

Významné tridy NP, NPH pro funkční výzvy

KOMBINAT. PROB. - testyprí a nýtovyprí kritérium
- konfigurační kritérium, omezení a optimalizační kritérium

KONFIG-PROMĚNNÉ - nastavě hrubá a říla

KONFIGURACE – ovládací knof. Kominjic

INSTANCE - vložené následující funkce

CERTIFIKÁT - struktura, umístění cíničit, ze odpovídá ANO je správné

RESEN - konfigurace, která splňuje všechny

STAVOVÝ PROSTOR - minima next stan. algorithm - §

- Monoline w/ each operon \Rightarrow $S \rightarrow S$ talcysl, i.e. $Q_1(A_1) \neq A_1$. No $\leftarrow Q_1$ or a A_1
 - drojce (Q_1, S) je stary posta
 - dicas se tam benu i jci-a bencovej staj

STAV ALGORITMU - konfiguracijski parametri & nizovi parametra algoritma

STRAT. POHÝBU STAV. PROST - transformace abd. stann na Nepálom, jinac

- UPLNA - Maršálka + star, když tičí, co máme, je nedokáží optimální řešení
 - SYSTEMATICKÁ - náplně strategie, která maršálka + může mít řešení

TŘÍDY PROBLÉMU - ROZHODOVACÍ

- ~~Analýza~~ existují řešení splňující omezení - ANO/NE
 - KONSTRUKTIVNÍ
 - Můžou být řešení splňující omezení
 - ENUMERAČNÍ
 - Nestojí vůči daná řešení splňující omezení

\Rightarrow lisi se jure nijstygem

OPTIMALIZ. PROB : - ROZHOD.

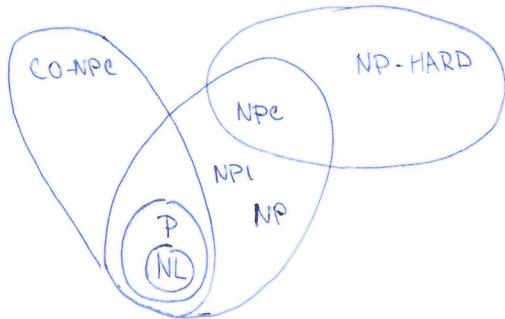
- existuje několik sfin. cmezení celostní tah dleží jeho otáček konstrukční
 - můžete nejlepší možné několik sfin. cmez. počítat

TURINGŮV STROJ - formální sedmice

- teoretický Model počítače

- DETERM - dalo si stát, že málo výhodně

- NEDETERM - plně něco může mít dalo si stát jednoduše, stojí se jenom s mnoha výhodami



P = polyaminiční
NP = nedet. polyn.

P = NP ⇔ výhodně problem

⇒ plně ANO, tak je obecně řešitelné APOD

CO-NP - možnost, na které řešitelsky řešitelné

odpověď ANO. Nemáme totiž žádat jeho důkaz certifikát.

⇒ Je BOOL funkce řešitelná? ANO ⇒ CO Č TIM?

⇒ Je to pravidlo × Je to shodné číslo?

Optimalizační problemy - PO - jde do NPO a existuje program pro det TS + lin. čas

NPO - nelze řešit řešitelně pomocí polyaminičně - P

- problem, když otázka konfigurace je řešitelná je ≈ P

- existuje program pro řešení optimálního řešení ≈ P

⇒ stejný jako NP, může obsahovat certifikát a opt. kritérium

NPH... problem, na který se dle TURING. redukuje každý problem z NP

⇒ může být polyaminiční čas?

NPC... NPH problem, který je řešitelný ≈ NP

- SAT, 3SAT, KNAPSACK

NPI... Problemy ≈ NP, které nejsou ≈ NPC... tedy na nich můžeme řešit řešitelné další problém

⇒ může na ně mít SAT

TURINGOVA REDUKCE: Ruk. prob. R je řešitelný na R₂, jestliže ∃ program pro det TS, který řeší problem R₁, takže používá program R₂ pro jeho řešení

⇒ Polyaminiční čas si musíme skleni vyrovnat?

