Network protocols security and other vulnerabilities

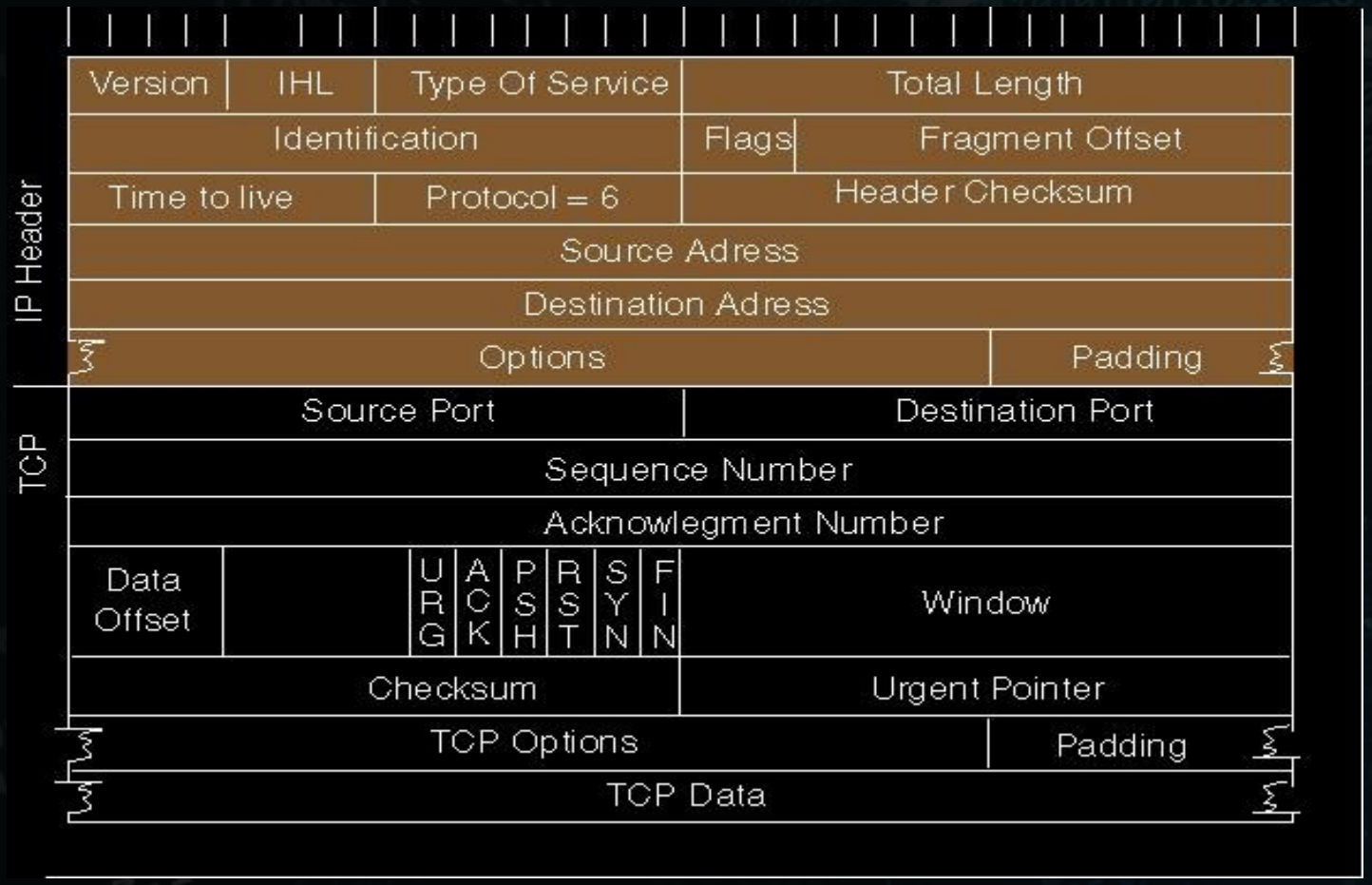
Marek Zachara
http://marek.zachara.name

Benefits of exploiting network vulner.

- Eavesdropping on communication (credentials are the most valuable spoils)
- Modification of the data in-transit
- Denial of Service and session hijacking
- Hiding of the transmission/data source
- A port of a multi-stage attack
- Challenge (?)

