Cryptography and Network Security II

Second Course

Lecture 13: Wireless security, Password management, Viruses

Overview of the course

- I. CRYPTOGRAPHY
  - Secret-key cryptography
    - Classical encryption techniques
    - DES, AES, RC5, RC4
  - Public-key cryptography
    - RSA
  - Key management
- II. AUTHENTICATION
  - MAC
  - Hashes and message digests
  - Digital signatures
  - Kerberos
- III. NETWORK SECURITY
  - Email security
  - Web security (SSL, secure electronic transactions)
  - IP security
  - Firewalls
  - Wireless security
- IV. OTHER ISSUES
  - Viruses
  - Digital cash
  - Secret sharing schemes
  - Zero-knowledge techniques

Wireless security

Wireless Security

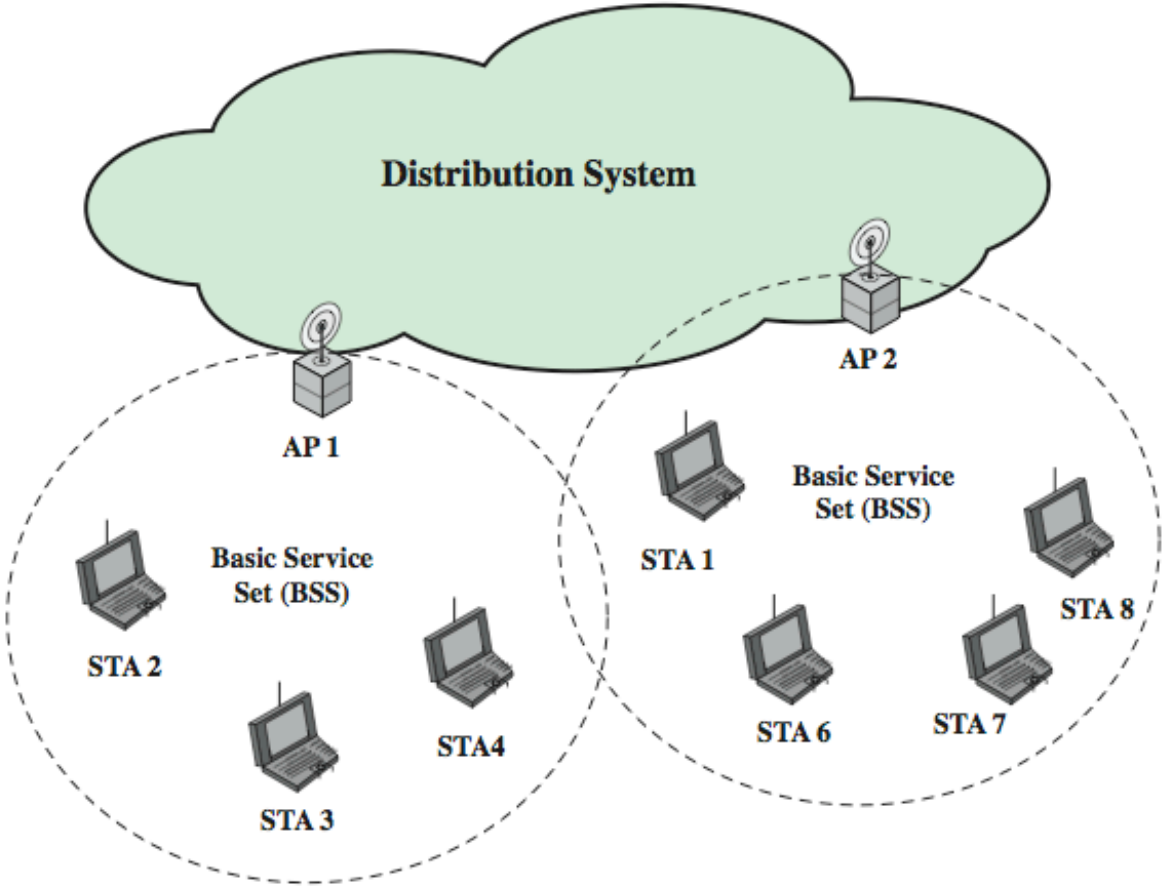
- Very convenient to have wireless connections
- Nightmare for security – the range for 802.11 is often a few hundred meters, so that one can spy on a company by leaving a laptop on in the parking lot
- Many problems arise from the vendors trying to make their products as friendly as possible – when you plug the device it starts working right away, usually with no security by default
- Take a look here at several ways wireless nets handle security
  - 802.11i
  - Bluetooth
  - Wap 2.0

802.11 security – key points

- IEEE 802.11 is a standard for wireless LANs
  - interoperable standard-compliant implementations are called Wi-Fi
- IEEE 802.11i specifies security standards for IEEE 802.11 LANs
  - interoperable implementations called Wi-Fi Protected Areas (WPA)
- Wireless Application Protocol (WAP) – standard to provide mobile use of wireless phones and other wireless devices access to telephony and information services, including the internet
- WAP security primarily provided by Wireless Transport Layer Security (WTLS) – provides security between the mobile device and the WAP gateway to the Internet

