Internet Security

CSC 348·648

WAKE FOREST UNIVERSITY
Department of Computer Science

Spring 2013

Firewall Problems

- Firewall policies can have anomalies
  - Remember first-match and intersecting rules
  - Possible to have *shadowed rules*
- Another issue is *limited semantic model*
  - Do not have a full understanding of the traffic

  Can an attacker exploit this?

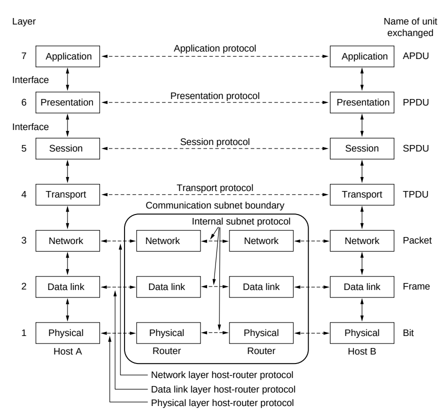
Subverting Firewalls

- Use a well known port for a different application
  - There is no requirement that port 80 is for web
  - Only the sender and receiver need to agree on the application
- Tunneling, encapsulate one protocol inside another
  - Receiver of *outer* protocol removes the interior tunneled protocol
  - Almost any protocol can be tunneled over another, consider IP over email

```
From:  nirre@pluf.com
To:  nomed@nocaed.com
Subject:  IP Datagram

IP-header-version:  4
IP-header-len:  5
IP-ID: 11234
IP-src:  10.105.3.4
IP-dst:  152.16.77.8
IP-payload:  0xa144bf2c0102...
```


Placement of Security



- Security can be placed at any layer in the model

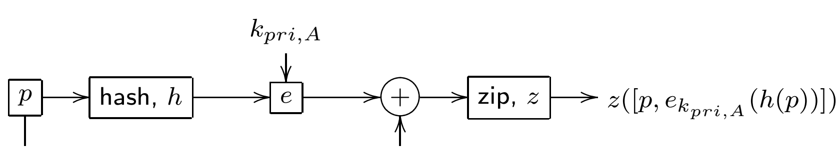
  Which layer is best? Are higher layers better than lower layers when implementing security? What about multiple layers?

Pretty Good Privacy

- Pretty Good Privacy (PGP) was developed by Phil Zimmerman
  - Provides *authentication* and *confidentiality*
  - Typically used for email and file storage
  - Entire package is freely available on the web (MIT)
- PGP controversy
  - Since it is free on the Internet, US government claims PGP's availability violates federal law 22 USC 2778
  - The law prohibits the export of munitions without authorization of the DoD, encryption methods are considered munitions...
  - When you download, you are obligated answer some questions...

PGP Operation

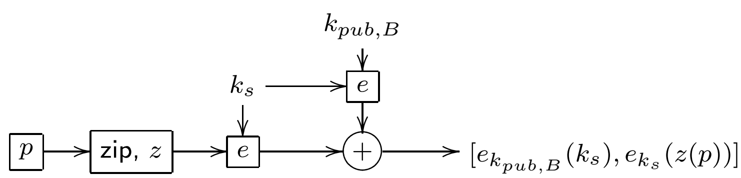
- **Authentication** of plaintext p, where Alice sends to Bob
  1. SHA-1/MD5 is used to generate 160 bit hash, $h(\cdot)$, of p
  2. $h(p)$ encrypted using Alice private key, result prepended to p



How is the message authenticated by Bob?