TCP implies trust of the incoming packet data

- Session packets are identified by Source/Destination IP + ISN



Access to the communication offer various possibilities

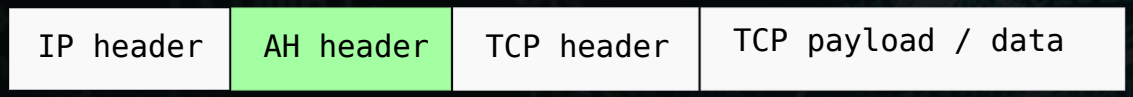
- Eavesdropping on the packets
- Hijacking the session (IP + ISN)
  - Injection of data
  - Connection termination (RST)

'Blind' attacks

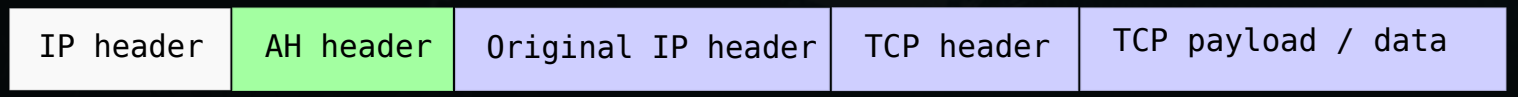
- There is only 2^{32} values of ISN
- Knowing SRC i DST IP it is possible to generate enough packets to inject data into the communication

Internet Protocol Security (ipsec)

- Utilizes encryption on a packet level
- Offers two modes of operation: transport



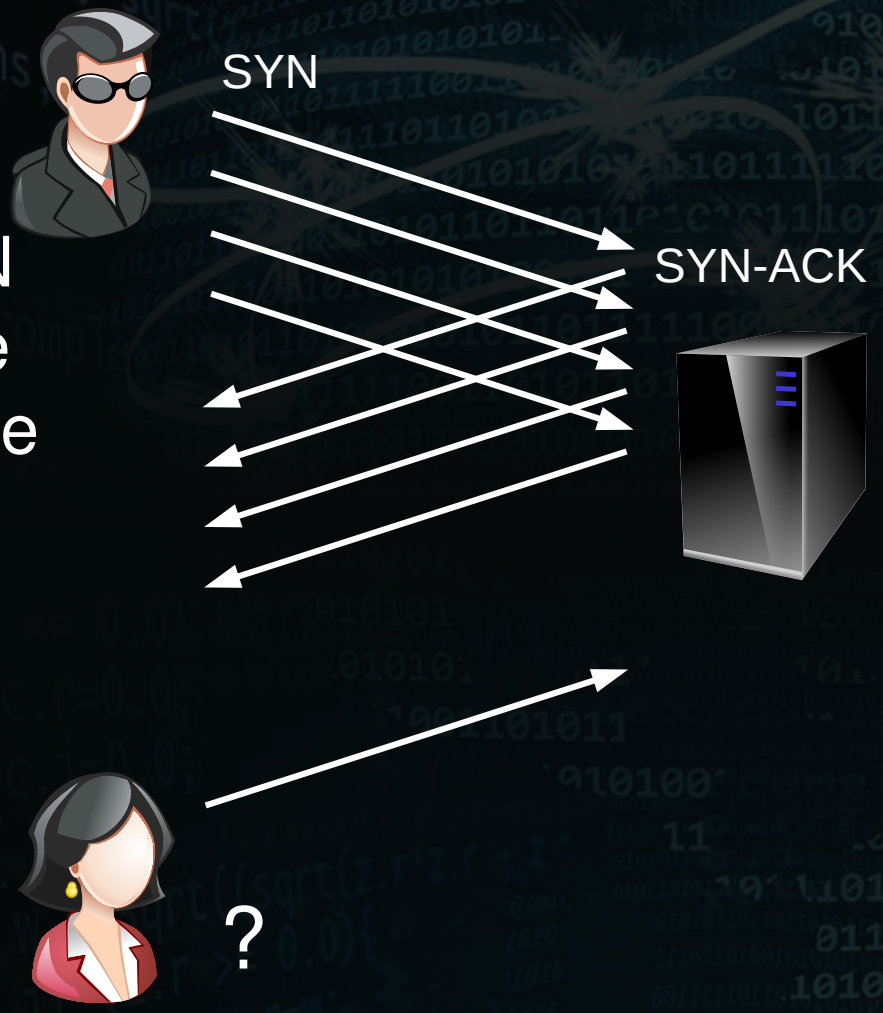
- Tunnel



- Encryption and signing of the packets
- Device support is required for IPv6, optional for IPv4

