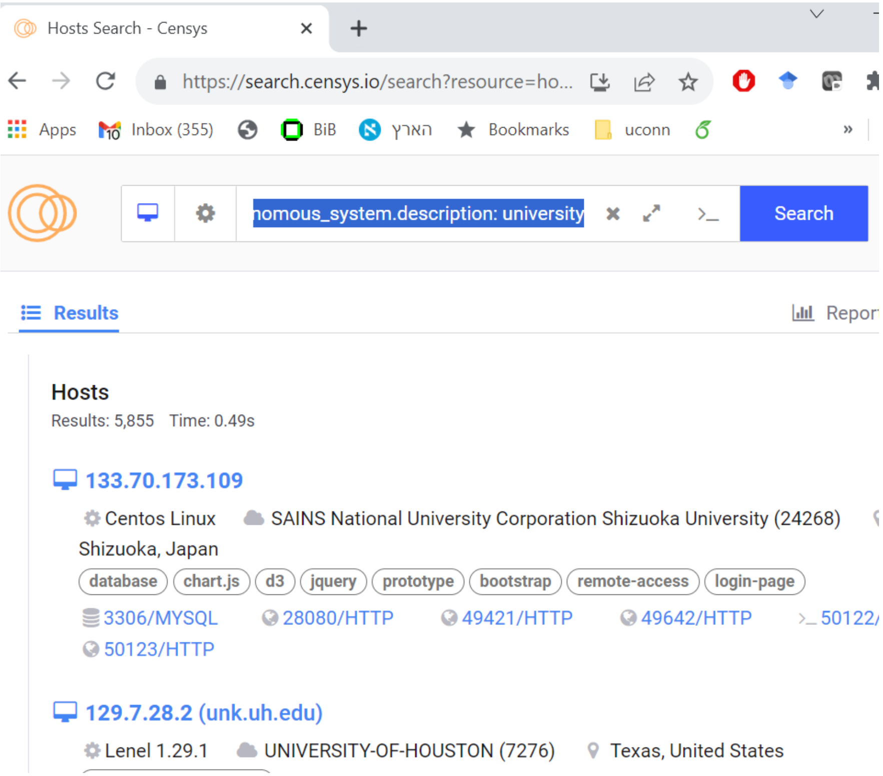
network.device

80/HTTP 443/HTTP 7547/HTTP

186.7.145.120 (120.145.7.186.f.dyn.claro.net.do)

Example: Censys Scanning Engine (2)

- Search in daily-ZMAP scans :
  - Hosts on public IPv4 space
  - X.509 certificates
- Servers running insecure TLS (1.0, 1.1):
  - services.tls.version_selected: {TLSv1_0, TLSv1_1}
- And AS is ‘university’
  - autonomous_system.description: university
- Lots of relevant info – both here and in Shodan.IO


Hosts Search - Censys
https://search.censys.io/search?resource=ho...
Apps
Inbox (355)
BiB
הארץ
Bookmarks
uconn
nomous_system.description: university
Search
Results
Repor

Hosts

Results: 5,855 Time: 0.49s

133.70.173.109

Centos Linux SAINS National University Corporation Shizuoka University (24268)

Shizuoka, Japan

database chart.js d3 jquery prototype bootstrap remote-access login-page

3306/MYSQL 28080/HTTP 49421/HTTP 49642/HTTP 50122/

50123/HTTP

129.7.28.2 (unk.uh.edu)

Lenel 1.29.1 UNIVERSITY-OF-HOUSTON (7276) Texas, United States

Reconnaissance - ‘Knowledge is Power’

- First step of black-box pen-testing and of attacks
- **Passive/public reconnaissance**
  - Open general-info: Google, ChatGPT / Bard, social networks…
  - Open (free/pay) specific-info:
    - DNS, WhoIs, Caida, …
    - Internet-wide network scan engines: Shodan.IO, **Censys.IO**
- **Active reconnaissance:**
  - Spyware
  - Phishing: email, social-networks-contacts
  - Web reconnaissance, crawling
  - **Network scans**
    - Tools: NMAP (classic), ZMAP (efficient, used by Censys), …
    - We’ll study **methods**

Network Scans: Goals

- Goal 1: effectiveness: discover all relevant information
- Goal 2: efficiency
  - Time (speed)
  - Resources: communication, state
- Goal 3: resiliency, availability, minimal requirements
  - Resiliency: avoid blocking by FW etc.
  - Agent: puppet / user-zombie / admin-zombie
- Goal 4: **no attribution, detection (‘stealthy scan’)**
  - Weakly-stealthy scan: avoid logged events, attribution
  - Off-path stealthy: **no exposure of IP** to scan-target
- Goal 5: ethics [a goal for white-hat scanners]
- Reality: if you’re connected, you’re scanned…
  - Attackers, scan engines, pen-testers, researchers

Network Scans: for what information?

- Resources: in general, and for attacks
- Vulnerabilities in victim network
- Behaviors and configurations

Network Scans: for what information?

- Resources: in general, and for attacks, e.g.:
  - Vulnerable hosts that can be exploited (worm)
  - Peer/slave bots, CnC center
  - For DDoS: amplifiers, e.g., open DNS resolvers
  - For stealthy scans+attacks, e.g., IP-ID incrementing hosts
  - For off-path side-channels, e.g., rate-limiting nets/hosts
- Vulnerabilities in victim network
- Behaviors and configurations

Network Scans: for what information?

- Resources: in general, and for attacks

- Vulnerabilities in a (victim/customer) network:
  - Vulnerable product/version, identify by ‘banner’ or fingerprint
  - Vulnerable configurations, e.g.:
    - Vulnerable services, often identified by specific open port
    - DNS vulnerabilities: ???, ??? l port, …
    - Vulnerable web servers: vulnerable TLS / cipher-suite, …
    - Unprotected networks, e.g., no egress filtering

- Behaviors and configurations:
  - Deployed products, configurations; e.g., validating DNSSEC
  - Users of (‘forbidden’) site/service (e.g., Tor or other)

Ethical Research-Scanning

- Researchers scanning non-owned networks (IPs)…
- Be open
  - Publish goals, policy, contact
    - Include clear identification in probes (where possible)
- Opt-out mechanisms:
  - Scan-specific and standard (e.g., robots.txt)
- Be considerate: do no harm
  - Limited experiment before large-scale scanning
  - Avoid side-effects
  - Rate-limit, load-balance
    - Also important to avoid target's rate-limiting !

Categories of Network Scans

- **Direct (on path):** send requests, inspect responses
  - 👎 Visible: exposes scanner’s IP, logged
    - 👎 Weakly-stealthy scans: expose IP but avoid log ?
  - Essential (only?) when response required:
    - Version, header, options
    - `Fingerprint’ of OS, version
      - TTL, TCP init window size, MSS,IP-ID, retransmit pattern…
    - Amplification (is response really required?)
- **Off-path (spoofed):** do not expose scanner’s IP!
  - Often via side channels, e.g., IP-ID of 3rd party
    - Usually requires raw sockets
    - What’s this IP-ID? Well, it begins with IP fragmentation...

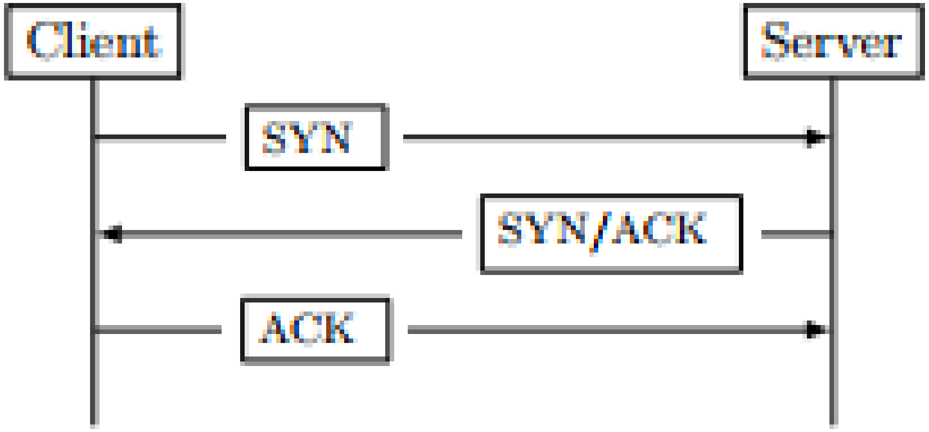
Reconnaissance: Agenda

- Introduction
- **TCP scans**
- UDP scans
- DNS scans

We discuss specific scans; you should learn principles and techniques, to be able to apply to other scans

