Week 11
Network and Application Layer Security

Attacks on TCP/IP

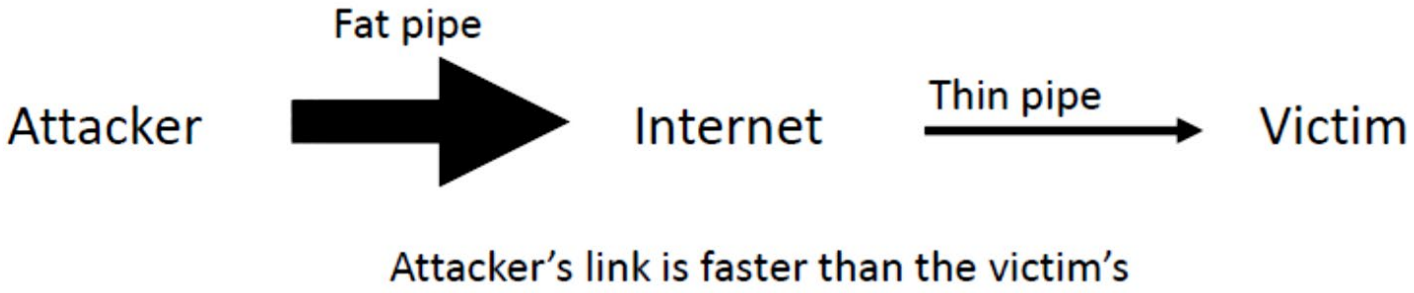
THE UNIVERSITY OF SYDNEY

Denial of Service Principles

- Find a resource (any resource) and use it up
  - Bandwidth
  - CPU or router processing ability
  - Memory, disk space
  - File descriptors, sockets (or other OS resources)
  - Cognitive limits of humans
- Own as many attackers as possible
- Find amplifiers (or post to slashdot.org)
- Choose amplifiers with abundant bandwidth

SYN Flooding

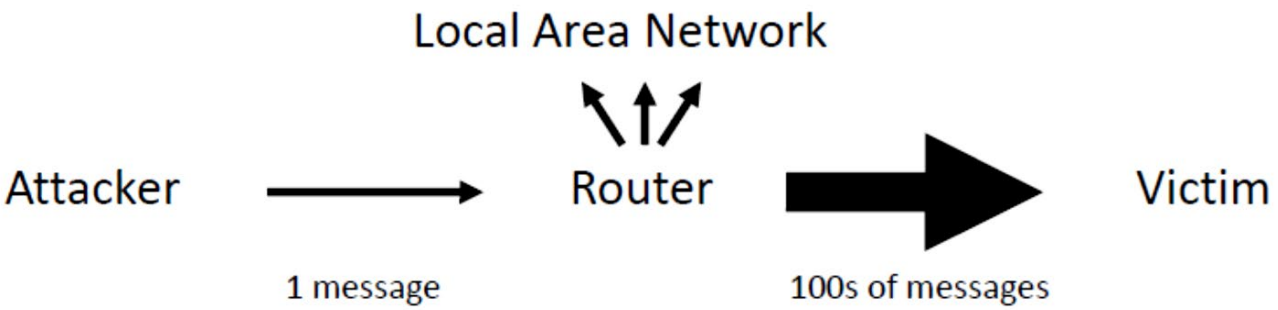
- A simple denial-of-service attack on TCP/IP



- Alice → Bob: SYN(ISN_A)
- Bob → Alice: ACK(ISN_A + 1), SYN(ISN_B)
- Bob, the victim, allocates resources (memory, a process, a socket) to store details from Alice
- If Alice, the attacker, never completes the handshake, eventually all of Bob's resources are used up

Smurfing

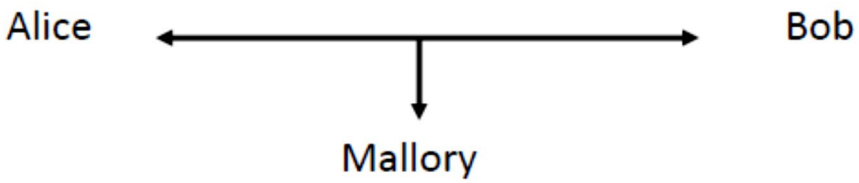
- Another simple denial-of-service attack



- Attacker uses broadcast facility of ICMP echo (i.e. “ping”)
- All hosts respond to single message
- Attacker forges the source address of the victim
- Amplifier machines do not need to be compromised!