Syn Flood

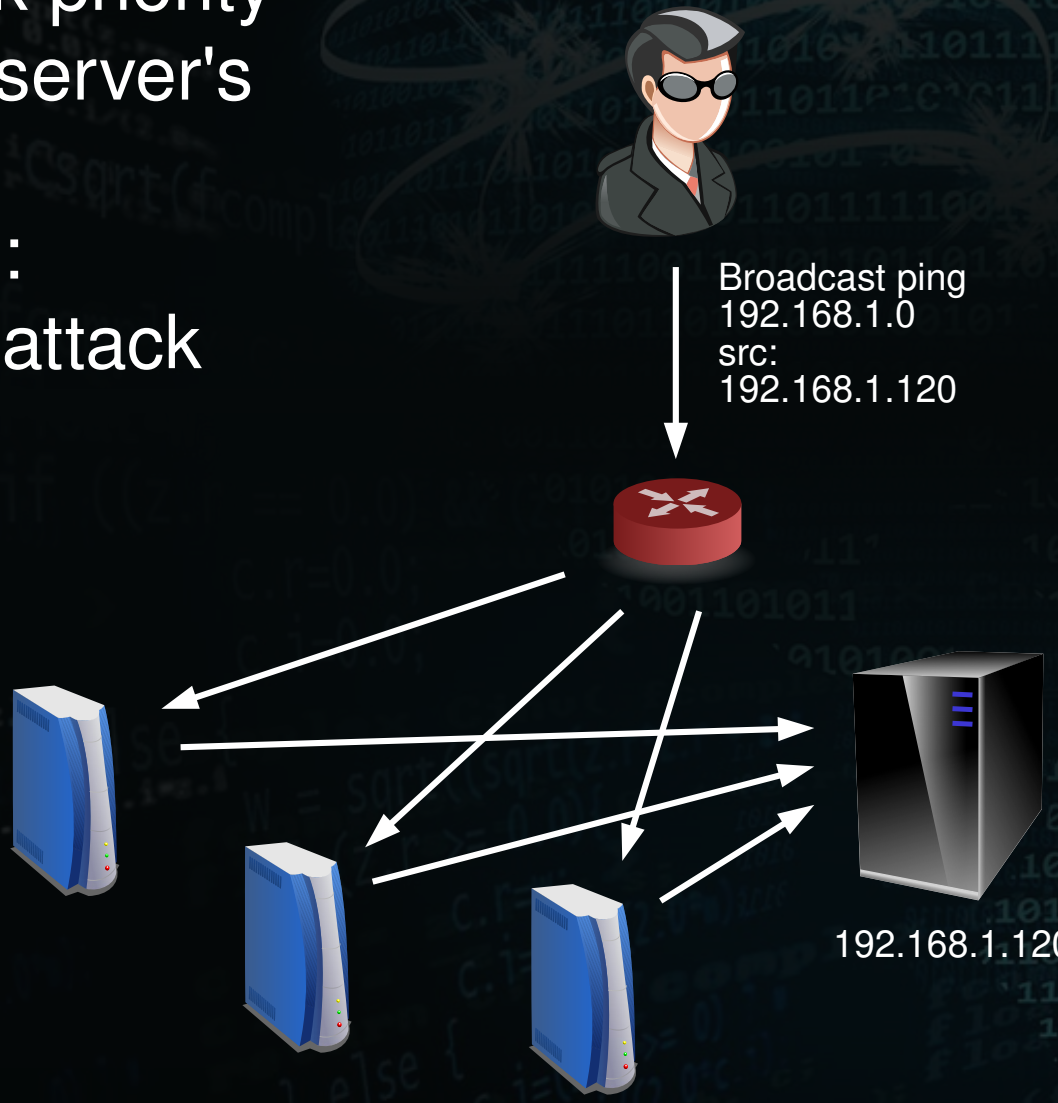
- TCP connection starts with a 3-way handshake
- Sending a lot of SYN packets can saturate the connection queue
- Attack can be conducted simultaneously from different machines



A possible solution: store SYN data in a „SYN Cookie” and put it aside until ACK arrives