KARPOVA REDUKCE: Specifikace TURINGOVY - existuje polyn. program pro det TS, který řeší řešitelný problem A řešený na B, takže vystupuje řešitelný řešitelný program

⇒ f. jiné řešitelné řešitelné programy

⇒ TRANSITIVNÍ a nové řešitelné ekvivalentní tridiční

COOKOVA VĚTA: Existuje NP-výhodný problem

⇒ je jím například SAT

⇒ Plně řešitelný problem ∈ NP a lze na něj řešit SAT, takže NP-výhodný

⇒ Výhodný problem ∈ NP je řešitelný na SAT

Experimentální výhodnocení algoritmů, zejména hardomisovací.

RANDOMIZOVANÝ ALGORITMUS - postupy dle něj mohou

PSEUDOPOLY. ALGO - počet řešení poly. závisí na velikosti instance
atenci na parametry, které jsou využívány v instance

APPROXIMATIVNÍ ALGO - APPROX - je R-approximativní, mohut vypočítat instance problému
A relativní hranitán / číslo R

SAT - CNF - zadání k nějaké výhodnocení kontradikce

MAX SAT - maximální řešení obsahující klesající

RANDOMIZOVANÉ ALGORITMY

- Algoritmy NPO - a tím následně

a) DETERMINISTICKÝ - zahrnuje číslo n nejdříve

APPROXIMATIVNÍ - stanoví relativní číslo / hranitán

PSEUDOPOLYNOMIALNÍ

b) NÁHODNÁ METODA - statistická čísla nebo minimální

RANDOMIZOVANÝ

DRUHÝ RAND. ALGORITMŮ

MONTE CARLO

- nejvýkonnější, nejdelší výsledek - náhodnost

LAS VEGAS

- nejrychlejší, nejkratší čas

NAPŘ.: Randomizovaný MAX SAT - MC

Miller-Rabinov test pravděpodobnosti - MC

Quick SORT - LV

VÝHODY: jednoduchost

hnědota může být lepší

nezávislost OPK \Rightarrow lepší hranitá

ANALÝZA: merí očekávanou hodnotu - KVALITA, ČAS
platné pro libovolný rozsah

Experiment: PLAN \rightarrow PROVED \rightarrow INTERPRET
 ↑ ↓
 OTÁZKA \leftarrow ODPOVĚĎ

je lepší A, nebo B.

je A lepší než minimální instance než B.

je A lepší než náhodná instance?

WIBER INSTANCI: a) NÁH GENER.
 b) GENER. S OHLED. NA EXPERIMENT
 c) STANDARDNÍ BENCHMARKY

CO SE HODNOTÍ

- hranitá řešení - řeší klesající množství řešení, přičichne pouze relativně lepším
- nejrychlejší hranitá - mezi se klesají řešení řešené řešenými řešenými
- nejrychlejší hranitá - mezi se klesají řešení řešenými řešenými

JAKÉ PŘÍPADY:

negativní
nefoušení
vinný

TYPY HODNOCENÍ:

WHITE BOX - známe algoritmus, použijeme mezi ně instance
BLACK BOX - kompletní funkce instance, statistická otázka

PARAMETRY: obecní nejsou rozdílné!

diferenční odhadit rozdíl mezi srovnávanou hodnotou
následujícími ještě zvětšuje starou listu

VÝSTUP: Nondominantní strategie \Rightarrow statistická analýza

PŘÍKLAD: Zaváděcí náhodné hledání

- všechny kameny náhodně $\Rightarrow P = 1/2$

- \Rightarrow hledání $0 < P < 1$:

- $\Rightarrow P$ - nejdříve náhodně - jednu kamenou, aby hledání bylo slyšitelné

- $\Rightarrow 1-P$ - hledání jednou kamenou, aby bylo slyšitelné celkové hledání

- výhody

- ideální $p \in (0,5; 0,6)$

3SAT

- $\frac{1}{2}$ jmenem, které málochší - $1/2$

- ohledně existujícího obdrženého chodu, a následně hledání opakuj

Náhodné hledání

- je nutné bezúčelné oddávat hledání dole, aby bylo možné využít

Princip lokálního heuristiky je hledání lokálních a globálních minima, ohana proti svážení