Network components and architectural model

- BSS
  - Client stations do not talk to each other directly
  - everything relayed through the AP
  - connections between a stations and a BSS are dynamic
- Independent BSS (IBSS)
  - client stations talk directly to each other
  - typically an ad-hoc network



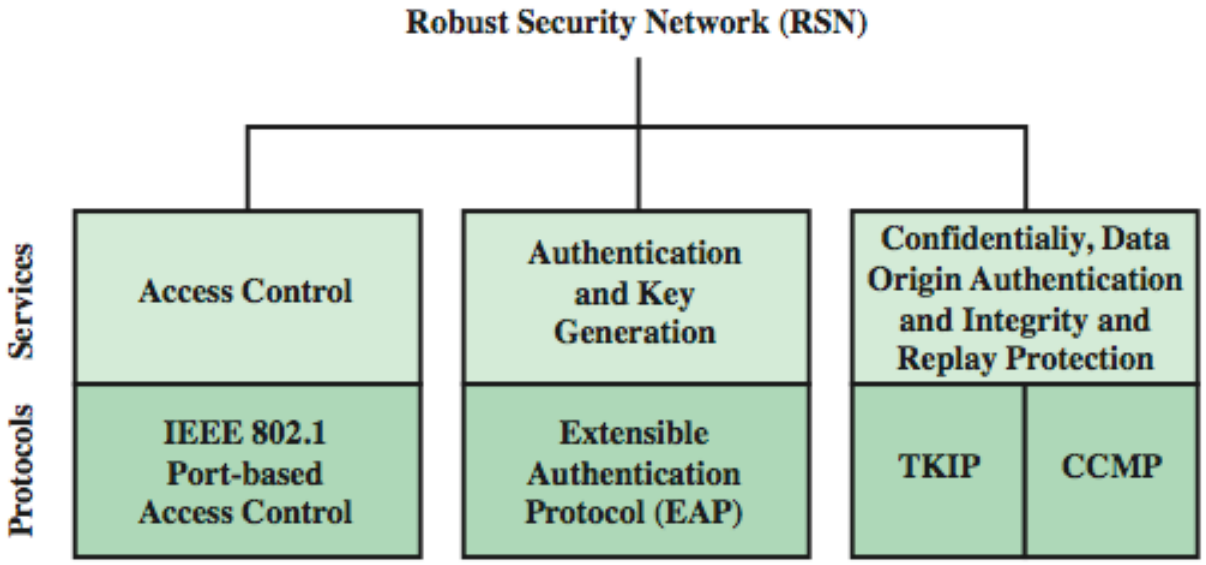
Stallings, fig 17.3

802.11 security

- 802.11 standard prescribed a data link-level security protocol called WEP (Wired Equivalent Privacy) designed to make the security of a wireless LAN as good as that of a wired LAN
- When 802.11 security is enabled, each station has a secret key shared with the AP – not specified how keys are distributed
- WEP encryption uses a stream cipher based on RC4 (generate a key stream that is XORed with the data), the IV used to augment the RC4 key is sent in plain
  - WEP has been broken already in July 2001 (Borisov et al.)
- Solution
  - Replace WEP with WPA (Wi-fi Protected Access) or WPA2
  - Final proposal in 802.11i: Robust Security Network (RSN)
    - The Wi-Fi Alliance certifies vendors in compliance with the full 802.11i specification under the WPA 2 program

