Network and Internet Vulnerabilities

Computer Security Lecture 8

David Aspinall

School of Informatics
University of Edinburgh

31st January 2008

Outline

Internet attacks and defences

- General pattern in serious Internet incidents:

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect
  3. Short-term workarounds; specific detection/recovery

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect
  3. Short-term workarounds; specific detection/recovery
  4. Proper repairs to software or protocols are issued

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect
  3. Short-term workarounds; specific detection/recovery
  4. Proper repairs to software or protocols are issued
  5. Over time, most sites implement repairs.

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect
  3. Short-term workarounds; specific detection/recovery
  4. Proper repairs to software or protocols are issued
  5. Over time, most sites implement repairs.
  6. Remaining sites may be black-listed and outlawed.

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect
  3. Short-term workarounds; specific detection/recovery
  4. Proper repairs to software or protocols are issued
  5. Over time, most sites implement repairs.
  6. Remaining sites may be black-listed and outlawed.
- The last step happens in the most serious cases, especially where security flaws may be exploited to attack other, well-managed sites.

Internet attacks and defences

- General pattern in serious Internet incidents:
  1. Someone finds an exploit (usually via software bug)
  2. Exploit is seen in the wild, possibly to large effect
  3. Short-term workarounds; specific detection/recovery
  4. Proper repairs to software or protocols are issued
  5. Over time, most sites implement repairs.
  6. Remaining sites may be black-listed and outlawed.
- The last step happens in the most serious cases, especially where security flaws may be exploited to attack other, well-managed sites.
- Internet security is a **distributed community-wide responsibility**. Black-listing is a socioeconomic countermeasure. Black lists may be useful for crackers as well as good guys (they list hosts which may have security holes), so systems which are not repaired find themselves being attacked and isolated from the rest of the network.

Outline

SYN flooding

- Here is the basic three-part handshake used by Alice to initiate a TCP connection with Bob, and send initial sequence numbers:

$$
\begin{aligned}
A \to B: &\quad \mathtt{SYN},\ X \\
B \to A: &\quad \mathtt{ACK},\ X+1;\ \mathtt{SYN},\ Y \\
A \to B: &\quad \mathtt{ACK},\ Y+1
\end{aligned}
$$

SYN flooding

- Here is the basic three-part handshake used by Alice to initiate a TCP connection with Bob, and send initial sequence numbers:

$$
\begin{aligned}
A \to B: &\quad \texttt{SYN}, X \\
B \to A: &\quad \texttt{ACK}, X+1;\ \texttt{SYN}, Y \\
A \to B: &\quad \texttt{ACK}, Y+1
\end{aligned}
$$

- A DoS attack is **SYN Flooding**. Alice sends many SYN packets, without acknowledging any replies. Bob accumulates more SYN packets than he can handle. Large-scale attacks were seen in 1996.

SYN flooding

- Here is the basic three-part handshake used by Alice to initiate a TCP connection with Bob, and send initial sequence numbers:

$$
\begin{aligned}
A \to B: &\quad \mathtt{SYN}, X \\
B \to A: &\quad \mathtt{ACK}, X+1; \mathtt{SYN}, Y \\
A \to B: &\quad \mathtt{ACK}, Y+1
\end{aligned}
$$

- A DoS attack is **SYN Flooding**. Alice sends many SYN packets, without acknowledging any replies. Bob accumulates more SYN packets than he can handle. Large-scale attacks were seen in 1996.
- A protocol implementation fix called **SYNcookie**, is for Bob to send out Y as encrypted version of X, so he doesn't need to keep state. This is implemented in Linux and some other systems.

Spoofing: forged TCP packets

- **Responses to attacks**. Victim and Internet community want to find attack source, so corporate network administrators or ISP can be notified and given the chance to prevent it (or risk being isolated). Tracing may also allow legal action.

Spoofing: forged TCP packets

- **Responses to attacks**. Victim and Internet community want to find attack source, so corporate network administrators or ISP can be notified and given the chance to prevent it (or risk being isolated). Tracing may also allow legal action.
- Tracing a packet to its source is therefore important. But **forging source addresses** of IP packet is easy!

