# Security I – retired slides

Markus Kuhn

UNIVERSITY OF
CAMBRIDGE
Computer Laboratory

Lent 2013 – Part IB

http://www.cl.cam.ac.uk/teaching/1213/SecurityI/

# TEA, a Tiny Encryption Algorithm

TEA is a 64-bit block cipher with 128-bit key and 64-round Feistel structure, designed at the Computer Lab by David Wheeler and Roger Needham for 32-bit processors. The aim was to find a cipher so simple that the implementation can be memorised, not maximum performance:

```
void code(long *v, long *k)
{
  unsigned long y = v[0], z = v[1], sum = 0;
  unsigned long delta=0x9e3779b9, n = 32;
  while (n-- > 0) {
    sum += delta;
    y += ((z << 4) + k[0]) ^ (z + sum) ^ ((z >> 5) + k[1]);
    z += ((y << 4) + k[2]) ^ (y + sum) ^ ((y >> 5) + k[3]);
  }
  v[0]=y ; v[1]=z;
}
```

ftp://ftp.cl.cam.ac.uk/users/djw3/tea.ps

# Network security

> "It is easy to run a secure computer system. You merely have to disconnect all connections and permit only direct-wired terminals, put the machine in a shielded room, and post a guard at the door." — Grampp/Morris

Problems:

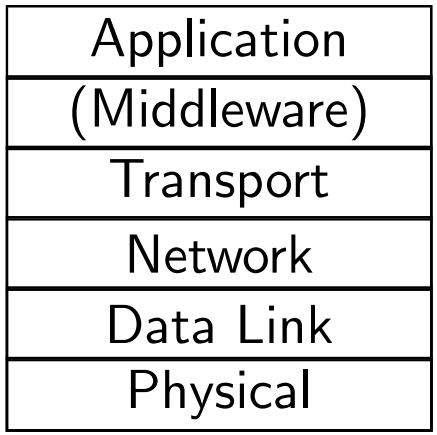
- Wide area networks allow attacks from anywhere, often via several compromised intermediary machines, international law enforcement difficult
- Commonly used protocols not designed for hostile environment
  - authentication missing or based on source address, cleartext password, or integrity of remote host
  - missing protection against denial-of-service attacks
- Use of bus and broadcast technologies, promiscuous-mode network interfaces
- Vulnerable protocol implementations
- Distributed denial-of-service attacks

# TCP/IP security

TCP/IP transport connections are characterised by:

- Source IP address
- Destination IP address
- Source Port
- Destination Port

Network protocol stack:




IP addresses identify hosts and port numbers distinguish between different processes within a host. Port numbers $< 1024$ are "privileged"; under Unix only root can open them. This is used by some Unix network services (e.g., rsh) to authenticate peer system processes.

Example destination ports:
20–21=FTP, 22=SSH, 23=telnet, 25=SMTP (email), 79=finger, 80=HTTP, 111=Sun RPC, 137–139=NETBIOS (Windows file/printer sharing), 143=IMAP, 161=SNMP, 60*xx*=X11, etc. See `/etc/services` or `http://www.iana.org/assignments/port-numbers` for more.

# Address spoofing

IP addresses are 32-bit words (IPv6: 128-bit) split into a network and a host identifier. Destination IP address is used for routing. The IP source address is provided by the originating host, which can provide wrong information ("address spoofing"). It is verified during the TCP 3-way handshake:

$$
\begin{aligned}
S \to D: &\quad \texttt{SYN}_x \\
D \to S: &\quad \texttt{SYN}_y, \texttt{ACK}_{x+1} \\
S \to D: &\quad \texttt{ACK}_{y+1}
\end{aligned}
$$

Only the third message starts data delivery, therefore data communication will only proceed after the claimed originator has confirmed the reception of a TCP sequence number in an ACK message. From then on, TCP will ignore messages with sequence numbers outside the confirmation window. In the absence of an eavesdropper, the start sequence number can act like an authentication nonce.