Sequence Number Prediction

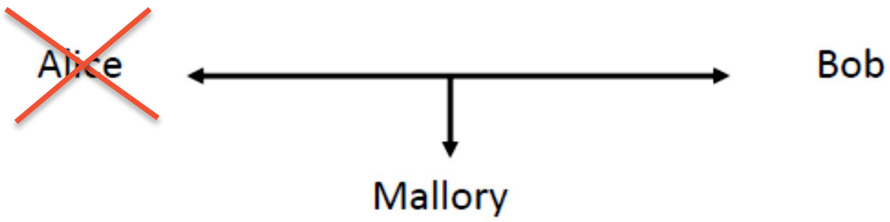
- Say Bob trusts Alice (e.g. through /etc/hosts.equiv)



- Say Alice is alive and Mallory is remote, and he can't see reply packets
  - Mallory → Bob: SYN(ISN_A)
  - Bob → Alice: ACK(ISN_A + 1), SYN(ISN_B)
  - Alice → Bob: RST # wasn't me!
- Alice will tear down the connection

Sequence Number Prediction

- Say Bob trusts Alice (e.g. through /etc/hosts.equiv)

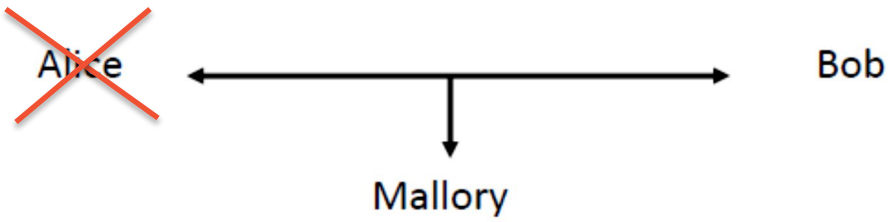


  - Mallory → Alice: SYN flood
  - Mallory → Bob: SYN(ISN_A)
  - Bob → Alice: ACK(ISN_A + 1), SYN(ISN_B)

- Mallory can't see reply packets (he is blind)
- Mallory needs to know ISN_B to complete the connection

Sequence Number Prediction

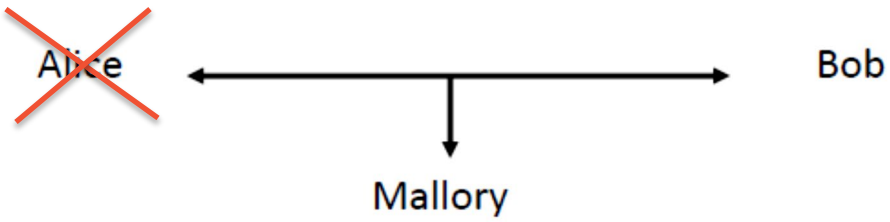
- Say Bob trusts Alice (e.g. through /etc/hosts.equiv)



- Remember that sequence numbers are incremented 128,000 times per second and by 64,000 every new connection
- Mallory can open a connection to Bob earlier to obtain an estimate of the current value of the pointer then guess the current value (or send a flood of guesses)
- Mallory can then send data on the final handshake packet even though he is blind and can't see replies

Sequence Number Prediction

- Say Bob trusts Alice (e.g. through /etc/hosts.equiv)



- Mallory → Alice: SYN flood # bye bye Alice

- Mallory → Bob: SYN(ISN_M) # hi it's Mallory
- Bob → Mallory: ACK(ISN_M + 1), SYN(ISN_B) # welcome

- Mallory → Bob: SYN(ISN_A) # hi it's Alice
- Bob → Alice: ACK(ISN_A + 1), SYN(ISN_B) # welcome
- Mallory → Bob: ACK(ISN_B + 1), PSH(DATA) # execute code

Sequence Number Prediction

- The problem here is authentication by source IP address
- Poor ISN generation also contributes to the problem
- As technology increases, the bounded latency of networks and computer systems becomes more accurate, making this attack easier
- In addition to initiating a connection impersonating someone else, the attacker may gain capability to inject packets into an existing TCP connection or to prematurely close an existing TCP connection
- Note that attacker does not need to see any reply

Session Hijacking

- Session hijacking is where a connection between two parties is hijacked by an attacker (after authentication). Effectively becoming the man in the middle

- In TCP, packets are checked by sequence numbers. i.e., Alice accepts a packet from Bob because it has her IP address and a correct sequence number.