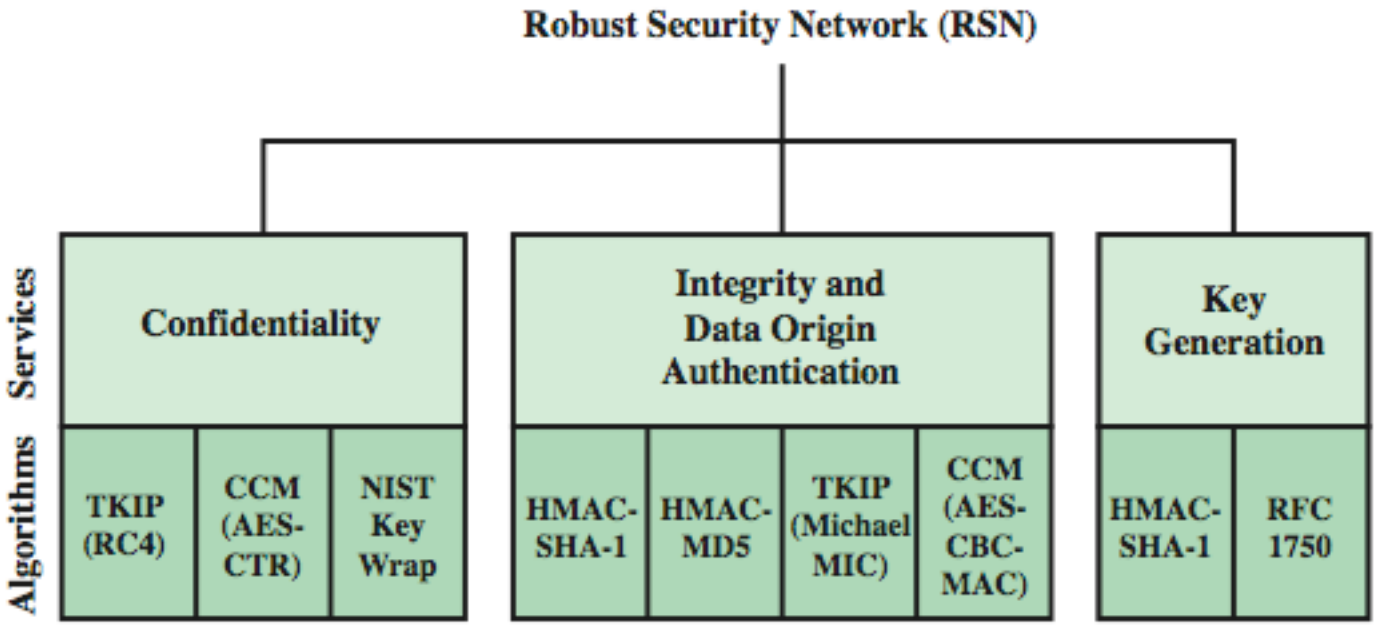
802.11i RSN Services and Protocols

- Authentication: provides mutual authentication and generates temporary keys to be used between user and AS
- Access control: enforces the use of authentication function, routes the messages properly and facilitates key exchange
- Privacy with message integrity



Stallings, fig 17.4a

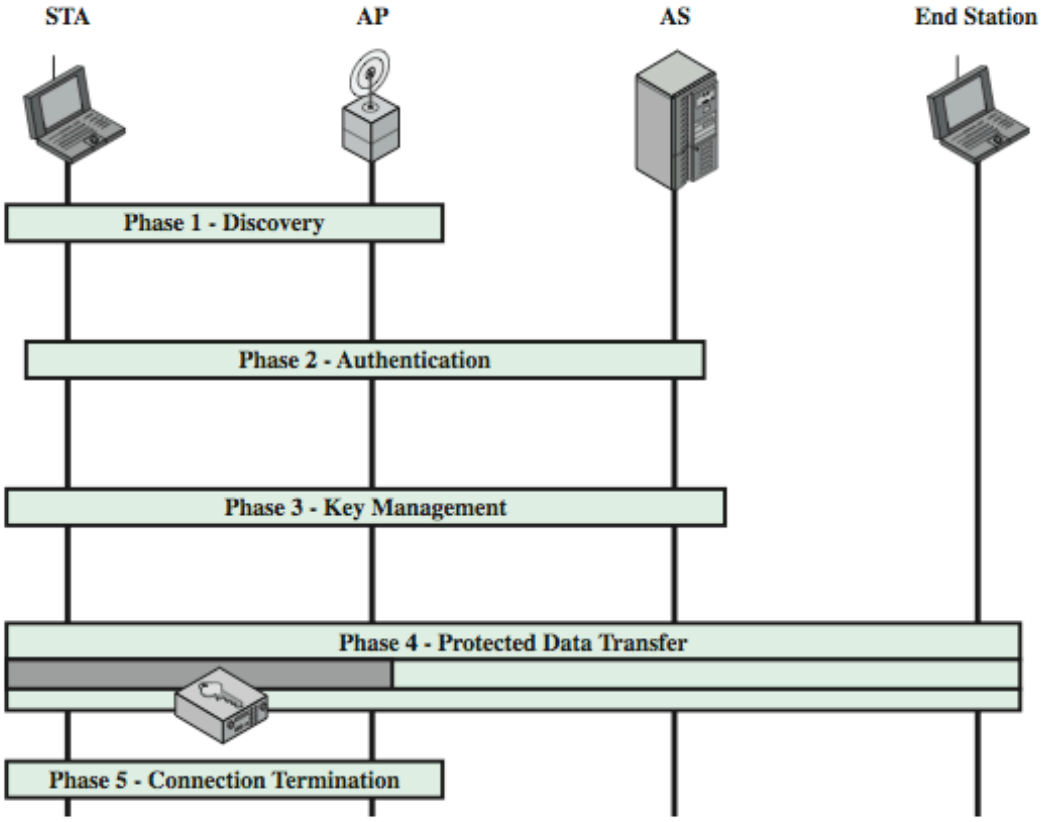
802.11i RSN Cryptographic Algorithms



Stallings, fig 17.4b

802.11i Phases of Operation

- **Discovery**: An AP uses messages called Beacons and Probe Responses to advertise its IEEE 802.11i security policy. STA uses them to identify an AP for a WLAN with which it wishes to communicate
- **Authentication**: STA and AS prove their identities to each other. The AP blocks non-authentication traffic between the STA and AS until the authentication transaction is successful. The AP does not participate in the authentication transaction other than forwarding traffic between the STA and AS.
- **Key generation and distribution**: AP and the STA perform several operations that cause cryptographic keys to be generated and placed on the AP and the STA
- **Protected data transfer**: Frames are exchanged between the STA and the end station through the AP. Secure data transfer occurs between the STA and the AP only
- **Connection termination**: AP and STA exchange frames. During this phase, the secure connection is torn down and the connection is restored to the original state.