# Examples of TCP/IP vulnerabilities I

- The IP *loose source route* option allows $S$ to dictate an explicit path to $D$ and old specifications (RFC 1122) require destination machines to use the inverse path for the reply, eliminating the authentication value of the 3-way TCP handshake.
- The connectionless *User Datagram Protocol (UDP)* has no sequence numbers and is therefore more vulnerable to address spoofing.
- Implementations still have predictable start sequence numbers, therefore even without having access to reply packets sent from $D$ to $S$, an attacker can
  - impersonate $S$ by performing the entire handshake without receiving the second message ("sequence number attack")
  - disrupt an ongoing communication by inserting data packets with the right sequence numbers ("session hijacking")

# Examples of TCP/IP vulnerabilities II

- In many older TCP implementations, *D* allocates a temporary data record for every half-open connection between the second and third message of the handshake in a very small buffer. A very small number of `SYN` packets with spoofed IP address can exhaust this buffer and prevent any further TCP communication with *D* for considerable time ("SYN flooding").
- For convenience, network services are usually configured with alphanumeric names mapped by the *Domain Name System (DNS)*, which features its own set of vulnerabilities:
  - DNS implementations cache query results, and many older versions even cache unsolicited ones, allowing an attacker to fill the cache with desired name/address mappings before launching an impersonation attack.
  - Many DNS resolvers are configured to complete name prefixes automatically, e.g. the hostname *n* could result in queries *n*`.cl.cam.ac.uk`, *n*`.cam.ac.uk`, *n*`.ac.uk`, *n*. So attacker registers `hotmail.com.ac.uk`.

# Firewalls I

Firewalls are dedicated gateways between intranets/LANs and wide area networks. All traffic between the "inside" and "outside" world must pass through the firewall and is checked there for compliance with a local security policy. Firewalls themselves are supposed to be highly penetration resistant. They can filter network traffic at various levels of sophistication:

- A basic firewall function drops or passes TCP/IP packets based on matches with configured sets of IP addresses and port numbers. This allows system administrators to control at a single configuration point which network services are reachable at which host.
- A basic packet filter can distinguish incoming and outgoing TCP traffic because the opening packet lacks the `ACK` bit. More sophisticated filtering requires the implementation of a TCP state machine, which is beyond the capabilities of most normal routing hardware.

# Firewalls II

- Firewalls should perform plausibility checks on source IP addresses, e.g. not accept from the outside a packet with an inside source address and vice versa.
- Good firewalls check for protocol violations to protect vulnerable implementations on the intranet. Some implement entire application protocol stacks in order to sanitise the syntax of protocol data units and suppress unwanted content (e.g., executable email attachments $\rightarrow$ viruses).
- Logging and auditing functions record suspicious activity and generate alarms. An example are port scans, where a single outside host sends packets to all hosts of a subnet, a characteristic sign of someone mapping the network topology or searching systematically for vulnerable machines.

Firewalls are also used to create encrypted tunnels to the firewalls of other trusted intranets, in order to set up a *virtual private network (VPN)*, which provides cryptographic protection for the confidentiality and authenticity of messages between the intranets in the VPN.

# Limits of firewalls

- Once a host on an intranet behind a firewall has been compromised, the attacker can communicate with this machine by tunnelling traffic over an open protocol (e.g., HTTPS) and launch further intrusions unhindered from there.
- Little protection is provided against insider attacks.
- Centrally administered rigid firewall policies severely disrupt the deployment of new services. The ability to "tunnel" new services through existing firewalls with fixed policies has become a major protocol design criterion. Many new protocols (e.g., SOAP) are for this reason designed to resemble HTTP, which typical firewall configurations will allow to pass.

Firewalls can be seen as a compromise solution for environments, where the central administration of the network configuration of each host on an intranet is not feasible. Much of firewall protection can be obtained by simply deactivating the relevant network services on end machines directly.

**“Is this product/technique/service secure?”**

Simple Yes/No answers are often wanted, but typically inappropriate. Security of an item depends much on the context in which it is used.

Complex systems can provide a very large number of elements and interactions that are open to abuse. An effective protection can therefore only be obtained as the result of a systematic planning approach.

# Security Management and Engineering II

**"No need to worry, our product is 100% secure. All data is encrypted with 128-bit keys. It takes billions of years to break these."**

Such statements are abundant in marketing literature. A security manager should ask:

- What does the mechanism achieve?
- Do we need confidentiality, integrity or availability of exactly this data?
- Who will generate the keys and how?
- Who will store / have access to the keys?
- Can we lose keys and with them data?
- Will it interfere with other security measures (backup, auditing, scanning, ...)?
- Will it introduce new vulnerabilities or can it somehow be used against us?
- What if it breaks or is broken?
- ...

# Security policy development

## Step 1: Security requirements analysis

- Identify assets and their value
- Identify vulnerabilities, threats and risk priorities
- Identify legal and contractual requirements

## Step 2: Work out a suitable security policy

The security requirements identified can be complex and may have to be abstracted first into a high-level **security policy**, a set of rules that clarifies which are or are not authorised, required, and prohibited activities, states and information flows.

**Security policy models** are techniques for the precise and even formal definition of such protection goals. They can describe both automatically enforced policies (e.g., a mandatory access control configuration in an operating system, a policy description language for a database management system, etc.) and procedures for employees (e.g., segregation of duties).