- One form of session hijacking can occur is through connection desynchronisation.

Session Hijacking by Desynchronisation

- Mallory listens for a connection between Alice and Bob.
- At an opportune time (say just after Alice enters her password), Mallory sends packets to both Alice and Bob that increment the sequence numbers on each end such that further packets between Alice and Bob will be regarded as old (outside the window).
- Mallory is now effectively the man in the middle.

Null Data Desynchronisation

- Mallory listens for a connection between Alice and Bob.
  - Alice → Bob: ACK(ISN_B), PSH (DATA)
  - Bob → Alice: ACK(ISN_A), PSH (DATA)

  - Mallory → Bob: ACK(ISN_B + 1), PSH (DATA) # NOP
  - Mallory → Alice: ACK(ISN_A + 1), PSH (DATA) # NOP
  - […]

- Mallory now has a connection to both Alice and Bob

Early Desynchronisation

- Mallory listens for a connection between Alice and Bob.
  - Alice → Bob: SYN(ISN_A)
  - Bob → Alice: ACK(ISN_A + 1), SYN(ISN_B)
  - Mallory → Bob: ACK(ISN_B + 1), RST

    Alice is not aware the connection is closed. Mallory can continue to talk to Alice Pretending to be Bob.
  - Mallory → Bob: SYN(ISN_{AM})
  - Bob → Mallory: ACK(ISN_{AM} + 1), SYN(ISN_{BM})
  - Mallory → Bob: ACK(ISN_{BM} + 1), PSH(DATA)

    Get data from Alice and forward to Bob.
- Mallory now has a connection to both Alice and Bob

Source Routing

THE UNIVERSITY OF SYDNEY

Source Routing

- In both IPv4 and IPv6, sender is allowed to specify packet routes using source routing (or “path addressing”).

- Strict source routing the sender specifies each hop that the packet takes through the network.
    - IP Header: SSRR
- Loose source routing the sender only specifies the group of hosts the packet must transfer through.
    - IP Header: LSRR

- This allows a remote attacker to facilitate non-blind attacks. Previously, they were only allowed to mount blind attacks because they never got a “reply packet”.

Source Routing

- Source routing can be turned off in the kernel.
- Many kernels are configured to ignore source routing.
- Many firewalls and routers block source routed packets and may optionally trigger some alarms.

Port Scanning and Fingerprinting

THE UNIVERSITY OF SYDNEY

Port Scanning & Fingerprinting

- Port Scanning is the process of sending packets to all ports on a machine (or machines) to audit the available (open) services.

Example:

```
# nmap 192.168.0.1-255
Starting nmap V. 2.3BETA14 by fyodor@insecure.org ( www.insecure.org/
    nmap/ )
Interesting ports on cosmic.spectre.net (192.168.0.1):
Port    State   Protocol    Service
22      open    tcp         ssh
139     open    tcp         netbios-ssn

Interesting ports on orbital spectre.net (192.168.0.1):
Port    State   Protocol    Service
7       open    tcp         echo
9       open    tcp         discard
21      open    tcp         ftp
25      open    tcp         smtp
42      open    tcp         nameserver
53      open    tcp         domain
80      open    tcp         http

Nmap run completed -- 255 IP addresses (2 hosts up) scanned in 10
    seconds
```

Port Scanning Detection

- Port scanning is a very “noisy” activity, in that it generates a large number of packets on the network.
- This “noise” can be detected on a network by automated services such as an Intrusion Detection System (IDS).
- These systems listen for suspicious activity on the network and provide alerts to network administrators.

