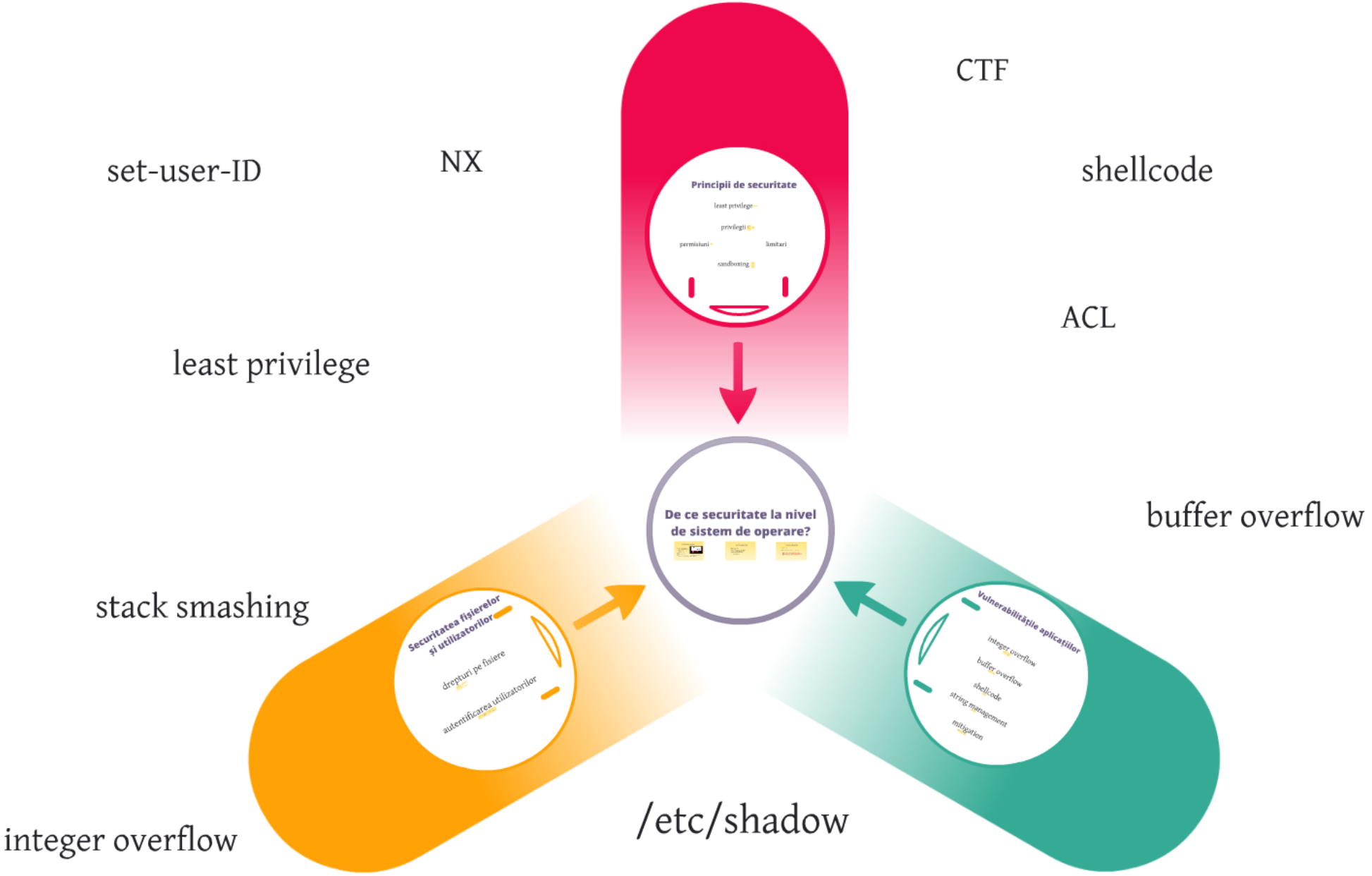