Spoofing: forged TCP packets

- **Responses to attacks**. Victim and Internet community want to find attack source, so corporate network administrators or ISP can be notified and given the chance to prevent it (or risk being isolated). Tracing may also allow legal action.
- Tracing a packet to its source is therefore important. But **forging source addresses** of IP packet is easy!
- SYN flooding attacks usually have forged source addresses. The ACK is either impossible (address not reachable) or targets another machine, sending meaningless ACK packets.

Spoofing: forged TCP packets

- **Responses to attacks**. Victim and Internet community want to find attack source, so corporate network administrators or ISP can be notified and given the chance to prevent it (or risk being isolated). Tracing may also allow legal action.
- Tracing a packet to its source is therefore important. But **forging source addresses** of IP packet is easy!
- SYN flooding attacks usually have forged source addresses. The ACK is either impossible (address not reachable) or targets another machine, sending meaningless ACK packets.
- The SYNcookie fix doesn’t prevent flooding. As a countermeasure to assist tracing, network providers should implement **ingress filtering** on edge routers (RFC 2267). This ensures packets entering the Internet have source addresses within their origin network fragment, restricting forged packets.

Smurfing (directed broadcast)

- The **smurfing** attack exploits the ICMP (*Internet Control Message Protocol*) whereby remote hosts respond to echo packets to say they're alive (`ping`). Some implementations respond to pings to broadcast address (idea: ping a LAN to find hosts). A bunch of hosts that do it is a *smurf amplifier*.

Smurfing (directed broadcast)

- The **smurfing** attack exploits the ICMP (*Internet Control Message Protocol*) whereby remote hosts respond to echo packets to say they're alive (`ping`). Some implementations respond to pings to broadcast address (idea: ping a LAN to find hosts). A bunch of hosts that do it is a *smurf amplifier*.
- Attack: make packet with forged source address containing the victim's IP number. Send to smurf amplifiers, who swamp target with replies.

Smurfing (directed broadcast)

- The **smurfing** attack exploits the ICMP (*Internet Control Message Protocol*) whereby remote hosts respond to echo packets to say they're alive (`ping`). Some implementations respond to pings to broadcast address (idea: ping a LAN to find hosts). A bunch of hosts that do it is a *smurf amplifier*.
- Attack: make packet with forged source address containing the victim's IP number. Send to smurf amplifiers, who swamp target with replies.
- Fix: standards change August 1999, ping packets sent to broadcast addresses aren't answered. Number of smurf amplifiers is reducing. Black-listing: "concerned sysadmins" at `netscan.org` published name-and-shame list of misconfigured nets.

Smurfing (directed broadcast)

- The **smurfing** attack exploits the ICMP (*Internet Control Message Protocol*) whereby remote hosts respond to echo packets to say they're alive (`ping`). Some implementations respond to pings to broadcast address (idea: ping a LAN to find hosts). A bunch of hosts that do it is a *smurf amplifier*.
- Attack: make packet with forged source address containing the victim's IP number. Send to smurf amplifiers, who swamp target with replies.
- Fix: standards change August 1999, ping packets sent to broadcast addresses aren't answered. Number of smurf amplifiers is reducing. Black-listing: "concerned sysadmins" at `netscan.org` published name-and-shame list of misconfigured nets.
- A **fraggle**: similar attack with UDP packets (port 7, or other ports). Also attacks using TCP.

netscan.org on 5th Feb 2004

netscan.org

Current count: **10,901** broken networks.
Average amplification: **3**x

Welcome to **netscan.org**. This site contains a searchable and browsable list of broadcast ICMP ("smurf") amplifiers.

netscan.org on 5th Feb 2004

netscan.org

Current count: **10,901** broken networks.
Average amplification: **3**x

Welcome to **netscan.org**. This site contains a searchable and browsable list of broadcast ICMP ("smurf") amplifiers.