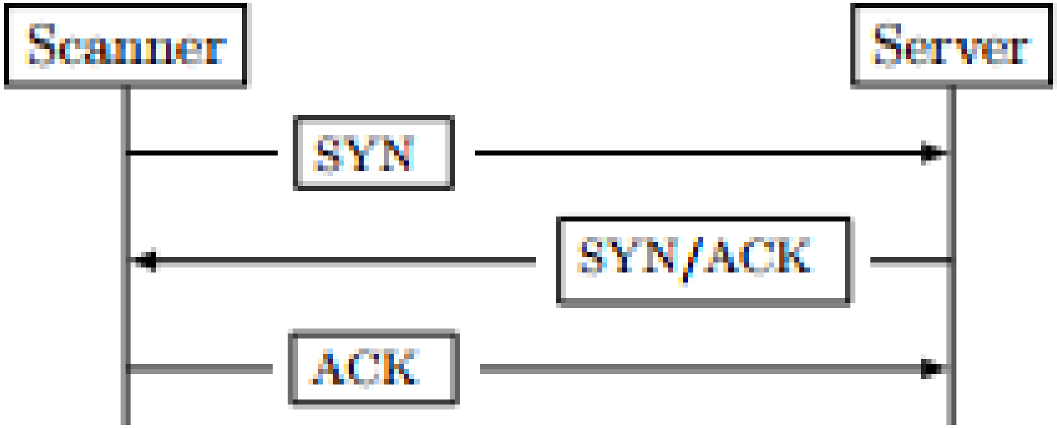
Recall: TCP three-way handshake

- TCP uses 3-way handshake to setup connection:
  - Allocate buffers (or abort, if unavailable)
  - Agree on client, server's ISNs (Init Seq Number)
    - Reliability for this connection - & separate from others
  - Agree on options, e.g., MSS (maximal segment size)



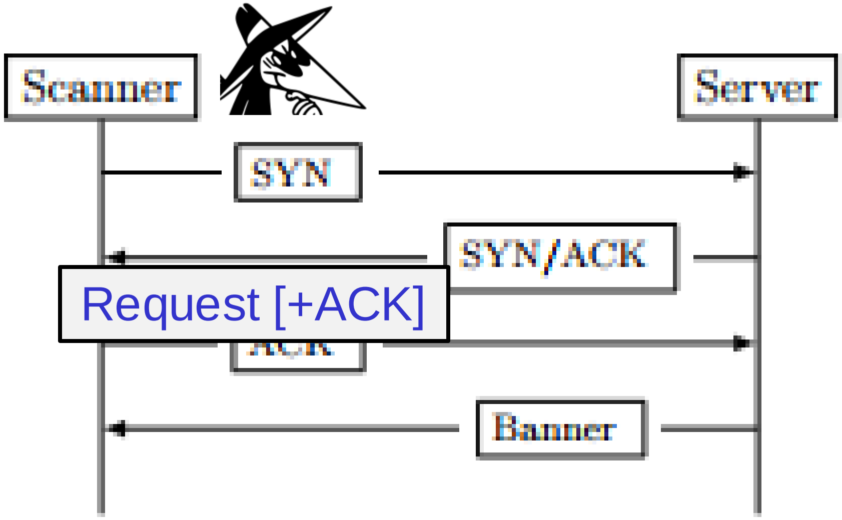
TCP Connect Scan

- Scan using ‘standard TCP process’
  - Detect if connection succeeds of fails
  - Use standard TCP sockets
    - If receiving SYN/ACK, respond with ACK (and succeed)
    - If not: resend SYN, eventually time-out (and fail)
- Pro: easy to deploy: uses standard TCP sockets
- Cons? overhead, ???



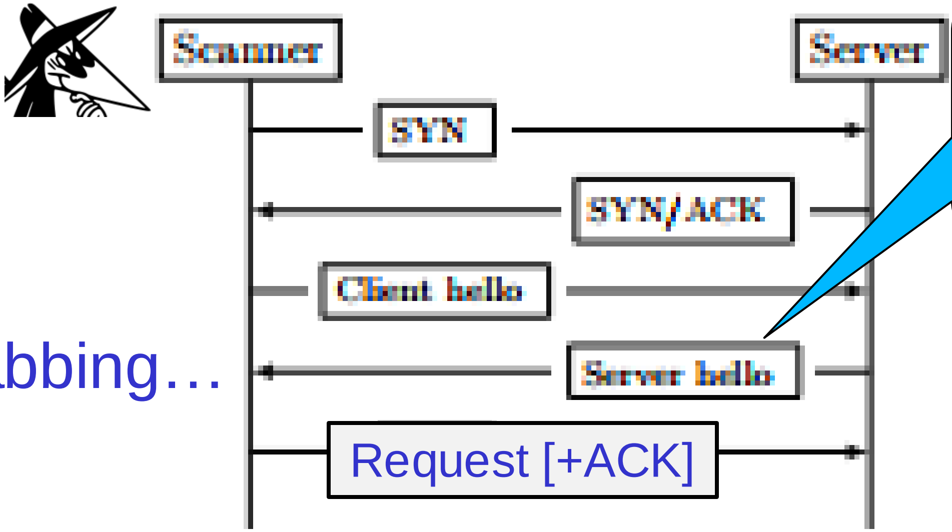
Application Scan and Banner Grabbing

- Complete TCP connection, then…
  - HTTP: send request(s), wait for response
  - SMTP: wait for ‘ready’ from server (220 OK)
  - May continue handshake to get more responses
- Allows detection of specific application, behavior
- Application may respond with useful data
  - E.g., ‘Banner’ identifying software



TLS-Hello Scan

- Application scan allows completing appl handshake
- TLS-hello scan: receive server-hello message, incl:
  - Server’s protocol version, cipher-suite responses
    - Need to send different client versions to ‘learn’ server
  - Server’s extensions, response to client extensions
  - Other (e.g., DH groups, cert)

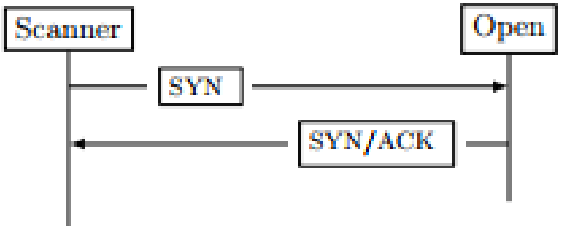


Ciphersuite, version, extensions, …

Or also do banner-grabbing…

The weakly-stealthy TCP SYN scan

- Scanner sends SYN to target IP:port
  - Target reachable, port open: SYN/ACK [scanner doesn’t respond!]
  - Target reachable, port closed: RST
  - Unreachable: ICMP ‘unreachable’ response or timeout
  - Filtered/non-existing target: no response (timeout)



TCP On-Path Scans

Connection scan:

- Full handshake
- Full (logged) connection
- TCP Socket library
  - Resend SYN till Time-Out
- Visible, attributable
- Easy to deploy (sockets)

SYN scan:

- Only SYN handshake
- Half-open connection
  - Not logged? Suspect?
- Requires raw socket
- Weakly-stealthy, deniable

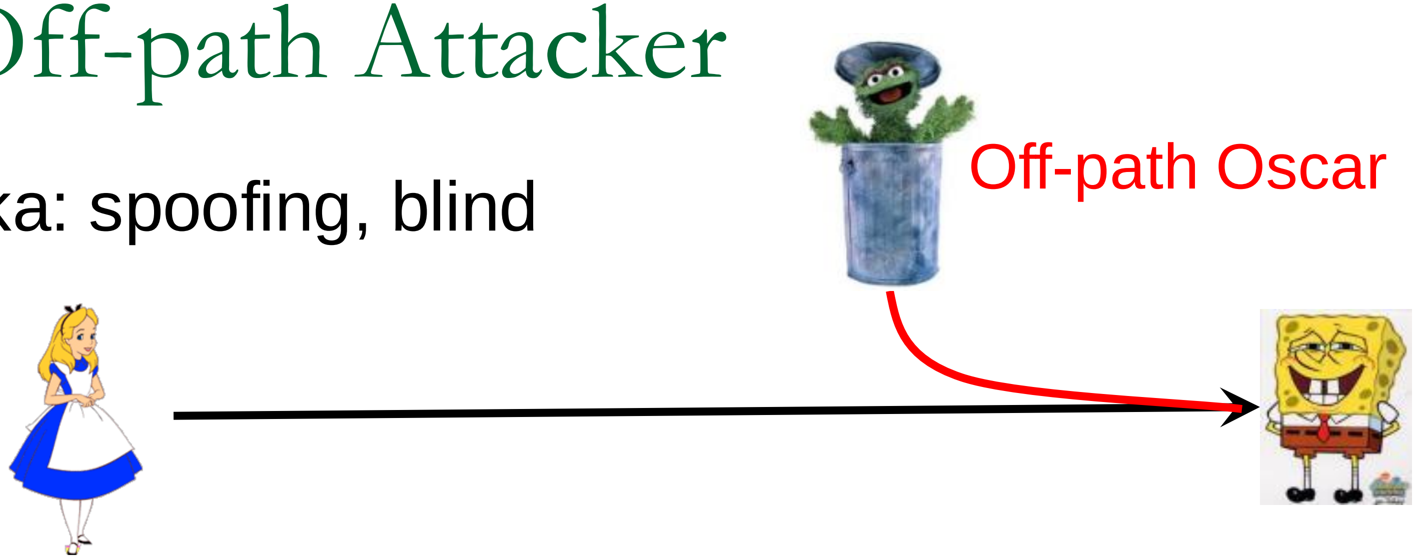
Next: a stealthy, off-path TCP scan

Other TCP on-path scans:

- NULL (no flag), FIN and XMAS (URG, PSH and FIN all set)
- Standard response: RST if port closed, none if open
- Raw socket, obvious attack
- Weakly-stealthy, deniable

Off-path Attacker

- Aka: spoofing, blind



Off-path Oscar

Can ~~eavesdrop~~, **inject**, ~~modify~~

Spoofed sender IP address (‘sender: Alice’)

Cannot receive responses (or original packets)

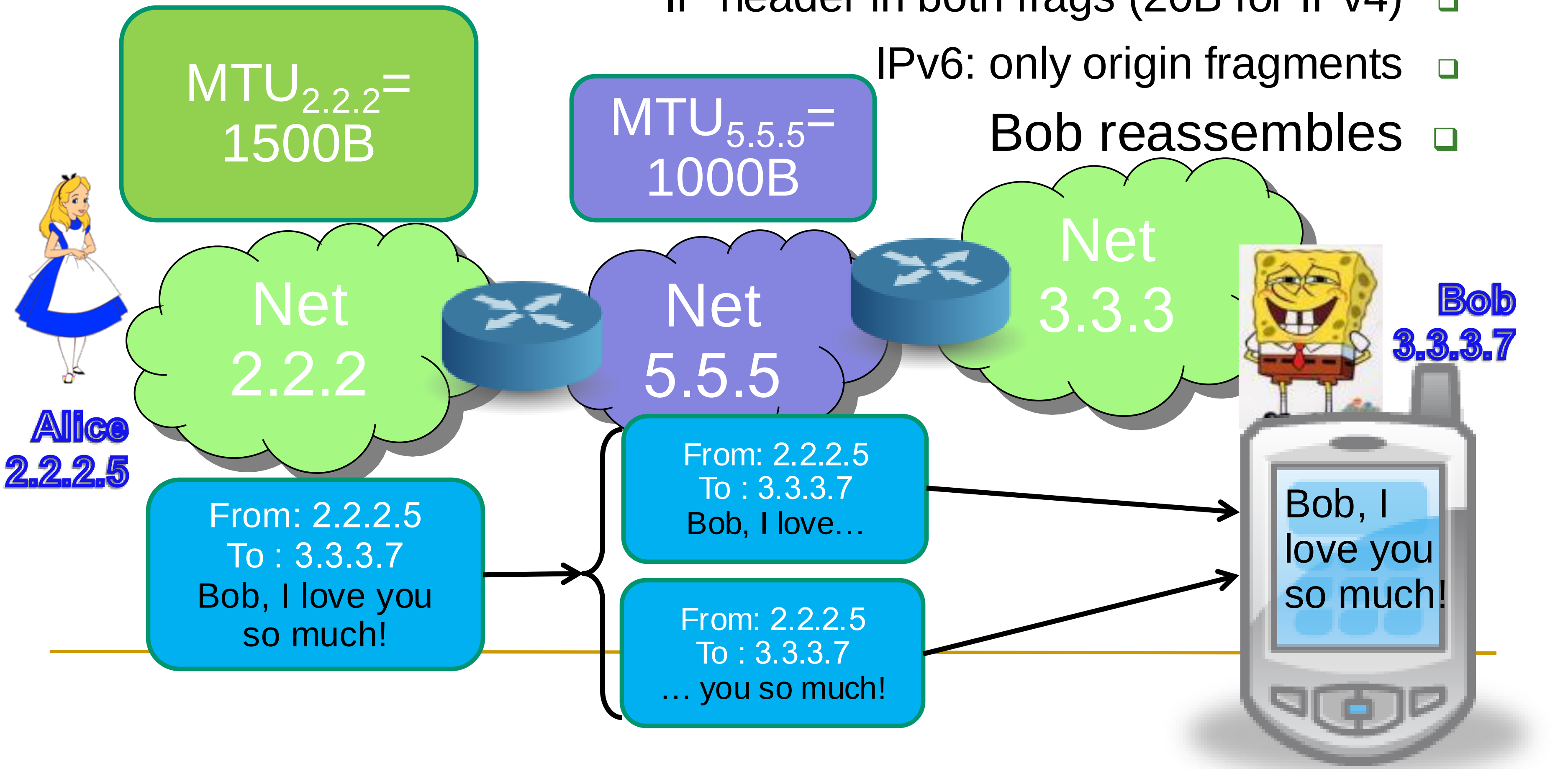
ISPs should prevent: ‘ingress filter’ → many don’t

Off-path ('Idle') TCP Stealthy Scan

- Goal: identify Open/Closed TCP ports
  - Without exposing scanner's IP address
- Idea: use IP-ID Incrementing hosts
  - What's IP-ID?
    - Hint: part of IP header
  - 16-bit field in IP header, used for fragmentation
  - IP-ID Incrementing hosts: increment IP-ID upon sending packets
    - Global-incrementing: one IP-ID counter for all dest-IPs
    - Per-dest incrementing: IP-ID counter per each dest-IP
  - Useful for many stealthy attacks
    - Later: a scan to **find** IP-ID incrementing hosts

The Internet Protocol: Fragmentation

- Every network has a size-limit on packet size (MTU)
- What if we need to send more?
- Solution: Fragmentation
  - IP header in both frags (20B for IPv4)
  - IPv6: only origin fragments
  - Bob reassembles



From: 2.2.2.5
To : 3.3.3.7
Bob, I love you so much!

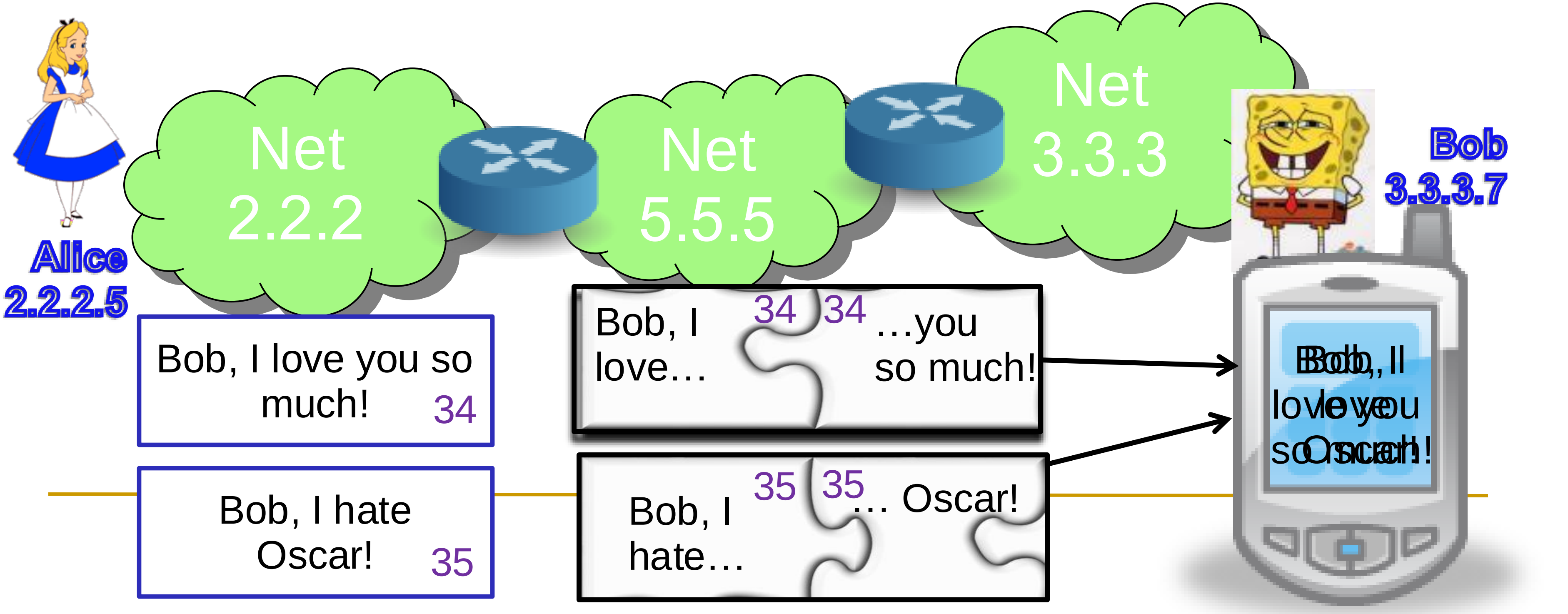
From: 2.2.2.5
To : 3.3.3.7
Bob, I love…

From: 2.2.2.5
To : 3.3.3.7
… you so much!