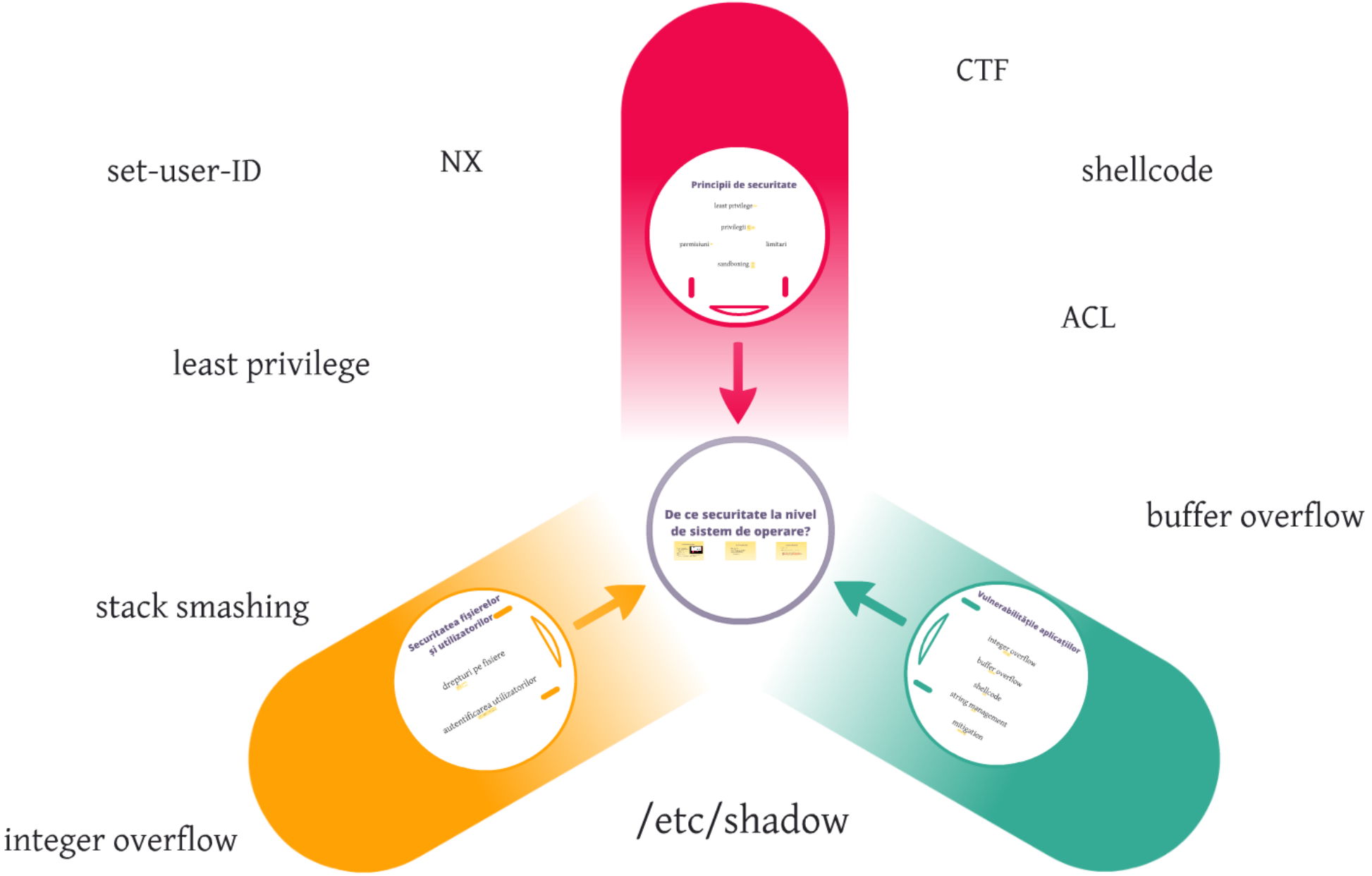