- 16th Feb 2003: 46k broken networks; 10 in UK.
- 3rd Feb 2005: 2k broken networks reported.
- 29th Jan 2007: `www.powertech.no/smurf/` replaces `netscan.org`, only 231 broken
- 31st Jan 2008: 124 broken (2448867 scanned)

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:
  - Distribute servers over server farm (expensive)

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:
  - Distribute servers over server farm (expensive)
  - Dynamically relocate network under heavy attack (tricky/ineffective)

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:
  - Distribute servers over server farm (expensive)
  - Dynamically relocate network under heavy attack (tricky/ineffective)
  - *Pushback*: try to dynamically restrict likely DoS packets, by rate-limiting with a *congestion signature* (experimental)

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:
  - Distribute servers over server farm (expensive)
  - Dynamically relocate network under heavy attack (tricky/ineffective)
  - *Pushback*: try to dynamically restrict likely DoS packets, by rate-limiting with a *congestion signature* (experimental)
- General defences, by improving packet integrity/traceability:

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:
  - Distribute servers over server farm (expensive)
  - Dynamically relocate network under heavy attack (tricky/ineffective)
  - *Pushback*: try to dynamically restrict likely DoS packets, by rate-limiting with a *congestion signature* (experimental)
- General defences, by improving packet integrity/traceability:
  - Community responsibility: filtering out forged source addresses

DDoS attacks

- In a **distributed denial of service attack**, a large number of machines are subverted with malicious code (e.g., via worm or virus), and then synchronized to attack a target together.
- Specific defences:
  - Distribute servers over server farm (expensive)
  - Dynamically relocate network under heavy attack (tricky/ineffective)
  - *Pushback*: try to dynamically restrict likely DoS packets, by rate-limiting with a *congestion signature* (experimental)
- General defences, by improving packet integrity/traceability:
  - Community responsibility: filtering out forged source addresses
  - Have routers add extra *ICMP traceback messages* with a low probability, e.g., 1 in 20,000. Then sysadmins can trace large-scale attacks back to responsible machines (even if IP spoofing is used).

Sequence number attacks

- Suppose Mallory knows Alice and Bob are hosts on a target LAN, and wants to masquerade as Alice to Bob (in one direction). Here's a strategy:

Sequence number attacks

- Suppose Mallory knows Alice and Bob are hosts on a target LAN, and wants to masquerade as Alice to Bob (in one direction). Here's a strategy:
  1. Take Alice down with a DoS attack (optionally)

Sequence number attacks

- Suppose Mallory knows Alice and Bob are hosts on a target LAN, and wants to masquerade as Alice to Bob (in one direction). Here's a strategy:
  1. Take Alice down with a DoS attack (optionally)
  2. Initiate a new connection with Bob, by sending a `SYN` packet.

Sequence number attacks

- Suppose Mallory knows Alice and Bob are hosts on a target LAN, and wants to masquerade as Alice to Bob (in one direction). Here's a strategy:
  1. Take Alice down with a DoS attack (optionally)
  2. Initiate a new connection with Bob, by sending a SYN packet.
  3. Mallory doesn't get Bob's ACK, but may be able to **guess the sequence number** *Y* used by Bob.

