CSE543 Computer and Network Security

Module: Network Security

Professor Trent Jaeger
Fall 2010
Networking

• Fundamentally about transmitting information between two devices

• Direct communication is now possible between any two devices anywhere (just about)
  ‣ Lots of abstraction involved
  ‣ Lots of network components
  ‣ Standard protocols
  ‣ Wired and wireless
  ‣ Works in protection environment

• What about ensuring security?
Network Security

• Every machine is connected
  ‣ What is trust model of the network?

• Not just limited to dogs as users
  ‣ What other ‘dogs’ are out there?
Exploiting the network ...

• The Internet is extremely vulnerable to attack
  ‣ it is a huge open system ...
  ‣ which adheres to the *end-to-end* principle
    • smart end-points, dumb network

• Can you think of any *large-scale attacks* that would be
The End-to-End Argument

- Clark et. al discussed a property of good systems that says features should be placed as close to resources as possible
  - In communication, this means that we want the middle of the network to be simple, and the end-points to be smart (e.g., do everything you can at the end-points
  - “Dumb, minimal network”
  - This is the guiding principle of IP (Internet)
  - Q: Does this have an effect on security?

- Note: this is a departure from the early networks which smart network, dumb terminals
Network security: the high bits

• The network is …
  ‣ … a collection of interconnected computers
  ‣ … with resources that must be protected
  ‣ … from unwanted inspection or modification
  ‣ … while maintaining adequate quality of service.

• Another way of seeing network security is …
  ‣ … securing the network infrastructure such that the integrity, confidentiality, and availability of the resources is maintained.
The network ...
The big picture ....

• Internet Protocol (IP)
  ‣ Really refers to a whole collection of protocols making up the vast majority of the Internet

• Routing
  ‣ How these packets move from place to place

• Network management
  ‣ Administrators have to maintain the services and infrastructure supporting everyone’s daily activities
• Bellovin’s observations about security problems in IP
  ‣ Not really a study of how IP is misused, e.g., IP addresses for authentication, but really what is inherently bad about the way in which IP is setup

• A really, really nice overview of the basic ways in which security and the IP design is at odds
TCP/IP uses a *three-way handshake* to establish a connection

1. \( C \rightarrow S: Q_C \)
2. \( S \rightarrow C: Q_S, \text{ack}(Q_C) \) where sequence number \( Q_S \) is nonce
3. \( C \rightarrow S: \text{ack}(Q_S) \) … then send data

2. However assume the bad guy does not hear msg 2, if he can guess \( Q_S \), then he can get \( S \) to accept whatever data it wants (useful if doing IP authentication, e.g., “rsh”)

![Diagram showing client, server, and adversary connections](image)
Sequence Number Prediction (fixes)

• The only way you really fix this problem to stop making the sequence numbers predictable:
  ‣ Randomize them -- you can use DES or some other mechanism to generate them randomly
  ‣ There is an entire sub-field devoted to the creation and management of randomness in OSes

• Also, you could look for inconsistencies in timing information
  ‣ Assumption: the adversary has different timing than
  ‣ OK, maybe helpful, but far from definitive
Routing Manipulation

• RIP - routing information protocol
  ‣ Distance vector routing protocol used for local network
  ‣ Routers exchange reachability and “distance” vectors for all the sub-networks within (a typically small) domain
  ‣ Use vectors to decide which is best, notification of changes is propagated quickly

• So, the big problem is that you receive vast amounts of data that a router uses to form the routing table
  ‣ So, just forge that, and the game is up
  ‣ Manipulate paths, DOS, hijack connections, etc.

• Solutions:
  ‣ Authenticate data, but this is less than obvious how to do this efficiently (a whole lot of people are trying)
Internet Control Message Protocol (ICMP)

- ICMP is used as a control plane for IP messages
  - Ping (connectivity probe)
  - Destination Unreachable (error notification)
  - Time-to-live exceeded (error notification)

- These are largely indispensable tools for network management and control
  - Error notification codes can be used to reset connections without any

- Solution: verify/sanity check sources and content
  - ICMP “returned packets”

- Real solution: filter most of ICMP, ignore it
The “ping of death” …

• In 1996, someone discovered that many operating systems, routers, etc. could be crash/rebooted by sending a single malformed packet
  ‣ It turns out that you can send a IP packet larger than 65,535 ($2^{16}$), it would crash the system
  ‣ The real reason lies in the way fragmentation works
    • It allows somebody to send a packet bigger than IP allows
    • Which blows up most fixed buffer size implementations
    • … and dumps core, blue screen of death, etc.
  ‣ Note: this is not really ICMP specific, but easy (try it)
    % ping -l 65510 your.host.ip.address
• This was a popular pastime of early hackers
Address Resolution Protocol (ARP)

- Protocol used to map IP address onto the physical layer addresses (MAC)
  1) ARP request: who has x.x.x.x?
  2) ARP response: me!
- Policy: last one in wins
- Used to forward packets on the appropriate interfaces by network devices (e.g., bridges)

- Q: Why would you want to spoof an IP address?
ARP poisoning

- Attack: replace good entries with your own
- Leads to
  - Session hijacking
  - Man-in-the-middle attacks
  - Denial of service, etc.