Stallings, fig 17.5

Bluetooth Security

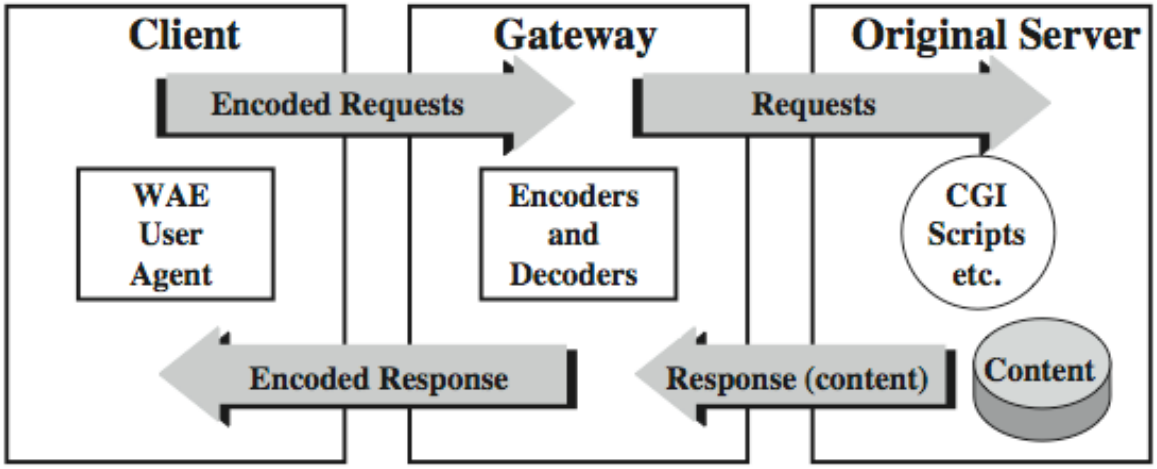
- Considerably shorter range than 802.11 – cannot be attacked from the parking lot but feasible to attack it from next office
  - An attacker next door can read the signals from one's keyboard or the data sent to the Bluetooth printer in the next office
- Bluetooth has 3 security modes ranging from nothing to full data encryption and integrity control – many users have security turned off
  - Two devices must share a secret key – perhaps the user types a PIN in both
  - They negotiate the channel to be used and establish a 128-bit session key (some bits made public due to government restrictions)
  - Encryption uses a stream cipher called E0, integrity control uses SAFER+, both classical block-ciphers (Bluetooth finalized before AES cipher was chosen, SAFER+ was one of the contenders) – E0 is similar to A5/1 cipher used in GSM and broken in 2000
- Many reports about various attacks on Bluetooth
- Concerns about many devices left on and visible by default

WAP (Wireless Application Protocol) 2.0 security

- Introduced for handheld-devices connectivity (mobile phones)
- Uses standard protocols in all layers
- It is IP-based and it supports IPsec in the network layer
- TCP connections protected by TLS in the transport layer
- Uses HTTP client authentication
- Probably better than 802.11 and Bluetooth because it only relies on well-known security standards