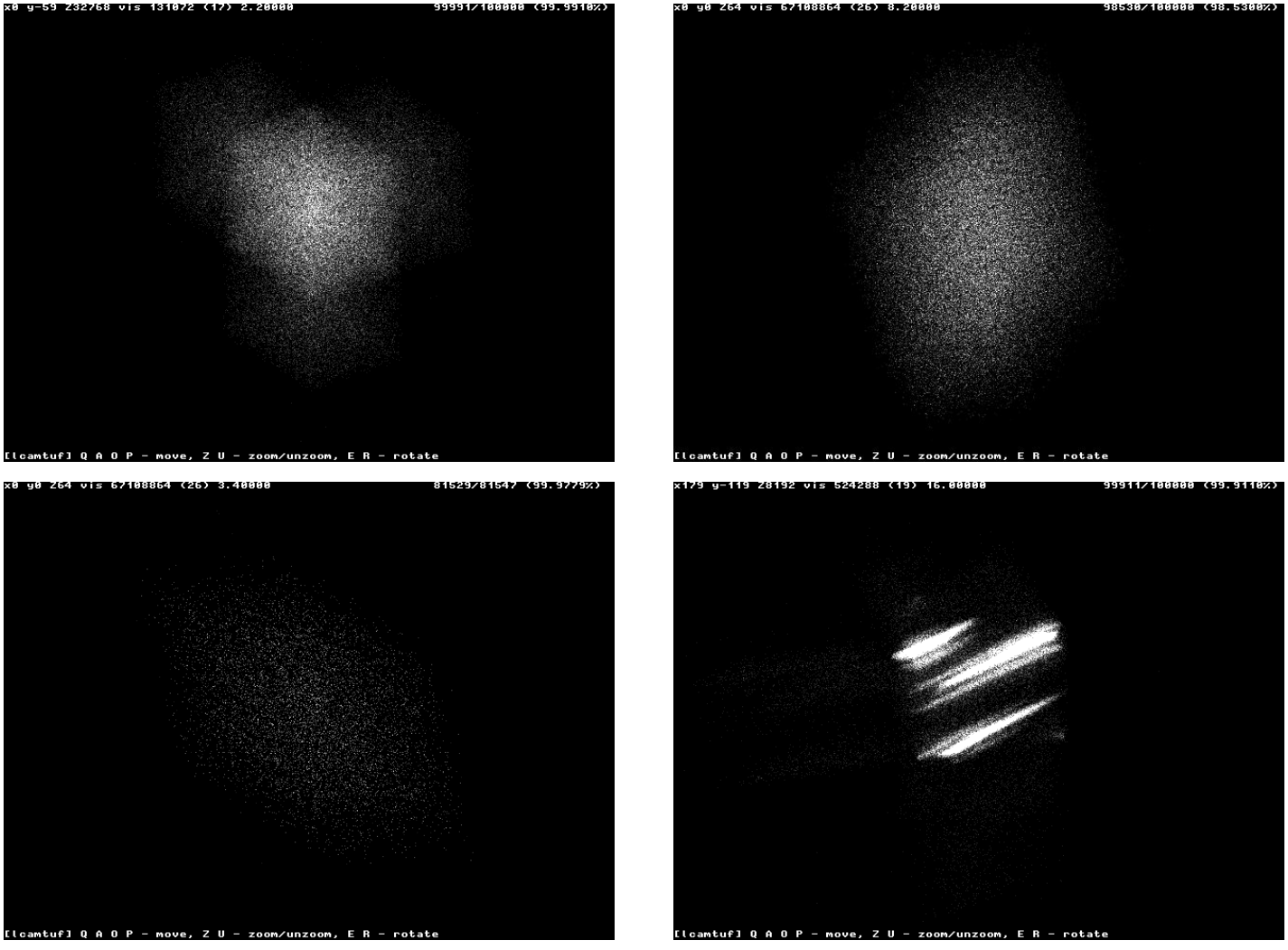
Sequence number attacks

- Suppose Mallory knows Alice and Bob are hosts on a target LAN, and wants to masquerade as Alice to Bob (in one direction). Here's a strategy:
  1. Take Alice down with a DoS attack (optionally)
  2. Initiate a new connection with Bob, by sending a SYN packet.
  3. Mallory doesn't get Bob's ACK, but may be able to **guess the sequence number** *Y* used by Bob.
- Initial sequence numbers may be somehow predictable, so Mallory could make his own connection with Bob and a while later use a related *Y* value when masquerading as Alice. Good IP stack implementations (e.g., most since 2001) use random increments or random values. Many older implementations were not random, or not random enough.

Sequence number attacks

- Suppose Mallory knows Alice and Bob are hosts on a target LAN, and wants to masquerade as Alice to Bob (in one direction). Here's a strategy:
  1. Take Alice down with a DoS attack (optionally)
  2. Initiate a new connection with Bob, by sending a SYN packet.
  3. Mallory doesn't get Bob's ACK, but may be able to **guess the sequence number** *Y* used by Bob.
- Initial sequence numbers may be somehow predictable, so Mallory could make his own connection with Bob and a while later use a related *Y* value when masquerading as Alice. Good IP stack implementations (e.g., most since 2001) use random increments or random values. Many older implementations were not random, or not random enough.
- A complex attack, but can be scripted.

ISN Predicability



- Plots in 2002 for WinXP (tl), Linux (tr), OS/400 (bl), UNICOS (br).
- See `http://lcamtuf.coredump.cx/newtcp`

Routing attacks

- Protocols like OSPF (*Open Shortest Path First*) let routers tell their neighbours about hosts they can reach, and cost metrics (hops). The *transitivity of trust* in routing protocols makes security difficult.