- Lots of other ways to abuse ARP.
- Nobody has really come up with a good solution
  - Except smart bridges, routers that keep track of MACs
- However, some not worried
  - If adversary is in your perimeter, you are in big trouble
  - You should never should validate the source of each pack
Legacy flawed protocols/services

• Finger user identity (my advisor hated this)
  ‣ host gives up who is logged in, existence of identities

```
PSU.local Presentations > finger megan
Login: megan                  Name: Megan Smith
Directory: /Users/megan       Shell: /bin/bash
Last login Mon 23 Aug 13:19 (EDT) on console
No Mail.
No Plan.
PSU.local Presentations >
```

• This is horrible in a distributed environment
  ‣ Privacy, privacy, privacy …
  ‣ Lots of information to start a compromise of the user.
POP/SMTPT/FTP

- Post office protocol - mail retrieval
  - Passwords passed in the clear (duh)
  - Solution: SSL, SSH, Kerberos

- Simple mail transport protocol (SMTP) - email
  - Nothing authenticated: SPAM
  - Nothing hidden: eavesdropping
  - Solution: your guess is as good as mine

- File Transfer protocol - file retrieval
  - Passwords passed in the clear (duh)
  - Solution: SSL, SSH, Kerberos
DNS - The domain name system

- DNS maps between IP address (12.1.1.3) and domain and host names (ada.cse.psu.edu)
  - How it works: the “root” servers redirect you to the top level domains (TLD) DNS servers, which redirect you to the appropriate sub-domain, and recursively ....
  - Note: there are 13 “root” servers that contain the TLDs for .org, .edu, and country specific registries (.fr, .ch)
A DNS query

1. User PC
   - www.patrickmcdaniel.org?

2. ISP Nameserver
   - www.patrickmcdaniel.org?

3. a-root-servers.net
   - redirect

4. a.gtld-servers.org
   - redirect

5. ns-patrickmcdaniel.org
   - redirect

6. 207.140.168.131
   - www.patrickmcdaniel.org?

7. DNS Cache
   - www.patrickmcdaniel.org = 207.140.168.131

8. User PC
   - 207.140.168.131

www.patrickmcdaniel.org
“Glue” information

• Suppose you ask a name server for a record and it redirects you to another name server (NS record)
  ‣ e.g., if you ask a root for a **NS** (name server) record for NET, it returns NS records for the authoritative servers for .net
• It will also give you the **A** (resource) record for the authoritative servers you were directed to
  ‣ avoid looking them up
  ‣ This is known as the “glue” records

```
.;NET referrals
/* Authority section */
NET. IN NS A.GTLD-Servers.NET.
IN NS B.GTLD-Servers.NET.
IN NS C.GTLD-Servers.NET.
  ...
IN NS M.GTLD-Servers.NET.

/* Additional section - "glue" records */
A.GTLD-Servers.net. IN A 192.5.6.30
B.GTLD-Servers.net. IN A 192.33.14.30
C.GTLD-Servers.net. IN A 192.26.92.30
  ...
M.GTLD-Servers.net. IN A 192.55.83.30
```
DNS Vulnerabilities

• Nothing is authenticated, so really the game is over
  ‣ You can not really trust what you hear …
  ‣ But, many applications are doing just that.
  ‣ Spoofing of DNS is really dangerous

• Moreover, DNS is a catalog of resources
  ‣ Zone-transfers allow bulk acquisition of DNS data
  ‣ … and hence provide a map for attacking the network

• Lots of opportunity to abuse the system
  ‣ Relies heavily on caching for efficiency -- cache pollution
  ‣ Once something is wrong, it can remain that way in caches for a long time (e.g., it takes a long time flush)
  ‣ Data may be corrupted before it gets to authoritative server
A Cache Poisoning Attack