Bob, I love you so much!

Packet Reassembly: Careful!

- Bob receives fragments of multiple packets
- How to reassemble without mixing?
- Identify each packet
  - By Src, Dst addresses and protocol
  - And: IP-ID (16bit in IPv4; 32bit in IPv6)



Typical methods to choose IP-ID

- Basic goal: avoid collision with an old fragment
- Security goal: unpredictable IP-ID [16b in IPv4]
- Common methods:
  - **Random**
    - Con: ‘birthday paradox’: if >255 packets are in transit (even low), collision occurs with probability ~ ½ !! [16b]
    - Also, good randomization is often hard
  - **Globally-incrementing** [from random initial value]
  - **Per-destination incrementing** [random initial value]
  - ‘**Zero**’: use one of above but only for long packets
    - For short pkts, send IP-ID of zero (or other fixed value)
    - Defeats some IP-ID prediction/exposure attacks

Knowing IP-ID facilitates off-path attacks

Attacker models → Security goals	MitM	Off-Path attack with unknown IP-ID	Off-Path attack Exploiting known IP-ID
Confidentiality and privacy	Broken (without crypto)	Expected	2nd Frag interception attack
Integrity and authentication	Broken (without crypto)	Expected: spoofing, but no modification	2nd Frag spoofing attack
Availability (and efficiency)	Broken	Expected (except by clogging)	Frag-based packet drop and overhead attacks
Stealthy scan	Broken	Expected	Off-path stealthy TCP scan

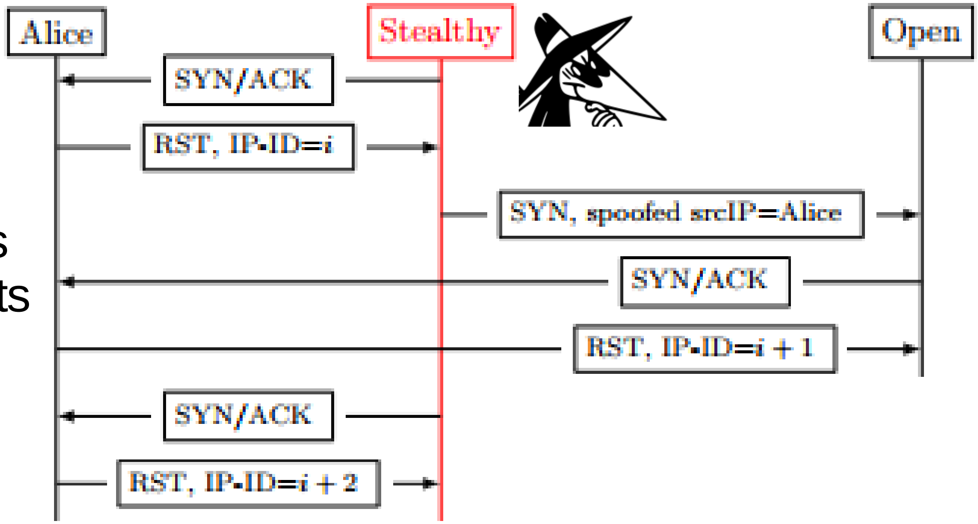
Off-path (‘Idle’) TCP Stealthy Scan

- Goal: identify Open/Closed TCP ports
  - Without exposing scanner’s IP address
  - Using IP-ID global-incrementing hosts (‘helpers’ or `useful idiots’)
    - Global-incrementing: one IP-ID counter for all dest-Ips
- Pros: off-path-stealthy, hard to detect and very hard to attribute, deniable
- Cons: slow, more pkts, raw socket

TCP Idle Off-path Stealthy Scan: Open

Alice: global-incrementing IP-ID host

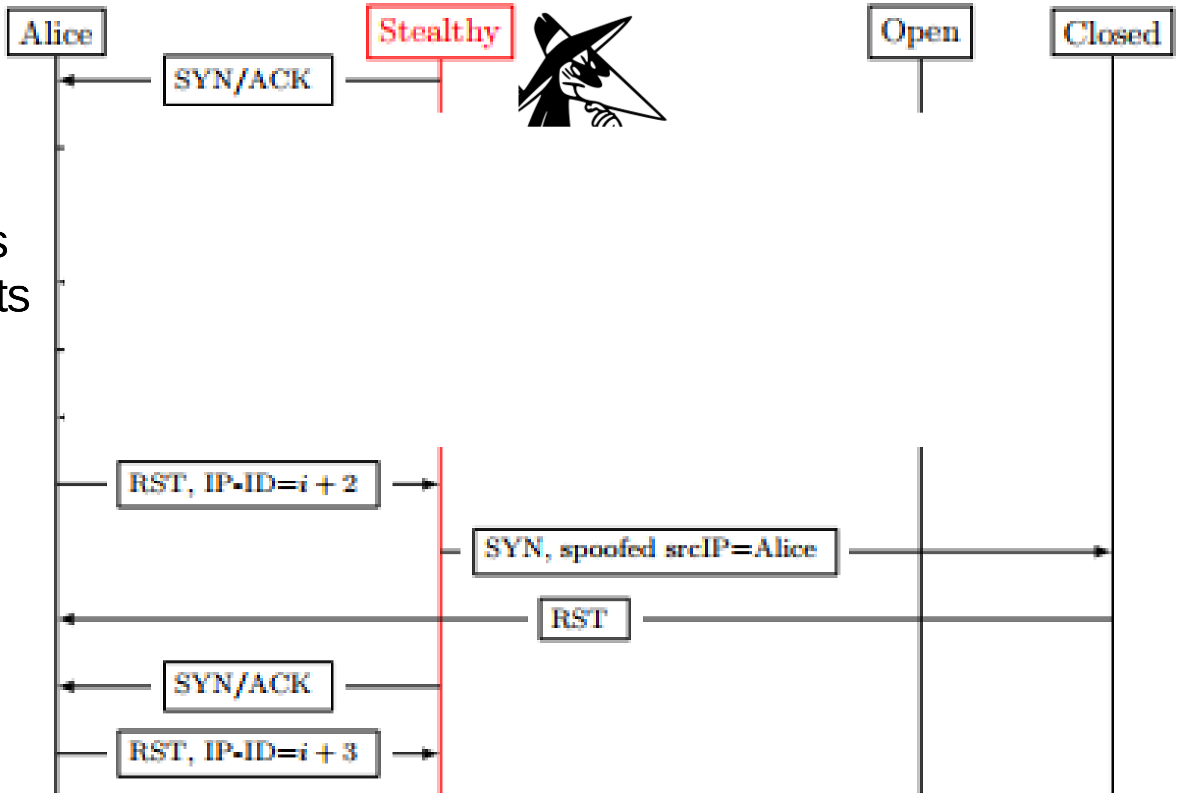
Used to be windows hosts; now: IoT hosts