Firewalls

THE UNIVERSITY OF SYDNEY

Firewalls

- A firewall is a packet filtering gateway, aiming to limit the number of services exposed on a connection. *Think of a boundary wall with gateways.*

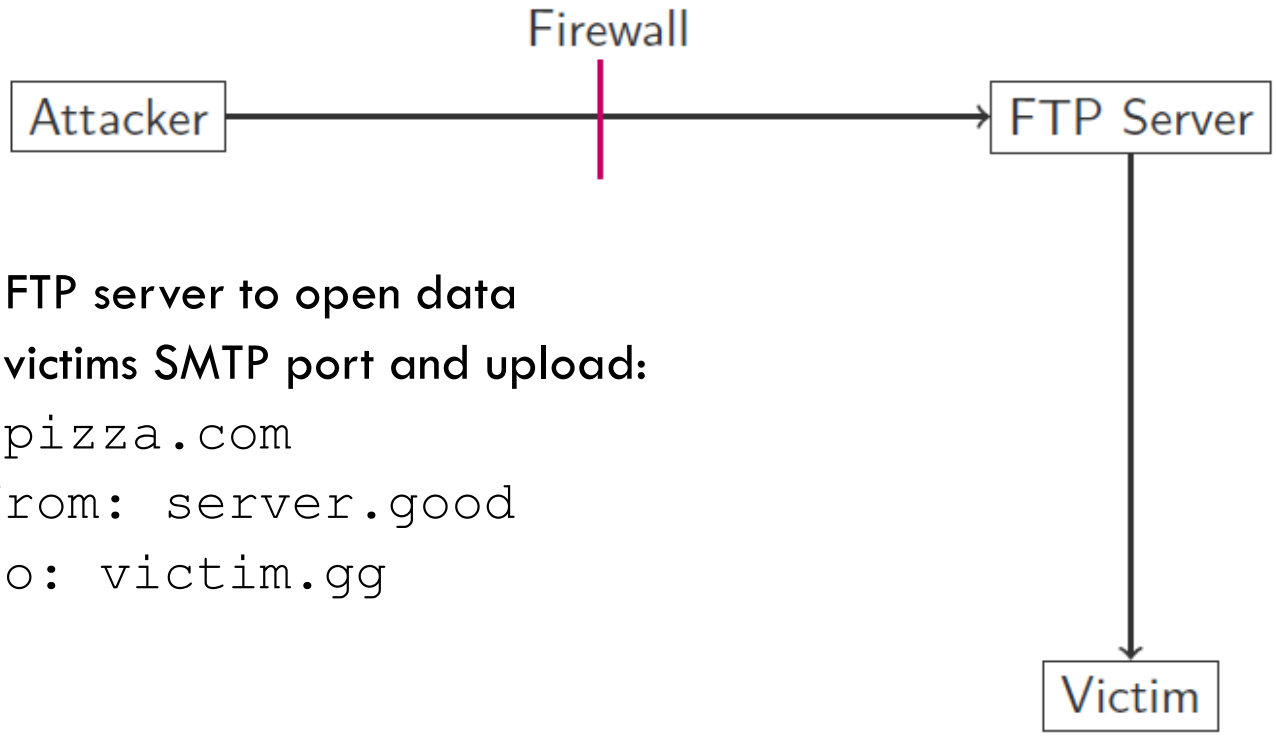
- Static Packet Filtering Gateway
  - Looks at a set of static rules known as access control lists (ACLs). Static packet filters are fast but weak. (And tedious to maintain)
- Dynamic Packet Filtering Gateways
  - Aims at being more intelligent about what packets to allow (stateful inspection of packet headers).
- Application-Level Gateways
  - Attempt to enhance the security further.
    - In short: acts as a proxy (user authentication, no direct IP connections between inside and out).
    - Doesn't support all services (so it isn't used as often).

FTP Bounce Attacks

- FTP Servers can be used to launch bounce attacks.
  - This is an example of an attack that a firewall can't help against.

- Example
  - 1. Attacker finds FTP server located behind a firewall - allowing write to a directory.
  - 2. Attacker logs in, uploads a file containing SMTP commands for a spoofed mail message.
  - 3. Attacker uses the PORT command to point to a victim's mail port.
  - 4. Attacker uses the RETR to initiate the file transfer.
  - 5. The FTP server will then connect to the victim's mail port, uploading valid mail commands.
  - https://en.wikipedia.org/wiki/List_of_FTP_commands

FTP Bounce Attacks



Attacker tells FTP server to open data connection to victims SMTP port and upload:

```
hello pizza.com
mail from: server.good
rcpt to: victim.gg
data
...
end
```