Securitatea sistemelor de operare

SO Curs 12



Securitatea sistemelor de operare

SO Curs 12

De ce securitate la nivel de sistem de operare?

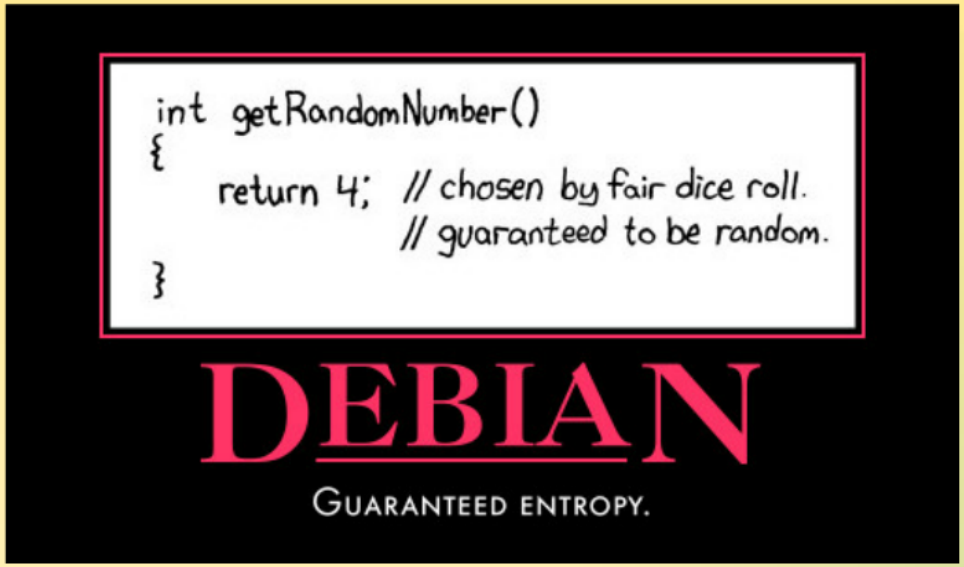
Debian Open SSL Bug

aparut in 2006, descoperit prin 2008
- http://lwn.net/Articles/282038/

cum a aparut bug-ul?
- test Valgrind peste OpenSSL
- unitialized memory

decizie
- stergerea a două linii de cod

problema
- o linie importanta pentru entropia de numere aleatoare (RNG)



http://www.gergely.risko.hu/debian-dsa1571.en.html

Linux vmsplice bug

Linux kernel 2.6.17-2.6.24.1
11 februarie 2008
combinatie de integer overflow si buffer overflow in subsistemul de memory management al nucleuluiz
http://www.milw0rm.com/exploits/5092

oferea prompt de root

Password breaking

Crack me if you can, DEFCON

- http://contest.korelogic.com/

http://arstechnica.com/security/2012/08/passwords-under-assault/

> *This $12,000 computer, dubbed Project Erebus v2.5 by creator d3ad0ne, contains eight AMD Radeon HD7970 GPU cards. Running version 0.10 of oclHashcat-lite, it requires just 12 hours to brute force the entire keyspace for any eight-character password containing upper- or lower-case letters, digits or symbols. It aided Team Hashcat in winning this year's Crack Me If You Can contest.*

NX

A

Principii de securitate

least privilege

privilegii

permisiuni

limitari

sandboxing

De ce securitate la nivel de sistem de operare?

fișierelor

atorilor

Vulnerabilitățile aplicați

Principii de securitate

least privilege

privilegii

permisiuni

limitari

sandboxing

Kernel mode and user mode

Instructiunile privilegiate sunt executate in spatiul kernel

- accesul la I/O
- alocarea de resurse
- handler-ele de întrerupere
- gestiunea sistemului

Suportul procesorului

- niveluri de privilegiu (rings)
- x86: nivelul 0 (kernel), nivelul 3 (user)

Privilegii de utilizator

root (UID 0): drepturi complete in sistem
poate realiza toate tipurile de actiuni

privilegii utilizator: notiunea de owner

- schimbarea drepturilor de acces pe fisierul detinut

privilegiile sunt atribuite doar agentului, nu au legatura cu obiectul

Capabilitati

o cheie asociata unor actiuni privilegiate
pot fi interschimbate intre agenti
nu este un lucru obisnuit in sistemele de operare actuale

capabilitati POSIX (IEEE 1003.1e)