- All requests have a unique query ID
- The nameserver/resolver uses this information to match up requests and responses
- If an adversary can guess the query ID, then it can forge the responses and pollute the DNS cache
  - 16-bit query IDs (not hard)
  - Some servers increment IDs (or use other bad algo.)
  - First one in wins!!!
- Note: If you can observe the traffic going to a name server, you can pretty much arbitrarily own the Internet for the clients it serves.
Kaminsky DNS Vulnerability

1. Query a random host in a victim zone, e.g., 1234.cse.psu.edu

2. Spoof responses* as before, but delegate authority to some server which you own.

   1. The glue records you give make you authoritative

3. You now own the domain.

*the original attack exploited poor ID selection
Kaminski Fixes

• Make the ID harder to guess (randomized ports)
  ‣ Amplified ID space from $2^{16}$ to $2^{27}$

• Prevent foreign requests from being processed
  ‣ E.g., filter requests from outside domain

• Observe and filter conflicting requests
  ‣ E.g., if you see a lot of bogus looking requests, be careful

• All of this treats the symptoms, not the disease.
  ‣ Lack of authenticated values
  ‣ Thus, if you can observe request traffic, prevent legitimate responses, or are just plain patient, you can mount these attacks.
DNSsec

• A standard-based (IETF) solution to security in DNS
  ‣ Prevents data spoofing and corruption
  ‣ Public key based solution to verifying DNS data
  ‣ Authenticates
    • Communication between servers
    • DNS data
      ‣ content
      ‣ existence
      ‣ non-existence
    • Public keys (a bootstrap for PKI?)
DNSsec Mechanisms

• **TSIG**: transaction signatures protect DNS operations
  ‣ Zone loads, some server to server requests (master -> slave), etc.
  ‣ Time-stamped signed responses for dynamic requests
  ‣ A misnomer -- it currently uses shared secrets for TSIG (HMAC) or do real signatures using public key cryptography

• **SIG0**: a public key equivalent of TSIG
  ‣ Works similarly, but with public keys
  ‣ Not as popular as TSIG, being evaluated

• Note: these mechanisms assume clock sync. (NTP)
DNSsec Mechanisms

• Securing the DNS records
  ‣ Each domain signs their “zone” with a private key
  ‣ Public keys published via DNS
  ‣ *Indirectly* signed by parent zones
  ‣ Ideally, you only need a self-signed root, and follow keys down the hierarchy

![Diagram showing DNSsec mechanism]

- root
- .edu
- psu.edu
- cse.psu.edu
DNSsec challenges

• Incremental deployability
  ‣ Everyone has DNS, can’t assume a flag day

• Resource imbalances
  ‣ Some devices can’t afford real authentication

• Cultural
  ‣ Most people don’t have any strong reason to have secure DNS ($$$ not justified in most environments)
  ‣ Lots of transitive trust assumptions (you have no idea how the middlemen do business)

• Take away: DNSsec will be deployed, but it is unclear whether it will be used appropriately/widely
Transport Security

• A host wants to establish a secure channel to remote hosts over an untrusted network
  ‣ Not Login – end-users may not even be aware that protections in place (transparent)
  ‣ Remote hosts may be internal or external

• The protection service must …
  ‣ Authenticate the end-points (each other)
  ‣ Negotiate what security is necessary (and how achieved)
  ‣ Establish a secure channel (e.g., key distribution/agreement)
  ‣ Process the traffic between the end points

• Also known as *communications security*. 
IPsec (not IPSec!)

- Host level protection service
  - IP-layer security (below TCP/UDP)
  - De-facto standard for host level security
  - Developed by the IETF (over many years)
  - Available in most operating systems/devices
    - E.g., XP, Vista, OS X, Linux, BSD*, …
  - Implements a wide range of protocols and cryptographic algorithms
- Selectively provides ….
  - Confidentiality, integrity, authenticity, replay protection, DOS protection
IPsec and the IP protocol stack

- IPsec puts the two main protocols in between IP and the other protocols
  - AH - authentication header
  - ESP - encapsulating security payload
- Other functions provided by external protocols and architectures

<table>
<thead>
<tr>
<th>HTTP</th>
<th>FTP</th>
<th>SMTP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
TCP    | UDP |
```