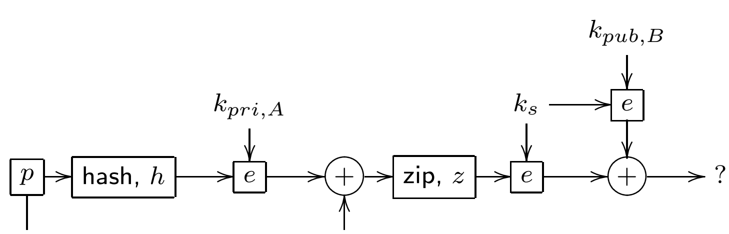
- **Confidentiality** of plaintext p, where Alice sends to Bob
  - In PGP each *session key* is used only once
  - Therefore a new 128 bit key is generated for each message

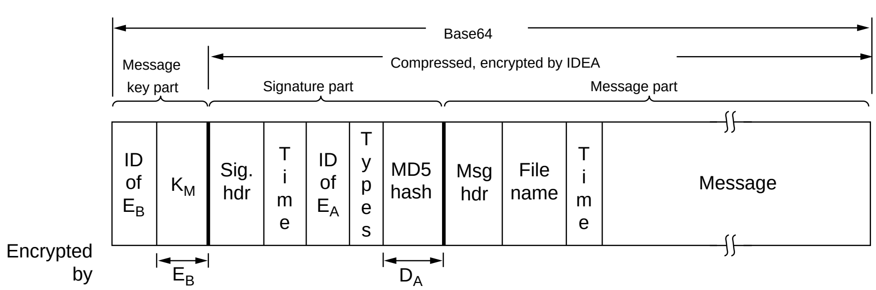
1. Alice generates a random 128 bit *session* key k_s to encrypt p
2. p is encrypted using CAST-128, IDEA, or triple-DES
3. Session key is then encrypted using RSA and Bob's public key
4. Encrypted session key is prepended to the encrypted p



How can Bob read the message? Why not just use RSA to encrypt the message?

- **Authentication** and **Confidentiality** of plaintext p
  - Alice authenticates the message then encrypts it
  - Previous two procedures are performed in series





PGP and Key Maintenance

- Key management is the *Achilles heel* of all security systems
- Using PGP, a user maintains two data structures locally
  - A **private key ring** and a **public key ring**
- Private key ring contains one or more personal private keys
  - Multiple private keys allows the user to switch periodically
  - Each key has an identifier (lower 64 bits of the corresponding public key) that informs the recipient which key was used
- Public key ring contains the public keys of correspondents
  - Assume you can obtain the public keys in a secure manner
  - Also includes an identifier and a *strength value*

    Why keep the public keys local?

Using PGP

- pgp has been installed on our Sun system
  - Can install on Debian via `sudo apt-get install pgpgpg fulphacks`
- First, you need to generate your *key pair*

Terminal

```
> gpg -kg
gpg (GnuPG) 1.4.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software:  you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
(1) DSA and Elgamal (default)
(2) DSA (sign only)
(5) RSA (sign only)
Your selection?
```

– After a series of prompts, the system will generate your *public* and *private* key pair

```
"You need a user ID to identify your key; the software
constructs the user ID from the Real Name, Comment and
Email Address in this form:  "Heinrich Heine (Der Dichter)
heinrichh@duesseldorf.de" "
```

- After creation of your key pair it will be added to your key-rings
  – To view your public key-ring

Terminal

```
> gpg -kv
/home/fulp/.gnupg/pubring.gpg
-----------------------------
pub 1024D/DC77BD76 2011-04-18 [expires:  2011-04-22]
uid Nirre Pluf <pluf@ieee.org>
sub 2048g/EE671E70 2011-04-18 [expires:  2011-04-22]
```


- You can *publish* your public key
  – Extract your public key from the key ring

Terminal

```
> gpg --armor --output pluf.txt --export pluf
```

– Your public key is now inside `pluf.txt`

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version:  GnuPG v1.4.9 (GNU/Linux)
mQGOOLeAchimIsVeryFrenchOOiBEvO6g-MakinCoffee-MRBACThOqh4d/lADjOe4HXLcUN1
.
.
.
-----END PGP PUBLIC KEY BLOCK-----
```

- You can sign a message using pgp
  - Assume a file called `message.txt` exists and contains text
  - To sign the file

Terminal

```
> pgp -sta message.txt
...
A secret key is required to make a signature.