TCP Idle Off-path Stealthy Scan:Closed

Alice: global-incrementing IP-ID host

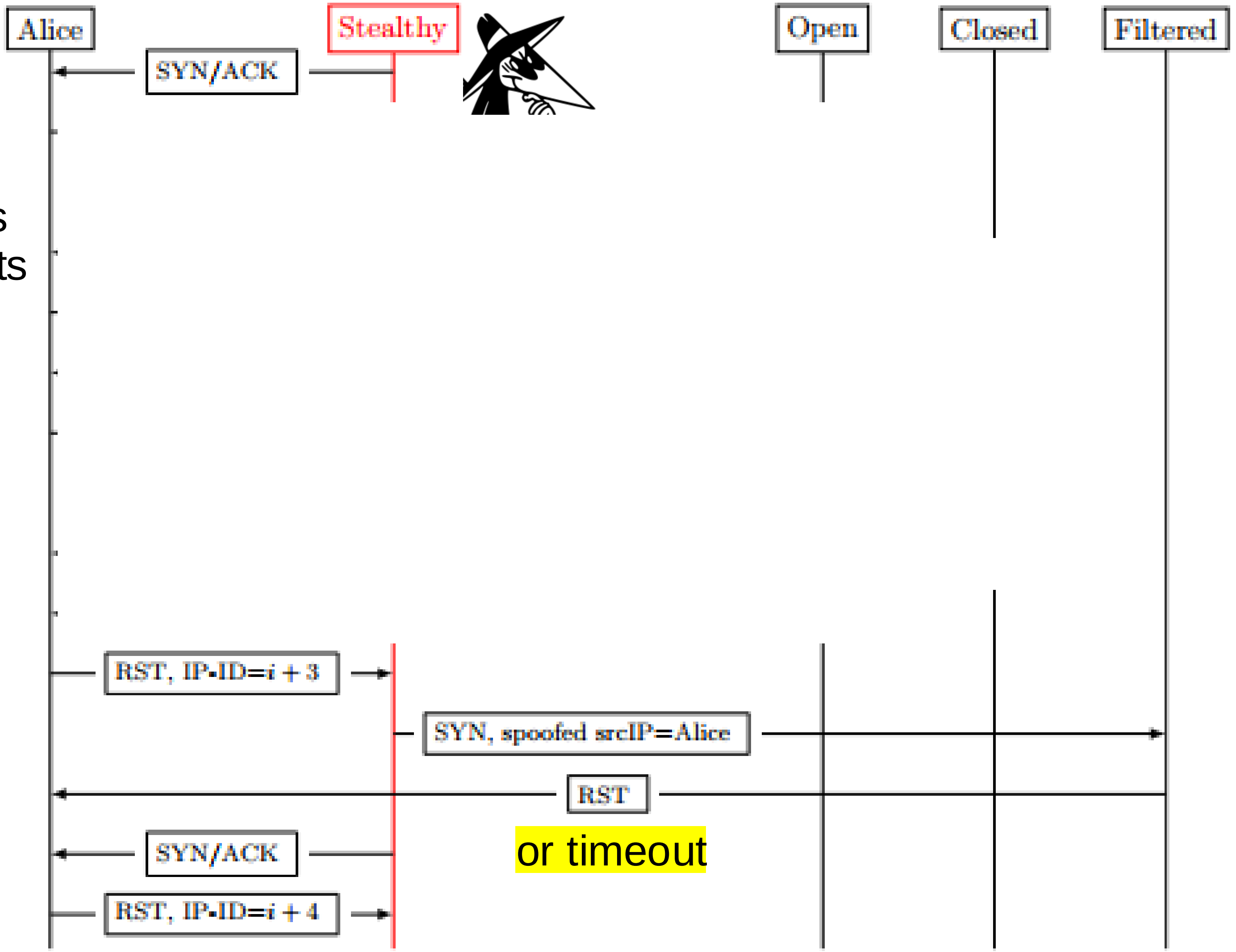
Used to be windows hosts; now: IoT hosts



TCP Idle Off-path Stealthy Scan: Filtered

Alice: global-incrementing IP-ID host

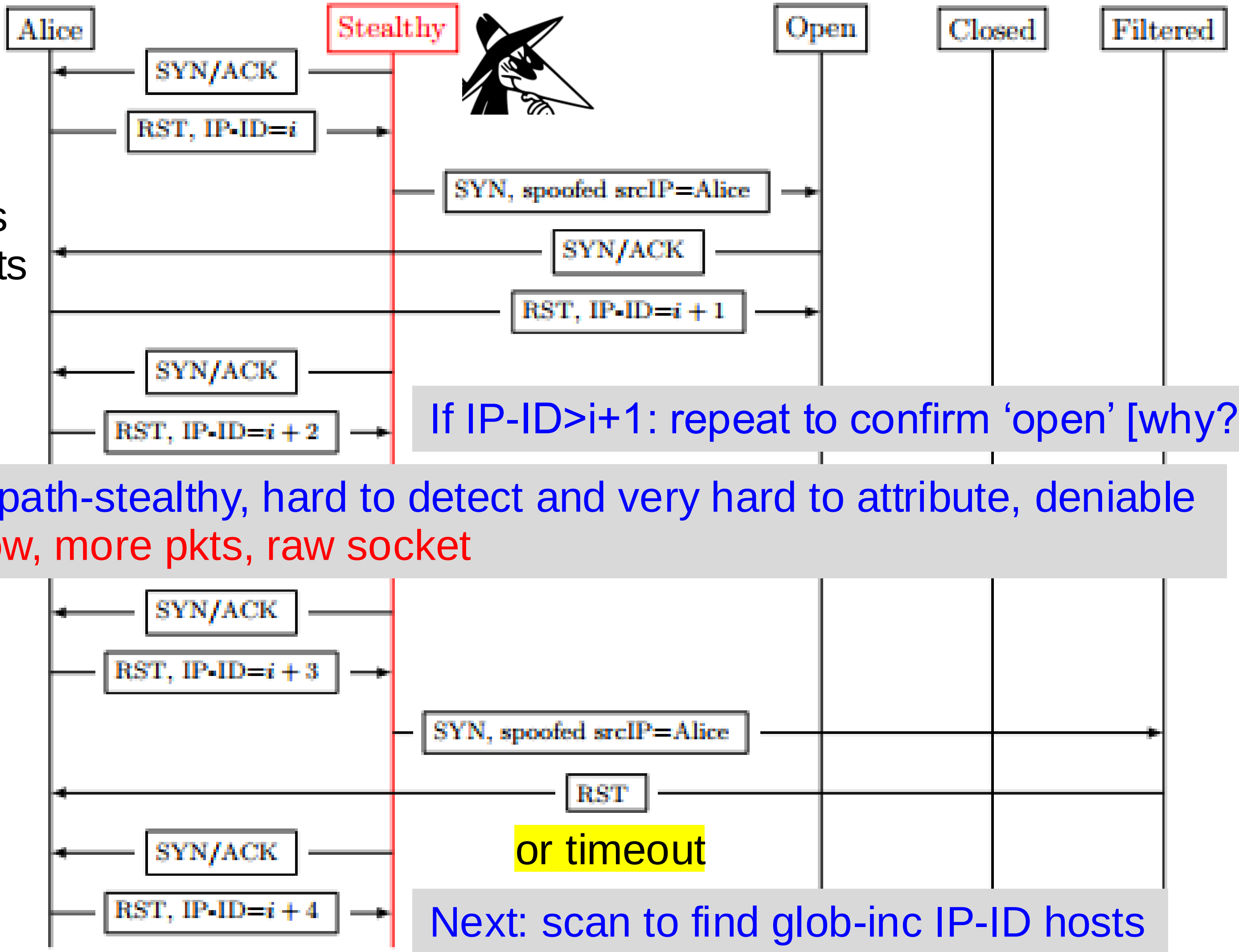
Used to be windows hosts; now: IoT hosts



TCP Idle Off-path Stealthy Scan: Filtered

Alice: global-incrementing IP-ID host

Used to be windows hosts; now: IoT hosts



If IP-ID>i+1: repeat to confirm 'open' [why?]

Pros: off-path-stealthy, hard to detect and very hard to attribute, deniable
Cons: slow, more pkts, raw socket

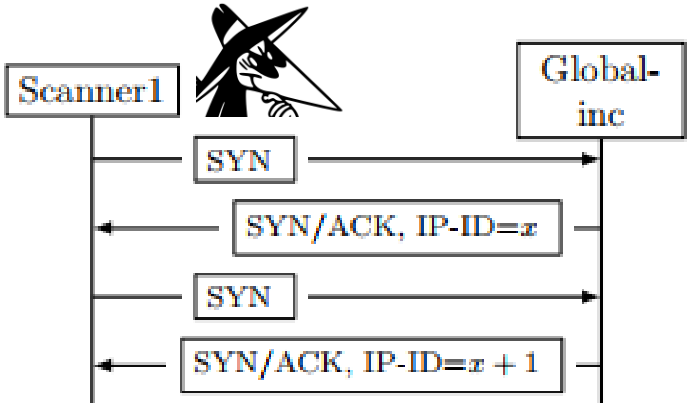
or timeout

Next: scan to find glob-inc IP-ID hosts

Scanning for Global-inc IP-ID helpers

- Basic goal: avoid collision with an old fragment
- Security goal: unpredictable IP-ID [16b in IPv4]
- Common methods:
  - **Random**
    - Con: ‘birthday paradox’: if >255 packets are in transit (even low), collision occurs with probability ~ ½ !! [16b]
    - IPv6 uses 32b IP-ID; collision for ~ 64K pkts < 100MB
  - **Globally-incrementing** [from random initial value]
  - **Per-destination incrementing** [random initial value]
  - ‘**Zero**’: use one of above but only for long packets
    - Send IP-ID of zero (or other fixed value) for short pkts
    - Defeats some IP-ID prediction/exposure attacks