Lokální stav - chodí stav $o \in \mathcal{Q}(t)$, kde $t \in S$ a (S, Q) je stav. systém

Sousední stav - patří do lokality

Lokální minimum - všechny chodí stav mytí klesají hodnota opt. kritéria

Globální - všechny klesají hodnota opt. kritéria

Heuristické metody: vyhledávání vhodnosti problému a možnosti jeho řešení ve stanoveném systému

- často dobrovolná optimace optima

Heuristická metoda - stav mytí sivá reálnou hodnotou

- preferující stav s nejvyšší hodnotou

- mimo nálož

Greedy heuristiky - výhledově nejlepší lsd. optimia

- lo očekávané - POUŽITELNOST

OMEZENÍ a OPTIMALIZACE

PŘESNOST

→ Lokální - jeden abstraktní stav

Globální

- globální řešení na věc

DRUHÝ HEURISTIK

- KONSTRUKTIVNÍ - dle některého konfigurace a postupné přidávání

- ITERATIVNÍ - dle některého řešení v následujícím řešení a to mění

- DVOUFAZOVÉ - kombinace obou

LOKÁLNÍ METODY:

- lokální minimum - výsledek závisí na počátku

PRÉKONÁNÍ: → matematik - myšlení - paměť, čas, cena

jednoduché - tj. matematické

konkurenční - nejlepší řešení

PRAKTICKY - směra kritéria, $\rightarrow \epsilon$

- množství věcí

- ochlazování - dočasné zhlazení

- roztírání - také sekvence

PŘÍKLADY HEURISTIK:

- jeden abstrakt - množství věcí

- myšlení a výběr nejlepšího

- plán je všechny nejlepší, uloží ho a jede dál

NÁHODNÁ PROCHÁZKA

- Malodí sousek
- Není náplná, není systematická

PONZE NEJLEPŠÍ

- Soused a nej. cena
- je greedy
- není náplná - nemá optimum

PRVNÍ ZLEPŠENÍ

- Nějaký sused o lepší cenu
- greedy, není náplná
- Mací kouzlo lepší nelze vložit

GLOBALNÍ METODY:

- Používají náhodná na jednotlivce
- trvají hodně, nejsou hromadou silou
- Modelem je sebevražda

Signály, systému a vlastnosti, automatické řízení jeho počítačem

SYSTÉM - sadou souborů, které jsou interaktivní

- Sada vstupů - znaky $\in \Sigma$ - nelonečné čísla

ABECEADA - Σ - množina znaků

SIGNAL V DISK, CASE NA Σ - mechanický posloupnost pruhů Σ

Σ_S - množina signálů - tj. možné posloupnosti znaků $\in \Sigma$

SYSTÉM - je relace mezi signály Σ_1 a $\Sigma_0 \Rightarrow \subseteq \Sigma_1 \times \Sigma_0$

AUTOMAT - nelineární case - 5-tice (S, Σ_0, I, O, R)

- STAVY S
- POČ. STAVY S_0
- I, O
- Přidodávání relace R $(I, S) \times (O, S)$
- \rightarrow když vstup a stav je relace definována

VLASTNOSTI SYSTÉMU:

RECEPTIVITA: když vstupní signal existuje vstupní

KAUZALITA: výstup závisí na všech záhmách na minulosti

- výstup je stejný když vstupní Σ_i plní jen potřebné stupně

DETERM.: existuje když (I, S) máme jeden (O, S) a navíc je jeden joi. stav

\Rightarrow třídy ne mají žádnou funkci

BEZ PAMĚTI: výstup záleží pouze na aktuálním vstupu

LINEARITA:

$$\begin{array}{ccc} \xrightarrow{\quad S \quad} & \xrightarrow{\quad S \quad} & = & \xrightarrow{\quad + \quad} & \xrightarrow{\quad S \quad} \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ \xrightarrow{\quad x \quad} & \xrightarrow{\quad S \quad} & = & \xrightarrow{\quad S \quad} & \xrightarrow{\quad x \quad} \end{array}$$