```
AH     | ESP |
```

```
IP     |
```
Modes of operation

- **Transport**: the payload is encrypted and the non-mutable fields are integrity verified (via MAC).

- **Tunnel**: each packet is completely encapsulated (encrypted) in an outer IP packet.
  - Hides not only data, but some routing information.
Tunneling

• “IP over IP”
  ‣ Network-level packets are encapsulated
  ‣ Allows traffic to avoid firewalls
IPsec Protocol Suite

Policy/Configuration Management

(SPSC) Security Policy System

IKE  
Internet Key Exchange

Key Management

Manual

Packet Processing

ESP  
Encapsulating Security Payload

AH  
Authentication Header
Internet Key Exchange (IKE)

• Built on of ISAKMP framework
• Two phase protocol used to establish parameters and keys for session
  ‣ Phase 1: authenticate peers, establish secure channel
  ‣ Phase 2: negotiate parameters, establish a security association (SA)
• The details are unimaginably complex
• The SA defines algorithms, keys, and policy used to secure the session
IPsec: Packet Handling (Bump ...)

IP Protocol Stack

SADB  IPsec

Application
Presentation
Session
Transport
Network (IP)
Data Link
Physical
Authentication Header (AH)

- Authenticity and integrity
  - via HMAC
  - over IP headers and data
- Advantage: the authenticity of data and IP header information is protected
  - it gets a little complicated with _mutable_ fields, which are supposed to be altered by network as packet traverses the network
  - some fields are _immutable_, and are protected
- Confidentiality of data is _not_ preserved
- Replay protection via AH sequence numbers
  - note that this replicates some features of TCP (good?)
Authentication Header (AH)

• Modifications to the packet format
### IPv4 AH Packet Format

<table>
<thead>
<tr>
<th>IPv4 Header</th>
<th>Authentication Header</th>
<th>Higher Level Protocol Data</th>
</tr>
</thead>
</table>

### AH Header Format

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Length</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Security Parameter Index
- Authentication Data (variable number of 32-bit words)
IPsec Authentication

- **SPI**: (spy) identifies the security association for this packet
  - Type of crypto checksum, how large it is, and how it is computed
  - Really the policy for the packet

- **Authentication data**
  - Hash of packet contents include IP header as specified by SPI
  - Treat transient fields (TTL, header checksum) as zero

- **Keyed MD5 Hash is default**
Encapsulating Security Payload (ESP)

- Confidentiality, authenticity and integrity
  - via encryption and HMAC
  - over IP payload (data)

- Advantage: the security manipulations are done solely on user data
  - TCP packet is fully secured
  - simplifies processing

- Use “null” encryption to get authenticity/integrity only

- Note that the TCP ports are hidden when encrypted
  - good: better security, less is known about traffic
  - bad: impossible for FW to filter/traffic based on port
Encapsulating Security Payload (ESP)

- Modifications to packet format
**IPv4 ESP Packet Format**

- **IP Header**
- **Other IP Headers**
- **ESP Header**
- **Encrypted Data**

**ESP Header Format**

- Security Parameter Identifier (SPI)
- Opaque Transform Data, variable length

**DES + MD5 ESP Format**

- Security Parameters Index (SPI)
- Initialization Vector (optional)
- Replay Prevention Field (incrementing count)
- Payload Data (with padding)
- Authentication checksum
Practical Issues and Limitations

- IPsec implementations
  - Large footprint
    - resource poor devices are in trouble
    - New standards to simplify (e.g., JFK, IKE2)
  - Slow to adopt new technologies
  - Configuration is really complicated/obscure

- Issues
  - IPsec tries to be “everything for everybody at all times”
    - Massive, complicated, and unwieldy
  - Policy infrastructure has not emerged
  - Large-scale management tools are limited (e.g., CISCO)
  - Often not used securely (common pre-shared keys)
Network Isolation: VPNs

• Idea: I want to create a collection of hosts that operate in a coordinated way
  ‣ E.g., a virtual security perimeter over physical network
  ‣ Hosts work as if they are isolated from malicious hosts

• Solution: Virtual Private Networks
  ‣ Create virtual network topology over physical network
  ‣ Use communications security protocol suites to secure virtual links “tunneling”
  ‣ Manage networks as if they are physically separate
  ‣ Hosts can route traffic to regular networks (split-tunneling)
VPN Example: RW/Telecommuter

Internet

LAN

(network edge)

Physical Link

Logical Link (IPsec)
VPN Example: Hub and Spoke

Internet

LAN

Physical Link

Logical Link (IPsec)

(network edge)
VPN Example: Mesh

- Physical Link
- Logical Link (IPsec)
Virtual LANs (VLANs)

- VPNs built with hardware
  - Physically wire VPN via soft configuration of a switch crossbar
  - No encryption – none needed
  - “wire based isolation”
  - Many switches support VLANs
  - Allows networks to be reorganized without rewiring

- Example usage: two departments in same hallway
  - Each office is associated with department
  - Configuring the network switch gives physical isolation
  - Note: often used to ensure QoS