IT/ICT bezpečnost

Ing. Michal Dočekal

Neunikly vaše přihlašovací údaje?

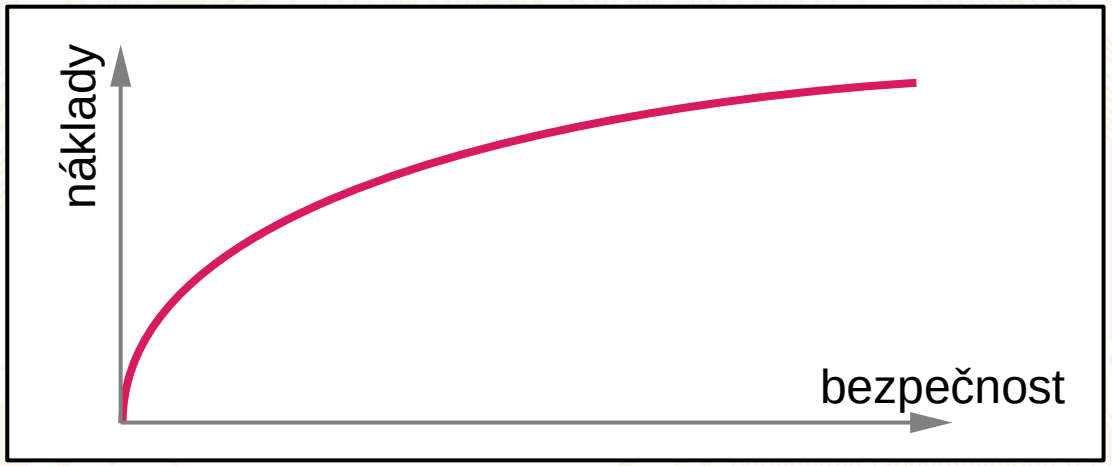
https://haveibeenpwned.com/

IT/ICT bezpečnost

- různé názvy: počítačová bezpečnost, IT bezpečnost, ICT bezpečnost, kybernetická bezpečnost, CyberSecurity, atd.
- různé definice bezpečnosti

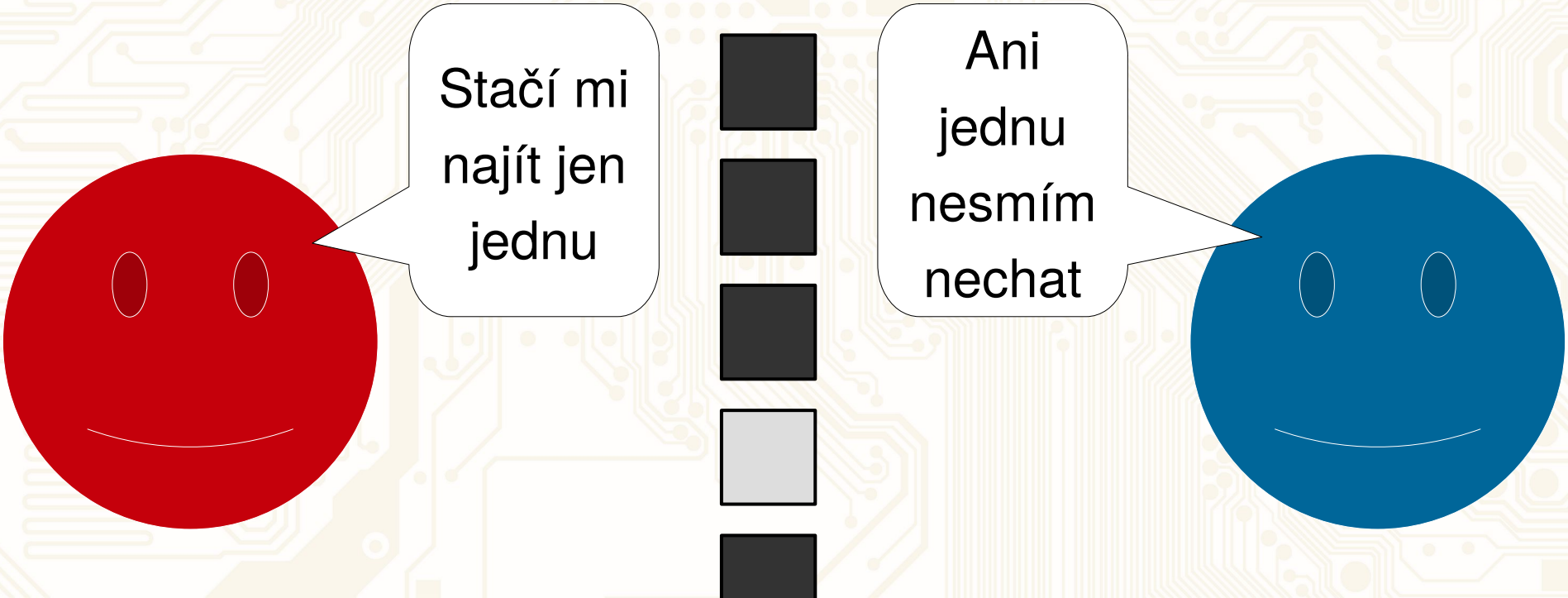
Bezpečnost

- bezpečnost je **kompromis** (trade-off) mezi *náklady* a *dosaženou* úrovní bezpečnosti
- do bezpečnosti lze investovat neomezené množství prostředků, avšak **100% bezpečnosti nelze nikdy dosáhnout**



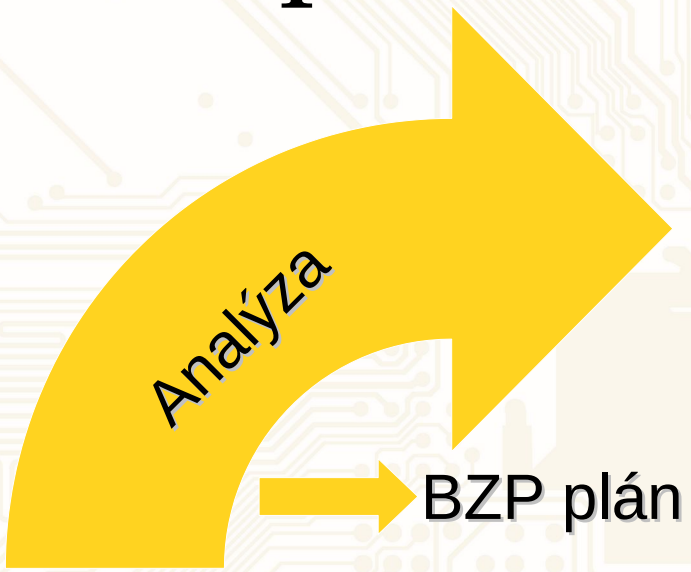
Postavení útočníka a obránce

- útočník je vždy ve výhodě, neboť mu stačí najít jedinou skulinku k průniku do systému
- obránce musí utěsnit všechny možné skulinky a žádnou neopomenout

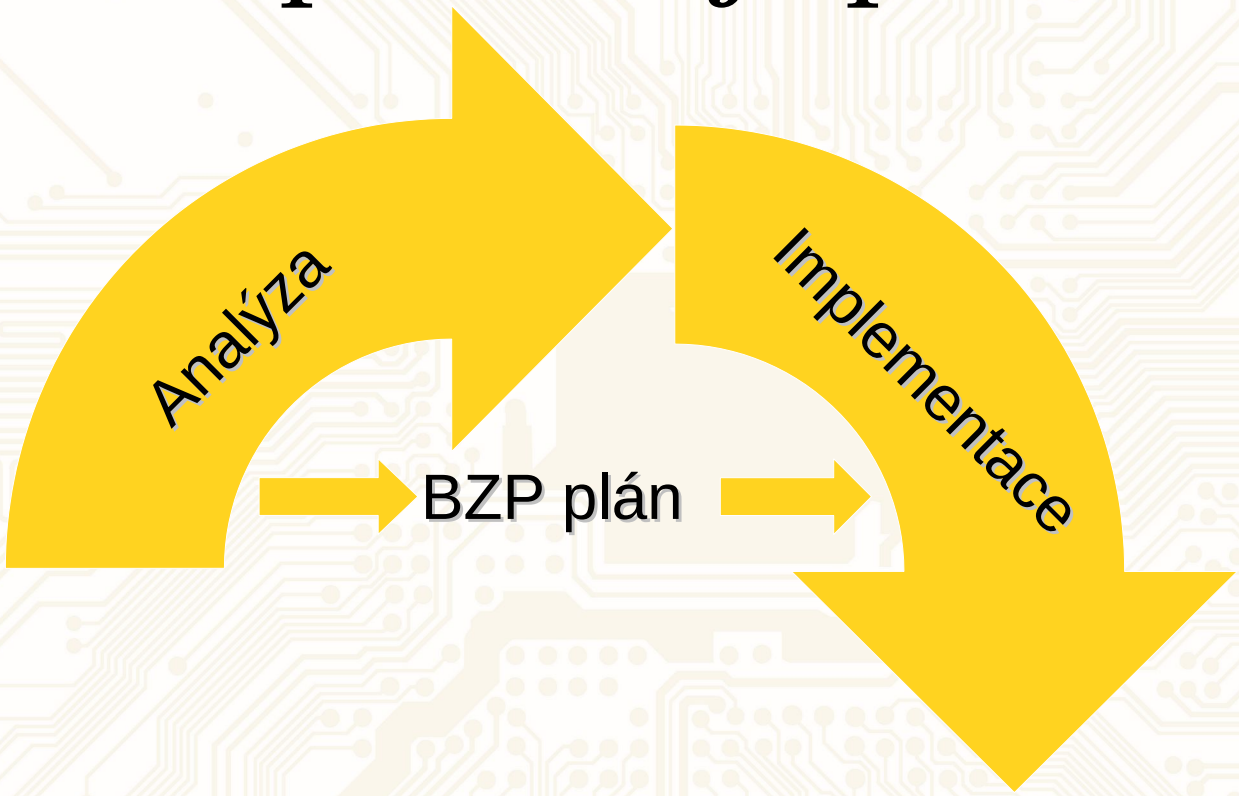


Bezpečnost je proces...

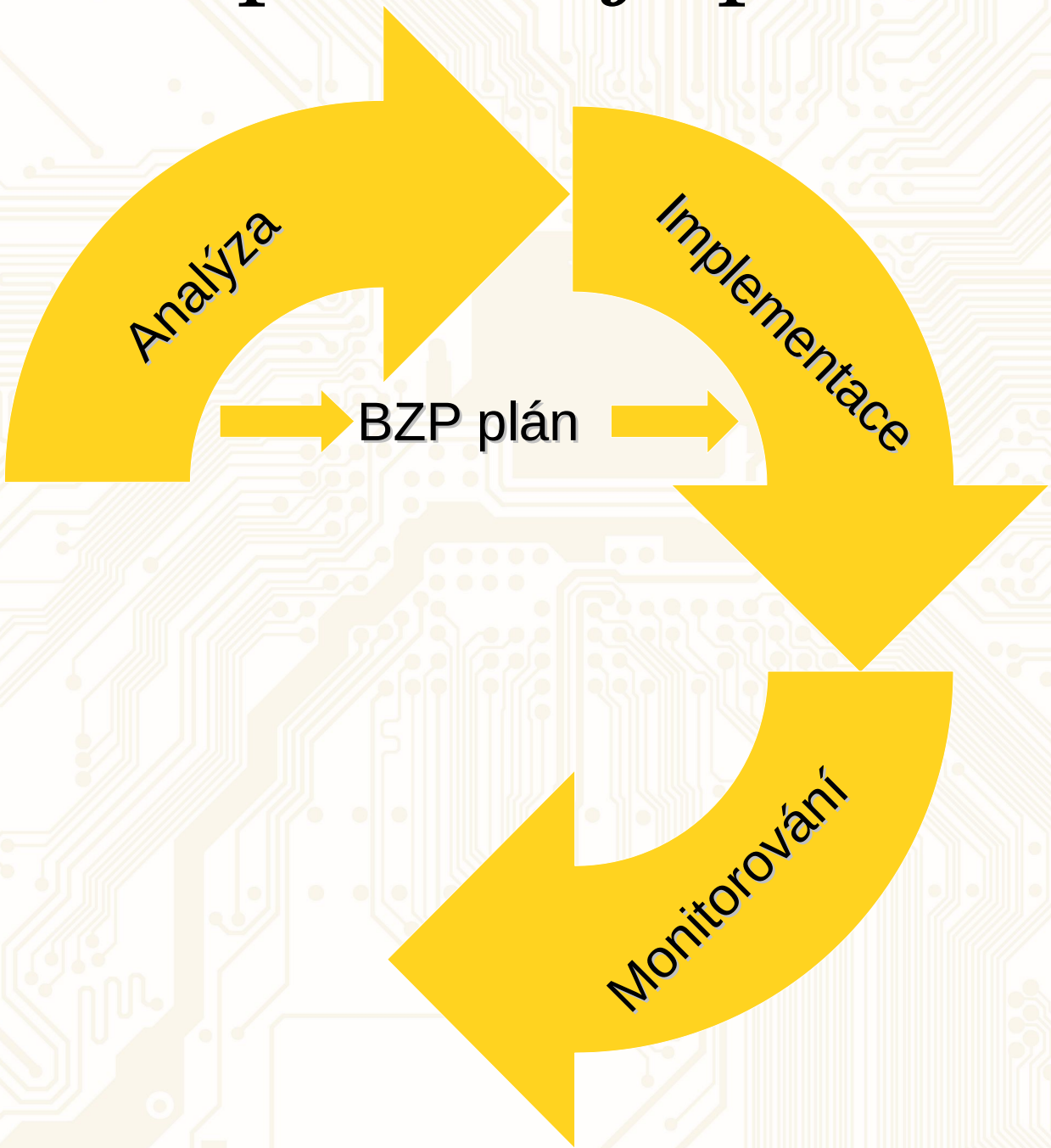
Bezpečnost je proces...