Victim gets (believable) email, spoofed

Traceroute

- Traceroute (on Unix) is a network debugging utility designed to map out the pathway between hosts over IP by monotonically increasing the time-to-live (TTL) field in the IP header.
- The TTL field is used to limit the number of hops a packet has through a network before it expires.
- On expiry, an ICMP error message is generated (the time to live exceeded).
- By increasing the TTL field, we will receive error messages for each hop, tracing the path taken to the destination.
- Windows equivalent - tracert

Firewalking

- Firewalking is the process of determining the access control lists of packet filtering gateways (firewalks, routing...) similar to traceroute.
  - Works by sending packets with a TTL one greater than targeted gateway.
  - If the gateway allows traffic, it will forward the packets to the next hop where it will expire and we will get a message back.
  - If the gateway doesn't allow traffic, it will most likely drop and no response will come back.
  - Through scanning, ACLs for the gateways and firewalls can be determined.

IP Security

THE UNIVERSITY OF SYDNEY

IPSec

- IPSec is a secure network protocol suite aiming at securing IPv4 and IPv6. Two main features:
  - 1. Authentication Header (AH)
    - Authentication and Integrity
  - 2. Encapsulated Security Payload (ESP)
    - For confidentiality

- Packets can use AH and/or ESP. IPsec doesn't protect against.
  - Traffic Analysis.
  - Non-Repudiation.
  - Denial-of-Service.

- IPsec is used to set up virtual private networks (VPNs)

Authentication Header

- The Authentication Header (AH) provides authentication (and possibly integrity) only.
  - Transport Mode is applicable only to host implementations and provides protection for upper layer protocols, in addition to selected IP header fields.
  - Tunnel Mode is where it protects the entire inner IP packet, including the entire IP header.

Encapsulating Security Payload

- The Encapsulating Security Payload (ESP) provides confidentiality for packets.
- Similar to the AH, it can be used in both transport (between two hosts) or tunnel (IP tunnel between gateways) modes.
- ESP operates directly on top of IP.

Address Resolution Protocol

THE UNIVERSITY OF SYDNEY

Address Resolution Protocol

- Address Resolution Protocol (ARP) is used to map IP addresses to hardware addresses.
- A table called the ARP cache is used to store the MAC address and its corresponding IP address.
- When a packet is sent to a node on a network:
  1. Goes to the router.
  2. Queries the ARP for the MAC address matching the destination IP.
  3. ARP lookup returns the MAC from the cache.
     - If it finds the address - ARP program provides it.
     - If no entry found, ARP broadcasts a special packet to all nodes.
       - The machine recognises the IP and responds with its MAC.
       - The ARP cache is updated.

Spoofing ARP

- ARP is one of the simplest, but most fundamental protocols on the Internet.

- Lack of strong authentication means that manipulating ARP is trivial, allows for powerful attacks to be accomplished, including many higher, more secure protocols (ssh, ssl...)
  - Poisoning the ARP cache of targets.
  - MAC flooding.
  - Man-in-the-Middle.
  - Connection hijacking.
  - Denial-of-Service.
  - Cloning.

ARP Attacks

- As time passes, networks are migrating towards being fully switched. Each host is on a separate cable\channel, so the number of machines sharing a connection are minimised.

- Implications:
  - Increases network performance.
  - Increases security: sniffing link will only give traffic for one host, not all on the network.

ARP Attacks

- ARP facilitates Mallory to trivially launch man-in-the-middle attacks against Alice and Bob:
  - Mallory poisons the ARP cache of Alice and Bob.
  - Alice associates Bob's IP with Mallory's MAC.
  - Bob associates Alice's IP with Mallory's MAC.
  - All of Alice and Bob's traffic will now pass through Mallory.
- This works even if the network is fully switched!

- What if Bob is a gateway or router?
  - All the traffic flowing through that router, now comes to Mallory. He can see everything!

Other ARP Attacks

- MAC Flooding
  - Where an attacker sends spoofed ARP replies at a high rate to the switch – eventually overflowing the port/MAC table. Most switches then revert to "full broadcast" mode.
- Denial-of-Service
  - ARP caches are updated with non-existent MAC addresses, causing valid frames to be dropped.
- Connection hijacking
- Cloning

Preventing ARP Spoofing

- ARP Spoofing is very difficult to prevent:
  - Enable MAC binding at a switch.
  - Implementing static ARP tables.