Ping Flood

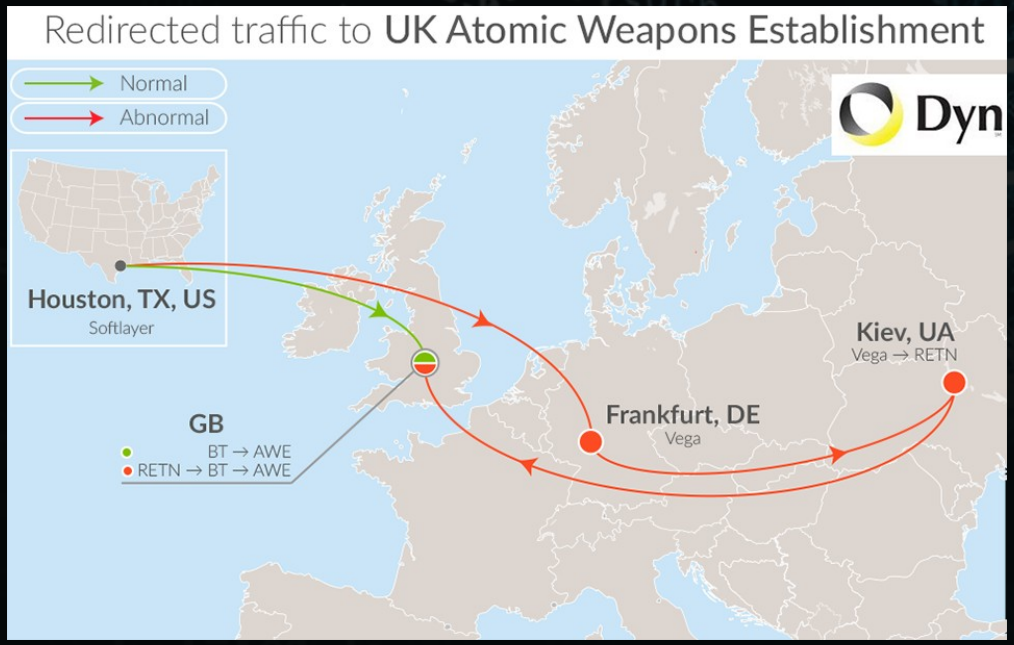
- ICMP packets usually have high network priority
- Targets the server's bandwidth
- Modification: a “SMURF” attack

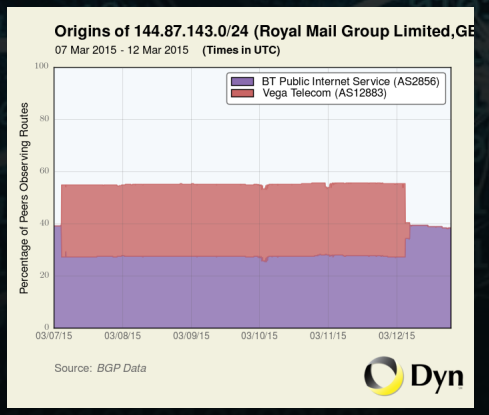


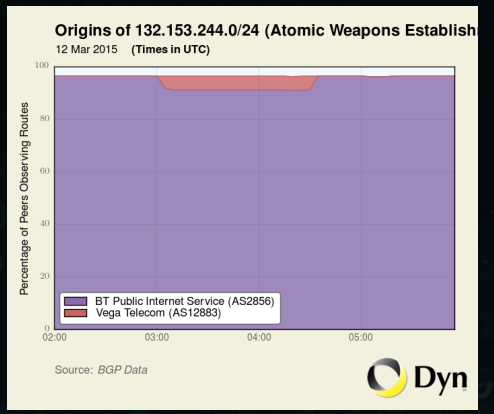
BGP Poisoning

- Border Gateway Protocol is used to determine the best routes for packets between ASs
- Announcement of a false/nonexistent route may lead to traffic loss - “black hole”
- Routing loops
- Announcement of a specific route as preferred will result in redirecting the traffic and allows for eavesdropping
- “Link flapping” degrades the quality of communication.