WAP Programming Model

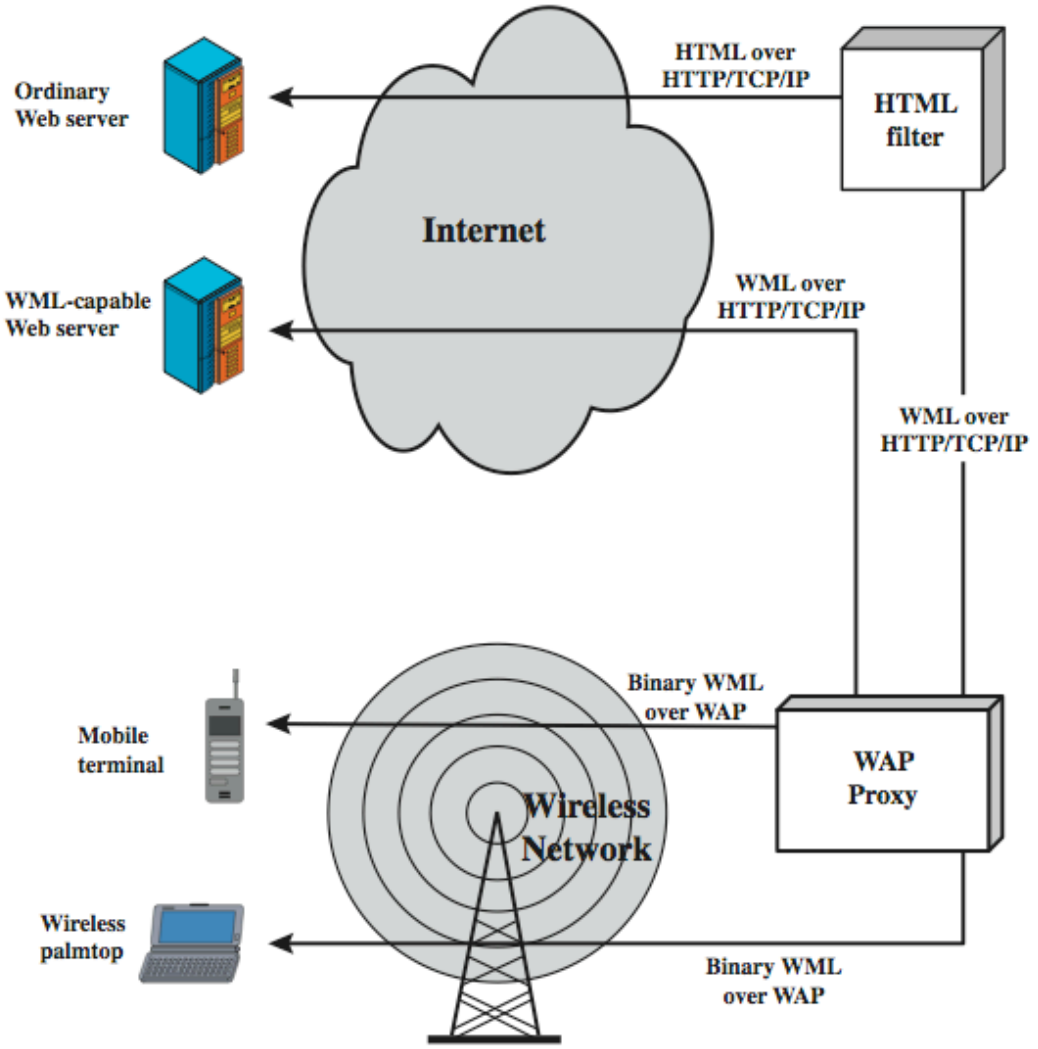
- HTTP used between gateway and the original server
- Gateway acts as a proxy server for the wireless domain
- Gateway provides a number of services (DNS services, conversion between WAP and WWW stacks, encodes into more compact form) to offload the limited computing capabilities of mobile clients



Stallings, fig 17.11

WAP Infrastructure

- WAP architecture is designed to cope with the two principal limitations of wireless Web access:
  - the limitations of the mobile node (small screen size, limited input capability)
  - the low data rates of wireless digital networks.



Stallings, fig 17.12

System security

Intruder detection

Password management

Viruses

Intruders

- More of a system (computer) security problem, rather than a *network* security problem but does have influence on it
- Hostile or unwanted trespass by users or software can pose threats to the whole network
- Three general classes of intruders
  - Masquerader – individual who is not authorized to use the computer, exploits a legitimate user's account (attacker is likely to be an outsider)
  - Misfeasor – legitimate user who accesses data, programs, or resources for which such access is not authorized (attacker generally an insider)
  - Clandestine user – individual who seizes supervisory control of the systems and uses it to evade auditing and access control or to suppress audit collection (can be insider or outsider)

Example

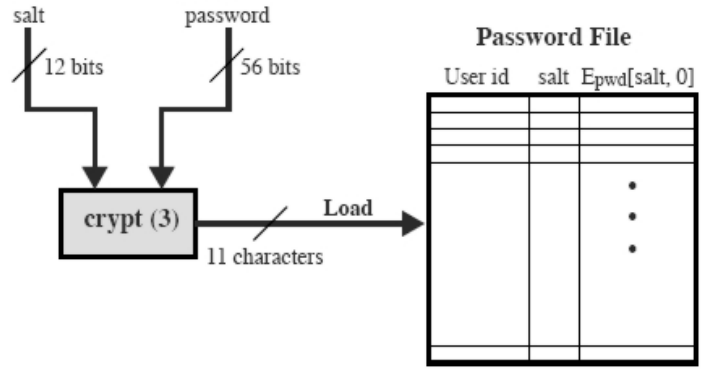
- 1992, Texas A&M University
  - Computer center notified that one of its machines was being used to attack computers at another location via the Internet
  - Several intruders were involved running password-crackers on several computers in the local campus
  - Center disconnected the machines, fixed known security holes and resumed normal operations
  - Few days later detected that the intruder attack resumed also
  - Extremely sophisticated attack:
    - several hundred passwords broken already, some on major and supposedly secure servers,
    - one of the local machines setup as a hacker bulletin board reporting techniques and progress
    - Two types of attackers involved: sophisticated users with good knowledge of security and “foot soldiers” with available computers and time to spare
    - They were attacking those computers that were reported weak by a legitimate security team of the University