- In MAC binding, once an address is assigned to an adapter it cannot be changed without authorisation.

- Static ARP management is only realistically achieved in a very small network. In a large, dynamic network, it would be impossible to manage the task to keep the entries updated.

- `arpwatch` for Unix based systems monitor changes in the ARP cache and alert admins to the change.

DNS

THE UNIVERSITY OF SYDNEY

DNS

- Many of the earlier problems we have discussed are a result of authentication through source IP address (and spoofing).
- Many other applications also extend trust to other hosts based on their names (known as name addresses).
- The Domain Name Service (DNS) performs the mapping between IP address and name address.

Attacks on DNS

- Similar to ARP, DNS by default does not have any form of authentication.

- The ability to subvert DNS through hacking the nameserver or poisoning the cache leads to many potential attacks:
  - Berkeley r-commands, NFS (file sharing), /etc/hosts.equiv and other transitive trust relationships.
  - Impersonation attacks (e.g., webserver).
  - Denial-of-Service.

DNSSEC

- Suite of IETF extensions to secure DNS using digital signatures, which include:
  - Origin of authentication of DNS data.
  - Authenticated denial of existence.
  - Data integrity.

- The DNSSEC-bis is a subsequent improvement for scalability.

- However, although DNSSEC has been available for years, it still has low adoption rate.
  - Why? Would you sacrifice traffic loss in the event of a small error?

DNSSEC

- New DNS record types:
  - RRSIG
  - DNSKEY
  - DS
  - NSEC
  - NSEC3
  - NSEC3PARAM

- Each lookup returns RRSIG DNS record, a digital signature that can be verified via public key in DNSKEY. DS Record is used in authentication of DNSKEYs using the chain of trust. NSEC and NSEC3 records are used for robust resistance.

DNSSEC

Example:

```
1  fedoraproject.org.  46  IN  RRSIG  AAAA 5 2 60 20170620150112
       20170521150112 7725 fedoraproject.org.
       kFn2QXx6KFbEBkjw2qpZgklAH7cuMyVtpyyR9iB/WmXDoEIVlbZ5ucvq iJPAHiLp/
       TDM8miFSftRu5FpKEUrZQjwHRg1Hd1kovl/YQM3AOpbMmko ukZ7/
       kiUDedKFnvwmSoYWipp5sI80bXrSG414BiaigYz2Cn+7BHXZ6zf yGQ=
2  fedoraproject.org.  46  IN  RRSIG  A 5 2 60 20170620150112
       20170521150112 7725 fedoraproject.org. tcWndX34Xom8DnKJF9+
       rcf0OMaiuJYN0YzH8IJvbgzP5QP7TddQ8/MaT ta7Zr/
       NOBxoqzFJoIfYG9Nt92E1wfGsZT5YSMLQbi/uLcFOss/J4Susn
       IfcZ7cpQLlmAm2RKWBpMxr7+kal1EApU5Tz536zD2YCMn9Dkta3zCWqv B7k=
3  fedoraproject.org.  86386  IN  RRSIG  NSEC 5 2 86400 20170620150112
       20170521150112 7725 fedoraproject.org. i8xbJU+4
       POERYspzwDt9v5uKQkzEwNu9t1RIIdchByAmDzKbvXCehH/9
       yyQLN1jAONkzNmzjvTUiTu6MpmfQblojVLUgIOGeVadfhSqG46xobX9U 9
       uAaK12FbSSmbWPLXwIbxmyxWMvPWtzQH/gBqMpyho1f4UKznBdM1ujt 7kQ=
4  fedoraproject.org.  286 IN  RRSIG  DNSKEY 5 2 300 20170620150112
       20170521150112 7725 fedoraproject.org.
       bAu4GOlsTMQuutwtpBi5V5Jik3gNDSbNt+dcrvrMMBXuoy818XWAaKVv /Te75/
       Li2wiC8fAgnlwj/UjsOCYlMQQr6rByiK2kqP63p63t/X4dFAfb lbOhm8kRA4t+
       DIooJm7AjBLUIP2HxdOlsMgtjHtLodHYWQu4vuEnHE/u O/o=
```

Infrastructure Attacks

- The Internet Control Message Protocol (ICMP) is used to communicate error messages and network conditions across IP. Like other infrastructure protocols, strong authentication is absent.
  - ICMP error messages can be spoofed, telling targets the host (victim) is unavailable.