Routing attacks

- Protocols like OSPF (*Open Shortest Path First*) let routers tell their neighbours about hosts they can reach, and cost metrics (hops). The *transitivity of trust* in routing protocols makes security difficult.
- An attacker who controls routing protocols can monitor, intercept, and modify much traffic. E.g., malicious node *M* announces low cost route to hosts *A* and *B*, and thereby diverts traffic through itself.

Routing attacks

- Protocols like OSPF (*Open Shortest Path First*) let routers tell their neighbours about hosts they can reach, and cost metrics (hops). The *transitivity of trust* in routing protocols makes security difficult.
- An attacker who controls routing protocols can monitor, intercept, and modify much traffic. E.g., malicious node *M* announces low cost route to hosts *A* and *B*, and thereby diverts traffic through itself.
- Packet switched networks route return data flow independently. Using network addresses for authentication falsely trusts integrity of return path, allowing *masquerading*. Circuit-switched networks less risky, but switches are new trust points.

Routing attacks

- Protocols like OSPF (*Open Shortest Path First*) let routers tell their neighbours about hosts they can reach, and cost metrics (hops). The *transitivity of trust* in routing protocols makes security difficult.
- An attacker who controls routing protocols can monitor, intercept, and modify much traffic. E.g., malicious node *M* announces low cost route to hosts *A* and *B*, and thereby diverts traffic through itself.
- Packet switched networks route return data flow independently. Using network addresses for authentication falsely trusts integrity of return path, allowing *masquerading*. Circuit-switched networks less risky, but switches are new trust points.
- TCP includes **source routing**, for bypassing network outages. Source-routed packets escape the (weak) authentication of the return address. Forged ICMP **redirect messages** can have similar effect.

Outline

DNS attacks

- Many protocols, including most email and web protocols (e.g., `smtp` and `http`) assume that lower levels are secure. The most they will do to authenticate is check source or destination addresses using DNS look-ups of hostname or reverse look-ups of IP addresses.

DNS attacks

- Many protocols, including most email and web protocols (e.g., `smtp` and `http`) assume that lower levels are secure. The most they will do to authenticate is check source or destination addresses using DNS look-ups of hostname or reverse look-ups of IP addresses.
- If the DNS can be corrupted somehow, DNS checks may be unreliable, leading to **address forgery**, **spam**, in general, powerful **spoofing attacks** (e.g. “pharming”).

DNS attacks

- Many protocols, including most email and web protocols (e.g., `smtp` and `http`) assume that lower levels are secure. The most they will do to authenticate is check source or destination addresses using DNS look-ups of hostname or reverse look-ups of IP addresses.
- If the DNS can be corrupted somehow, DNS checks may be unreliable, leading to **address forgery**, **spam**, in general, powerful **spoofing attacks** (e.g. “pharming”).
- The attack called **DNS cache poisoning** is based on feeding false information into locally cached DNS tables. It means that, within some network portion, a web site can be redirected elsewhere, for example, completely outwith the web-site server’s control.

Connection hijacking

- An attacker who observes the current sequence number of a connection can inject phony packets.

Connection hijacking

- An attacker who observes the current sequence number of a connection can inject phony packets.
  1. Alice logs in to a server. Mallory watches.

Connection hijacking

- An attacker who observes the current sequence number of a connection can inject phony packets.
  1. Alice logs in to a server. Mallory watches.
  2. At the right moment, he disconnects/disables Alice

Connection hijacking

- An attacker who observes the current sequence number of a connection can inject phony packets.
  1. Alice logs in to a server. Mallory watches.
  2. At the right moment, he disconnects/disables Alice
  3. Then he takes over the session; if he gets the sequence number correct, it is accepted by the server.

Connection hijacking

- An attacker who observes the current sequence number of a connection can inject phony packets.
  1. Alice logs in to a server. Mallory watches.
  2. At the right moment, he disconnects/disables Alice
  3. Then he takes over the session; if he gets the sequence number correct, it is accepted by the server.
- Session hijacking may be detected by the server, if the acknowledgement packet sent by Mallory cites data that it never sent. The server ought to reset the connection; instead sometimes an error condition is assumed, and current sequence numbers are resent.