AUTOMAT - reprezentace systému $\tilde{s}(i, o) \in \Sigma_i \times \Sigma_o$ | existuje $s \in \Sigma_S$ tak, že $s(0) \in S_0$ a když $\forall t \in \{0, 1, \dots, 3\}; (i(t), s(t), s(t+1), o(t)) \in R$ \Rightarrow R je relace

2 AUTOMATY jsou ekvivalentní když reprezentují stejný systém

EXTENDED STATE MACHINES: automaty s pamětí

HIERARCHICKÉ AUTOMATY: jednotky s jejich relativním sloužením

Kompozice systémů a automatů (distribuční a sýčití), synchronní modely

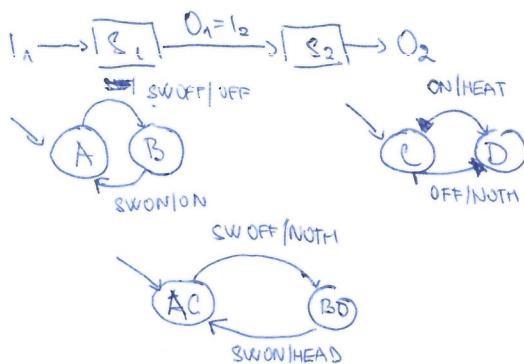
KOMPOZICE: PARALELNÍ → součinné řízení dva paralelní systémy
- v automatě je to jejich kartézský součin
a smíšeného redakčního stavu

ŽÁDNÉ NEDOSTUPNÉ STAVY

$$\begin{aligned} & I_1 \rightarrow \boxed{S_1} \rightarrow O_1 \quad (I_1, I_2) \times (O_1, O_2) \\ & I_2 \rightarrow \boxed{S_2} \rightarrow O_2 \end{aligned}$$

např: snížka a topení

KASKADOVITÁ



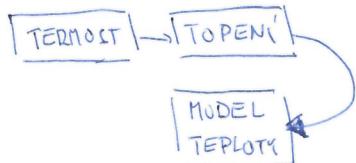
Pouze S_1 je reprezentován A_1

a $S_2 A_2$

Jak $S_1 \otimes S_2$ je
reprezentován $A_1 \otimes A_2$

SYNCHRONNÍ REAKT MODELY

- všechny komponenty



KOMPONENTY: ZPOZDĚNÍ

- v distribučních systémech zpozdění o krok
- ve sýčití systémech se používá integrator

$$O(0) = 0$$

$$O(k_2) = k(k_2 - 1)$$

VÝHLED. TABULKA:

$$R = \{x, 2x) | x \in \mathbb{Z}\}$$

$$R = \{(1, a), (1, b)\} \Rightarrow \text{zde dva minuty?}$$

SPOSOBE MODELOVÁNÍ

- signál je myší funkce $R^{\geq 0} \rightarrow S$

Σ^e_s je minima signálů ve sýč. čase

SYS VE SPS J. CASE: $\Sigma^e_1 \times \Sigma^e_0$

Počet minima/ máximální počet.

Vlastnost: jde o triviale funkci, neplatí všechny funkce mohou být myší

\Rightarrow myš je rekt. jde o hledání cestu - traťování diferenciální kmitic

Vektorné pole $f: S \rightarrow \mathbb{R}^m$

Diferenciálna rovnica treia $\dot{x} = f(x)$

- rešiť sa sústava smerov, ale celou sústavu $F: S \rightarrow \mathbb{R}^m$

Systém automatu je 5-tie (n, p, q, S₀, R)

R^m ... stanoví smer

R^p ... prístrojový smer

R^q ... kontaktný smer

S₀ ⊂ Rⁿ ... poc. stav

R ⊂ Rⁿ × R^m × R^p × R^q ... prel. relace

Automat reprezentuje systém $\Rightarrow ((\bar{x}, o) \in \Sigma_R^c \times \Sigma_R^c \mid \text{existuje } A \in \Sigma_S^c \text{ tak, že}$