Example of a BGP based attack







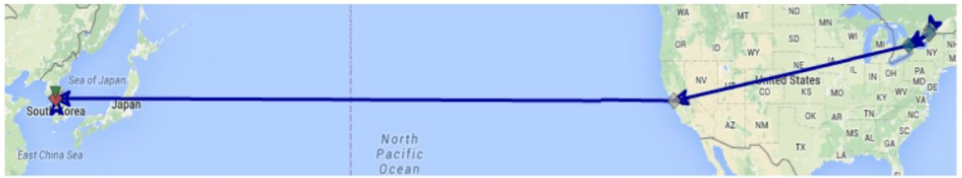


Figure 2a: The normal and shortest route from Canada to Korea before the hijack.

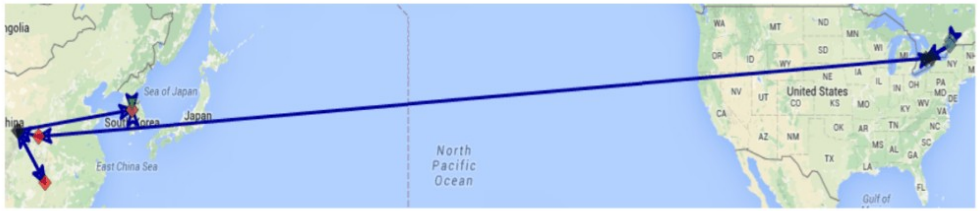


Figure 2b: The hijacked route through the CT PoP in Maryland – a long way from Canada to Korea.

source: http://research.dyn.com/2015/03/uk-traffic-diverted-ukraine/
https://www.cyberdefensemagazine.com/experts-detailed-how-china-telecom-used-bgp-hijacking-to-redirect-traffic-worldwide/

Wi-Fi security

- Increasingly popular, both private and in offices
- Users (and many administrators alike) are not aware of the risks involved
  - The communication layer is not under control anymore (the signal is accessible from various places)
- Possible attack vectors:
  - Access Points
  - Users

Sample attacks against Wi-Fi networks

- Eavesdropping on unencrypted connections
- Collection of data for cryptoanalysis
- MAC spoofing

Attacks targeting users

- Fake Access Point / impersonation
  - Retrieval of authentication data
  - “Man in the Middle”

DoS attack against an AP

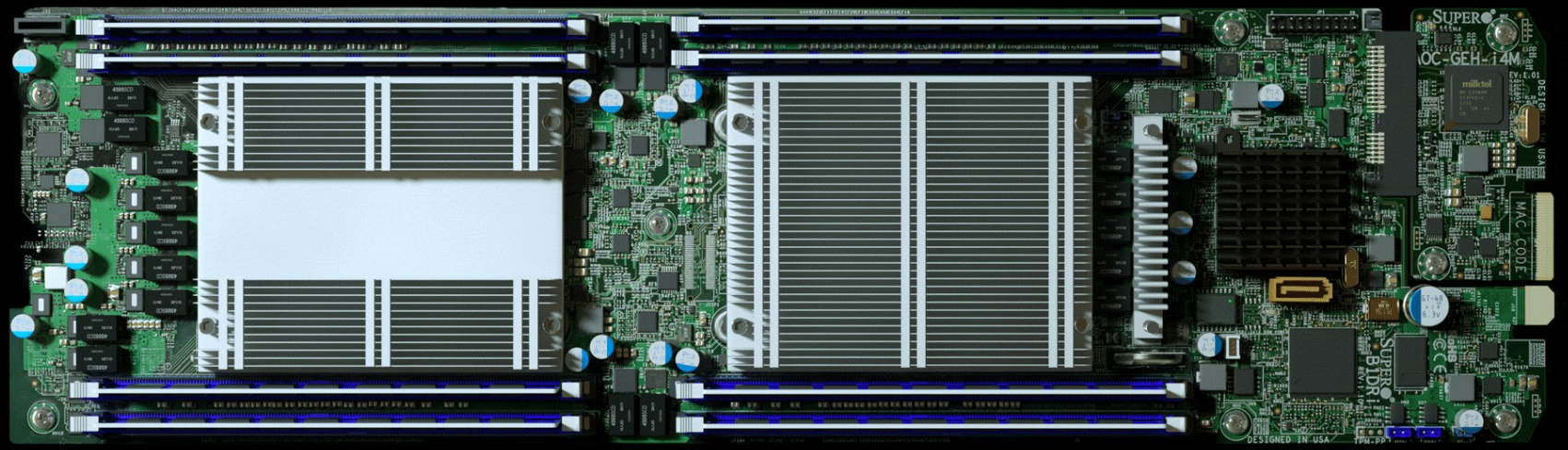
- Connection initiation and key generation are resource intensive