Intruder techniques – password security

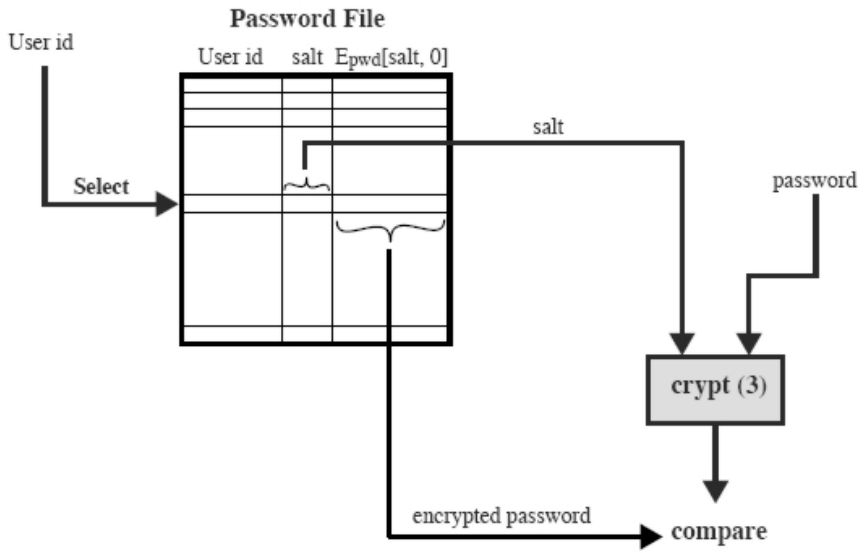
- In most cases, intruders need to acquire protected information, most often passwords
- Password security is a big problem on any system: hackers may take over someone's user account and use it in a major attack inside that system or against a totally different system – liability may involve the careless user
- The system protects the user passwords in two ways
  - Keeps them "encrypted" on the disk – in fact, one keeps on the disk hashes of the password: in this way, nobody can attack the system by trying to "decrypt" the password file
  - The file with the user personal information should be public so that when the user initiates login, the login process can check his data
  - The password file should be "hidden": /etc/passwd, /etc/shadow in Linux
- To break the password file, the attacker essentially has to "guess" the password of a user, hash it, and then compare it with the entry in the password file

Intrusion techniques – password crackers

- Protecting the passwords
  - In UNIX, the 8 characters of the password are converted to a 56-bit string that serves as key for (modified with 12-bit “salt”value) DES – start with 64 bits all 0 and iterate DES encryption 25 times: the result will be the hash of the password and will be stored in the file
  - The salt prevents duplicate passwords from showing in the password file (chosen based on the time when the password was assigned to the user)



(a) Loading a new password



(b) Verifying a password

Figure 18.4 UNIX Password Scheme

Intrusion techniques – password crackers

- Attackers can try to run a password cracker program once they gained access to the system (perhaps using a guest account)
- Better yet: try to guess the password:
  - Try default passwords used for standard accounts – some system administrators do not bother to change them when they buy the computer
  - Exhaustively try all short passwords
  - Try words in the system's online dictionary and also a list of likely passwords
  - Collect information about users: full name, names of their spouse and children, pictures in their office, books showing their hobbies, etc
  - Try user's phone number, social security numbers, room numbers
  - Try all legitimate license plate numbers for this state
  - Use a Trojan horse

Password selection strategies

- **Users are generally stubborn and insist using bad passwords** – the reason for this is that they actually have to remember them, also they may be bad judges of what is a good password
  - Capitalizing the first/last letter of your friend's name does not make a good password
- **Computer-generated passwords are better**
  - Disadvantage: difficult to remember
  - Programs exist to generate random passwords that can be pronounced, but still difficult to remember
- **Systems have sometimes password checking programs**
  - Check from time to time for bad passwords – ask those users to change them immediately
  - At the time of password selection, check if the password is easy to guess – if so, ask the user to provide another one – most users will not complain and they will realize that there are actually many good passwords they can give
    - E.g., ask all passwords to have length 8
    - Always include an uppercase, lowercase, digit, and a punctuation mark
    - Compile a large collection of bad passwords

Intrusion detection

- **Statistical methods**: how likely is it for this user to connect at this hour, using this computer, and trying to access this type of resources?
  - Need to find a balance – many false positive alarms will annoy legitimate users who try to work at unusual times and will induce system administrator to ignore alarms, many false negatives allow attackers to penetrate the system
- **Rule-based penetration identification** – based on expert system technology: collect suite of known penetration scenarios and evaluate the likelihood of such a scenario taking place for each alarm. Rules include:
  - Users should not read files in other users' personal directories
  - Users must not write other users' files
  - Users who login after hours often access same files as they used earlier
  - Users do not generally open disk devices directly, rely on higher-level system utilities
  - Users should not be logged in more than once to the same system
  - Users do not make copies of system programs