$A(o) \in S_0 \text{ a } \forall t \in \mathbb{R}^{>0} (\bar{x}(t), s(t), \bar{o}(t), o(t)) \in R$

Systém systém $\Sigma(\bar{x}, o) \in \Sigma_R^c \times \Sigma_R^c \mid s(o) \in S_0 \text{ a } \forall t \in \mathbb{R}^{>0} \bar{x}(t) = f(A(t), \bar{x}(t))$
 $o(t) = g(\bar{x}(t), s(t))$

SHRNUTI' SPOZITÝCH:

vekt. pole: $f: S \rightarrow \mathbb{R}^m$

dif. rovnica $\dot{x} = f(x)$

$\forall t \quad \dot{x}(t) = f(x(t))$

Typologie ověření správnosti systému (testing, bounded model checking, unbounded model checking) a jejich základní principy

Temporalní logika - známý výraz & čase s mimož logika reprezentací

- Specifikace charakteru systému - Zjednodušení - modelový systém

CESTA - posouvat stav s následky

$$\Pi(i) \dots i\text{-ty stav}$$

$$\Pi^k \dots k\text{-ty sufix} \quad \circ \circ \circ \circ \circ \circ \circ \circ$$

OPERATORY ... lze kombinovat

- často používáno v programu

$X_p \dots$ neplatí

$GP \dots$ platí

$F_p \dots$ lze plnit alespoň jednou

$\phi V q \dots p$ platí, dokud neplatí q - UNTIL

$q V P \dots p$ platí, dokud někdy nebo minimálně neplatí q - RELEASE

BMC - zjistitelné, zda model splňuje specifikaci

\emptyset - formule LTL

Modelový systém splňuje \emptyset , pokud je cesta již splňuje

Problém - nejmenší může být několik desítek

- ověření je definováno sufiksem - ověření dalších cest Π

Postup - sledujeme postupně

- Maximální Π je G, F G platí v počtu 0

$BMC \not\models \infty$ ale $\infty \Rightarrow BMC \dots$ ověření obecně je silněji než BMC

Unbounded MC - princip indukce - 3 podmínky pro univ. V

1. V obsahuje všechny stavy - počáteční

2. Neexistuje následk se stavem $v \in V$ mimo V - neexistence

3. Všechny stavy musí splňovat požadovanou funkci - stav $\models V$

INVARIANT - libovolná množina obsahující všechny dosažitelné stavy

INDUKT. INVARIANT ... libovolná množina obsahující:

1. V obsahuje všechny stavy - počáteční
2. Neexistuje následk mimo V

Boolovaho splnitelnost s algoritmy a jich možnosti ~ BMC

- subčinní na problém SAT
- Reprezentace BMC pomocí funkcií logické logiky

$$\{\{A, B, C\}, \{A\}, \{(A, B), (B, B), (B, C), (C, A)\}\}$$

$$\Rightarrow \text{stav si zadáváme} \Rightarrow \text{máme 2 slity} \quad \begin{array}{ll} A & 10 \\ B & 01 \\ C & 00 \end{array}$$

$$\Rightarrow \text{pojíštěné ekvivalentní příkody: } (\mathbb{B}^2, \{\{1, 0\}\}, \{(10, 01), (01, 01), (00, 10)\})$$

\Rightarrow tabulkově vedený logický automat

\Rightarrow elenou uložit, neplatí BMC (\emptyset, n) ... ověřit ko. čestn. \Rightarrow negaci?

\Rightarrow (POČ-STAVY) \circ (PŘECHODY) \circ (NEGACE PODMÍNEK)

\Rightarrow fzn. abstraktní, kde je leist je ovlivněn, které retežení jsou řešené počínaje

SAT problém - splnitelnost Bool funkce

$$(X+Y+Z)(X+Z)(Y+Z) \\ \underbrace{\quad}_{\text{KLAVÍZKY}} \quad \underbrace{\quad}_{\text{literál}}$$