IP-ID Scan: find global-inc hosts



Scanner1

Global-inc

SYN

SYN/ACK, IP-ID=x

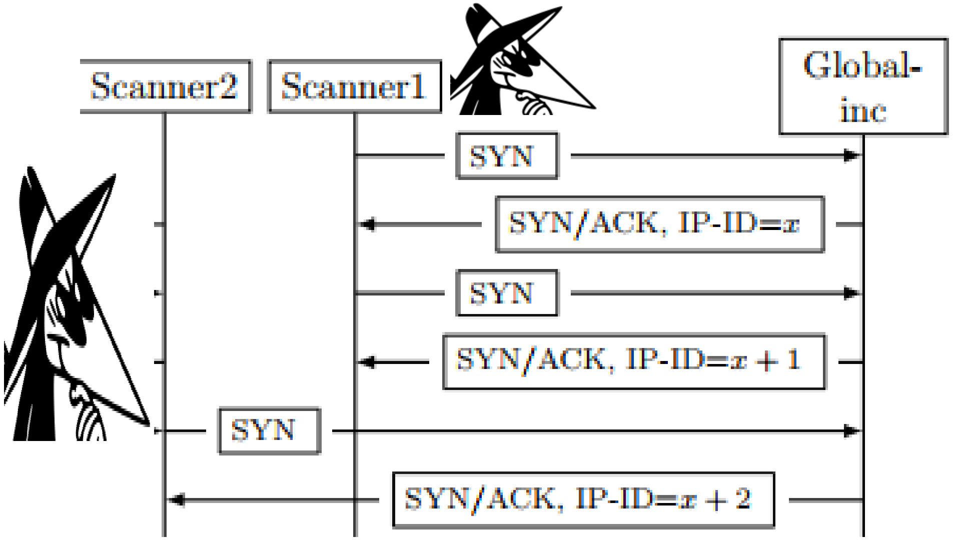
SYN

SYN/ACK, IP-ID=$x+1$

Problem:

???

IP-ID Scan: global-inc or per-dest-inc?

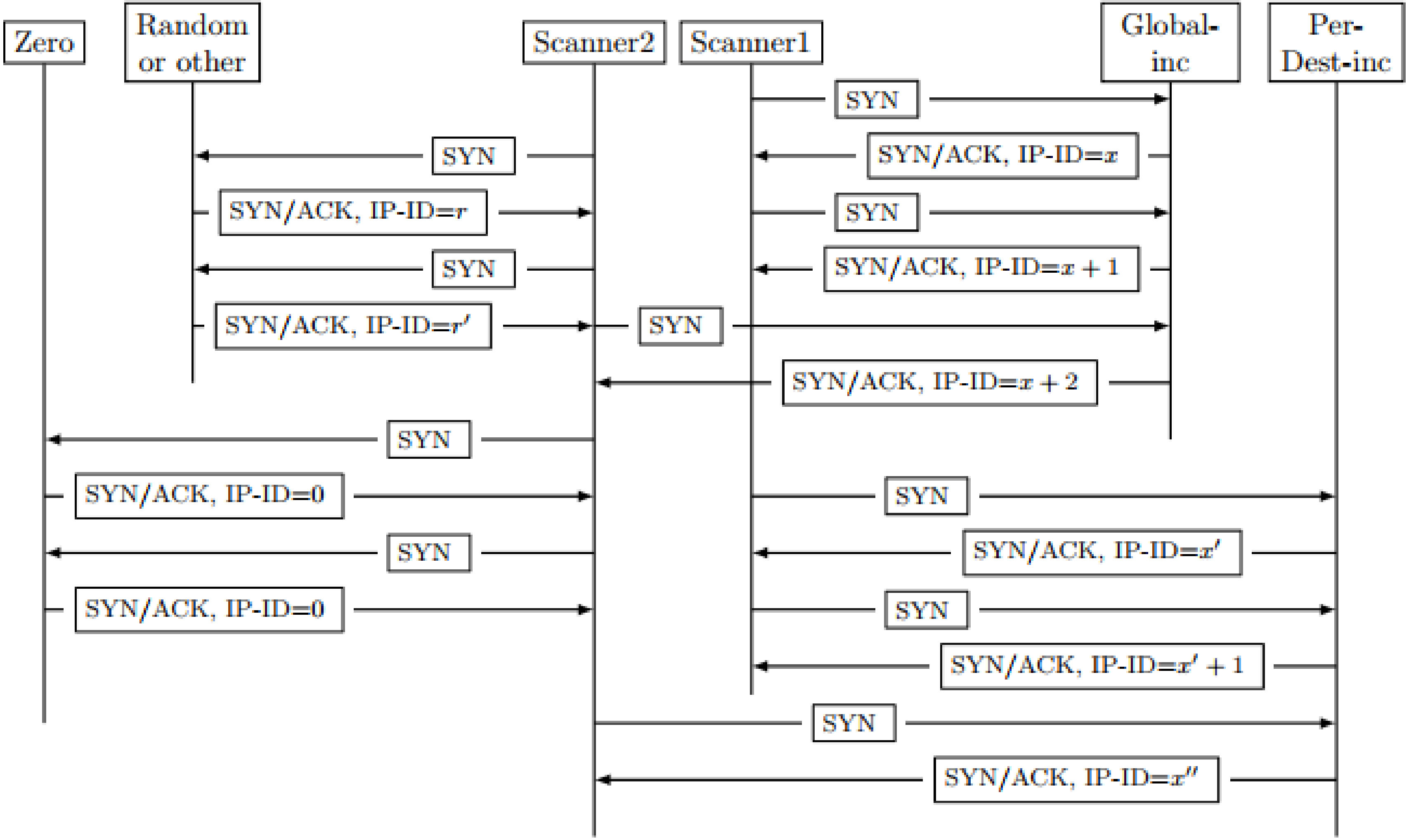


Great, but some hosts are not IP-ID incrementing, mainly:

- Always use IP-ID=0 (for short packets)
- Other ('random')

[there's also a common option which is btw global and per-dest, but we'll ignore it]