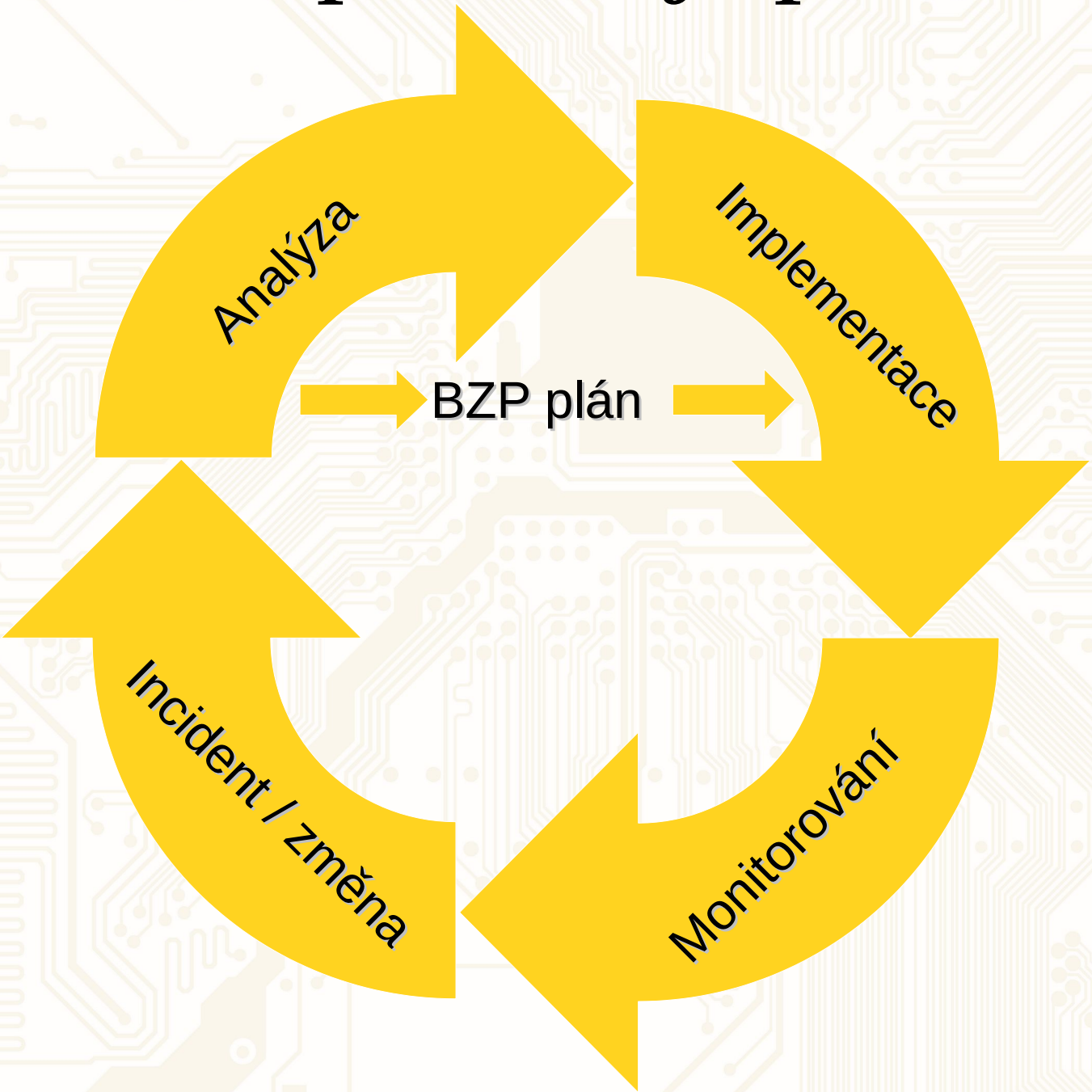
Bezpečnost je proces...



Bezpečnost je proces...



Bezpečnost je proces...



Analýza: 3 zásadní otázky

- co je třeba chránit
  - aktiva
- před kým nebo čím to chráníme
  - hrozby
- jak to ochránit
  - optimální poměr vynaložených nákladů a dosažené BZP

Analýza

- **Aktivum** – co chráním (data, systém, služba, ...)
- **Zranitelnost** – slabé místo aktiva
- **Hrozba** – příčina možného narušení bezpečnosti (využití zranitelnosti)
- **Dopad** – následky daného narušení bezpečnosti
- **Riziko** – kombinace dopadu a pravděpodobnosti

Co chráníme?

- **informace**
  - ***utajení***: ochrana před přečtením/zkopírováním
    - třeba chránit i jednotlivá drobná data, která jsou sama o sobě neškodná, ale dohromady mohou být použita k získání jiných důvěrných informací
  - ***integrita***: ochrana před ztrátou, ale i změnou nebo poškozením dat
    - účetní záznamy, zálohy, časové údaje o souborech (vznik, modifikace, atd.), dokumentace

Co chráníme?

- **systémy a služby**:
  - systém je bezpečný, chová-li se dle očekávání
  - služby: business procesy i síťové služby
  - ***dostupnost***: služba/systém funguje a je dostupná pro ty, pro které má být dostupná
    - i nedostupná služba může služby ohrozit podobně jako smazání dat

Co chráníme?

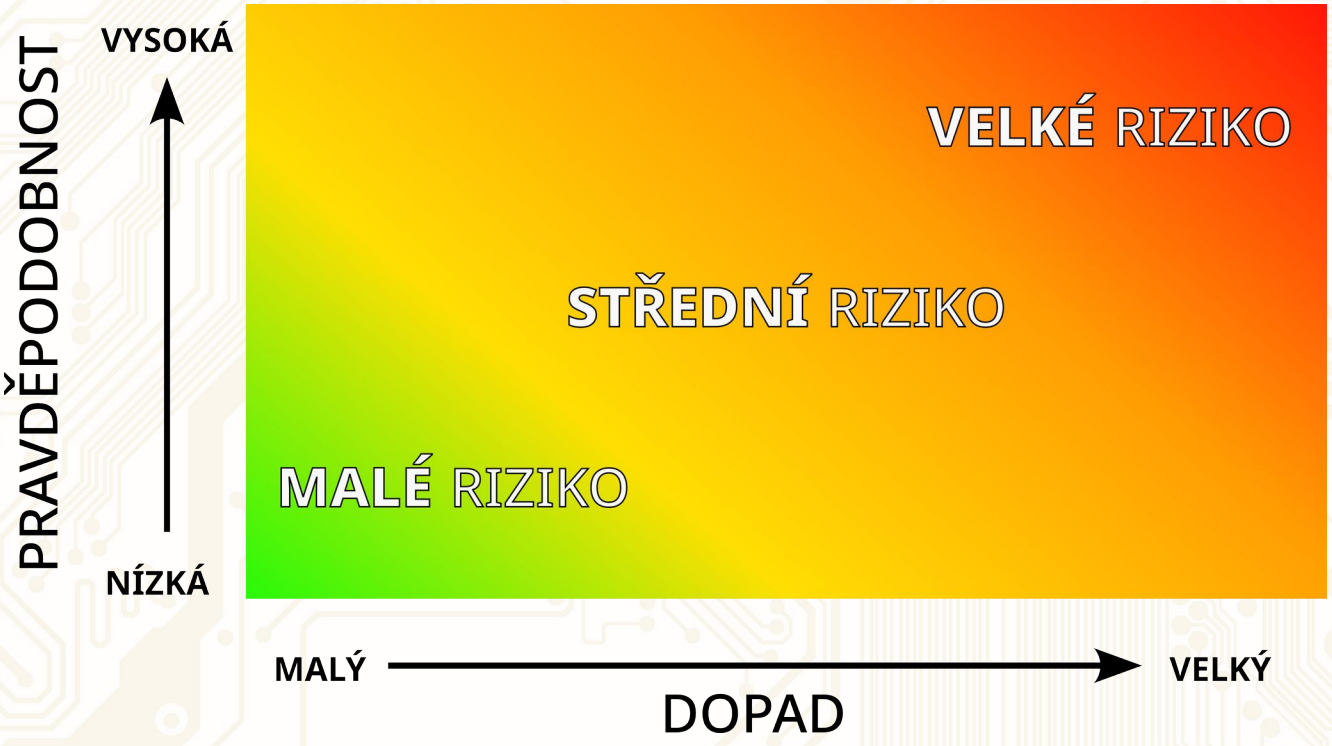
- **majetek**:
  - *hmotný majetek*
    - počítače, média, manuály, instalační média komerčního SW
    - komunikační vybavení a kabeláž
    - záznamy (osobní, účetní, smlouvy, atd.), dokumenty, atd.

Co chráníme?

- **majetek**:
  - *nehmotný majetek*
    - bezpečnost a zdraví lidí, soukromí uživatelů, hesla
    - data, informace, know how, image, pověst
    - agenda zákazníků a klientů
    - patenty, ochranné známky, autorská práva, patenty, apod.
    - konfigurace/nastavení systémů

Hodnocení hrozeb

- pro každou **hrozbu** určujeme **riziko**
- **riziko** je kombinace **dopadu** hrozby **a pravděpodobnosti**, že k ní dojde



Optimální zabezpečení

- smyslem je dosáhnout optimálního kompromisu mezi vynaloženými náklady a dosaženou BZP
- **riziko nikdy není nulové!**
  - selhat může jak zabezpečení, tak zabezpečení zabezpečení
- cena **ztráty** vs cena **prevence**

Cena ztráty a prevence

- **cena ztráty**:
  - jednoduše: cena za opravu
  - komplexně: cena za opravu + náklady (vč. ušlé příležitosti) za dobu, kdy je služba/zařízení offline, náklady na dodatečné školení a cenu dalších postupů a nákladů v souvislosti se ztrátou

Cena ztráty a prevence

- **cena prevence**:
  - cena UPS vs cena ztráty za krátkodobý výpadek
  - náklady na BZP mohou být nejen finanční, ale i snížení efektivity práce, popřípadě prostoje vlivem BZP
    - implementace MFA a zaměstnanec zapomene druhý faktor doma, nebo se přestane odhlašovat(!)
  - některá BZP opatření mohou vést i ke snížení BZP
    - stále měním heslo? Aktuální si nalepím na monitor...

Kategorie škod

- nedostupnost na kratší dobu
- nedostupnost na delší dobu
- náhodná částečná ztráta nebo zničení
- záměrná částečná ztráta nebo zničení
- trvalá ztráta nebo zničení
- neautorizované odhalení uvnitř organizace
- neautorizované odhalení vně organizace

Implementace plánu BZP

- implementace bezpečnostního SW/HW
- úprava konfigurace stávajících systémů a aplikací
- implementace systému řízení přístupu ke zdrojům, přiřazení rolí, práv a odpovědnosti uživatelům
- **úprava procesů a chování uživatelů**
- **školení uživatelů**
  - rozšiřování znalostí uživatelů o BZP i BZP postupech
  - testování uživatelů(!)

Řízení přístupu

- **autentizace** – ověření identity uživatele
- **autorizace** – ověření, že daný uživatel má mít přístup k danému zdroji
- **privilegovaný přístup** – správa chování, práv a monitorování činnosti administrátorů
  - omezení administrátorů, aby jeden nespokojený admin nepoložil firmu

SW/HW nástroje bezpečnosti

- logování + sběr a analýza dat z logů
- firewally, antiviry/anti-malware
- nástroje pro vlastní šifrování a utajování dat
- IDS/IPS
- nástroje pro penetrační testování
- manažerské systémy usnadňující analýzu a tvorbu plánu BZP, systémy pro správu aktiv, apod.

Monitorování

- kontrola implementace (bylo implementováno a nastaveno všechno správně?)
- monitorování BZP
  - prověřování logů, hledání známek narušení bezpečnosti
  - sledování odvětví IT bezpečnosti (nové hrozby, nové útoky, nové metody zabezpečení, atd.)
- prověřování účinnosti aktuálních BZP opatření
  - penetrační testování, audity, prověřování techniky i lidského faktoru
- prověřování provádění i obnovitelnosti záloh!

Incident / změna

- v reakci na incident nebo významnou událost či změnu v oblasti bezpečnosti se proces bezpečnosti opakuje (analýza, úprava plánu BZP, implementace změn, monitorování)
- v případě incidentu je třeba prověřit:
  - jak se útočník dostal do systému
  - k čemu měl útočník přístup
  - co prováděl
  - k jakému dalšímu narušení bezpečnosti došlo

Typologie incidentů

- kybernetický útok
- havárie (selhání zařízení)
- živelná pohroma (i požár či vytopení)
- oprávněný uživatel/admin, který udělal chybu
- oprávněný uživatel/admin, který úmyslně škodí

Dojde-li k incidentu

Nepanikařte!