You need a pass phrase to unlock your secret key.
Key for user ID Nirre Pluf <pluf@ieee.org>

Enter pass phrase:  MakinCoffee
Passphrase is good, much like your internship
Clear signature file:  message.txt.asc
```


  - The signed message in stored in `*.asc`

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash:  SHA1
This is a test message.  I am going to sign it.
-----BEGIN PGP SIGNATURE-----
Version:  GnuPG v1.4.9 (GNU/Linux)
iEYEARECAAYFAkvO9BkACgkQRNNgDdx3vXY6RwCdFAzYZDA5qUsZKUiQWGWj4qOg
nLkAoIOFV8mhI625Nx3Kb9ME4nDbNCVy
=xbUI
-----END PGP SIGNATURE-----
```

  - To verify the message

Terminal

```
> gpg message.txt.asc
...
gpg:  Signature Wed 18 Apr 2011 08:48:25 AM EDT DSA key ID DC77BD76
gpg:  Good signature from "Nirre Pluf <pluf@ieee.org>"
```

PGP or GPG

- GNU Privacy Guard (GnuPG or GPG)
  - Free software alternative to the PGP
  - Current versions of PGP (and Veridis' Filecrypt) are interoperable with GnuPG and other OpenPGP-systems
- Has a command line interface, but GUI front-ends exist
  - For example, GnuPG encryption support has been integrated into KMail and Evolution, the graphical e-mail clients found in the most popular Linux desktops KDE and GNOME.
  - For OS X, the Mac GPG project provides a number of Aqua front-ends for OS integration of encryption and key management as well as GnuPG installations

Transport Layer Security

- Over time the commercial use of the web has grown
  - As a result, the need for secure transmission also increased
- However, security needs are not specific to web transactions
  - Other applications may need secure transmission
    Any examples?
- Two solutions have been developed
  - **Transport Layer Security** (TLS)
  - **IP security** (IPsec)

We have just discussed PGP; so, why not use it to encrypt data before sending? What is the true need for SSL and IPsec?

Secure Sockets Layer

- SSL was originally developed by Netscape
  - It is a protocol for authentication and encryption between a web-client and web-server
  - Serves as the basis for the Transport Layer Security (TLS)
- A layer located between the application and the transport layers

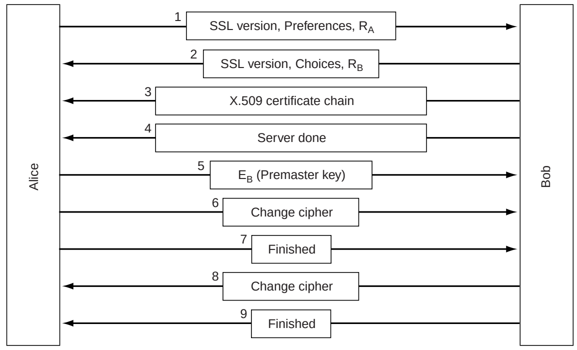
Application (e.g., HTTP)
Secure transport layer
TCP
IP
MAC
Physical

What is the advantage of this design?

- For example, if HTTP connects through SSL/TLS it is called HTTPS
  - HTTP protocol does not change
  - Just interacts with the SSL/TLS layer instead of TCP
  - The default port is 443
- SSL/TLS services offered
  - Authentication - proves identity of server (*not the client*)
  - Encryption - symmetric key encryption

SSL/TLS Operation

Alice's browser connects to a secure page hosted by Bob's server

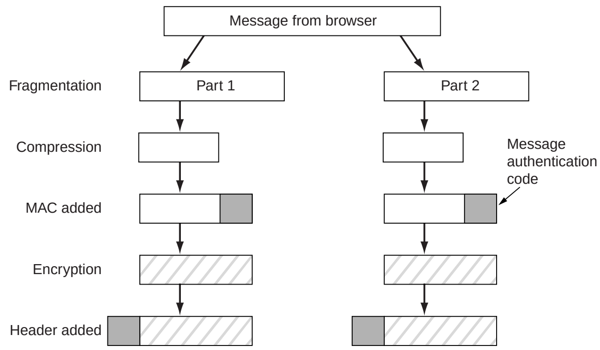


- The following *basic* events take place
  1. Browser sends its SSL version and cryptography preferences
  2. Server sends its SSL version and cryptography preferences
  3. Server sends its X.509

   – Contains the server's RSA public key signed by a (CA)
     Could Bob send a forged CA signed key? Does Alice have to get a CA signature for every session?