- Most firewalls block ICMP into and out of the network.

- Likewise, other protocols (RIP, EGPR, BGP, OSPF) are open to attacks. Common attacks with domain names involve subverting the process of domain name registrars, in order to change root name server records!

Conclusion

- We need to “rebuild” the Internet from the ground up using secure building blocks.
  - Build strong authentication into protocols, pretty much every component.
  - Add audit trails and authentication for packets.
  - Out-of-band network control.
  - Stronger languages, better programming practices, better quality control.
  - Secure default configs out of the box.

Application Layer: Web Security

THE UNIVERSITY OF SYDNEY

Application Layer: Web Security

- Websites are a complex interaction of:
  - Languages (JavaScript, SQL, Python, PHP, Ruby...)
  - In-band Mark-up Languages (HTML, CSS)
  - Third Parties (Google Analytics, Facebook, PayPal, VPS...)
  - Protocols (HTTP(S), JSON, DNS...)

- The machines that host websites are accessible from all over the world at any time. If they go down, they are rushed to get back online - security..?

- These websites are built upon common frameworks and tools. Neither machines nor the frameworks are kept up to date.

- A vulnerability in any of these can lead to failure of all the security

Web Vulnerabilities Overview

- There are many vulnerabilities. We will discuss some:
  - SQLi: SQL Injection
  - XSS: Cross-Site Scripting
  - CSRF: Cross-Site Request Forgery
  - SSLStrip: The HTTP to HTTPS bridge
  - Insecure Frameworks: Insecure or old frameworks and code.
  - Insecure machines: Insecure machines

- Many vulnerabilities are due to in-band signalling where content is mixed with control signals. Prime culprits where unsanitised input may result in control signals, e.g., HTML, JavaScript, XSS, SQL Injection

Web Server Architecture

THE UNIVERSITY OF SYDNEY

Application Servers

- App servers do the actual processing of HTTP requests. They contain:
  - Back-end website code
  - Business logic
  - Database interface methods
  - Caching layers

- This is where a majority of the work lies in a typical web architecture.

App Server Languages

- The app server could be written in many different languages.
- Since there are lots of standard web-server functions, developers choose to work within a web framework rather than re-inventing the wheel.
- To name a few languages and some associated web frameworks:
  - Python: Flask, Django
  - PHP: CodeIgniter, CakePHP
  - Ruby: Ruby On Rails
  - JavaScript: Express.js (on Node.js)
  - ASP.NET

Application Databases

- The app servers typically read information from a databases.

- There are many types and flavours of database. We will split them into two main types:
  - SQL Relational databases maintain and enforce relationships between data. Examples: MySQL, PostgreSQL, SQLite, Microsoft SQL Server
  - NoSQL Non-relational databases store data more like loose pages and does not store data in the same relational manner. Examples: MongoDB, Cassandra, CouchDB

- Databases can reside on the same host as the app server, or separately.

Internet Facing Web Servers

- An internet-facing web server can directly receive HTTP requests from clients.

- Usually, the first recipient of the HTTP request is not the service that provides the response. Instead, it is often:
  - A load balancer (LB) routes requests to appropriate servers for processing.
    - Load balancers allow a 'pool' of servers to handle incoming requests by forwarding a request to an available resource.
  - DDoS mitigation services often serve as the first-layer for handling requests on large websites. Their purpose is to simply ensure that incoming requests aren't part of a cyber attack, e.g. CloudFlare.
    - Legit packets are forwarded (often to the regular LB)
    - Attack packets are dropped

Web Vulnerabilities

THE UNIVERSITY OF SYDNEY

SQL Injection

- Very common
- Leads to complete compromise
- Many automated tools

- SQL Injection arises from failing to sanitise input before inserting it into a database string.

- If input can modify SQL queries, then it is possible to do any database queries.

SQL Injection Example

```
1   // Get input
2   $id = $_REQUEST[ 'id' ];
3
4   // Check database
5   $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id
        ';";
6   $result = mysql_query($query)
7
8   // Get results
9   while( $row = mysqli_fetch_assoc( $result ) ) {
10      // Get values
11      $first = $row["first_name"];
12      $last  = $row["last_name"];
13
14      // Feedback for end user
15      $html .= "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {
            $last}</pre>";
16  }
```

SQL Injection Example

- If we access the website with:

  http://example.com?id=15

  the code will respond normally, adding the details for the user with id 15.