Dojde-li k incidentu

- **zachovejte chladnou hlavu** a **rozmyslete** další kroky
- o čas nejde, neboť k prolomení BZP již došlo a útočníci měli čas dosáhnout svých cílů
- dodatečné škody za čas strávený zvážením dalšího postupu jsou většinou minimální
- panika vede ke špatným rozhodnutím, které mohou způsobit větší škody

Disaster Recovery Plan

- v případě, že jste jako součást BZP plánu sestavili i DRP pro různé scénáře a některý ze scénářů nastal, můžete postupovat podle DRP
- výhoda: na nic důležitého nezapomenete a DRP plán sestavujete v období klidu, takže máte čas vše rozmyslet

Obecná reakce na BZP incident

- analýza útoku a jeho dopadů, identifikace pachatele nebo pachatelů a slabých míst v zabezpečení
- zamezení přístupu útočníkům
  - nemělo by být nutně první reakcí na útok(!), pak toho útoku a útočnících zjistíte méně
- informování relevantních subjektů (uživatelé, CERT/CSIRT, ÚOOU, policie, apod.)
- obnova bezpečného stavu
  + revize a úprava BZP plánu

Obnova bezpečného stavu

- kompromitované systémy a systémy podezřelé z kompromitování se obvykle znovu kompleteně reinstalují (nelze jim již věřit) a data se obnovují ze záloh
  - to předpokládá, že zálohy nebyly útočníkem pozměněny nebo se nezálohovala útočníkova aktivita
- revize BZP plánu, implementace změn, pečlivý monitoring, atd.

Kybernetické hrozby

- malware: malicious software (škodlivý software)
  - **virus**: autonomně se šířící malware, typicky infikuje spustitelné soubory, jejichž spuštění pak aktivuje virus
  - **boot virus**: virus přežívající v boot sektoru média, dnes méně rozšířené
  - **červ**: malware se schopností šířit se a napadat systémy na síti (Morrisův červ, Stoned, Stuxnet, atd.)
  - **trojský kůň (trojan)**: malware, který je aktivován uživatelem samotným (sociální inženýrství)

Kybernetické hrozby

- malware: malicious software (škodlivý software)
  - **spyware**: software sbírá data bez vědomí uživatele a odesílá je zpravidla přes síť na sběrné místo útočníka
  - **ransomware**: software narušující fungování systému nebo integritu dat a požadující „výkupné“ za obnovu funkčního stavu (zablokování počítače, zašifrování souborů, atd.)
  - **rootkit**: sada nástrojů pro ovládnutí napadeného systému, maskování vlastní přítomnosti a útočných aktivit (skrývání souborů, procesů, činnosti)
  - **keylogger**: software nebo hardware zaznamenávající stisky kláves pro odchycení hesel a jiných informací

Kybernetické hrozby

- malware: malicious software (škodlivý software)
  - **scareware**: software předstírající problémy (typicky vyskakovací okno), aby nabídl jejich řešení za peníze nebo stažení řešícího softwaru, což je ovšem malware (např. falešný antivirus)
  - **infostealer**: integrované a customizovatelné malware řešení určené ke krádeži citlivých dat (přihlašovací údaje, čísla platebních karet, ale i běžící programy, cookies nebo relační tokeny pro session stealing)
  - **MaaS**: malware jako služba: útočníci nabízí malware řešení jako infostealery za peníze s možností customizace a snadného použití

Kybernetické hrozby

- **spam** – masivně šířená nevyžádaná pošta či nevyžádané zprávy (sociální sítě, atd.)
  - zneužíváno pro reklamy, podvody, phishing a krádež citlivých dat, distribuci malwaru, atd.
  - založeno na tom, že i když většina lidí na spam nereaguje, několik málo obětí se chytne
  - e-mailové adresy uhodnuté, získávané přes web scraping nebo z uniklých databází uživatelů webových služeb, atd.

Kybernetické hrozby

- **sociální inženýrství** – manipulace oběti útočníkem ve snaze dosáhnout svých cílů
  - **phishing** – podvodné e-maily předstírající identitu nějakého důvěryhodného subjektu s cílem přimět uživatele předat osobní a přihlašovací údaje útočníkovi
    - **spear phishing**: zaměřuje se více individuálně, užívá personalizované zprávy s informacemi zjištěnými o oběti
    - **whaling**: zaměřuje se na výše postavené členy cílové organizace (typicky vedoucí pracovníci)

Kybernetické hrozby

- **sociální inženýrství**
  - **vydírání** (extortion) – útočník hrozí oběti buď smyšlenou nebo reálnou hrozbou a typicky nutí k zaplacení
    - **sexuální vydírání** (sextortion): útočníci hrozí zveřejněním intimních fotografií nebo videí (reálných či fiktivních), mohou vyžadovat peníze nebo pořízení dalších kompromitujících materiálů

Jak reagovat na vydírání

- na vydírání vůbec **nereagujte** – nemáte žádnou záruku, že vyděrač hrozbu přesto neuskuteční nebo nebude žádat o něco dalšího, jakmile bude vědět, že jste ochotni jeho požadavky plnit
- **sbírejte důkazy**, pořizujte hlasové a jiné záznamy, schovejte si maily, apod.
- vydírání i sexuální nátlak jsou **trestnými činy**, lze se obrátit na policii
- využívejte psychické i jiné podpory osob, kterým důvěrujete
- ideálně nepořizujte kompromitující materiály – co není pořízeno, nemůže uniknout
  - pokud je pořizovat přesto chcete, nezabírejte nic co umožňuje vaši snadnou identifikaci (obličej, znaménka, tetování, pozadí bytu, apod.)

Kybernetické hrozby

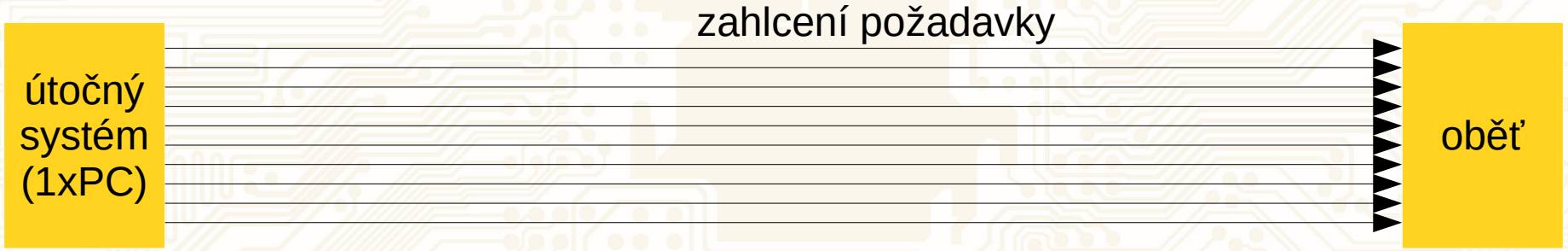
- **zombie** – nezabezpečený počítač částečně (neprivilegovaný přístup) nebo zcela (privilegovaný přístup) ovládnutý útočníkem
  - používají se pro páchání dalších útoků, nejen DDoS
- **botnet** – sada zombií ovládaná řídícím (C&C) serverem
  - pull systém – malware na zombii se připojí na řídící server a stáhne si, co má dělat

Kybernetické hrozby

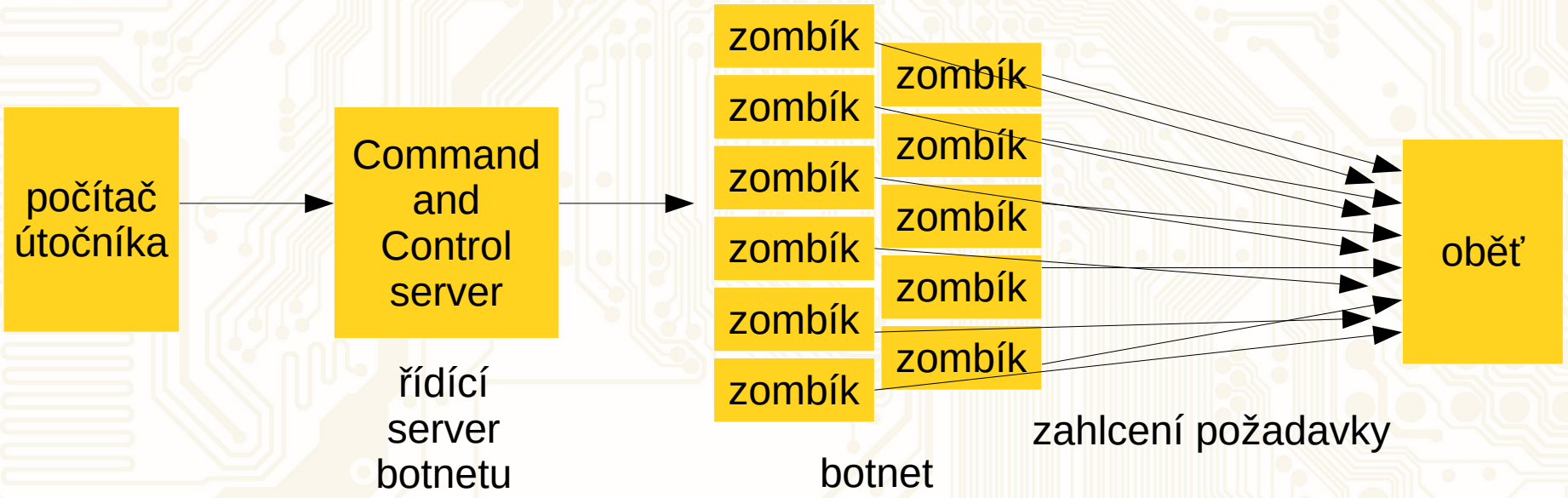
- DoS (**Denial of Service**) – útok na službu, který způsobí její nedostupnost
  - DoS může být způsoben přetížením vlivem přílišného množství požadavků, nebo využití zranitelnosti, která odstaví cílovou službu (služba spáchá neplatnou operaci a bude uknočena nebo se zasekne)
- DDoS (**Distributed Denial of Service**) – přetížení cílového systému (nebo sítě!) požadavky z mnoha zombií zpravidla zapojených do botnetu; jelikož zombie odesílají malé množství regulerních požadavků, těžko se odlišují od legitimního provozu
- **Botnet as a service**: některé botnety nabízí své služby za peníze

DoS versus DDoS

- **Denial of Service**

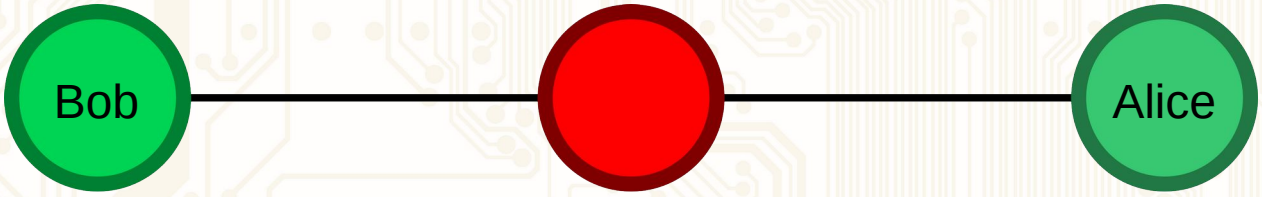


- **Distributed Denial of Service**



Kybernetické hrozby

- MITM: **M**an **I**n **T**he **M**iddle útok
  - jakýkoliv komunikační mezičlánek může odposlouchávat nebo dokonce vstupovat do nezabezpečené komunikace
  - s tím souvisejí různé oslabovací útoky na zabezpečení (downgrade verze TLS, SSL strip,...)
  - mnoho různých forem MITM útoku



MITM

Kybernetické hrozby

- chyby v SW poškozující data nebo ničící systém
  - útočník nebo malware mohou úmyslně ničit data
    - mohou ničit data i neúmyslně, když se spletou nebo je v malwaru chyba
  - takové chyby se ale mohou nacházet i v legitimním SW
  - dopad takových chyb může být vážný, zejména, pokud se na to hned nepřijde

Bumblebee a /usr

- projekt určený pro podporu hybridních grafických adaptérů pod Linuxem, kterému se podařilo v instalačním skriptu zanechat extra mezeru na řádku 351:

```
rm -rf /usr /lib/nvidia-current/xorg/xorg
```

Bumblebee a /usr

- projekt určený pro podporu hybridních grafických adaptérů pod Linuxem, kterému se podařilo v instalačním skriptu zanechat extra mezeru na řádku 351:

```
rm -rf /usr /lib/nvidia-current/xorg/xorg
```

mezera!