Malicious software

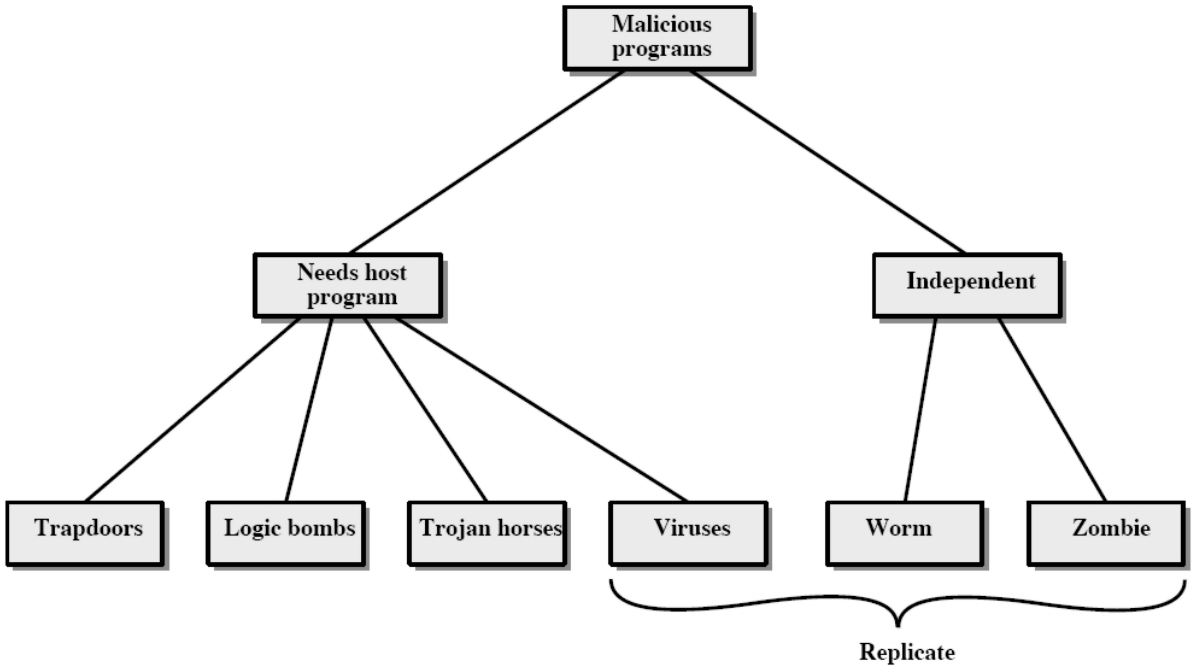


Figure 19.1 Taxonomy of Malicious Programs

Malicious software

- Trap doors
  - Secret entry point into a program that allows somebody aware of it to gain access without going through the usual security access procedures
  - Have been used legitimately for many years by programmers for debug and test
  - Trap door is a code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events
  - Difficult to counter

Malicious software

- Logic bomb
  - One of the oldest types of program threat, known before viruses and worms
  - Code embedded in some legitimate programs that is set to explode when certain conditions are met: presence or absence of certain files, a particular day of the week or date, a particular user running the application, etc.
  - Once triggered, the bomb may alter or delete data or entire files, cause a machine to halt, or do some other damage

Malicious software

- Trojan horses
  - Useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function
  - Can be used to perform functions that a user could not perform directly – induce another user to run the Trojan horse (by advertising it as a useful computer tool) – when run, the Trojan horse will have the right of the process owner, i.e., of the attacked user
  - Examples: listing somebody's files into a file accessible to the attacker
  - Another example: a compiler that inserts additional code into certain programs as they are compiled, e.g., a system login program to create a trapdoor into it

Malicious software

- Zombie
  - Program that secretly takes over a Internet-based computer and then uses that computer to launch attacks that cannot be traced to the zombie's creator
  - Used in denial-of-service attacks, typically against Web servers

Viruses

- A program that can infect other programs by modifying them, including a copy of the virus program
  - Similar to biological viruses
- Can do anything that other programs can
  - Attaches itself to another program and executes when the program executes
- The stages of a virus life are
  - Dormant phase – virus is idle, will be awaken by a certain trigger – a date, the presence of another program or file, the capacity of the disk exceeding some limit, etc. Not all viruses have this phase
  - Propagation phase – virus places an identical copy of itself into other programs or into certain system areas on the disk – each infected program will carry a clone of the virus which itself will enter a propagation phase
  - Triggering phase – virus is activated to perform its function, can be triggered by a number of events, including a count of its clones
  - Execution phase: function is performed

Types of viruses

- Stealth viruses
  - Compression virus – compresses part of its code so that it does not modify the size of the infected file
  - Also, can intercept I/O routines so that when there is an attempt to read the infected files using those routines, the virus will present the original uninfected program
- Polymorphic viruses – create virus clones that are functionally identical but different in code so that a scan will not show identical clones of the virus
  - Interchange the order of independent pieces of code
  - Use encryption – write the key in plain in the code of the virus and use it to encrypt the rest of the code, change the key for each clone
- Macro viruses – make up nowadays two thirds of all viruses
  - Platform independent
  - Infect documents not executable code
  - Easily spread, e.g., by email
- E-mail viruses – send itself to all people on the local address book and then do some local damage