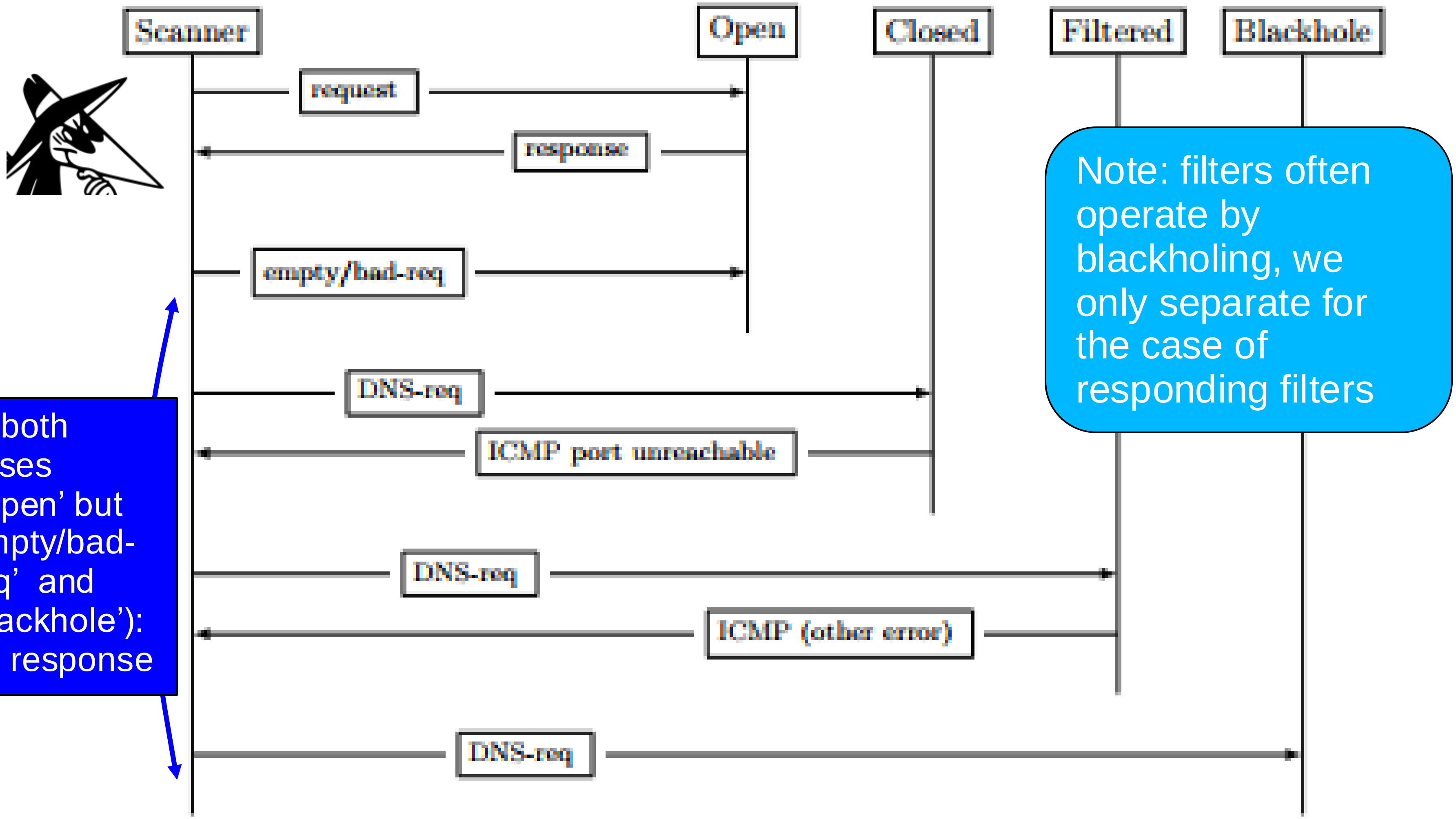
IP-ID Scan


Zero
Random or other
Scanner2
Scanner1
Global-inc
Per-Dest-inc
SYN
SYN/ACK, IP-ID=x
SYN
SYN/ACK, IP-ID=r
SYN
SYN/ACK, IP-ID=$x+1$
SYN
SYN/ACK, IP-ID=r'
SYN
SYN/ACK, IP-ID=$x+2$
SYN
SYN/ACK, IP-ID=0
SYN
SYN/ACK, IP-ID=x'
SYN/ACK, IP-ID=0
SYN
SYN/ACK, IP-ID=$x'+1$
SYN
SYN/ACK, IP-ID=x''

Reconnaissance: Agenda

- Introduction
- TCP scans
- **UDP scans**
- DNS scans

UDP Scan



In both cases ('open' but empty/bad-req' and 'blackhole'): no response

Note: filters often operate by blackholing, we only separate for the case of responding filters

UDP scan: rate-limiting challenge

- Rate limiting provided in all routers
- Traffic policing: limit incoming TCP, UDP packets
  - Mostly against Denial of Service (DoS)
  - Also limit incoming/outgoing ICMP packets
- ICMP `port unreachable` sent upon receiving UDP packet to a closed port, and other ICMP errors for filtered
  - ‘Good’ hosts will not send more pkts to closed/filtered port
- Many systems rate-limit such ICMP messages
  - Typically to one per second; higher rate → no response
- What to do?
  - Slow-down UDP scan – delay between packets

UDP scan: rate-limiting challenge

- Many systems rate-limit ICMP error messages
  - Typically to one per second; higher rate → no response
- What to do?
  - Slow-down UDP scan – delay between packets
  - Challenge: stealthy, high-rate UDP scan
    - Also allows stealthy scan for amplifiers
    - Hint: use DNS scans… (next)

ICMP Rate limiting

- ICMP-Response rate limits (global or per IP)
  - Mainly for ICMP error messages (ICMP NACKs)
    - Type 3: destination unreachable
      - Destination: net, host, protocol, port
      - Also: fragmentation required and don't-fragment set
    - Type 11: time (or TTL) exceeded
- Typical limit: 1 ICMP/second (globally)
  - Send in rate<0.39/second ➔ almost no rate limiting
  - Don't exceed for good scan
  - Or… abuse as a side-channel

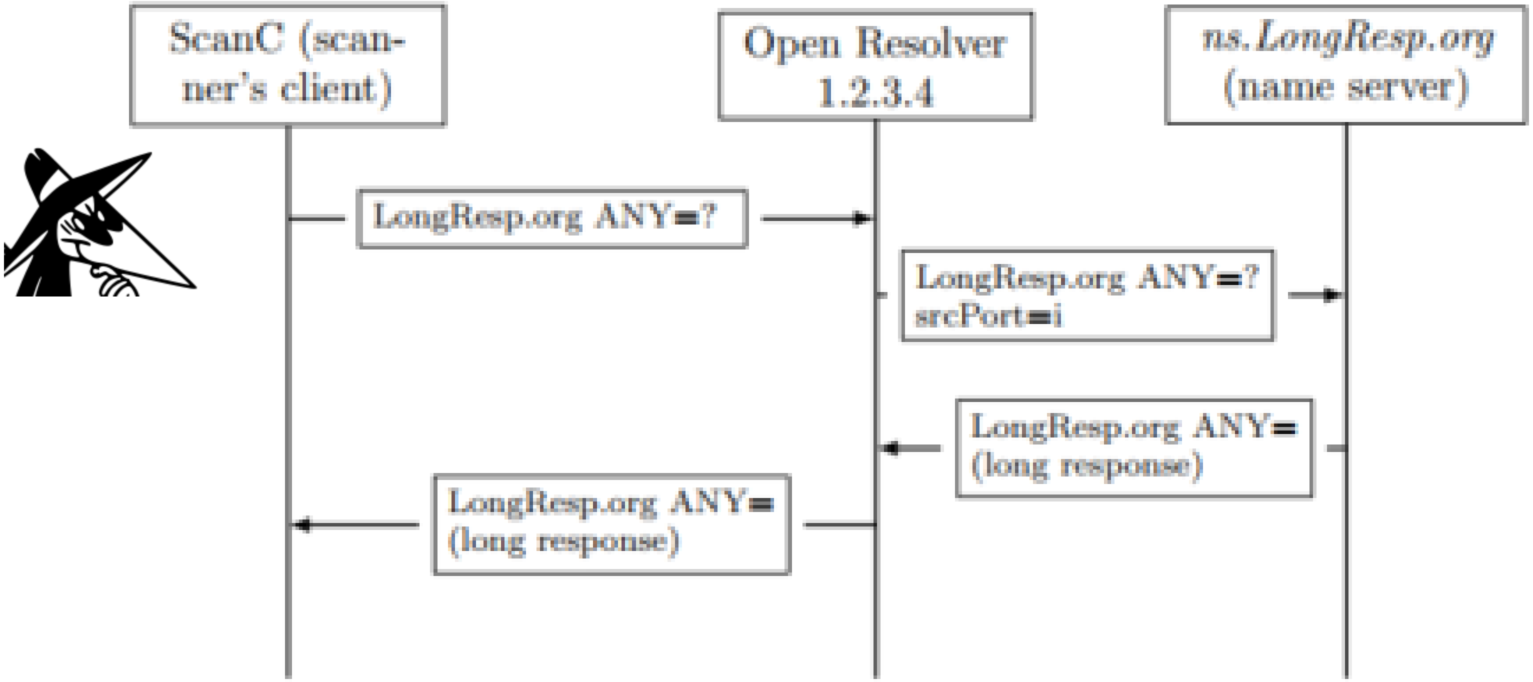
Reconnaissance: Agenda

- Introduction
- TCP scans
- UDP scans
- **DNS scans**
  - Goals: find vulnerable and/or ‘useful’ DNS servers
  - Find open DNS resolvers
  - Find (global/per-IP) incrementing IP-ID DNS servers
  - Find (global/per-IP) incrementing Src-port resolvers

Scans for Open DNS Resolvers

- Goal 1: use for BW-DoS attacks (amplification)
  - Send short request using (spoofed) src-IP of victim
  - DNS resolver sends long response to victim
  - ‘Amplification’: bw-to-victim/bw-by-attacker
  - Or: send many requests to victim name servers
- Goal 2: DNS poisoning
  - Easy if also using open/fixed source port (detect!)
- Goal 3: use for stealthy UDP-scans
  - Use open DNS resolvers with glob-inc-IP-ID
  - Hint: <u>use</u> ‘don’t send to unreachable port’

Scan for Open DNS Resolvers



ScanC (scanner's client)

Open Resolver 1.2.3.4

ns.LongResp.org (name server)

LongResp.org ANY=?