Bumblebee a /usr

- projekt určený pro podporu hybridních grafických adaptérů pod Linuxem, kterému se podařilo v instalačním skriptu zanechat extra mezeru na řádku 351:

```
rm -rf /usr
rm -rf /lib/nvidia-current/xorg/xorg
```

Bumblebee a /usr

- projekt určený pro podporu hybridních grafických adaptérů pod Linuxem, kterému se podařilo v instalačním skriptu zanechat extra mezeru na řádku 351:

```
rm -rf /usr
```

Bumblebee a /usr

- projekt určený pro podporu hybridních grafických adaptérů pod Linuxem, kterému se podařilo v instalačním skriptu zanechat extra mezeru na řádku 351:

  ```
  rm -rf /usr
  ```

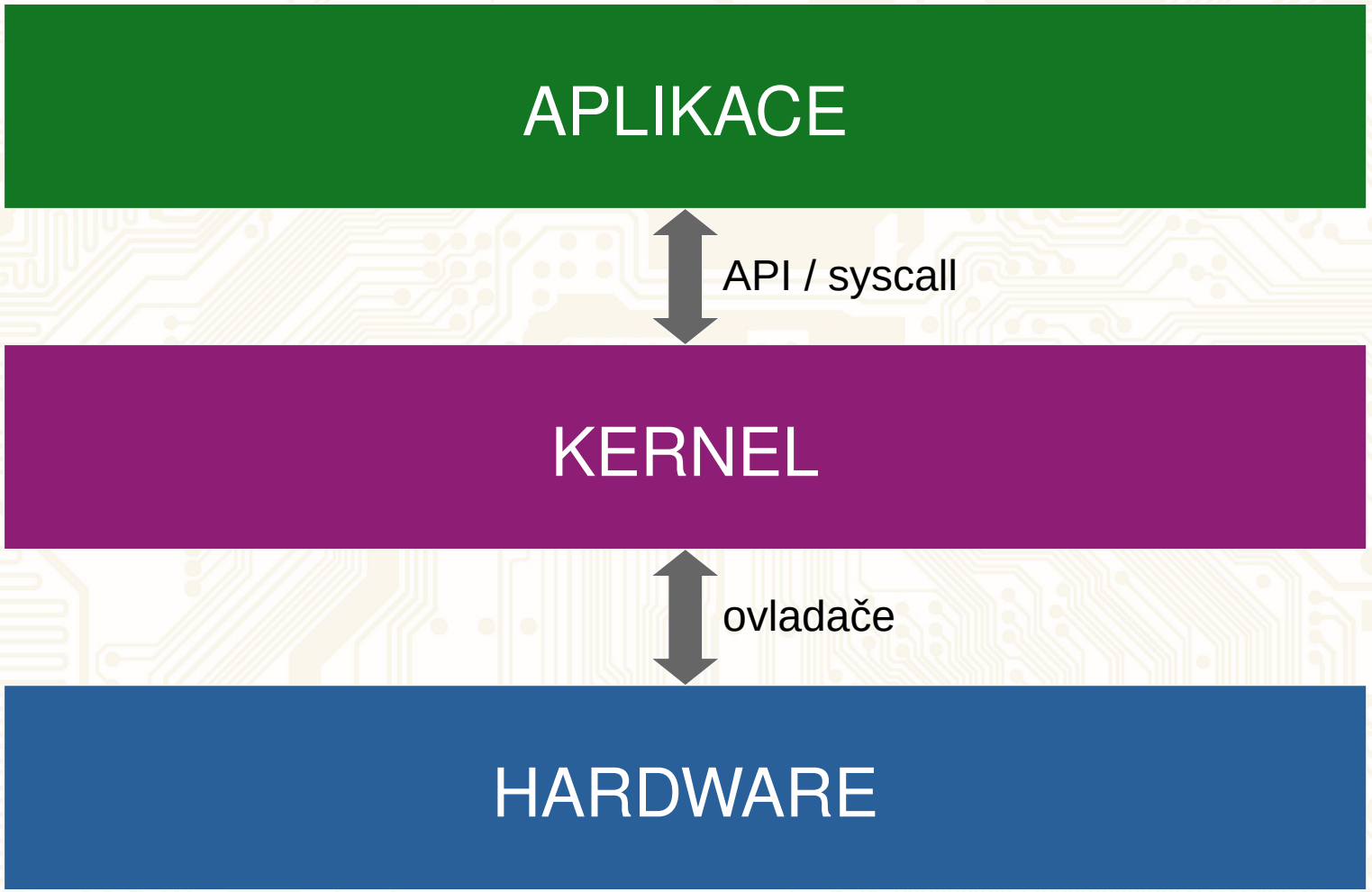
  smaž všechny spustitelné soubory a knihovny operačního systému

Kybernetické hrozby

- **silent data corruption** (tiché poškozování dat)
  - velice nebezpečné, protože trvá dlouho, než se na to přijde, a má to tendenci kontaminovat zálohy
- **data remanence**
  - data mohou zůstávat na médiu i po jejich vymazání, v horším případě i přepsání
- **data degradation** / **bit rot** (degradace zápisu)
  - technologie zápisu dat mohou časem, používáním i nepoužíváním akmulovat nekritické chyby, ze kterých se vyvinou fatální chyby čtení

Počítačová bezpečnost

Architektura SW, HW, OS



Protection rings (CPU)

nejméně privilegií

nejvíce privilegií

Protection rings (CPU)

aplikace

kernel

Úrovně přístupu v rámci OS

- **neprivilegovaný režim** – omezená práva
  - většina uživatelských aplikací
  - chybí přímý přístup k HW, procesy jsou izolované a nemohou zasahovat do procesů jiných uživatelů
  - systém přístupových práv
- **privilegovaný režim** – neomezená práva
  - kernel a služby operačního systému
  - přímý přístup k HW, CPU, paměti i všem procesům

Hrozby a úrovně přístupu v OS

- **útočník ovládne neprivilegovaný proces**
  - může přistupovat k datům vlastníka procesu(!) a k nechráněným datům jiných uživatelů(!)
  - nemůže přistupovat k HW a zdrojům jiných procesů
- **útočník ovládne privilegovaný proces**
  - může přistupovat ke všem datům v systému
  - může přistupovat k HW a OS
  - může skrývat svou přítomnost před uživateli a potenciálně i před OS a jeho službami (rootkity)

Zranitelnosti v SW

- **exploit**: programový prostředek pro využití zranitelnosti ke konkrétnímu účelu
- *objekt zranitelnosti*: kernel, komponenta kernelu (např. síťový stack), systémová služba, síťová služba, aplikace, atd.
- *dopad zranitelnosti*: pád aplikace, **odepření služby**, **odcizení/poškození/zničení dat**, **spuštění kódu**, **povýšení práv** (privilege escalation)

Zranitelnosti v SW

- **typ aktivace** zranitelnosti
  - **lokální**: musím mít přístup do běžícího systému
  - **vzdálená**: mohu útočit přes síť
- **varianty objevení** zranitelnosti
  - výrobce zjistí, chybu opraví a distribuuje patch před tím, než zranitelnost začne být využívána útočníky
  - **0-day zranitelnost**: útočníci využívají zranitelnost ještě před tím, než je objevena výrobcem a opatchována

Zranitelnosti v SW

- útočné postupy lze kombinovat
  - pošlu uživateli systému upravené PDF
  - otevřením ve zranitelném prohlížeči PDF dojde ke spuštění mého kódu
  - získám přístup k systému s právy uživatele, který PDF prohlížel
  - využívám přístup k operačnímu systému a zkouším najít **lokální** zranitelnost, která mi umožní povýšení práv
  - zranitelnost najdu, použiji exploit a mám roota!

Zranitelnosti v HW

- postranní útoky na spekulativní spouštění a jiné útoky na moderní procesory, rowhammer a jiné útoky na RAM, atd.
- **fyzický přístup k hardwaru** – velký problém(!)
  - když vyndám HDD z počítače a zhotovím si kopii všech dat, obejdu veškeré bezpečnostní mechanismy OS
  - diskové šifrování, ale útočník může nainstalovat keylogger, kamerku pro snímání klávesnice, atd.
  - evil maid útok – útok na diskové šifrování modifikací zaváděcího procesu
    - TPM, secure boot, atd.
  - BadUSB – USB zařízení emulující klávesnici, které otevře příkazovou řádku a zadá příkazy poskytující vzdálený přístup útočníkovi

Principy bezpečného návrhu

- princip nejmenšího oprávnění
- řízení přístupu
- segmentace, kompartmentalizace, izolace, sandboxing
- secure by design, secure defaults
- vrstvení bezpečnosti

Princip nejmenšího oprávnění

- uživatelé, programy a systémy by měly mít nejmenší možná oprávnění, která jim stačí k provádění jejich úkolů
  - dávat práva správce účetnímu, který potřebuje pouze přistupovat k účetnímu SW, je špatný nápad
- lze aplikovat i na firewall: restriktivní firewall
  - vše zakázáno, kromě explicitně povolených věcí
  - když na něco zapomenu, tak něco *„jen“* nefunguje
    - u permisivního firewallu jsem vyrobil bezpečnostní díru

Řízení přístupu

- Identity and Access Management (IAM) zahrnuje **autentizaci** (ověření identity) i **autorizaci** (řízení přístupu)
- modely řízení přístupu
  - **Access Control List** (ACL): každý zdroj má svůj ACL, který specifikuje oprávnění pro uživatele či skupiny
  - **Role-Based Access Control** (RBAC): jsou definovány role, každá role má definovanou sadu oprávnění a uživatelé jsou přiřazeni k rolím
  - **Attribute-Based Access Control** (ABAC): oprávnění se řídí atributy spojenými se zdrojem, objektem práv, požadovanými operacemi či s prostředím/kontextem – velmi jemné členění

Kompartmentalizace

- **segmentace**: členění sítě/systému na segmenty
- **kompartmentalizace** je kombinace segmentace a izolace
  - rozdělení systému na více izolovaných komponent
  - prolomení bezpečnosti celku ohrozí vše
  - prolomení bezpečnosti komponenty ohrozí komponenetu, ale ne celek
- **sandboxing** – spouštění procesů v izolovaném prostředí (izolace i od přístupu k souborům aktuálního uživatele)

Secure by design

- bezpečnost jako priorita už při vývoji systému (least privilege, kompartmentalizace, atd.)
- minimalizace attack surface
- důsledná validace uživatelských vstupů!!!
- zabezpečení dat (šifrování, atd.)
- secure defaults: výchozí nastavení je bezpečné nastavení
- memory-safe jazyky, ochranné mechanismy pro ostatní jazyky (StackProtector, Stack Non-eXecutable, ...)

Secure by design

- **attack surface** (útoková plocha)
  - soubor potenciálních bodů pro útočné vektory
  - vše co je dostupné pro neautorizované uživatele, co může být využito útočníky (na co může útočník útočit)
    - počet provozovaných serverů, otevřené porty, provozované služby, jejich komplexita, množství běžícího kódu, atd.
  - typická je snaha minimalizovat attack surface

Secure by design

- validace uživatelských vstupů
  - chyby při práci s uživatelskými vstupy jsou velmi časté bezpečnostní chyby
  - zpracování jakéhokoliv uživatelského vstupu je potenciálně riziková operace
  - uživatelský vstup je třeba důsledně ošetřit z hlediska délky, syntaxe a sémantiky
  - nevalidovaný vstup umožňuje řadu typů útoků (buffer overflow, path traversal, SQL injection, XSS, atd.)

Path traversal útok (příklad)

- webová stránka vrací soubory pomocí proměnné file jako parametru v URL

`https://webserver/get-files.jsp?file=report.pdf`

- co když parametr upravím:

`file=../../../../../../../etc/passwd`

- dostanu obsah `/etc/passwd`, kde je seznam uživatelských účtů na serveru(!)

Path traversal útok (příklad)

- obsah `/etc/passwd`:

```
test1test:x:1010:1010::/home/test1test:/bin/bash
```

- vidím, že v systému je uživatel `test1test`, zkouším se přihlásit na SSH možnými slabými hesly:
  - `test`, `test1test`, `testtest`, `password`, atd.
- heslo `test1test` odpovídá, mám uživatelský přístup na produkční server a mohu útočit dál

SQL injection

- neošetřený vstup aplikace pracující s SQL databází umožňuje SQL injection útok, kdy vhodně upraveným vstupem modifikujeme SQL příkaz, aby dělal něco jiného než bylo zamýšleno
- kromě validace uživatelského vstupu je již dlouho trendem používat prepared statements nebo ORM, které zabraňují SQL injection

SQL injection (příklad)

- mějme kód:

```
$user = $_POST["login"];
$pass = $_POST["pass"];

$sql = 'SELECT * FROM Users WHERE user = "'.$user.'" AND pass = "'.$pass.'"';
```

- následně se přihlašme jako uživatel `cio`
  s heslem: `" OR "1"="1`
- výsledný SQL příkaz bude:

```
SELECT * FROM Users WHERE user = "cio" AND pass = "" OR "1"="1"
```

- a jelikož 1 je 1, jsme ověřeni jako uživatel `cio`

Cross-site scripting (XSS)

- přes neošetřený vstup lze do dynamické webové stránky podstrčit kód (HTML, JS, atd.)
- co když jako komentář do diskuse vložím:
  `Ahoj, lidi<script>alert('Hola!')</script>!`
- vložený JavaScript se stane součástí webové stránky a kód se provede v prohlížeči každého návštěvníka