- However, if we access the website with

  http://example.com?id=%27%3B+DROP+TABLE+users%3B+--

  Then the query that gets executed will be:

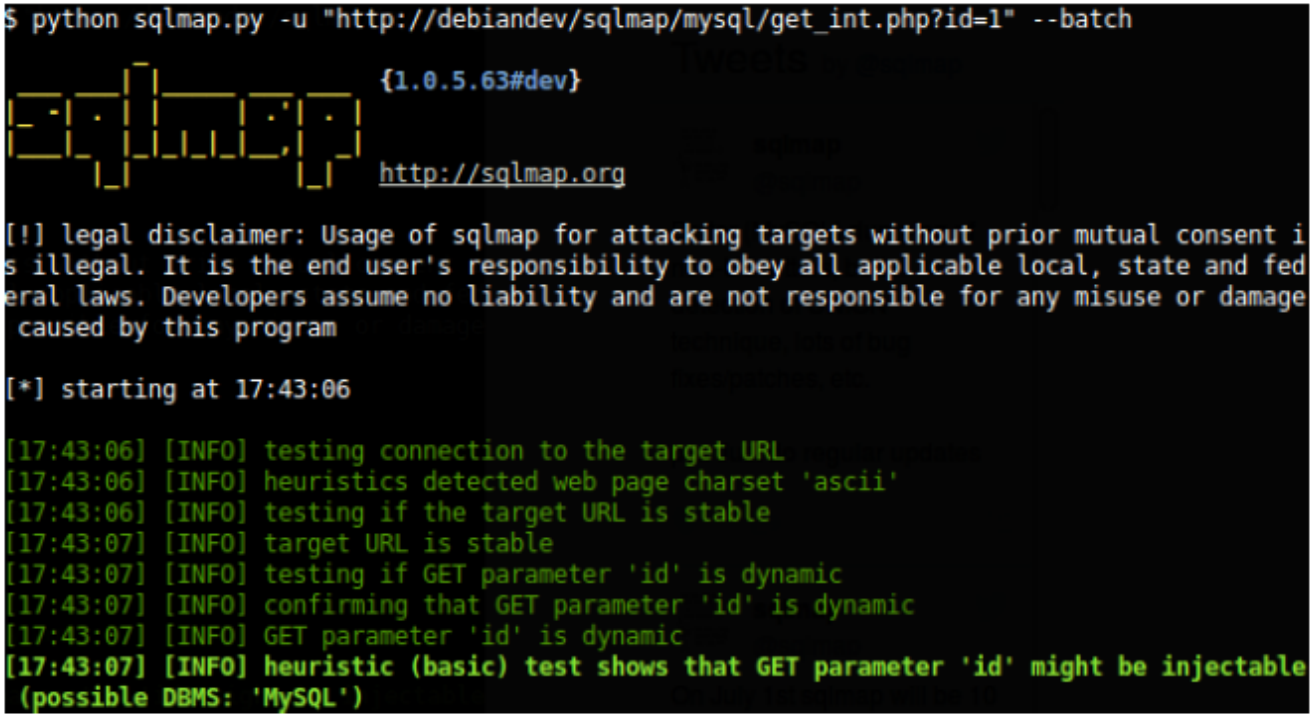
  SELECT first_name, last_name FROM users WHERE user_id =

  ' '; DROP TABLE users; --' ;

- SQL Comments
  - The -- characters indicate the rest of the query is a comment and should be ignored.

SQL Injection: sqlmap

- Performing SQL injection can be automated. A program called sqlmap automates most of the hard work involved when performing SQL injections. It is fully open source and written in Python. [sqlmap: automatic SQL injection and database takeover tool]



Real World Impact

- Real world applications experience a high number of SQLI attacks every hour which need to be stopped. There are a number of SQLI detection and exploit tools around.

- Very common in PHP applications due to the prevalence of direct database commands.

- Successful exploits can lead to:
  - Reading / Leaking sensitive information.
  - Modifying sensitive data (INSERT | UPDATE | DELETE)
  - Executing administrator operations (shutdown, DROP)
  - Reading files of the database systems hard drive.

THE UNIVERSITY OF SYDNEY