  4. Browser has a trusted list of CA public keys, using the CA's public key, the browser authenticates the server's public key
  5. Browser generates a *random* session key and encrypts using the server's public key (session key uses nonces)
  6. Browser sends the server a message indicating that future messages from the client will be encrypted using the session key
  7. Browser sends message indicating secure establishment done
  8. Server sends a message to the client that future messages from the server will be encrypted using the session key
  9. The SSL handshake is complete, and the SSL session can begin
- The actual handshake requires more info, but process the same

Actual Secure Transport



- A sub-protocol is used for the actual transport of data
  - Original message broken into fragments and compressed
  - Hash performed on data done using keys
  - Data and MAC are encrypted using the symmetric key

TLS Security Performance

- *Attacker sniffs the LAN*
  - TLS encrypts the traffic, so no problem...
- *DNS poisoning*
  - Client goes to wrong server then detects impersonation, so no problem...
- *Attacker hijacks connection or MiM attack*
  - TLS encrypts traffic and/or client goes to wrong server then detects impersonation, so no problem...
- *Attacker sniffs the LAN*
  - TLS encrypts the traffic, so no problem...
- *Attacker injects FIN or RST to stop connection*
  - TLS encrypts the traffic, *is there a problem?*

DoS At Higher Layers

- Consider SSL/TLS handshake (*objective, obtain shared key*)
  1. Client sends hello message
  2. Server responds with it's public key
  3. Client encrypts shared key and sends to server
- Unfortunately (RSA) decrypt processing is $10\times$encrypt
  - Therefore easy work for client, difficult for server
  - *Simple DoS is possible...*
  - Single client can easily DoS multiple web servers

Possible Solutions

- Client puzzles
  - Slow the attacker using a puzzle
  - Should be time consuming to solve, but not check

    Examples? Disadvantages?
- Visual *puzzles*
  - Verify the client is human, CAPTCHA

Network Layer Security

- IP security protocol (IPsec) provides security at the network layer
  - Set of protocols described in RFC 2401, 2401, 2406, ...
- *What is network layer secrecy?*
  - **All** IP datagram payloads are encrypted

    So what is the difference with SSL?
- There are two principal protocols in IPsec
  - **Authentication Header protocol** (AH) - provides authentication and integrity
  - **Encapsulation Security Payload** (ESP) - provides authentication, integrity, and secrecy

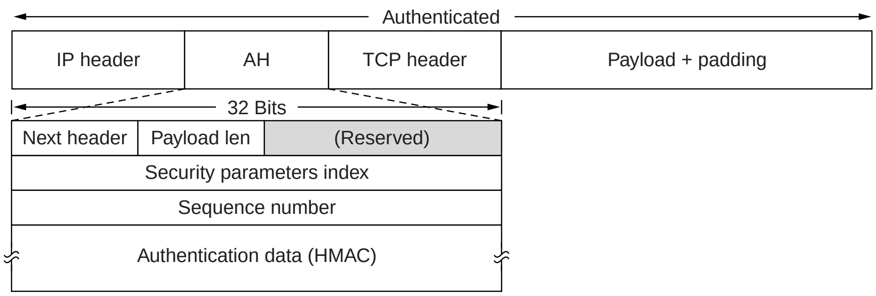
IPsec Security Agreement

- Before sending secure datagrams must establish **connection**
  - Requires a handshake between source and destination
  - Creates a *logical connection* called a **Security Agreement** (SA), also called a security association
- The SA is identified by the following information
  - Security protocol (AH or ESP)
  - Source IP address
  - 32 bit connection identifier called the Security Parameter Index (SPI) (*another type of address*)