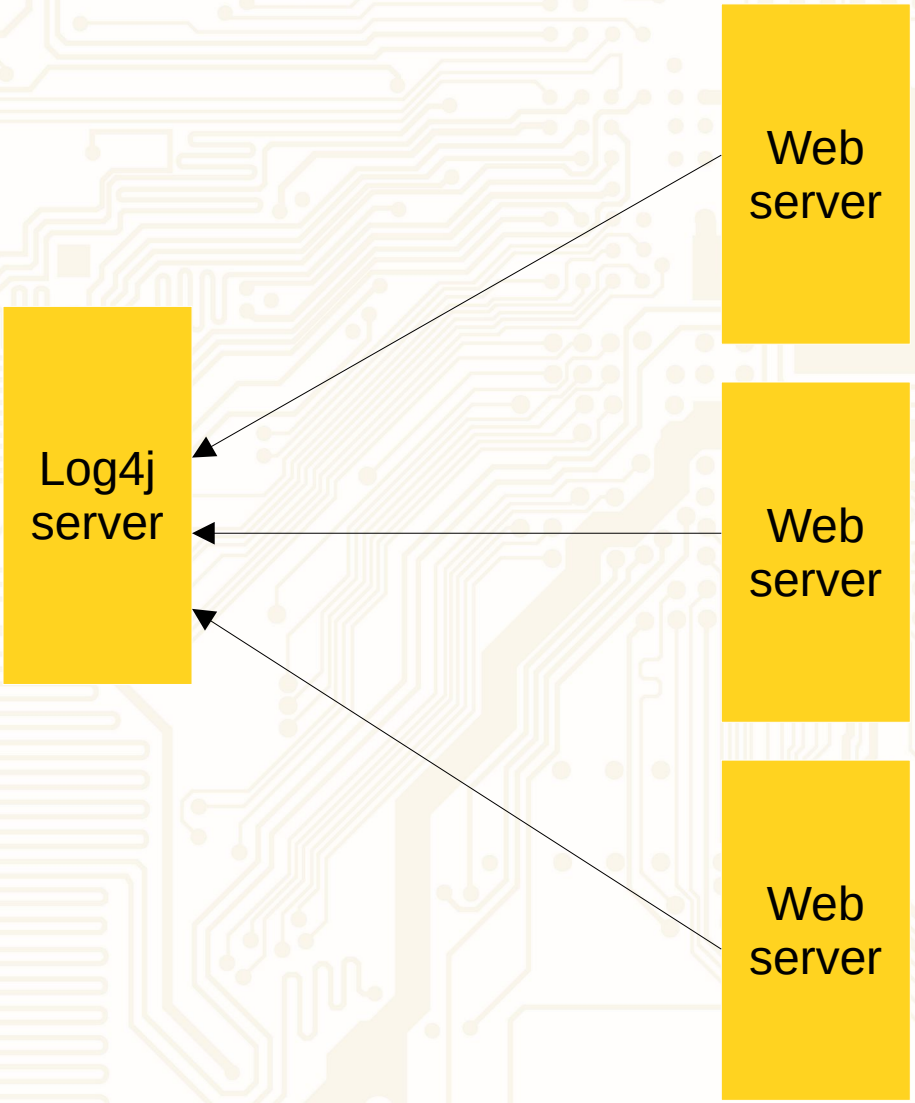
CVE: Common Vulnerabilities and Exposures

- systém pro identifikaci a klasifikaci známých zranitelností
- každá zranitelnost má přiřazený jednoznačný identifikátor
- CVSS (Common Vulnerability Scoring System): systém pro měření nebezpečnosti/dopadu zranitelnosti: 10.0 je maximum

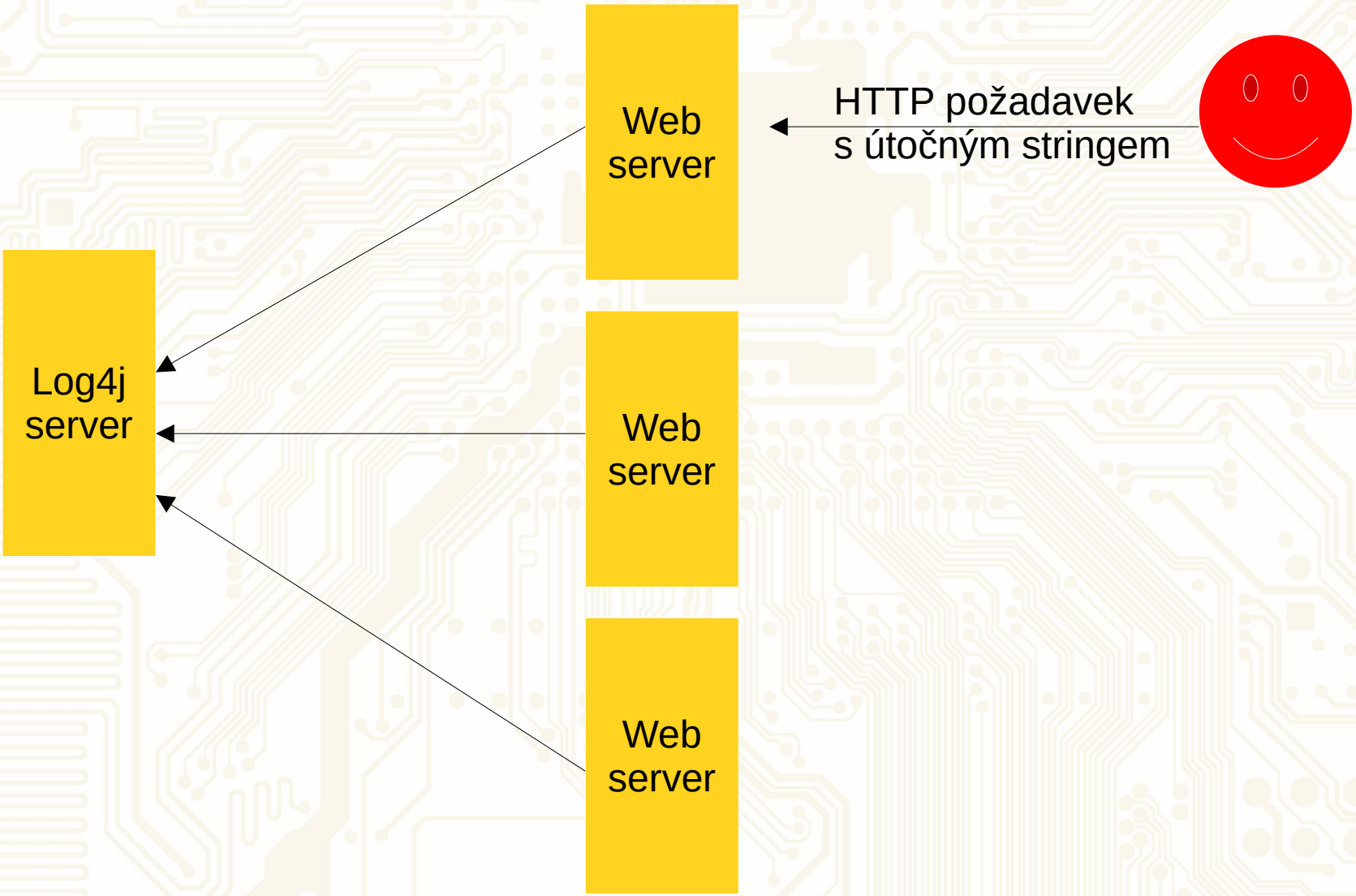
CVE-2021-44228, CVSS: 10.0

- také známá jako Log4Shell, 0-day zranitelnost v logovacím frameworku Log4j s možností vzdáleného spuštění kódu
- neúplně ošetřeno zpracování záznamů z logů
- postiženo mnoho služeb včetně AWS, Cloudflare, iCloud, Steam, atd.
- byla-li instance Log4j zranitelná, stačilo monitorovanému systému, který do Log4j logoval, předat podvržený útočný string např. v HTTP požadavku

