KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

IT for Law and Legal Science 2

PhD Program Law and Digital Technologies

doc. Mgr. Jan Outrata, Ph.D.

Purpose of this course

- **introduction to computer systems security and cybercrime**
- information technology (IT) and related legal view
- introductory, selected info – too broad and complex area, basic-to-intermediate level, supposed basic IT knowledge

Contents

Credit

= written analysis and prevention discussion of chosen given fictive computer systems security incident or cybercrime, from the IT and related legal point of view

Course web page at `outrata.inf.upol.cz`

Recommended literature

[1] Du W.: Computer Security: A Hands-on Approach (Computer & Internet Security), 3rd ed. Wenliang Du, 2022.

[2] Du W.: Internet Security: A Hands-on Approach (Computer & Internet Security), 3rd ed. Wenliang Du, 2022.

[3] Easttom W. II: Computer Security Fundamentals, 5th edition. Pearson IT Certification, 2023.

[4] Kaufman Ch., Perlman R., Speciner M., Perlner R.: Network Security: Private Communication in a Public World, 3rd edition. Addison-Wesley Professional, 2022.

[5] Brooks Ch. J., Grow Ch., Craig P. A. Jr., Short D.: Cybersecurity Essentials. Sybex, 2018.

[6] Kolouch J., Bašta P. a kol.: CyberSecurity. CZ.NIC, 2019.

[7] Kolouch J.: CyberCrime. CZ.NIC, 2016.

[8] Bandler J., Merzon A.: Cybercrime Investigations. CRC Press, 2022.

Additional literature

[9] Wilson D. C.: Cybersecurity. The MIT Press, 2021.

[10] Grubb S.: How Cybersecurity Really Works: A Hands-On Guide for Total Beginners. No Starch Press, 2021.

[11] Marsh N.: Practical Cybersecurity: A Fat-Free Guide to Network Security Best Practices. Independently published, 2023.

[12] Alexandrou A.: Cybercrime and Information Technology: The Computer Network Infrastructure and Computer Security, Cybersecurity Laws, Internet of Things (IoT), and Mobile Devices. CRC Press, 2021.

PART I.

Computer systems and software security

Computer systems and software security

- computer system = computer (PC or special) or network of computers with other devices (communication, data storage, print, multimedia etc.) and data
  - hardware = physical components of computers and devices, „bare metal“
  - software = applications/programs run in the system, specially interpreted data
  - data = form of (representation of) information processed by the system

- system security ∼ system and data protection against attacks, misusage, vulnerabilities exploits, errors etc.

→ applying and controling system security via:

+ user accounts and authentication (identity verification)
+ privilegies (permissions) and authorization (access control)
+ data encryption
+ monitoring and logging for solving security incidents
+ hardware and software control and maintenance – hardware and software security

! endless ongoing repetitive process, not one-and-for-all action

! legal implications: regulation (security enforcement) and law enforcement (demanding privilegies, censorship) vs. freedom of computer system usage, access to information and privacy

User accounts

= representation of user (human) and system/program task identities to the system

∼ user roles in the system

1. ordinary – for activities of human users: using the system and programs for common tasks (work, entertainment); no special privilegies needed
2. system – for activities of (possibly other) system parts and programs: data management (e.g. backup), hardware control, for program proper working (when using hardware), network activities roles etc.; special privilegies used
3. administrator – for system maintenance and management by human user; all privilegies granted!

- account info: login name, password (or other auth. data), location of user's data directory = „home“, login program = user interface for interaction with the system, ID, groups, user full name ...
- groups of users (accounts): ordinary, system, administrator(s); sharing, often additional, privilegies

Privilegies (permissions)

= form of authorization of (human) users, system parts and programs to: use the system (login), run programs (execution), access/manipulate data (read, write), use and control hardware and network (making connections, data exchange), maintain the system and programs (configuration) etc.

- defined and demanded for: parts of the system (user and program interfaces, maintenance services), programs (functions, using system services), data (files, directories/folders, disks, storages), hardware (particular functions), network (services) etc.

→ granted to users (accounts):

- permanently via association lists or groups of users with privilegies set
- temporarily via privilege delegation/elevation within an action only (e.g. „run as“, „sudo“)

Authentication

= verification and identification of user and his/her association to user account or system role, to enable using the system (according to authorization)

- based on (= factors): user something unique! 1 knows, 2 has or 3 is
  - all represented by data ⇒ copyable, reusable ⇒ securely stored with protected access! (with authorization)

  1 passwords, PINs, (cypher) keys, certificates . . . = data/software tokens – easily changeable (in case of leak) ⇒ suitable for remote (over network) auth., reusable or also one-time ⇒ different, randomly generated, used to protect access to 2

  2 auth. keys, ID cards (e.g. SIMs), auth. tokens (e.g. USB) . . . = devices/hardware tokens – worse changeable or copyable ⇒ securely store 1 (keys, certificates) and generate random 1 (passwords, keys), or remotely receive 1 (passwords, PINs)

  3 fingerprints, face, voice, eye or blood parts . . . = bio-markers – not changeable! ⇒ suitable for local use only, backed by 1, used to protect access to 2
- multi-factor = based on/using more factors simultaneously, usually 1/3 and 2 – e.g. password and mobile device (SIM, itself protected by password or fingeprint), Authenticator apps/services

[1] **1**–2, 3–7, [4] **1.10**, **9**–10, [5] 2

Monitoring and logging

- of activities of users: login/logout, data manipulation (final state), network communication (control info), bad usage and attack to the system (violating security) etc. – for solving security incidents
- of (automatic) system maintenance: hardware and software control and management (e.g. updates) – for possibility of reverting changes
- of system malfunction: hardware, network, programs (erroneous, malicious) – for error/vulnerability exploit or attack identification and recovery