Connection hijacking

- An attacker who observes the current sequence number of a connection can inject phony packets.
  1. Alice logs in to a server. Mallory watches.
  2. At the right moment, he disconnects/disables Alice
  3. Then he takes over the session; if he gets the sequence number correct, it is accepted by the server.
- Session hijacking may be detected by the server, if the acknowledgement packet sent by Mallory cites data that it never sent. The server ought to reset the connection; instead sometimes an error condition is assumed, and current sequence numbers are resent.
- Can also prevent Alice noticing, by sending synchronized empty packets instead of disconnecting, and letting her reconnect to server afterward.

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:
  1. none

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:
  1. none
  2. IP address

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:
  1. none
  2. IP address
  3. “magic cookie” — clear-text password

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:
  1. none
  2. IP address
  3. “magic cookie” — clear-text password
  4. cryptographic mechanisms

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:
  1. none
  2. IP address
  3. “magic cookie” — clear-text password
  4. cryptographic mechanisms

X Window protocol

- In X, a server runs the physical screen, keyboard, and mouse; applications connect and are allocated use of those resources. A malicious application can monitor all keystrokes, dump the screen, scribble on it, etc.
- X has several authentication mechanisms of varying quality:
  1. none
  2. IP address
  3. “magic cookie” — clear-text password
  4. cryptographic mechanisms

  If any of these authentication mechanisms are broken by an attacker, he can attach a malicious application to the server.

UDP. . . RPC, NFS, NIS

- UDP, the *User Datagram Protocol*, is connectionless. There isn't even the weak authentication from a return path, so source addresses **cannot be trusted at all**.

UDP. . . RPC, NFS, NIS

- UDP, the *User Datagram Protocol*, is connectionless. There isn't even the weak authentication from a return path, so source addresses **cannot be trusted at all**.
- Protocols built on UDP are therefore immediately at risk, unless they implement their own security mechanisms. Unfortunately, the most important, **RPC** (*Remote Procedure Call*) does not. The ordinary RPC authentication field is unsecure; the RPC crypto option is rarely used.

UDP. . . RPC, NFS, NIS

- UDP, the *User Datagram Protocol*, is connectionless. There isn't even the weak authentication from a return path, so source addresses **cannot be trusted at all**.
- Protocols built on UDP are therefore immediately at risk, unless they implement their own security mechanisms. Unfortunately, the most important, **RPC** (*Remote Procedure Call*) does not. The ordinary RPC authentication field is unsecure; the RPC crypto option is rarely used.
- RPC is used to implement **NFS** (*Network File System*), and **NIS** (*Network Information Service*).

UDP. . . RPC, NFS, NIS

- UDP, the *User Datagram Protocol*, is connectionless. There isn't even the weak authentication from a return path, so source addresses **cannot be trusted at all**.
- Protocols built on UDP are therefore immediately at risk, unless they implement their own security mechanisms. Unfortunately, the most important, **RPC** (*Remote Procedure Call*) does not. The ordinary RPC authentication field is unsecure; the RPC crypto option is rarely used.
- RPC is used to implement **NFS** (*Network File System*), and **NIS** (*Network Information Service*).
- NFS and NIS have had numerous additional security problems. NFS file-handles can be guessed. NIS may serve up password files, and NIS server responses can be faked. Newer replacements are recommended.

SNMP

- SNMP, the *simple network management protocol*, is used to configure network devices including routers and switches, and allows servers and devices to report status information.

SNMP

- SNMP, the *simple network management protocol*, is used to configure network devices including routers and switches, and allows servers and devices to report status information.
- Useful for hackers to obtain sensitive info about systems, for example, routing tables.

SNMP

- SNMP, the *simple network management protocol*, is used to configure network devices including routers and switches, and allows servers and devices to report status information.
- Useful for hackers to obtain sensitive info about systems, for example, routing tables.
- Later versions of SNMP have security features (MD5 authentication, DES encryption), but many devices only implement SNMPv1 which sends reports and passwords in clear text.

SNMP

- SNMP, the *simple network management protocol*, is used to configure network devices including routers and switches, and allows servers and devices to report status information.
- Useful for hackers to obtain sensitive info about systems, for example, routing tables.
- Later versions of SNMP have security features (MD5 authentication, DES encryption), but many devices only implement SNMPv1 which sends reports and passwords in clear text.
- Many reported flaws in particular implementations (libraries, specific network devices).