- CAP_NET_BIND_SERVICE
- CAP_SYS_CHROOT
- CAP_NET_RAW

man 7 capabilities

Principii de securitate

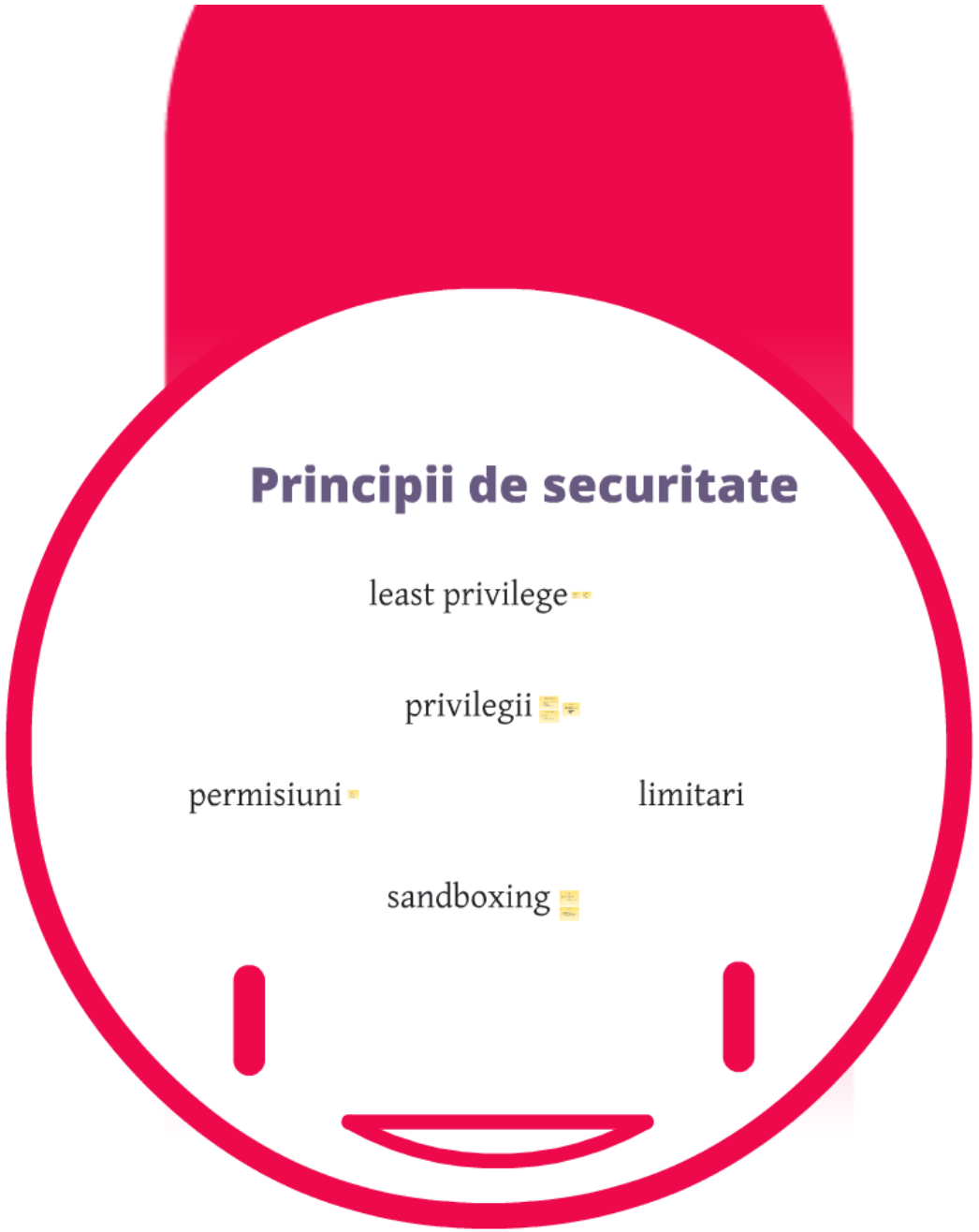
least privilege

privilegii

permisiuni

limitari

sandboxing



chroot

modifică directorul radacina asociat procesului
nu se poate accesa un director/fisier din afara ierarhiei impuse
chroot jail

comanda chroot[3]
apelul chroot: chroot("/var/spool/postfix");