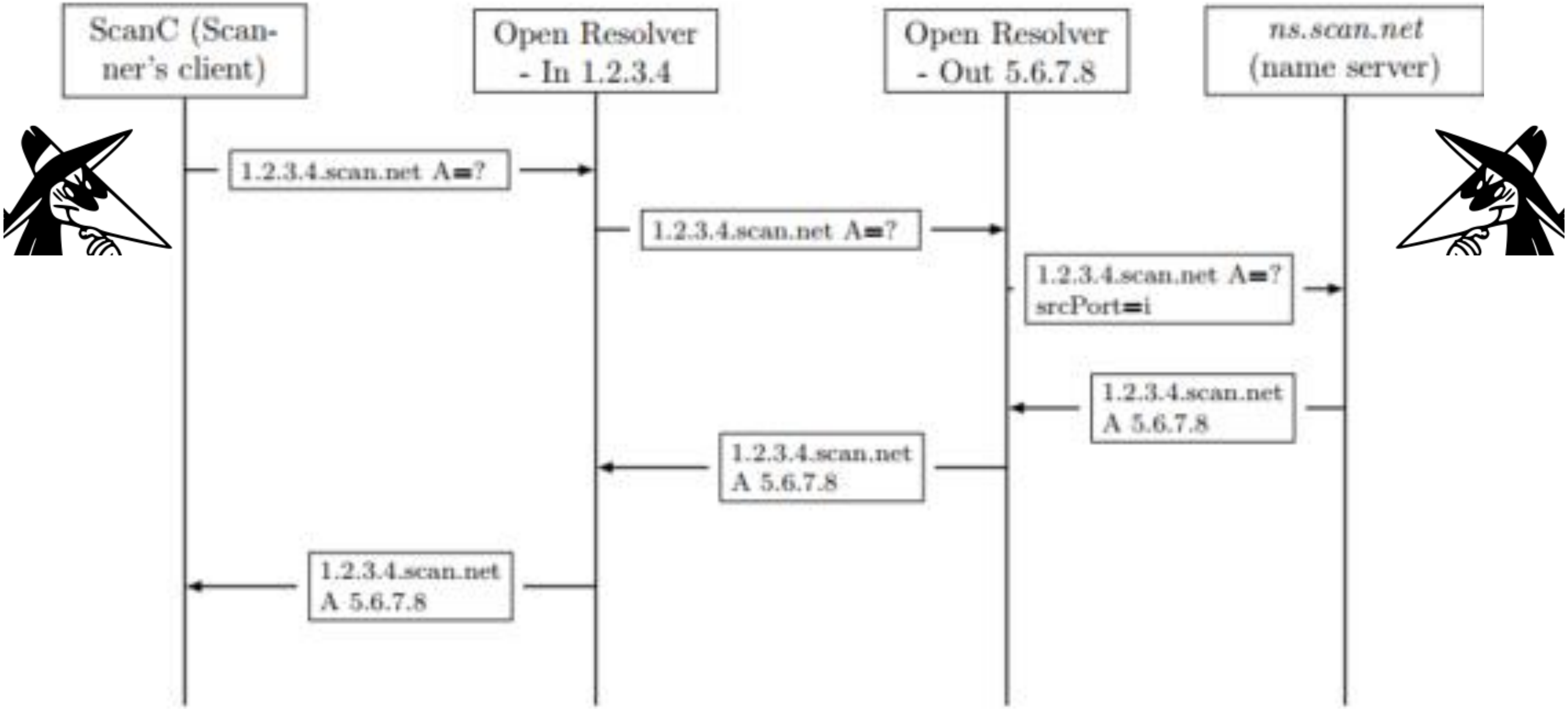
LongResp.org ANY=? srcPort=i

LongResp.org ANY= (long response)

LongResp.org ANY= (long response)

- Open resolvers that return long responses are abused for clogging DoS
- ANY DNS query returns all DNS records of the specified domain name

Scan for (in, out) open resolvers



ScanC (Scanner's client)

Open Resolver - In 1.2.3.4

Open Resolver - Out 5.6.7.8

ns.scan.net (name server)

1.2.3.4.scan.net A=?

1.2.3.4.scan.net A=?

1.2.3.4.scan.net A=? srcPort=i

1.2.3.4.scan.net A 5.6.7.8

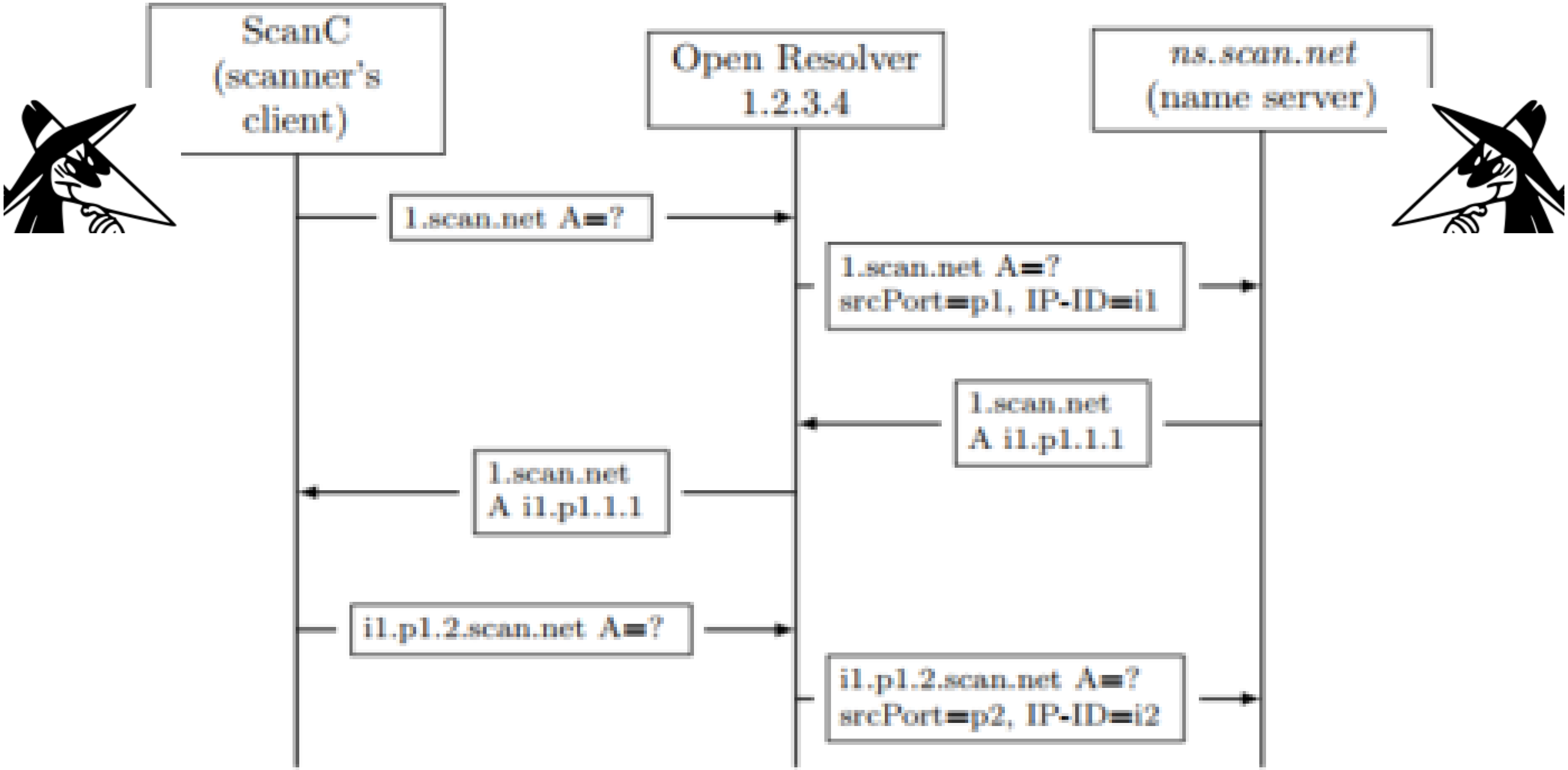
1.2.3.4.scan.net A 5.6.7.8

1.2.3.4.scan.net A 5.6.7.8

Finds mapping btw open-resolver's In IP-address and Out IP-address
Note: Scan-client can be off-path (stealthy scan, except use of *ns.scan.net*) !

Scan for Inc-src-port/IP-ID Resolvers

Detect if resolver uses (fixed or) incrementing IP-ID and/or source ports



ScanC (scanner's client)

Open Resolver 1.2.3.4

ns.scan.net (name server)

1.scan.net A=?

1.scan.net A=?
srcPort=p1, IP-ID=i1

1.scan.net
A i1.p1.1.1

1.scan.net
A i1.p1.1.1

i1.p1.2.scan.net A=?

i1.p1.2.scan.net A=?
srcPort=p2, IP-ID=i2

Detection rule? Distinguish btw `per-dest' and global incrementing?

Other scans…

- Scanning from a client visiting rogue website
  - Using Javascript, HTML5: error and/or timing side channel
  - Stealthy – and with access to internal network!
- **Fingerprinting:** identifying device, appl, version
  - Explicit (‘banner’ or in errors, e.g., SQL)
  - Behavior: TTL, options, MSS, retransmit pattern…
  - **Defense:** corrupt ‘fingerprint’ (by FW/IPS; ‘fuzzing’)
- **Scanning for other amplifiers… and more!!**