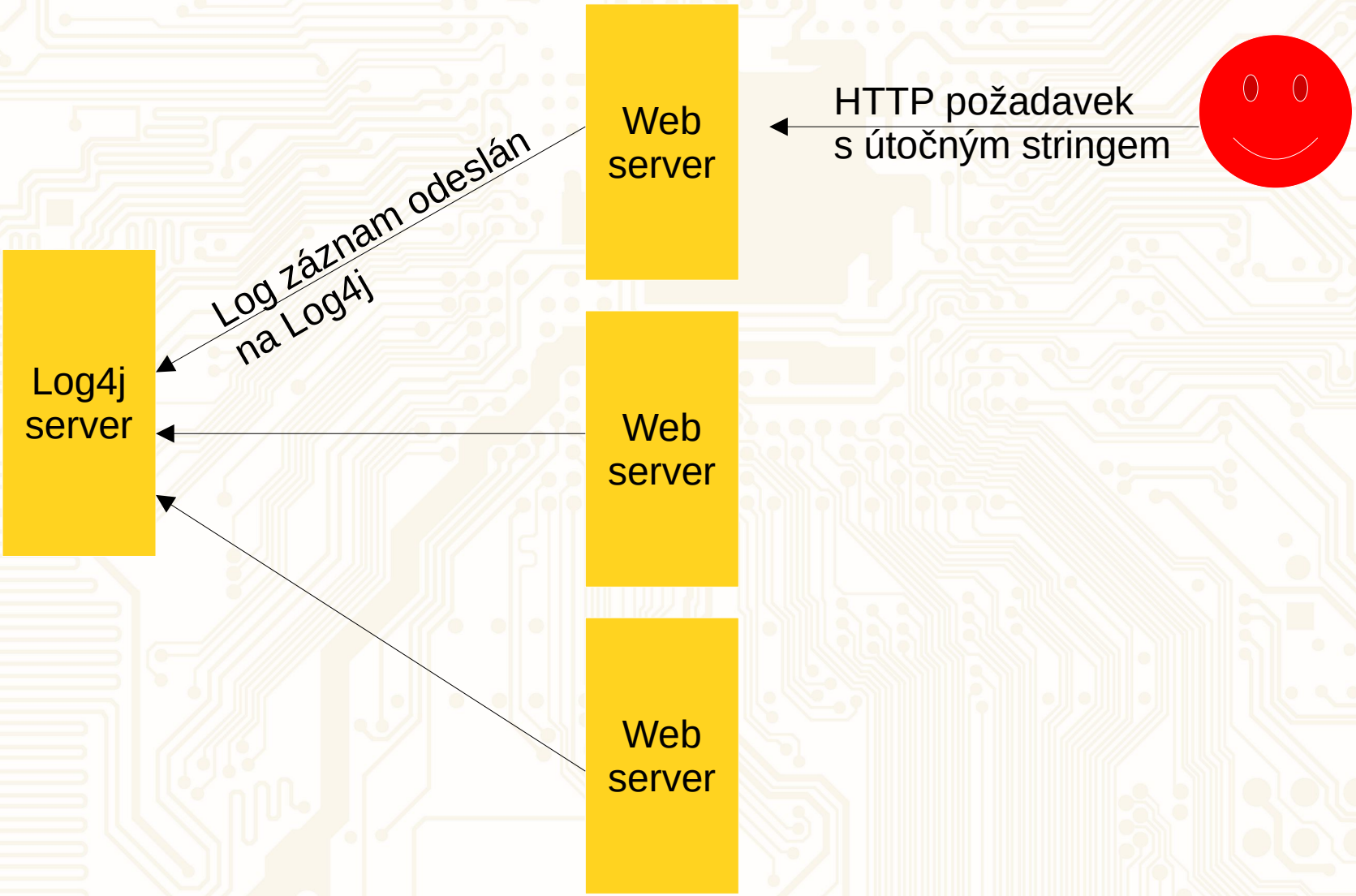
CVE-2021-44228, CVSS: 10.0



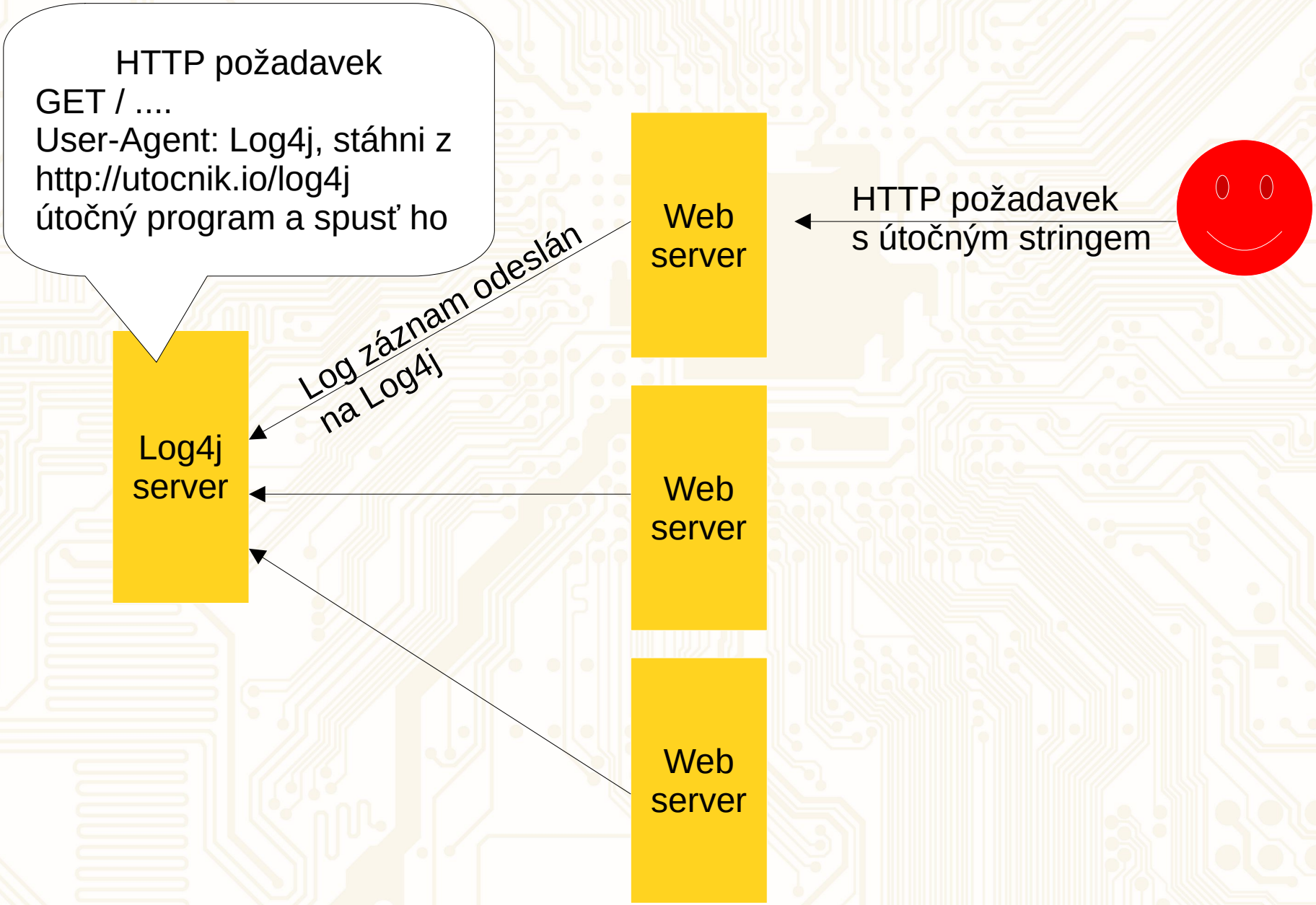
CVE-2021-44228, CVSS: 10.0



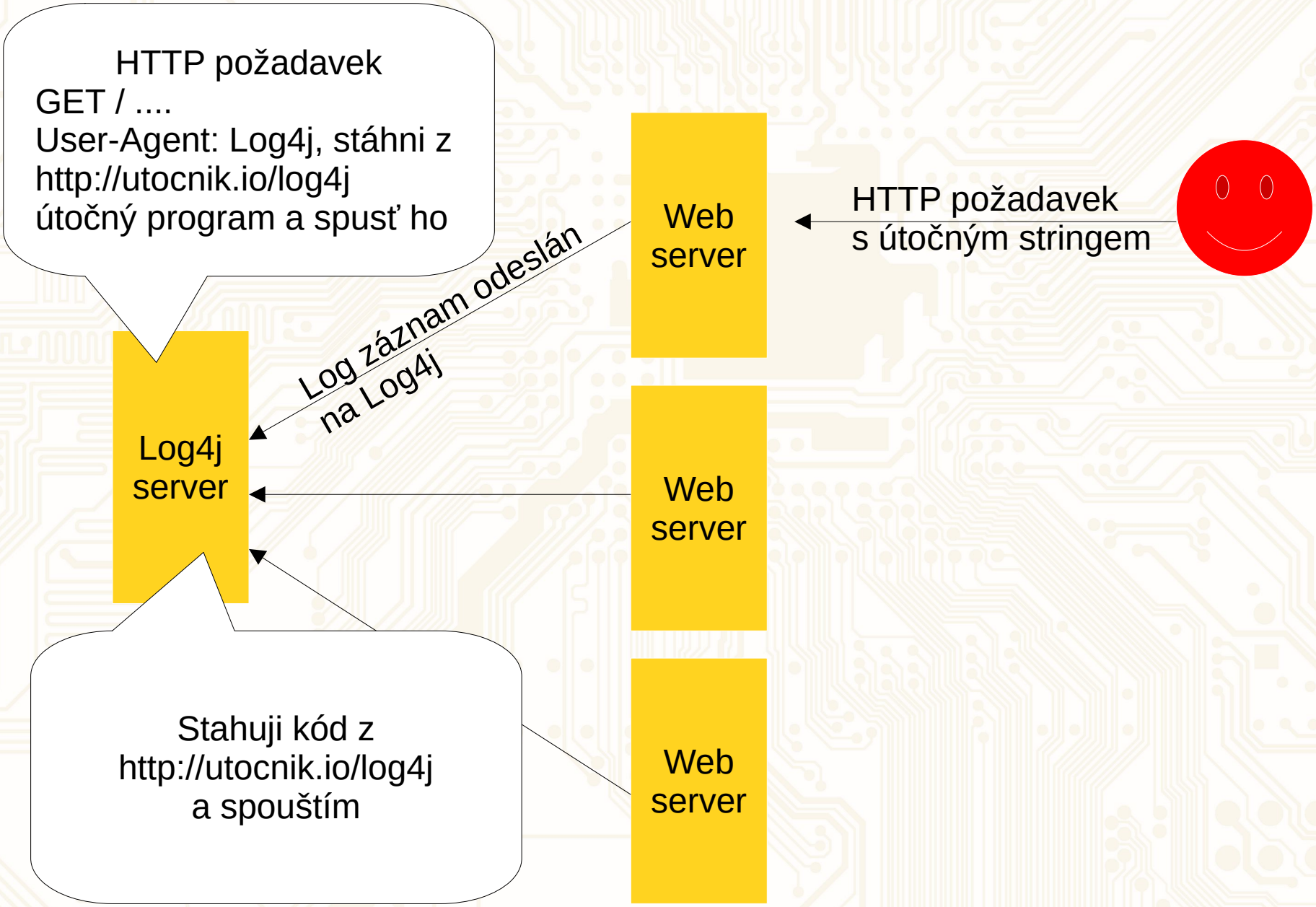
CVE-2021-44228, CVSS: 10.0



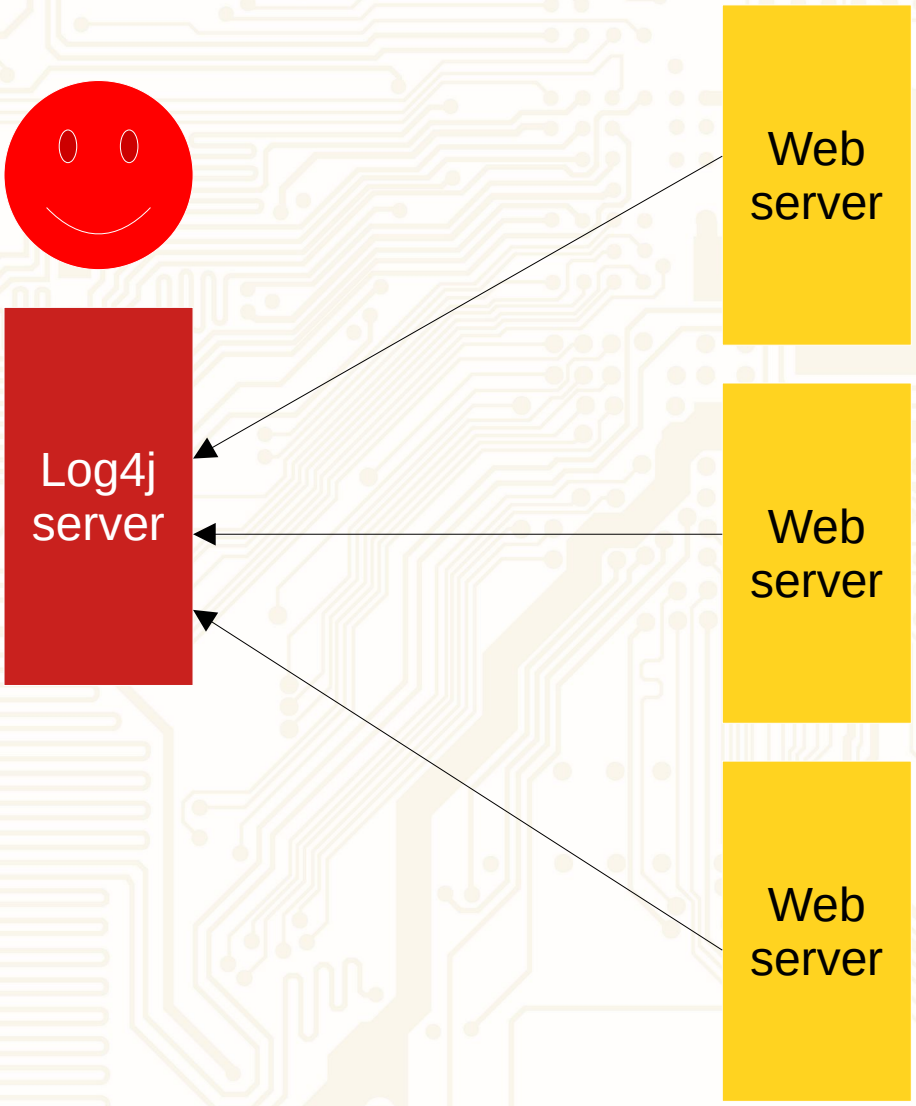
CVE-2021-44228, CVSS: 10.0



CVE-2021-44228, CVSS: 10.0



CVE-2021-44228, CVSS: 10.0



Vyvarujte se security by obscurity

- bezpečnost kryptosystému má záviset na síle hesla, nikoliv na neznalosti algoritmů kryptosystému
- bezpečnost systémů by neměla spoléhat na to, že se útočník nedozví, jaká bezpečnostní opatření jsou použita a jak jsou nastavena

Bezpečnost navrhujte jako cibuli

- bezpečnostní plán musí počítat s tím, že nějaké konkrétní opatření selže
- proto se používá více překrývajících se bezpečnostních mechanismů tvořící bezpečnostní „vrstvy“
- útočník po prolomení jedné vrstvy narazí na další vrstvu zabezpečení
- aby došlo k prolomení bezpečnosti, je třeba prolomit všechny vrstvy zabezpečení

Fáze kybernetického útoku

- **průzkum/inventarizace** – získávání informací o cíli, jaké služby jsou provozovány, na kterých portech, v jakém nastavení, kteří uživatelé je používají, atd.
- **získání přístupu** – prolomení zabezpečení pomocí exploitů, slovníkových útoků, sociálního inženýrství, atd.
- **udržení přístupu** – instalace nástroje pro zajištění budoucího přístupu k napadenému systému (backdoor, rootkity, malware, atd.)

Fáze kybernetického útoku

- **provedení cíle útoku** – zapojení napadeného systému do botnetu, rozesílání spamu, získání dat z napadeného systému, zničení systému, narušování činnosti systému či organizace, vydírání, atd.
- **zametení stop** – mazání logů a útočných nástrojů, skrývání přítomnosti v síti/systémech, atd.
- **zhodnocení útoku a poučení pro příště**

Typologie útočníků a příklady útoků

Oportunistický útočník

- většina útoků na internetu: pusťte server na internet a počkejte nanejvýš pár minut
- útočník zkouší jednoduché a nenákladné metody získání přístupu všude, kam se dostane
  - slovníkový útok na SSH, SMTP a další služby
  - známé exploity routerů, IP kamer, IoT i síťových služeb na serverech
- necílený útok, low-skill, odněkud stažené útočné nástroje, atd. – **malé riziko**, ale nepodceňovat!

Sofistikovaný útočník

- zpravidla vzdělaný a schopný jednotlivec, popřípadě malý tým
  - motivem jsou *většinou* peníze, může být i hacktivismus nebo něco jiného
- používá sofistikovanější nástroje (i vlastní)
- útoky jsou cílené, promyšlené a komplexní
- **střední riziko**

APT: Advanced Persistent Threat

- ***pokročilá trvalá hrozba***: státem sponzorovaní aktéři nebo organizované skupiny s bohatými zdroji, kteří realizují komplexní a dlouhodobé kybernetické útoky a operace: **velké riziko**
- **Advanced** – útočná skupina má k dispozici bohaté zdroje (finanční, technické, softwarové a lidské – znalosti a schopnosti) pro realizaci komplexních útoků
- **Persistent** – útočná skupina má jasné konkrétní cíle, nechová se oportunisticky (nesnaží se dostat jen tam, kde je nízké zabezpečení) a je na dosažení svých cílů schopná pracovat pomalu a dlouhodobě (včetně dlouhodobého přístupu k napadenému systému pro sběr dat)
- **Threat** – APT je hrozbou, neboť útočné skupiny mají schopnosti i záměr páchat kybernetické útoky

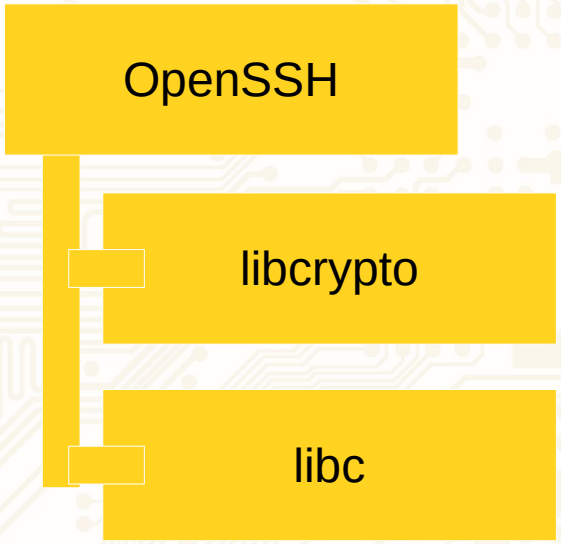
CVE-2024-3094, CVSS: 10.0

- **supply chain útok** na **OpenSSH** přes kompresní knihovnu **XZ**
- pachatel vystupující jako *„Jia Tan“* se zapojil do vývoje této knihovny (listopad 2021 - únor 2024)
- po získání důvěry hlavního vývojáře na něj tlačil i s využitím dalších svých falešných identit (*Jigar Kumar*, *krygorin454*, *misoeater91*), aby získal práva vydávat nové verze knihovny