Virtualizare

sandboxing complet
sistem de operare, hardware virtualizat
configuratiile din masina virtuala nu sunt vizibile in exterior

cgroups in Linux (LXC) pentru sandboxing

Principii de securitate

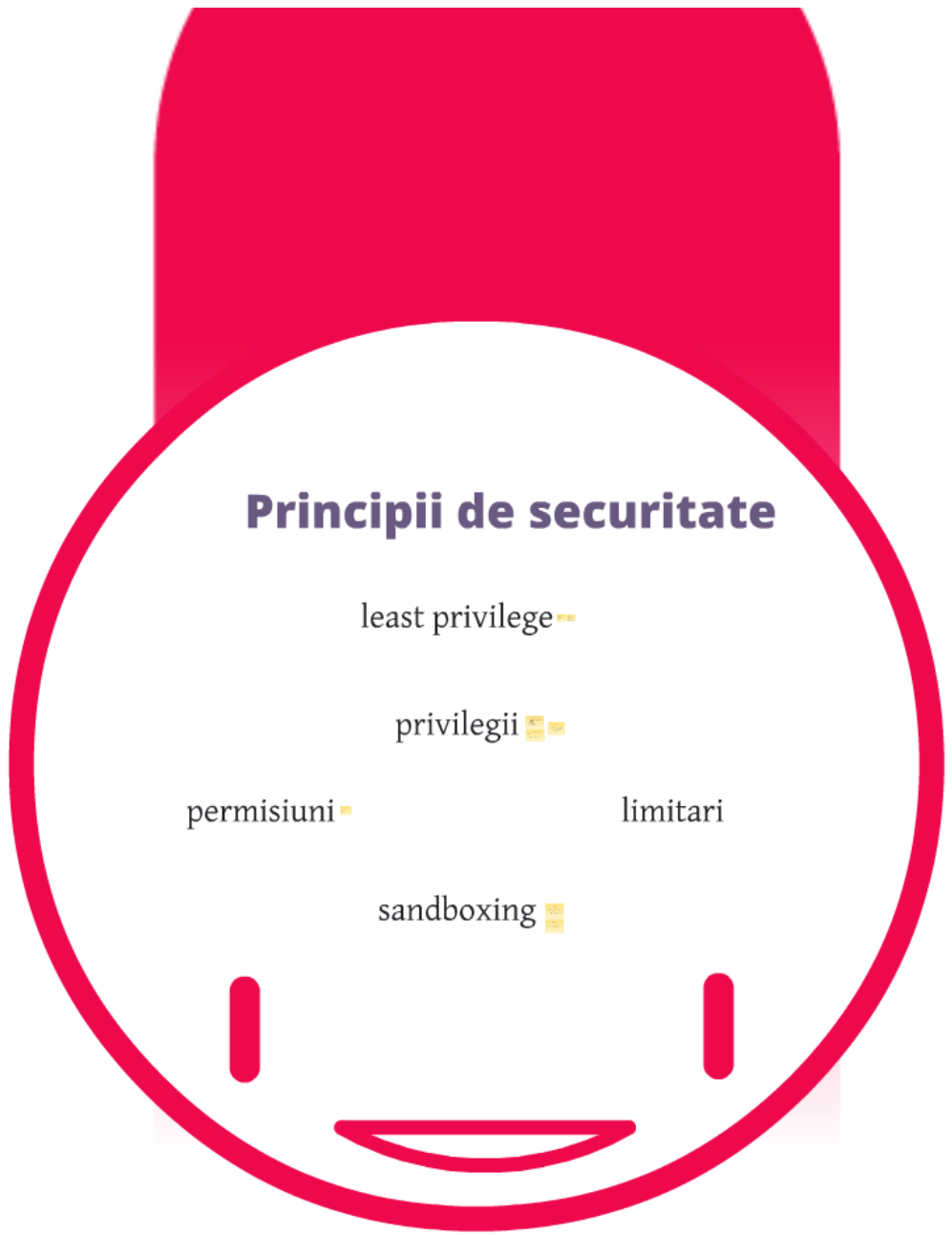
least privilege

privilegii

permisiuni

limitari

sandboxing



ACL

access control lists
permiuni aferente unui agent (subiect) si unui obiect
stocate in obiect in forma unui vector de perechi

- subiect
- permisiuni

implicit in Windows
forma redusa pe Unix

- grupare subiecti: user, group, others
- drepturi mai putine: read, execute, write

