Things missed in the paper

- Application layer security
    - network is less important than the ways applications can protect themselves (end to end encryption, TLS)
- Cloud

Takeaways from the paper

- Difference in perspective
    - Cryptographer
        - When A wants to communicate with B, assume the whole network (besides A and B) is a black box, and is malicious
    - Network security
        - When A wants to communicate with B, we care about the routers along the path between A and B
            - No need to worry about routers off path
        - TCP model: How much security can we get assuming the on path routers are not malicious?
            - vulnerable to on path attackers
        - Nowadays, most off path attacks addressed

BGP: route discovery protocol

- Designed to help find out how packet can get from A to B; decides which routers are on path
    - If attacker can control routing, they can route packets through their own malicious host
- How it works
    - Each node broadcasts which nodes they can reach, along with the distance
        - This information gets propagated through the network
    - Each nodes updates their own routing information based on what their neighbors tell them
- Case study 1 - DOS: How to take down Facebook?
    - DOS: Advertise a route to Facebook that is the shortest route (the advertised route includes a router under your control), drop all packets from the router you control
    - This attack has happened - Pakistan's government blocked YouTube
        - ISPs made BGP announcement that was meant to propagate within Pakistan, but accidently propagated worldwide
    - Possible defenses
        - Specify path within network packet, check to see if path is reasonable
            - Tricky to determine what a reasonable path is
        - Timing analysis, or other methods to detect implausible paths

- Case study 2 - Man in the Middle: eavesdrop, assuming lack of encryption
    - Same strategy as attack 1 does not work
        - After the attacker advertises a short route, routers will redirect packets to malicious router...but then what? Because the attacker advertised a short route, sending a packet to other routers will result in the packet coming back
        - Without being able to send the packet to its intended destination, no way to maintain the connection, and thus eavesdropping can't happen
    - Strategy that *does* work: advertise different hop counts to different links
        - Announce short path to target node (who the attacker wants to eavesdrop on), announce long path to other nodes (so those other nodes will propagate the packet and maintain the connection)
        - Could be unreliable, because the short path advertised to the target node could propagate and pollute the correct paths
            - The fix to guarantee one unpolluted router: malicious router picks one route to the destination. To the neighboring node N on that route, the malicious router announces a short path, but includes N in the AS path. N will not send the packet back to the malicious router to avoid a routing loop.
    - Defenses
        - VPN or end to end cryptography so that routers don't need to be trusted
        - Destination router signs the announcements that its neighboring routers can reach it in one hop, and routers recursively sign announcements for their neighbors
            - Prevents a router from advertising a fake short path
            - Recursive signature are required; if ach router signs its own announcements, it can still lie about the hop count
- Case study 3: Great firewall of China
    - Block internal Chinese host's access to certain sites
    - If B wants to access W, which is blocked, a simple on path attack is carried out
        - Router injects a forged RST packet to the browser that looks like it came from W
- Case study 4: Great cannon of China
    - DOS attack (uses infrastructure of great firewall) against Github
    - Took down Github from everyone in the world so Github removed the project
    - On path attack: client C is connected to Baidu
        - With probability .02, great firewall forged packets back to C with malicious Javascript embedded into Baidu page that initiated a connection to Github
        - Millions of people who visited Baidu were attacked and were unknowingly part of this DOS attack
    - Attack doesn't work with an encrypted HTTPS connection

IP ID Scanning

- Different IP address for each connection attempt when doing port scanning
- Motivation: port scanning is used see which services are running on a server
    - Easy to detect and block an attacker if this is done from one IP address, and it is not cheap to have many different IP addresses
- IP packet has a semi-unique 60 bit ID number
    - Depending on the OS, this ID is can be set randomly or is incremented
- A: attacker, S: server attacker wants to port scan, P: Patsy
    - Find an idle (not doing anything, not receiving much traffic) P where the IP ID is being incremented
    - At any point, A can send a packet to P to view the ID number
- Attack
    - A sends a forged packet to S, saying the source is P
    - If port is open, S tries to set up a connection with P. If port is closed, S sends a message to P that it cannot set up a connection.
    - From P's point of view, it gets a random packet.
        - According to TCP protocol, if P receives a packet to set up a connection that it did not initiate, it sends a reset packet. If P receives a packet that a connection failed, but P did not initiate the connection, then P does nothing.
- Attacker is not on path, but can infer whether port is open
    - If sequence number increases, the port was open
- Defenses by changing P
    - Change TCP protocol so P responds to both messages
    - Modify P so it doesn't use incrementing packet numbers, use random ones to close this side channel
    - Shortcoming: changing P doesn't work, and is difficult
        - If we tell Microsoft to implement this in Windows, this does not solve the problem
            - We should address S rather than P, because the attacker only needs to find one Patsy - there will be others out there!
        - There is no incentive for Microsoft to implement this
- Defense by changing S
    - ISP checks outgoing packets to make sure source is in the network
    - Same shortcoming: no benefit for ISP to prevent spoofing

Discussion: Protection Architecture for Enterprise Network

- Scenario: you want to build email service, so that only authenticated employees can access it
    - Firewalls - but hard to directly configure
        - Guests can access other services
        - Different people with different permissions

        - Common idea for wifi - have separate network for the guests
            - Separate networks are not suited for separating people on the network by which services they are allowed to access
        - Centralized service for access control
            - How to revoke access? Check on every request?
                - Something like TTL for when to check permissions
            - How to enforce permissions?
                - Email server can check permissions, but then every service would have to do the permissions checking
                - The network can check permissions
- How to prevent a single point of failure, if there's a centralized service?
    - Duplication, redundancy - classic solution