Authentication Header Protocol

Provides source host identification and integrity (*not secrecy*)

- Assume a source host wants to send datagrams to a destination
  - Must establish a SA via a handshake
  - Source can send *secure* datagrams to the destination
  - Datagrams include AH header (inserted in IP the payload)



- AH header includes
  - Next header field - indicates the transport protocol type
  - Security parameter index - identifies SA (connection identifier)
  - Sequence number - sequence number for each datagram
  - Authentication data - contains message signature

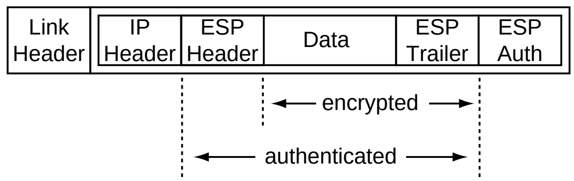
  The IP datagram already indicates the transport protocol and sequence number, so why repeat?

- The actual *signed data* would then follow

ESP Protocol

Provides secrecy and integrity

- Similar to AH, must establish SA first
  - Afterwards, send secure datagrams that include ESP header
  - The data is actually encapsulated



- When the destination receives the datagram
  - Data and ESP trailer is decrypted
  - The trailer then contains the *next header field*

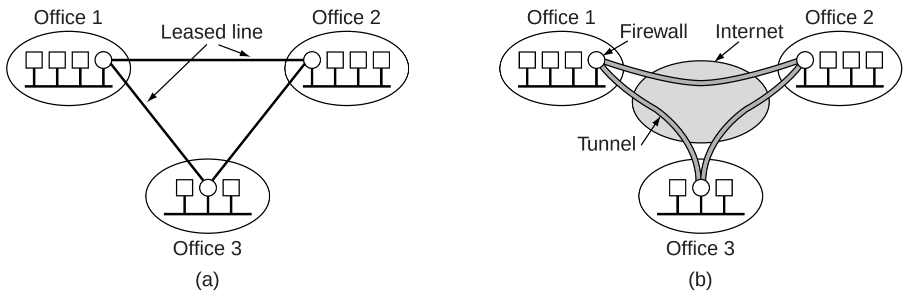
Misc IPsec Items

- For successful deployment of IPsec requires
  - Key management and handshake protocols
- Several have been defined, for example
  - Internet Key Exchange (IKE) algorithm [RFC 2409] is the default key management protocol for IPsec
  - Internet Security Association and Key Management Protocol (ISKMP) defines procedures for establishing SAs
- SA and ESP have **tunnel mode**, encapsulates the entire packet

  So we know the network layer is layer 3 and it is implemented in routers and switches. So does the use of IPsec impact these devices?

Virtual Private Networks

- Most companies have multiple offices at different locations
  - Not cost effective to lease lines between locations



- Virtual Private Network (VPN) can provide secure communication
  - Provide secure communication over insecure networks
  - Create secure tunnels between office pairs

- If IPsec is used for tunneling
  - Possible to aggregate all connections between offices
  - Only need one single authenticated, encrypted SA
- Often the firewalls will negotiate SA between the sites
  - Common to have firewalls, VPNs, and IPsec with ESP in tunnel mode

What is the advantage and disadvantage?

Title

- Item
  - Sub-item

The Cost of IPsec

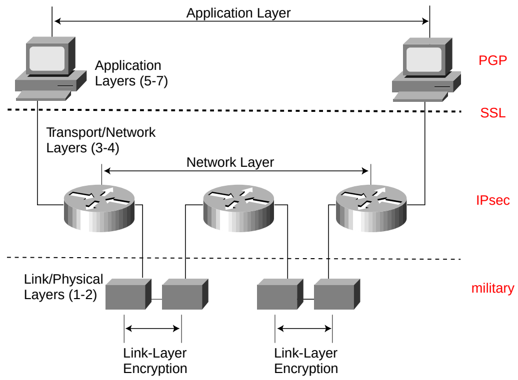
- There is a cost associated with security
  - The transmission speed of packets is reduced due to hashing, encryption, and decryption *What else slows throughput?*
- Below are the performance results of a IPsec prototype
  - Sending a large file between Sun workstations using 10MB/s Ethernet (*old technology and data*)