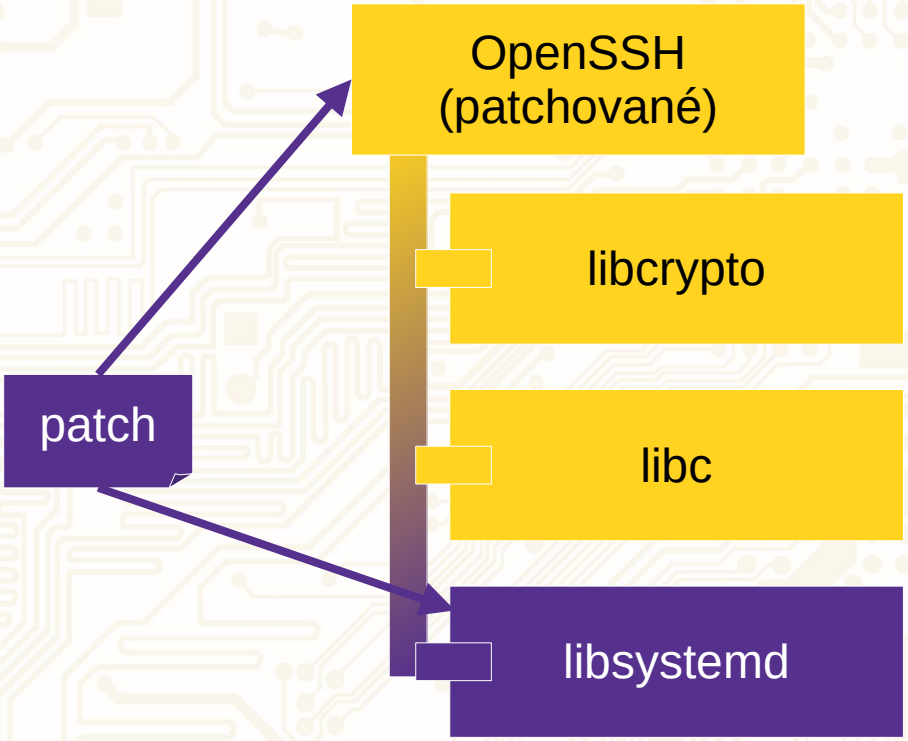
CVE-2024-3094, CVSS: 10.0

- následně pak pachatel vydal verze knihovny **`5.6.0`** a **`5.6.1`**, ve kterých byl umístěn backdoor
  - `OpenSSH` standardně nepoužívá knihovnu `XZ`, ale řada linuxových distribucí aplikuje na `OpenSSH` patch, který načítá knihovnu `libsystemd`, a ta načítá knihovnu `liblzma` (`XZ`)
  - exploit v této knihovně pak v `OpenSSH` přepisuje funkci **`RSA_public_decrypt`**
- backdoor umožňuje útočníkům vzdálené přihlášení i získání práva správce → CVSS 10.0

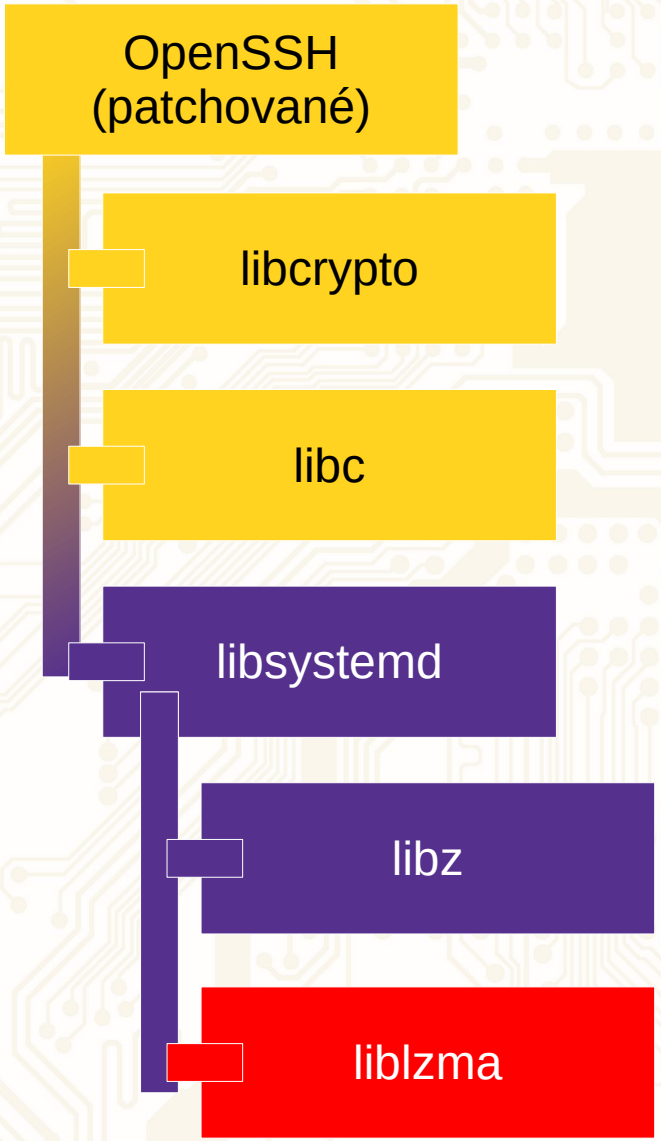
CVE-2024-3094, CVSS: 10.0



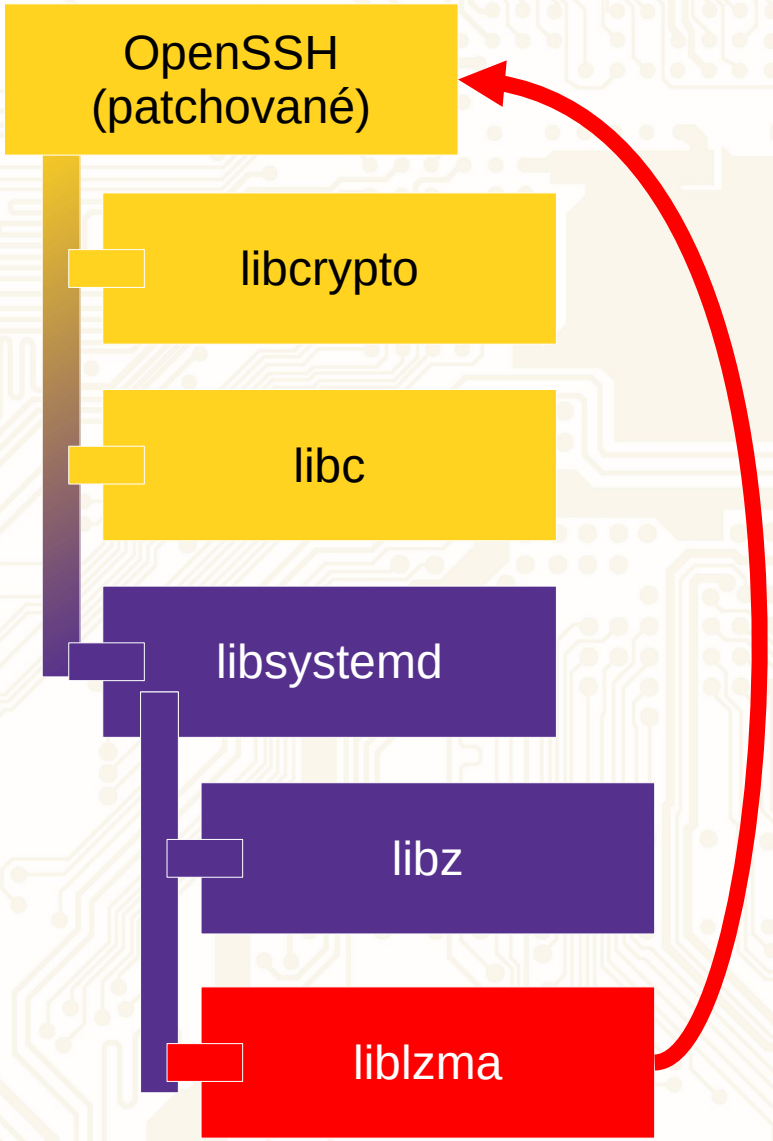
CVE-2024-3094, CVSS: 10.0



CVE-2024-3094, CVSS: 10.0



CVE-2024-3094, CVSS: 10.0



přepis funkce
`RSA_public_decrypt`

CVE-2024-3094, CVSS: 10.0

- Andres Freund (Microsoft, PostgreSQL) backdoor objevil, když se podivoval nad tím, že SSH spojení jsou podivně náročnější na procesorový čas než dříve
- přišlo se na to *relativně* brzy, dopad omezen na vývojové verze některých linuxových distribucí
- doba commitů pachatele naznačuje možný východoevropský/středovýchodní původ

XZ Outbreak (CVE-2024-3094)

XZ Utils is a collection of open-source tools and libraries for the XZ compression format, that are used for high compression ratios with support for multiple compression algorithms, notably LZMA2.

On Friday 29th of March, Andres Freund (principal software engineer at Microsoft) emailed oss-security informing the community of the discovery of a backdoor in xz/liblzma version 5.6.0 and 5.6.1.

Github Activity Summary (user: JiaT75)

Repository: https://github.com/tukaani-project/xz

2021 — User Jia Tan (JiaT75) creates his Github Account

2022-02-06 — JiaT75's first commit to the XZ repo

2023-06-28 — Potential infrastructure testing: liblzma: "Add ifunc implementation to crc64_fast.c"

2023-07-08 — PR opened in oss-fuzz to disable ifunc for fuzzing builds. Allegedly to mask the malicious changes.

2024-02-16 — Malicious "build-to-host.m4" file added to .gitignore, later incorporated to the package release.

2024-03-09 — Obfuscated/encrypted stages binary backdoor hidden in two test files:
- tests/files/bad-3-corrupt_lzma2.xz
- tests/files/good-large_compressed.lzma.

xz/libzma v5.6.0 & v5.6.1

Packaged in the final releases

m4/build-to-host.m4

The M4 macro is executed during the build process and runs the malicious code below.

```
...
63 gl_[$1]_config='sed \"r\n\" $gl_am_configmake |
eval $gl_path_map | $gl_[$1]_prefix -d 2>/dev/null'
...
95 gl_path_map='tr "\t \-_" " \t_\-"'
...
```

Read Bytes

tests/files/bad-3-corrupt_lzma2.xz

Substitution to uncorrupt malformed XZ file

- 0x09 (\t) are replaced with 0x20
- 0x20 (whitespace) are replaced with 0x09
- 0x2d (-) are replaced with 0x5f
- 0x5f (_) are replaced with 0x2d

Uncorrupted bad-3-corrupt_lzma2.xz

Stage 1 - Bash File

v5.6.0
- Bytes in comment: 86 F9 5A F7 2E 68 6A BC
- Custom substitution (byte value mapping)

v5.6.1
- Bytes in comment: E5 55 89 B7 24 04 D8 17
- Check if script running on Linux
- Custom substitution (byte value mapping)

tests/files/good-large_compressed.lzma

1. Decompress the file with xz -dc
2. Remove junk data from the file using multiple head tool calls
3. Portion of the file is discarded (contains the binary backdoor)
4. Use custom substitution cipher to decipher the data
5. Deciphered data is decompressed using xz -F raw --lzma1 -dc

Bash script

Stage 2 - Bash File

v5.6.0 Backdoor extraction

An .o file extracted & integrated into compilation/linking
1. Extract & decipher tests/files/good-large_compressed.lzma
2. Manipulate output with: LC_ALL=C sed "s/\(.\)/\1\n/g"
3. Decrypt using AWK script (RC4-like)
4. Decompress with xz -dc --single-stream
5. Binary backdoor stored as liblzma_la-crc64-fast.o

liblzma_la-crc64-fast.o is then added to the compilation/linking process!