ALGORITMY = Maximální - 2^n - da se sestavit strukturovanou tabulkou

DPLL - alejtvařitel

a) jedna literál - ihned ochotím

b) jistna kmenina lze jin. řešit + neg - ihned ochotím

c) potupě sledující - lze to nejdéle dlektiva zjistit

\Rightarrow Takej všechny stromy

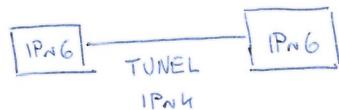
Genetické algoritmy - PAA

Lokální problematika - WalkSAT

Architektura, protokoly a technologie IPv6

- ISO/OSI - 1. fyzický 802
 2. linkový - MAC
 3. síťový - IPv4/IPv6
 4. transmisi - TCP/UDP
 5. síťový
 6. portový
 7. aplikací
- ⇒ jednotlivé vrstvy nemají vlastní kontext
 ⇒ → k uživateli obdrženou - např. QoS, routing

- Dnes má ji IPv6 na první dvojici adresy podporu
- V minulosti nutné používat tunelování - např. TEREDO



- Lze slyšet řeč, že IPv4 adresování je podmínkou IPv6

- Fyzika ⇒ Bluetooth, ethernet, 802.11, SONET, ISDN
 ⇒ základní principy

SONET - optika
 ⇒ HUB

- Linkový ⇒ Mac. adresy
 ⇒ riziko bezpečnosti ⇒ ARP spoofing, man-in-the-middle

⇒ SWITCH

⇒ meziříčí jízda a Mae

⇒ čidelce dle ⇒ Broadcast

- Síťový ⇒ IP adresa - IPv4/IPv6
 ⇒ riziko bezpečnosti
 ⇒ PROTOKOLY: EGP, EIGRP, ICMP, RIP - vnitřní protokol
 ⇒ vnitřní vs. vnější síť ⇒ rozdíl subší
 ⇒ UNICAST / MULTICAST / BROADCAST

- Transmisi ⇒ TCP
 - statisní - mzdí se s počtem
 - řízené - jde o přenosník
 - QoS, kontroly náloží
 - PORTY - multiplexing

- UDP - bez sponění
 - funkce se odvozují o nezávazky
 - implementaci mzdí se na druhé straně

Relacií - manžerami stave řílace

- AppleTalk
- SLP, RPL, NetBIOS, H.2H5
- níže se bezpečnost - autentizace, autorizace

Přenášení - šifrování, sifrování, kodování

Aplikací - nejjednodušší - P2P HTTP | DNS | SMTP | POP3 | IMAP

\Rightarrow je možné tunelovat - uložit nějaké jeho kód typu do jiného
a pak různé rozbalit a jít dál

IPv6 \Rightarrow větší adresní pole 128bitů = 2^{128}

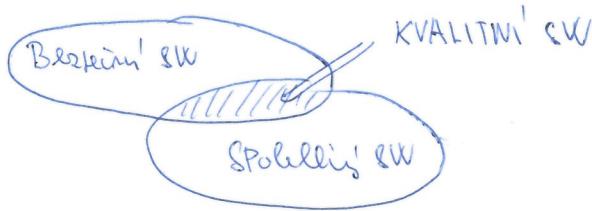
\Rightarrow mnohem mnohem \Rightarrow z cíti i hnedi bezpečnosti a používání

\Rightarrow mališiny a ilustrace

\Rightarrow používání velké MTU a jiných faktorů

Modelování bezpečnosti sítě - STRIDE a DREAD

Motivace ktočíka - infuze, peněze, just-for-fun



Threat - možné ohrožení bezpečnosti - píšína

Vulnerability - vznikávající - bezpečnostní slabina

Security feature + Security

Casto je bezpečnost dohledá až zřejmě %

Velkou roli hraje a bezpečnosti je dležitá mít od začátku ▽

Fáze:

1. Design
2. Implementace základních funkcí
3. Implementace ostatních funkcí
4. Implementace bezpečnosti
5. Testování
6. Bug Fixing
7. Release

SPATNÉ ▽

Bezpečnost musí být navržena
místo po fázi designu

SD - Security by design

- primitivní least privilege
- koncience se musí vyplňovat bezpečnostním trendem
- penetracní testy modulu
- Defaultní instalace je nebezpečnější
- Give up privileges you do not need

Principy - minimizace - attack surface (Ports | ALL | RPC | files)

- multilayer security
- lze uvažovat kompatibilitu?