Securing system (1)

- ✓ ordinary user account for „ordinary work“, administrator account only for system maintenance and management – prevention from „bad“ actions of users or malicious software
- ✓ appropriate (right $\rightarrow$ least) privilegies – to users, system parts, programs and data (prevention from harm)
- ✓ suitable (strong) authentication method: bio-markers (not changeable) or PINs $<$ good/strong passwords (= long, different category chars, random sequences) $<$ (cypher) keys and certificates $<$ authentication keys/tokens and ID cards $<$ multi-factor authentication
  - case study: Apple iCloud („Celebgate“, 2014): private data breach (photos of celebrities), weak passwords, no multi-factor authentication
- ✓ sensitive data encryption – authenticated access!

Securing system (2)

- ✓ software maintenance („care“):
  - + up-to-date system and software (free of known flaws/bugs) – regular/automatic updates
  - + avoiding use of „unknown, suspicious“ software (e.g. „free of use“) or hardware (e.g. found USB drives) – using trusted sources only
  - + anti-malware (antivirus) software – prevention from exploiting flaws and running malware
- ✓ monitoring and logging system and user activities, regular data backup (better several at geographicaly different locations)
- ✓ (last but not least) security education of users – training to awareness

[3] **10**, [5] 7–**9**, [6] **5**, **6.2**, 6.9

Software and hardware vulnerabilities (1)

= flaws/errors that render software and hardware vulnerable to security violations

- software/program flaws $\sim$ bugs = results of programming mistakes, errors, „optimizations“, bad decisions etc.
  - * buffer overflow = bad usage of memory, access „out of“ allocated memory
  - * race condition = wrong order of actions in program's multiple threads of execution running in parallel (= parallel program)
  - * (possibility of) code injection = absent validation of user input data enabling entering/manipulating program code, e.g. SQL injection or cross-site scripting (XSS) in websites which use databases
  - – zero-day = not broadly known with no fix available yet
- CVE (Common Vulnerabilities and Exposures): list of publicly known vulnerabilities and exposures (CVE ID and severity score)

[1] 3, **4**, 5, **6–7**, 13–14, [5] 24

Software and hardware vulnerabilities (2)

- hardware flaws (errors) = results of design or manufacturing failures, „optimizations“, bad decisions etc.
  - more difficult to patch/fix, by software – persistent, widespread
  - Spectre & Meltdown (2018) – CPU side-channel data leaks
  - Rowhammer (2014) – RAM data corruption
  - TPM, Intel ME, Wi-Fi, Bluetooth etc. device vulnerabilities – possibility of (side-channel) extract crypto secrets, backdoors, encryption flaws ...

[1] **III**

Malicious software (1)

- exploit = action, program or data to take advantage of software or hardware vulnerabilities to violate security (e.g. privilege escalation)

= malware = software to harm system (function, data, security) or human user (property, privacy)
  - using exploits to install into the system and spread
  - spread by „free of use“ software, emails (attachments), malicious websites ...
  - polymorphic = modifies itself to evade detection
  - botnet = for remote control of system (= bot) for network attacks
  * virus – embedded into programs/data, spread upon execution by user
  * worm – network spread by vulnerabilities without user action
  * Trojan Horse – disguised as legitimate, not spreading, for remote control or malware install
  * spyware – „attractive“, monitoring activity, gathering info for expolit or other's utility
  * adware – ad display to generate revenue, tracking, may install other malware
  * ransomware – data lock and demanding payment to unlock
  * keylogger – keystroke recording, capturing sensitive info
  * bootkit/rootkit/firmware – hiding as part of system to gain persistence and avoid detection, e.g. BIOS/UEFI, boot data
  * ...

[3] **5**, [4] **1.11**, [7] 4.2–4.4

Malicious software (2)

- case studies:
  - ILOVEYOU (2000): virus in email attachments (mils. computers)
  - WannaCry (2017): ransomware worm ($\sim$200 thous. computers), MS Windows SMB vuln.
  - Equifax (2017): personal and financial data breach ($\sim$150 mil. Americans), web vuln.
  - SolarWinds (2020): malware in software update used in gov. and corp. networks
  - Zeus: banking Trojan keylogger for capturing banking details
  - Emotet (2021): banking Trojan $\rightarrow$ malware-as-a-service
  - Mirai botnet (2016), Storm worm (2007), FinSpy spyware, Fireball adware, ...

- malicious hardware: keylogger, skimming device (capturing auth. info, e.g. camera, fake keypad), forged („rogue“) network device (disguising as legitimate, e.g. Wi-Fi AP) etc.
  - case studies: BadUSB (keystroke injection), Supermicro servers (2018, spy chips), ATM Skimmer (2018–2022)

Securing software

= design, development and maintenance with respect to security measures:
- \+ least (just enough) privilegies
- \+ proper authentication/access control
- \+ correct usage of system resources (CPU, memory, storage etc.)
- \+ user, data and system protection
- ∗ e.g. „sandbox“ mode = restricted privilegies and enforced protection

✓ best (secure) programming/development practices to prevent flaws/errors:
- \+ „higher“ language – with automatic memory management and strict code rules (Java, C#, Python, Rust)
- \+ testing and code review – code scan for vulnerabilities, functional, penetration tests, bug bounty programs
- \+ input validation and sanitization
- \+ fixing bugs in releases
- ! ignoring can be criminal

PART II.

Computer networks security

PART III.

Cybersecurity and Cybercrime