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:
  - ASCII based, similar to HTTP; uses MIME and S/MIME.

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:
  - ASCII based, similar to HTTP; uses MIME and S/MIME.
  - Data transport direct between end points (P2P)

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:
  - ASCII based, similar to HTTP; uses MIME and S/MIME.
  - Data transport direct between end points (P2P)
  - Voice traffic on another transport, e.g. RTP over UDP

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:
  - ASCII based, similar to HTTP; uses MIME and S/MIME.
  - Data transport direct between end points (P2P)
  - Voice traffic on another transport, e.g. RTP over UDP
  - Strong security provisions built in.

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:
  - ASCII based, similar to HTTP; uses MIME and S/MIME.
  - Data transport direct between end points (P2P)
  - Voice traffic on another transport, e.g. RTP over UDP
  - Strong security provisions built in.
- **Skype**, proprietary P2P protocol.

Telephony: H.323 and SIP

- Increasingly importance with rise of VoIP and linking existing telecoms networks to the Internet. Protocols must carry data channels and switching information, usually also allow teleconferencing.
- **H.323**
  - protocol based on ISDN signaling protocol Q.931
  - uses separate UDP ports, via intermediate server(s);
  - firewall must parse ASN.1 to find port numbers.
- **SIP**, the Session Initiation Protocol:
  - ASCII based, similar to HTTP; uses MIME and S/MIME.
  - Data transport direct between end points (P2P)
  - Voice traffic on another transport, e.g. RTP over UDP
  - Strong security provisions built in.
- **Skype**, proprietary P2P protocol.
- Various vulnerabilities reported by CERT/CC, UK NISCC, University of Oulou's PROTOS tool. Including DoS and worse.

Other attacks, mechanisms and tools

- **Packet sniffers** are eavesdropping tools which collect packets passing over the network, typically to skim plaintext login ids and passwords.

Other attacks, mechanisms and tools

- **Packet sniffers** are eavesdropping tools which collect packets passing over the network, typically to skim plaintext login ids and passwords.
- **Port scanning** tools or more generally **vulnerability scanners** can be used to find and investigate network hosts open to particular attacks. Useful to good guys as well as bad guys. Examples: `nmap`, SATAN, Nessus.

Other attacks, mechanisms and tools

- **Packet sniffers** are eavesdropping tools which collect packets passing over the network, typically to skim plaintext login ids and passwords.
- **Port scanning** tools or more generally **vulnerability scanners** can be used to find and investigate network hosts open to particular attacks. Useful to good guys as well as bad guys. Examples: `nmap`, SATAN, Nessus.
- **Authentication attacks** based on breaking authentication protocols using attacks we've seen before, or brute-force guessing passwords or keys. Can be easy: many network devices have default passwords or hidden "service" accounts.

Other attacks, mechanisms and tools

- **Packet sniffers** are eavesdropping tools which collect packets passing over the network, typically to skim plaintext login ids and passwords.
- **Port scanning** tools or more generally **vulnerability scanners** can be used to find and investigate network hosts open to particular attacks. Useful to good guys as well as bad guys. Examples: `nmap`, SATAN, Nessus.
- **Authentication attacks** based on breaking authentication protocols using attacks we've seen before, or brute-force guessing passwords or keys. Can be easy: many network devices have default passwords or hidden "service" accounts.
- **Software bug attacks** exploit bugs in particular network server (or client) program versions. Most incidents raised by CERT/CC are because of program bugs.

References

Surveys of (older) network attacks and defences are in the Wily Hacker book and Anderson's book. For more recent and practical information, look on the Internet, e.g., articles in the hacker magazine Phrack, `http://www.phrack.org`.

Ross Anderson. *Security Engineering: A Comprehensive Guide to Building Dependable Distributed Systems*.
Wiley & Sons, 2001.

William R Cheswick, Steven M Bellovin, and Aviel D Rubin. *Firewalls and Internet Security Second Edition: Repelling the Wily Hacker*.
Addison-Wesley, 2003.

Recommended Reading

Chapter 18 of Anderson.