IPsec Performance Measurement	
Transmission technique	**Bandwidth**
No IPsec (no STREAMS)	315 kb/s
IPsec with AH	26 kb/s
IPsec with transport-mode ESP	26 kb/s
IPSEC with transport-mode ESP and AH	20 kb/s

Location of Security

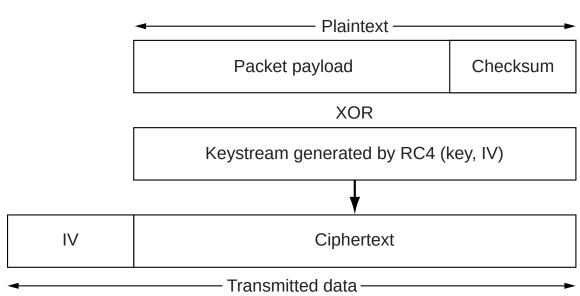
- We have considered security at different locations (layers)
  - Application, application/transport, transport

What are the advantages and disadvantages?



802.11 Security

- 802.11 (wireless) describes Wired Equivalent Privacy (WEP)
  - Supposed to make a wireless LAN as secure as wired LAN
  - Uses secret key, distributed in advance to station and computer



- WEP uses the RC4 stream cipher, plaintext is XORed with the key
  - Payload is checksummed then encrypted using the key
  - The IV used to start RC4 is sent to synchronize the receiver

Problems with WEP

- First common mistake is using the *factory* secret key
  - Most manufacturers use the same key for each device
- WEP can be broken even if the key is randomly set
  - IV is only 24 bits, eventually will be reused
  - Some messages are always sent during a session
    For example?
  - Attacker waits until the same IV and key is used again
  - Can then determine the keystream

Other Types of Wireless Threats

- Accidental association (*evil twins*)
  - Computer associates with a rogue Access Point (AP)
  - Also referred to as "accidental associations"... [David joke]
- Malicious association
  - For example, rogue AP setup within corporate offices...
  - Software on laptop can fake an actual AP... need the SSID
- Ad-hoc networks
  - A type of Peer-To-Peer (P2P) networking at layer 2
  - Every device is a possible router... *enough said*
    There are actually a large number of routing attacks...
- Non-traditional networks
  - Bluetooth, barcode readers, RFID, etc...

- MAC spoofing
  - Identify computer with certain *privledges*
  - Spoofing the MAC allows the attacker to pass ACL
    Who would notice?
- (Wo)Man-in-the-middle attacks
  - Attacker can spoof the AP, read messages, forward to real AP
  - Can force users to *"de-authenticate"* and reconnect
- Denial of service
  - Flood AP/network with bogus requests, successful/failure connection messages, etc...
  - *The objective?* forces all connections to reestablish, making cracking easier...

- Network injection
  - Use AP exposed to non-filtered network traffic, specifically broadcasting network traffic (Spanning Tree, OSPF, RIP)
  - Attacker injects fake networking re-configuration commands...
- Caffe Latte attack (*ask LeAhim*)
  - An off-line method to defeat WEP
  - Possible to obtain the WEP key from a remote Windoze client, sending a flood of encrypted ARP requests...

Making Wireless a Little Better

- Use MAC filtering
- Use static IP addresses, no DHCP
- Use Wi-Fi Protected Access (WPA) v1 or v2
  - An improvement over WEP, can use shared predetermined key

    "Weak PSK passphrases can be broken using off-line dictionary attacks by capturing the messages in the four-way exchange when the client reconnects after being deauthenticated. [stuff on how to crack it regardless] Still, WPA Personal is secure when used with good passphrases or a full 64-character hexadecimal key."
- Temporal Key Integrity Protocol (TKIP)
  - Implements per-packet key mixing with a re-keying system and also provides a message integrity check
- EAP, LEAP, and PEAP, are different authentication extensions