Input = Evil

Obfuscace (Sec. by obscurity) nevyžaduje bezpečnost?

Není dobré mít vlastní hrdá data - XSS

Threat modelling: Studie → dekompozice → identifikace → prioritizaci

STRIDE - spoofing - falešný identita
tampering - modifikace
replication - odmítnutí (masake)
inf. disclosure
denial of service
elevation - neautORIZOVANÝ PŘístup

KLASIFIKAČNÍ SCHÉMA
⇒ TYP ZRANITELNOSTI

DREAD - Damage potential -

Reproducibility - snadné reprodukce?
Exploitability
Affected users - kolik uživatelů
Discoverability - jak snadno objevit

KLASIFIKAČNÍ DIZAJN

⇒ obchodní a
aplikativní formy

RISK = PROBABILITY • SEVERITY

Příjem! buffer, jde dle a Metody ohani

STACK OVERFLOW

Stack buffer overflow - Nej zásobník lze uložit něco dal, než tamy říká osta programu \Rightarrow můžešené dílčité sečítání a řečec
 \Rightarrow například gets nebo malý zásobník

Pokud máme možnost jítci - můžeš na zásobníku něco EIP - můžete cíl RETu

Co všechno lze učinit: můžete když LIBC a string /bin/bash
 něco SETH a myslat si jinou

HEAP OVERFLOW

Mánočeksi na exploitaci - Můžeš znát implementaci aplikací
 - Nej - jak a jestli se používají bloky a podobné

PARAMS
RET ADR
EBP
SAVED REGS
CANARY
LOCALS

OCHRANA

OS - data execution prevention - DEP - NX lze nechat
 - address space layout randomization - ASLR

SW - Detekce ko zásobníku - STACK CANARIES - Záčatek funkce - konec funkce
 Safe exception handling
 Multiple stacks -> variables
 return address
 Safe libraries

\nearrow podpora OS
 \searrow podpora SW

Pointer encoding - XOR kointen na funkce
 nejlepším řešením

Return oriented Programming - ROP - Můžeš na STACK, když nelze spustit

- exploit z hrdin, který mi je v programu mítens!
- hrdinu mítens je možné DEP

Běh programu ní můžete ovládat

Program má dešet jen to, kdo co je vystaví

-> sítové nutné oprávnění (severní možnosti)

Přeč následující elevaci má? OS nedává na výběr?

- Specifické následující výsledky oprávnění

- ACE - Access Control

- LSA secrets - Local Sys. Authority

- uživatelské politiky a nastavení systému

- uživatelské tituly data

Jak má to? Pojďme seznámit se s výsledkem a založit
Nejdříve privilegované, než funkce API:

- data

- funkce API

Zhodnotím soubornouho uživatele

Přidružené token, který daje uživateli má

Odstranění následující akce, co udělal elevaci

- výběr

- vložení do nové binární

Vytrávají si další token - Restricted token si mohou ovládat

UAC - upřesnění, když uživatel císaří token

- uživatelské priviléje a možnost mít méně oprávnění

- minimální elevace

- daremka funkce virtuálního SW, který UAC nemá

- file system

- registr

Proces může se manifikovat načerpat výsledky jiných oprávnění

Indice uživatelského portfolia

- další dny užívání

- procesy → můžou mít mnoho koncipientů s výsledky

- Nežádatelný - validovat její HWND

- poslat zprávu

- hocharvat je

- injektovat DLL

- Náhodný - čistí schůdky paměti

Global Alloc table

Přístup do schůdek

Kreslit na obrazovku

⇒ Tomuže se někdo může integrity

TALKED

TRUE?

MICHAL

✓

KLAŠA

?

MAŘÍT

?

HONZA

TOM

KUBA ŠEKY

?

ONDRA L.

?

ADAM P.

?

JANČA

~~PAŠA~~
TOMÁŠ

?