Worms

- Propagates itself from system to system (much like an e-mail virus)
  - A worm will actively seek out more machines to infect without any expected trigger from the human
  - Once active in the system it behaves like a virus
  - To spread, worms use e-mail, remote execution capability, remote login capability
  - Unlike a virus, it need not attach to an existing program
- Same phase as a virus: dormant, propagation, triggering, execution

Worms – some examples

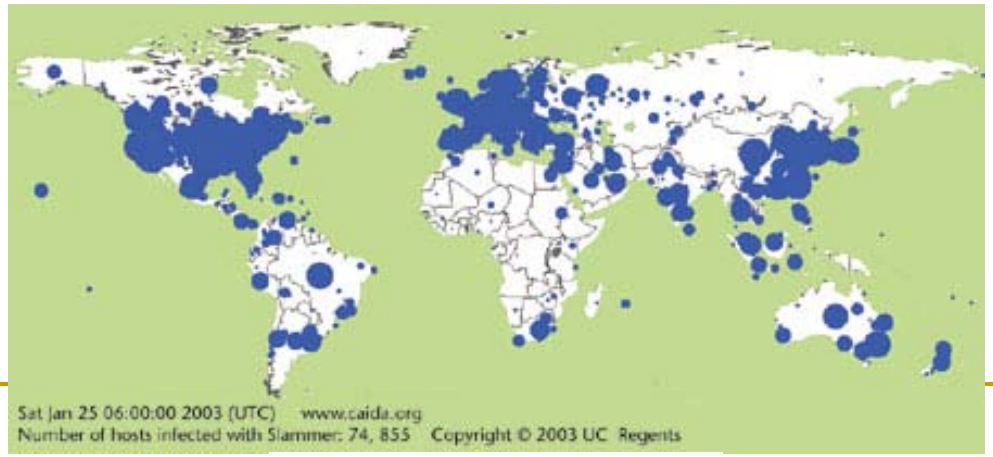
- Morris worm (1998)
  - Launched by a Cornell University computer science graduate
  - Spread on UNIX systems
  - First discover other hosts known from this host that would allow entry from this host: list and tables, mail forwarding files, etc
  - Try to access those systems: login as a legitimate user (first crack the local password file and then assume that some users have the same or close passwords on different systems), exploit a bug in the finger protocol, or exploit a trapdoor in the debug option of the remote process that receives and sends email
  - After getting communication with the operating system command interpreter, it sends a short boot-strap program, issued a command to execute it, then logged off; the bootstrap then called back and downloaded the remainder of the worm
  - Execute the worm
  - Affected about 10% of the Internet-connected computers at that time

Worms – some examples

- Code Red (July 2001) – exploited a security hole in the Microsoft Internet Information Server to penetrate and spread
  - Probes random IP addresses to spread to other hosts
  - For a while it only spreads
  - Then launch denial-of-service attack against a government website by flooding the site with packets from many hosts
  - Infected about 360.000 servers in 14 hours
  - Consumed enormous amounts of Internet capacity
- Code Red II – similar, plus installs a backdoor allowing a hacker access to the infected computers
- Nimda (late 2001) – spreads through email, open net shares, browsing, Microsoft IIS, backdoors left by Code Red II

Worms – some examples

- Slammer (also called Sapphire) – fastest computer worm at that time (January 25, 2003) – infected more than 75.000 hosts, 90% of them in the first 10 minutes
- Exploited a buffer-overflow in computers on the Internet running MS SQL Server or MS SQL Server Desktop Engine
- Lead to cancelled airline flights, interference with elections, ATM failures
- Bellow is the geographical spread of Slammer in the first 30 minutes



Worms – some examples

- Slammer had no malicious payload, just disrupted Internet communication, overloaded nets, disabled database servers – hardware failures were reported because of it
  - Failures included Bellevue, Washington's 911 emergency service, partially Bank of America's ATM network
- Randomly selected IP addresses
- Initially spread exponentially, slowed down afterwards
  - So fast because it did not wait for any answer from the attacked hosts (like Code Red did) – bombarded hosts with one UDP packet – 404 bytes altogether, compared with 4kb for Code Red or 60kb for Nimda
  - Much more damage was possible – the author had some programming errors
    - The pseudo-random number generator (for IP addresses) had a minor error, enough to try the same computers many times
    - Used OR instead of a XOR to clear some register

Virus countermeasures

- Antivirus approaches
  - Detection, identification, removal
  - Use heuristics instead of looking for a signature
  - Activity traps
  - Access control capability
- Advanced Antivirus techniques
  - Digital immune system (prototype of IBM): when a new virus enter, the immune system captures it, analyzes it, adds detection and shielding for it, removes it, and passes information about it to other registered digital systems
  - Behavior blocking software – block potentially malicious software before it does the damage: attempts to open, view, delete, modify files, attempts to format disks, modifications to the logic of executable files, modifications of critical system settings, initiation of network communication, etc.