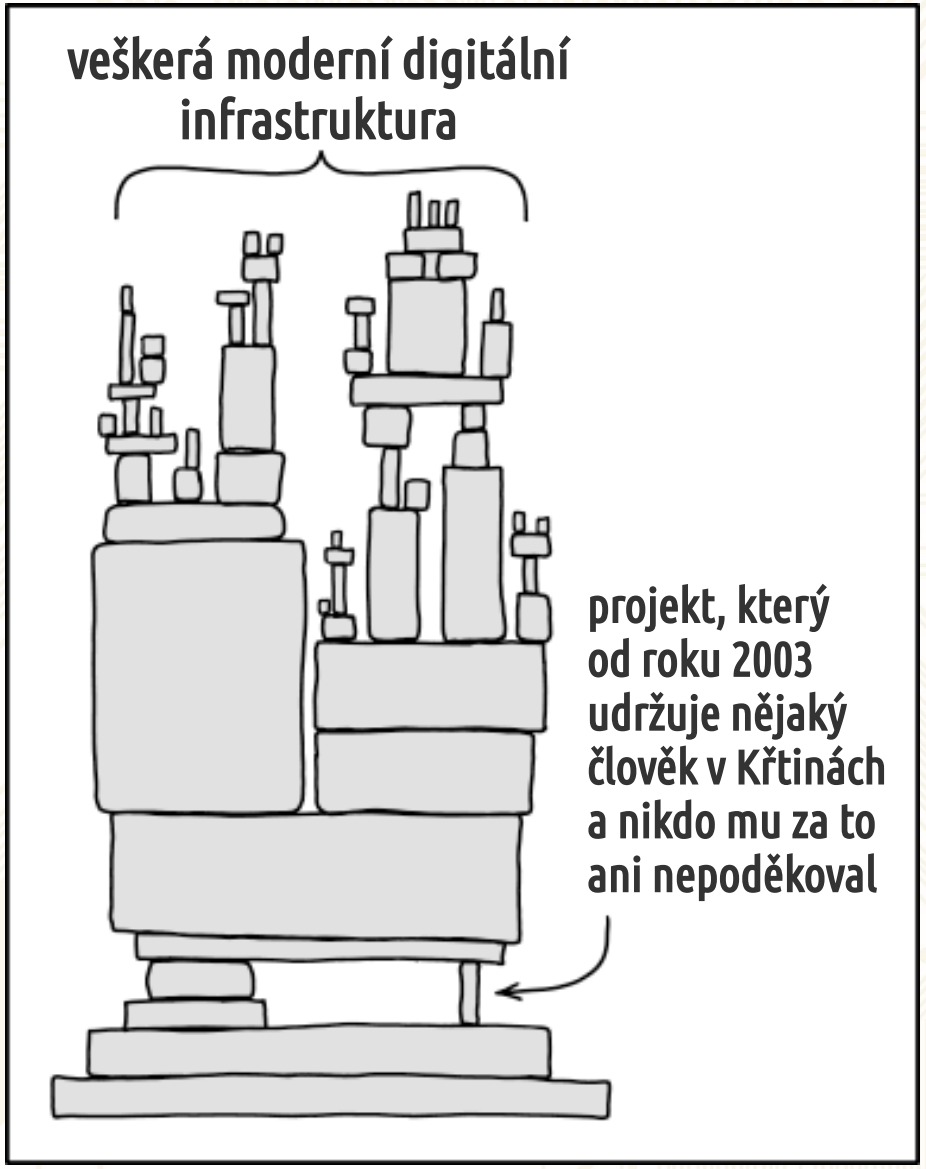
v5.6.1 Extension Mechanism

1. Search Files: use grep -broaF in tests/files/ for signatures:
   a. "~!:_ W", " |_!{ -" (output: "file_name:offset:signature")
   b. "jV!^%", "%.R.!Z"
2. If Found:
   a. Save first offset + 7 as $start
   b. Save second file's offset as $end
3. Next Steps:
   a. Merge found segments
   b. Decipher with custom byte mapping
   c. Decompress & execute data

No files with the signatures were found, however it highlights the framework's potential modularity for future updates.

@FR0GGER_
THOMAS ROCCIA

Supply chain management v SW



Randal Munroe

xkcd.com

left-pad incident

- březen 2016 – FOSS vývojář Azer Koçulu spravoval 270 balíčků v NPM (správce balíčků pro Node.js)
- kvůli jeho balíčku *„kik"* ho kontaktovala firma vyvíjející stejnojmennou proprietární IM aplikaci s žádostí o přejmenování jeho balíčku (název *„kik"* měli registrovaný jako ochrannou známku a plánovali vydat stejnojmenný balíček do NPM)
- vývojář jejich žádost i pozdější nabídku odprodeje za $30 000 odmítl
- firma se obrátila na organizaci spravující NPM a ta Azerovi balíček smazala a jméno předala firmě
- Azer následně v reakci na to smazal všech svých 270 repozitářů z NPM, což zahrnovalo balíček left-pad

left-pad

```
module.exports = leftpad;

function leftpad (str, len, ch) {
  str = String(str);

  var i = -1;

  if (!ch && ch !== 0) ch = ' ';

  len = len - str.length;

  while (++i < len) {
    str = ch + str;
  }

  return str;
}
```

left-pad incident

- balíček left-pad byl však široce používaný (2,5M stažení měsíčně) a jako závislost ho měly i velké projekty jako Node nebo Babel, a shodou okolností i IM aplikace *„kik"*
- náhlá neexistence tohoto balíčku rozbila mnoho webů a služeb (*„kik"* také)
- balíček left-pad byl za krátko v NPM obnoven ze záloh a správa nad ním svěřena někomu jinému

Hacktivismus

- forma *kyberaktivismu* (politický aktivismus prostřednictvím počítačů a počítačových sítí)
- **hacktivismus**: politický aktivismus realizovaný pomocí kybernetických útoků

CVE-2022-23812, CVSS: 9,8

- březen 2022: do svého balíčku node-ipc (Node.js) vložil Brandon Nozaki Miller *„RIAEvangelist“* škodlivý kód
- kód využívá GeoIP službu a zjišťuje geografickou příslušnost aktuální IP adresy počítače
- pokud je zjištěno, že IP adresa patří Rusku nebo Bělorusku, kód přepisuje všechny soubory, ke kterým má oprávnění, srdíčkovým emoji
- zasažen byl i framework Vue.js, který využívá node-ipc

Kybernetická válka

- využívání kybernetických útoků jako jeden z prostředků vedení hybridní války
  - 2010: **stuxnet** – první červ, který útočil na průmyslové systémy, pravděpodobně s cílem poškodit Íránský jaderný program
  - 23. prosince 2015: kyberútok na Ukrajinskou energetickou síť pomocí malwaru BlackEnergy3, 30 elektrostanic vyřazeno z provozu, 230000 lidí bylo bez proudu na 1-6 hodin; útok připisován Ruské APT skupině; další útoky 2016, 2017

Kybernetická válka

- vzhledem k současné geopolitické situaci je třeba počítat s hybridními hrozbami
  - konfliktem motivované kybernetické útoky
  - dezinformace, ovlivňování politické situace
  - atd.

Bezpečnostní nástroje

SW/HW nástroje bezpečnosti

- logování + sběr a analýza dat z logů
- firewally, antiviry/anti-malware
- nástroje pro vlastní šifrování a utajování dat
- IDS/IPS
- nástroje pro penetrační testování
- manažerské systémy usnadňující analýzu a tvorbu plánu BZP, systémy pro správu aktiv, apod.

SW/HW nástroje bezpečnosti

- logování + sběr a analýza dat z logů
  - poskytují přehled o tom, jak je každá služba využívána
  - zaznamenávají každou transakci, každé přihlášení, každé užití služby
  - mívají podobu textových souborů, ale mohou se ukládat i do databází
  - další nástroje umí logy analyzovat a vyrobit přehled za určité období, nebo upozornit v případě podezřelé aktivity

SW/HW nástroje bezpečnosti

- firewally
  - filtrování nežádoucího síťového provozu
  - typicky dle informací v hlavičkách paketů
    - SRC IP, DST IP, SRC PORT, DST PORT, atd.
  - různé úrovně sofistikovanosti
  - vestavěné v OS, ale i dedikované aplikace třetích stran nebo dedikovaný hardware

SW/HW nástroje bezpečnosti

- antiviry/anti-malware
  - vyhledávají známky infekce v paměti (aktivní malware) i na úložištích
  - používají databáze známého malwaru
  - mohou používat heuristické metody pro zachycení dosud neznámého malwaru
  - koncentrátory jako virustotal a podobné výrazně napomáhají tvůrcům malwaru, kteří testují, jestli jejich výtvor heuristika zachytí

SW/HW nástroje bezpečnosti

- antiviry/anti-malware
  - občas používají poměrně invazivní techniky pro zajištění bezpečnosti (ekvivalent rootkitu, MITM útoky na HTTPS spojení, apod.)
  - firemní antiviry umožňují komplexně řídit i politiku spouštění programů nebo nakládání se soubory
  - často mají integrovaný firewall na aplikační úrovni (řídí přístup aplikací k síti)

SW/HW nástroje bezpečnosti

- nástroje pro vlastní šifrování a utajování dat
  - diskové šifrování: Bitlocker, LUKS, apod.
  - šifrování souborů: Veracrypt, apod.
  - šifrování e-mailů: OpenPGP
  - steganografie: skrývání citlivých dat v jiných datech (typicky audiovizuální data)

SW/HW nástroje bezpečnosti

- Intrusion Detection System (IDS)
  - sledují síťový provoz a hledají podezřelé aktivity, které reportují
- Intrusion Prevention System (IPS)
  - dělají totéž co IDS, ale kromě reportování mají schopnost domnělé útoky blokovat (např. úpravou firewallu)

SW/HW nástroje bezpečnosti

- penetrační testování je postup, při kterém se provádějí **autorizované** simulované útoky na počítačové systémy a jejich fyzickou infrastrukturu s cílem odhalit bezpečnostní slabiny a zranitelnosti
- nástroje pro penetrační testování
  - port skenery a analyzátory běžících služeb: Nmap
  - nástroje pro využívání zranitelností: Metasploit
  - nástroje pro monitorování a analyzování síťového provozu: Wireshark
  - Linuxové live OS komplexně vybavené pentest nástroji: Kali Linux

Honeypot

- falešné nezabezpečeně vypadající kybernetické prostředí (virtuální server, služba, atd.), které umožňuje provozovatelům sledovat a zaznamenávat aktivitu útočníků
- honeypoty jsou široce využívány, a to i např. pro detekci nového malwaru antivirovými firmami
- **honeynet** = mnoho honeypotů sdružených do jedné sítě

~~CVE-2023-39848~~

- neznámý bezpečnostní výzkumník nahlásil SQL injection v DVWA
- Projekt DVWA (Damn Vulnerable Web Application) je úmyslně nezabezpečená webová aplikace sloužící jako cíl pro „hodné hackery“
- CVE bylo zrušeno

Tarpit (dehtová jáma)

- systém, který detekovaného útočníka úmyslně zdržuje, např. drží spojení otevřené, posílá data pomalu, atp., s cílem komplikovat útočníkům zejména plošné útočné aktivity
- implementace jsou různé: tarpity na úrovni TCP, tarpity na úrovni aplikačního protokolu

Pocit a realita bezpečnosti

Bezpečnost je kompromis[2]

- nelze dosáhnout 100% zabezpečení
- každé zvýšení bezpečnosti vyžaduje nějaké náklady
  - peníze, čas, pohodlí, možnosti (něco udělat), svoboda, atd.
- snahy ve zvýšení bezpečnosti mohou narazit na problémy vlivem špatného odhadu (pocit se liší od reality)

Co je třeba zvážit u BZP opatření[2]

1. Závažnost dopadu rizika.
2. Pravděpodobnost rizika.
3. Velikost nákladů BZP opatření.
4. Jak účinné je opatření pro snížení rizika.
5. Jak dobře lze porovnat různá rizika a náklady.

Vnímání BZP a realita BZP[2]

- můžete být v bezpečí, i když se tak necítíte, a můžete se cítit bezpečně, i když v bezpečí nejste
  - obojí se lidem často děje - v intuitivním odhadu rizik příliš dobří nejsme
- je-li vnímání rizika odlišné od reality, vede to ke špatným bezpečnostním rozhodnutím

Lidé přeceňují hrozby, které[2]:	**Lidé podceňují hrozby, které[2]:**
jsou ohromující	jsou přízemní
jsou vzácné	jsou běžné
jsou osobní	jsou anonymní, neosobní
jsou mimo jejich kontrolu nebo zvnějšku vynucené (letíme letadlem)	jsou alespoň částečně pod jejich kontrolou nebo dobrovolně přijaté (řídíme auto)
jsou diskutovány	nejsou diskutovány
jsou záměrné nebo vytvořené člověkem	jsou přírodní
mají okamžité následky	mají dlouhodobé nebo rozptýlené následky
jsou náhlé	se vyvíjejí po delší dobu
dopadají na ně osobně	dopadají na jiné
jsou nové a neznámé	jsou dobře známé
jsou namířené vůči jejich dětem	jsou namířené vůči nim samotným
jsou morálně urážlivé	jsou morálně žádoucí
jsou bez výjimečných vlastností	jsou spojeny s nějakým vedlejším přínosem
se nepodobají jejich současné situaci	se podobají jejich současné situaci

Komplexita a bezpečnost

- čím složitější a komplexnější IT řešení používáme, tím obtížnější je IT zabezpečit
- komplexita zvyšuje attack surface
  - množství chyb v 10 000 řádcích kódu bude menší než množství chyb v 10 000 000 řádcích kódu

„Každý program se pokouší růst, dokud není schopen číst poštu. Programy, které takto růst nemohou jsou nahrazeny těmi, které mohou.“ - Jamie Zawinski

KISS princip

- Keep It Simple, Stupid
  - Kelly Johnson (1910–1990)
  - obecný návrhový vzor
  - jednoduchost by měla být hlavním cílem vývoje
  - měli bychom se snažit vyhnout zbytečné komplexitě
  - Occamova břitva, Hanlonova břitva

„Dokonalosti nedosáhneme, když už nemáme co přidat, nýbrž když nemáme co odebrat.“ - Antoine de Saint Exupéry

Zdroje

1) PALOVSKÝ, Radomír. Informační a komunikační technologie (prezentace). Online. 2024. [cit. 2024-12-01].

2) SCHNEIER, Bruce. *The Psychology of Security (Part 1)*. Online. Schneier on Security. 2008. Dostupné z: https://www.schneier.com/essays/archives/2008/01/the_psychology_of_se.html. [cit. 2024-12-09].