Wi-Fi protocols

- WEP – oldest one, currently obsolete, uses RC 4 key (easy to decipher)
  There are widely available methods for retrieval/decryption of the keys (e.g. aircrack)
- WPA – designed as a 'drop-in' upgrade of WEP, uses TKIP which has a known exploit
- WPA2 – currently considered secure (when using CCMP/AES).
- WPA3 – in the process of being rolled out

Other vulnerabilities

Hidden circuits



- Large and distributed supply chains make it difficult to control all the components
- One of the supplier may include some undesired hardware or functionality
- Mostly state-sponsored

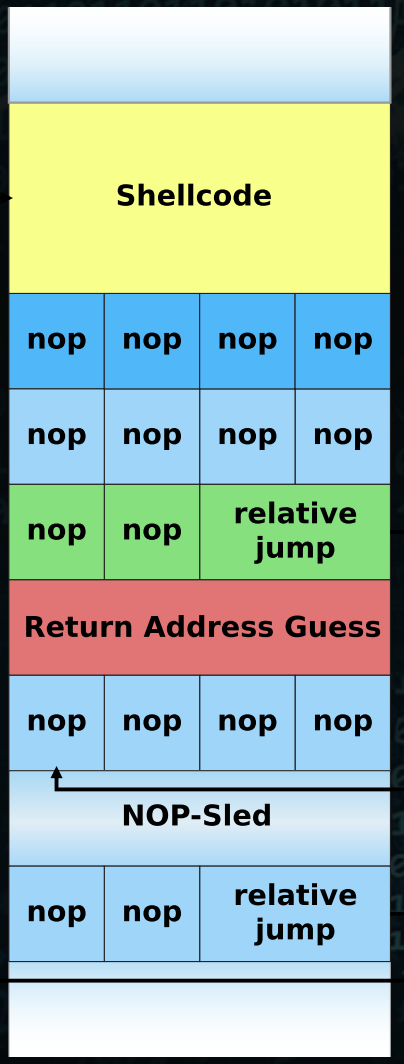
https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies

Meltdown & Spectre

- Serious design flaw in modern CPUs
- Allows one program to read memory belonging to another process
- Can lead to leaking of sensitive information
  - Even if the target program is considered safe
- Only software patches available
  - OS-level
  - Slowdown

Buffer overflow

- Can occur in applications written in languages that do not provide memory access controll (e.g. C/C++)
- Primary attack vectors:
  - Overwriting the stack
  - Overwriting the heap
- A sample of vulnerable code:
  char buffer[100];
  gets (buffer);



Results of buffer overflow

- Unpredictable – up to arbitrary code execution with root's privileges

Possible testing methods

- Limited. Usually fuzzing and long data inputs are used

Means of protection

- GRE security / PAX
- “Canaries”

Integer overflow

- An integer can only have a limited set of values (e.g. 0 – 4,294,967,296)
- Negative values are usually denoted by the most significant bit (e.g. U2)
- Direct casts can result in various problems
  - unsigned int → int; int → char
- As well as arithmetic operations (2 ^ 31 * 4)
- Potential risks
  - Malfunctions, DoS
  - Privilege elevation

Hazards (race conditions)

- The risk is related to several threads accessing the same region of memory (variable, other resources)
- Example:
  - Storing of pre-calculated report data in an application bean
  - AJAX – various parts of the page code can operate on the same dataset
- Warning: over-locking of the resource can lead to deadlocks (especially with intensive AJAX usage)

Faulty error handling

- Ignoring errors returned by methods/functions (e.g. 'catch all' – and ignore...)
- General (not specific) exception handlers
- Throwing the exceptions 'up' until they reach the UI (or as HTML comment)

Error logging

- Can it be done wrong?
- Yes: log forging

Thank you for your attention

Questions?

Wykorzystane materiały które nie miały wcześniej podanego źródła:

- Clipart – openclipart.org
- NOP sled, IPSec – Wikimedia Commons

Elementy licencjonowane (royalty-free), nie mogą być wykorzystywane oddzielnie:

- Tło prezentacji, awatary postaci