## Step 3: Security policy document

Once a good understanding exists of what exactly security means for an organisation and what needs to be protected or enforced, the high-level security policy should be documented as a reference for anyone involved in implementing controls. It should clearly lay out the overall objectives, principles and the underlying threat model that are to guide the choice of mechanisms in the next step.

## Step 4: Selection and implementation of controls

Issues addressed in a typical low-level organisational security policy:

- General (affecting everyone) and specific responsibilities for security
- Names manager who "owns" the overall policy and is in charge of its continued enforcement, maintenance, review, and evaluation of effectiveness
- Names individual managers who "own" individual information assets and are responsible for their day-to-day security
- Reporting responsibilities for security incidents, vulnerabilities, software malfunctions
- Mechanisms for learning from incidents

- Incentives, disciplinary process, consequences of policy violations
- User training, documentation and revision of procedures
- Personnel security (depending on sensitivity of job)

  Background checks, supervision, confidentiality agreement
- Regulation of third-party access
- Physical security

  Definition of security perimeters, locating facilities to minimise traffic across perimeters, alarmed fire doors, physical barriers that penetrate false floors/ceilings, entrance controls, handling of visitors and public access, visible identification, responsibility to challenge unescorted strangers, location of backup equipment at safe distance, prohibition of recording equipment, redundant power supplies, access to cabling, authorisation procedure for removal of property, clear desk/screen policy, etc.
- Segregation of duties

  Avoid that a single person can abuse authority without detection (e.g., different people must raise purchase order and confirm delivery of goods, croupier vs. cashier in casino)
- Audit trails

  What activities are logged, how are log files protected from manipulation
- Separation of development and operational facilities
- Protection against unauthorised and malicious software

- Organising backup and rehearsing restoration
- File/document access control, sensitivity labeling of documents and media
- Disposal of media

  Zeroise, degauss, reformat, or shred and destroy storage media, paper, carbon paper, printer ribbons, etc. before discarding it.
- Network and software configuration management
- Line and file encryption, authentication, key and password management
- Duress alarms, terminal timeouts, clock synchronisation, ...

For more detailed check lists and guidelines for writing informal security policy documents along these lines, see for example

- British Standard 7799 "Code of practice for information security management"
- German Information Security Agency's "IT Baseline Protection Manual"
  `http://www.bsi.bund.de/english/gshb/manual/`
- US DoD National Computer Security Center Rainbow Series, for military policy guidelines
  `http://en.wikipedia.org/wiki/Rainbow_Series`

# UK Computer Misuse Act 1990

- Knowingly causing a computer to perform a function with the intent to access without authorisation any program or data held on it $\Rightarrow$ up to 6 months in prison and/or a fine
- Doing so to further a more serious crime
  $\Rightarrow$ up to 5 years in prison and/or a fine
- Knowingly causing an unauthorised modification of the contents of any computer to impair its operation or hinder access to its programs or data $\Rightarrow$ up to 5 years in prison and/or a fine

The intent does not have to be directed against any particular computer, program or data. In other words, starting automated and self-replicating tools (viruses, worms, etc.) that randomly pick where they attack is covered by the Act as well. Denial-of-service attacks in the form of overloading public services are not yet covered explicitly.

`http://www.hmso.gov.uk/acts/acts1990/Ukpga_19900018_en_1.htm`

# UK Data Protection Act 1998

Anyone processing personal data must comply with the eight principles of data protection, which require that data must be

1. fairly and lawfully processed

   Person's consent or organisation's legitimate interest needed, no deception about purpose, sensitive data (ethnic origin, political opinions, religion, trade union membership, health, sex life, offences) may only be processed with consent or for medical research or equal opportunity monitoring, etc.

2. processed for limited purposes

   In general, personal data can't be used without consent for purposes other than those for which it was originally collected.

3. adequate, relevant and not excessive
4. accurate
5. not kept longer than necessary
6. processed in accordance with the data subject's rights

   Persons have the right to access data about them, unless this would breach another person's privacy, and can request that inaccurate data is corrected.

7. secure
8. not transferred to countries without adequate protection

   This means, no transfer outside the European Free Trade Area. Special "safe harbour" contract arrangements with data controllers in the US are possible.

## Some terminology:

"Personal data" is any data that relates to a living identifiable individual ("data subject"), both digitally stored and on paper.

A "data controller" is the person or organisation that controls the purpose and way in which personal data is processed.

`http://www.hmso.gov.uk/acts/acts1998/19980029.htm`
`http://www.ico.gov.uk/`
`http://www.admin.cam.ac.uk/univ/dpa/`