Principii de securitate

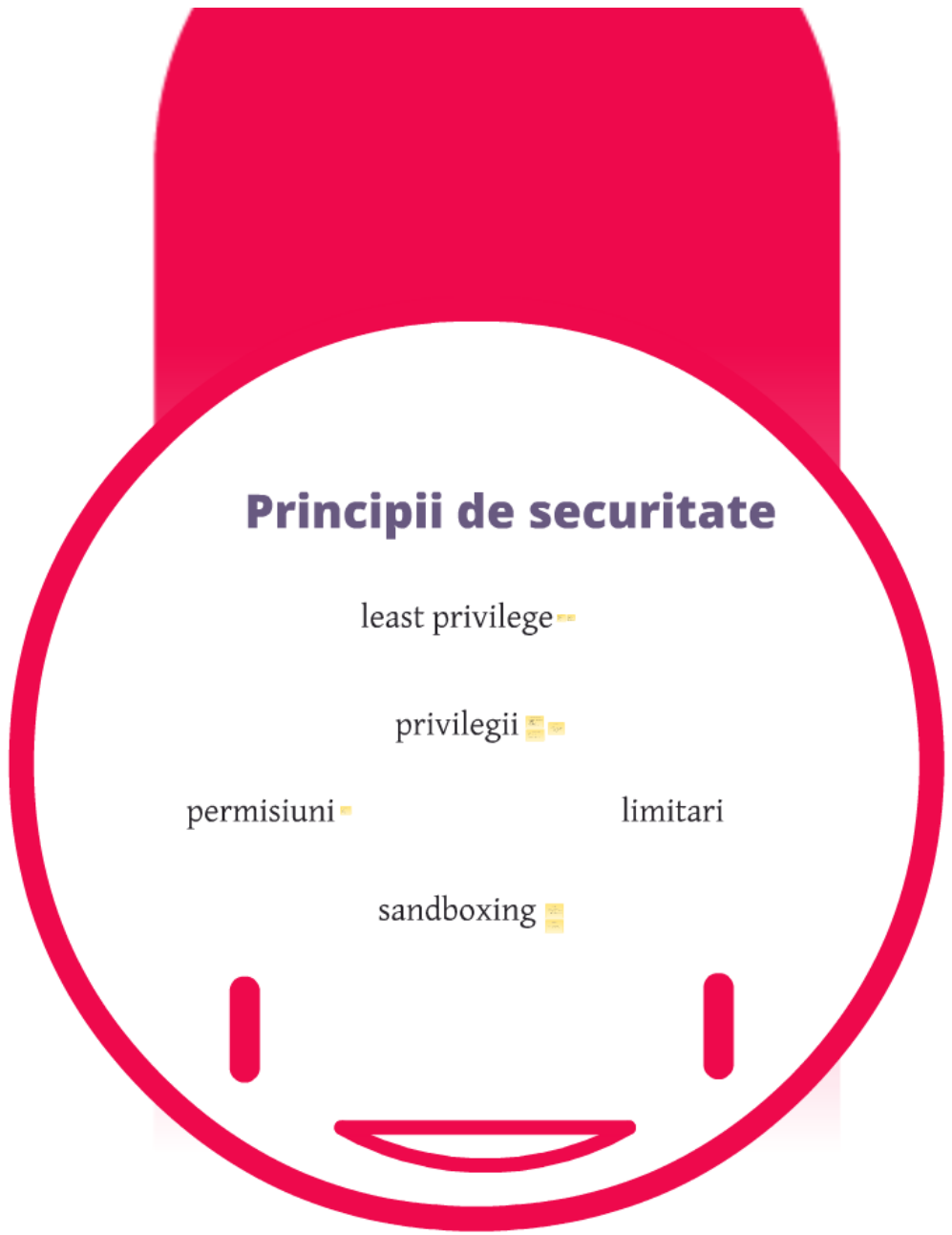
least privilege

privilegii

permisiuni

limitari

sandboxing



set-user-ID

privilege escalation

actiuni necesare unui utilizator obinsuit:

- scrierea in /etc/passwd
- citirea /etc/shadow
- lucrul cu socketi raw

un proces are

- real user ID (UID)
- effective user ID (EUID)

EUID-ul este cel al detinatorului pentru executabile cu bitul set-user-ID configurat

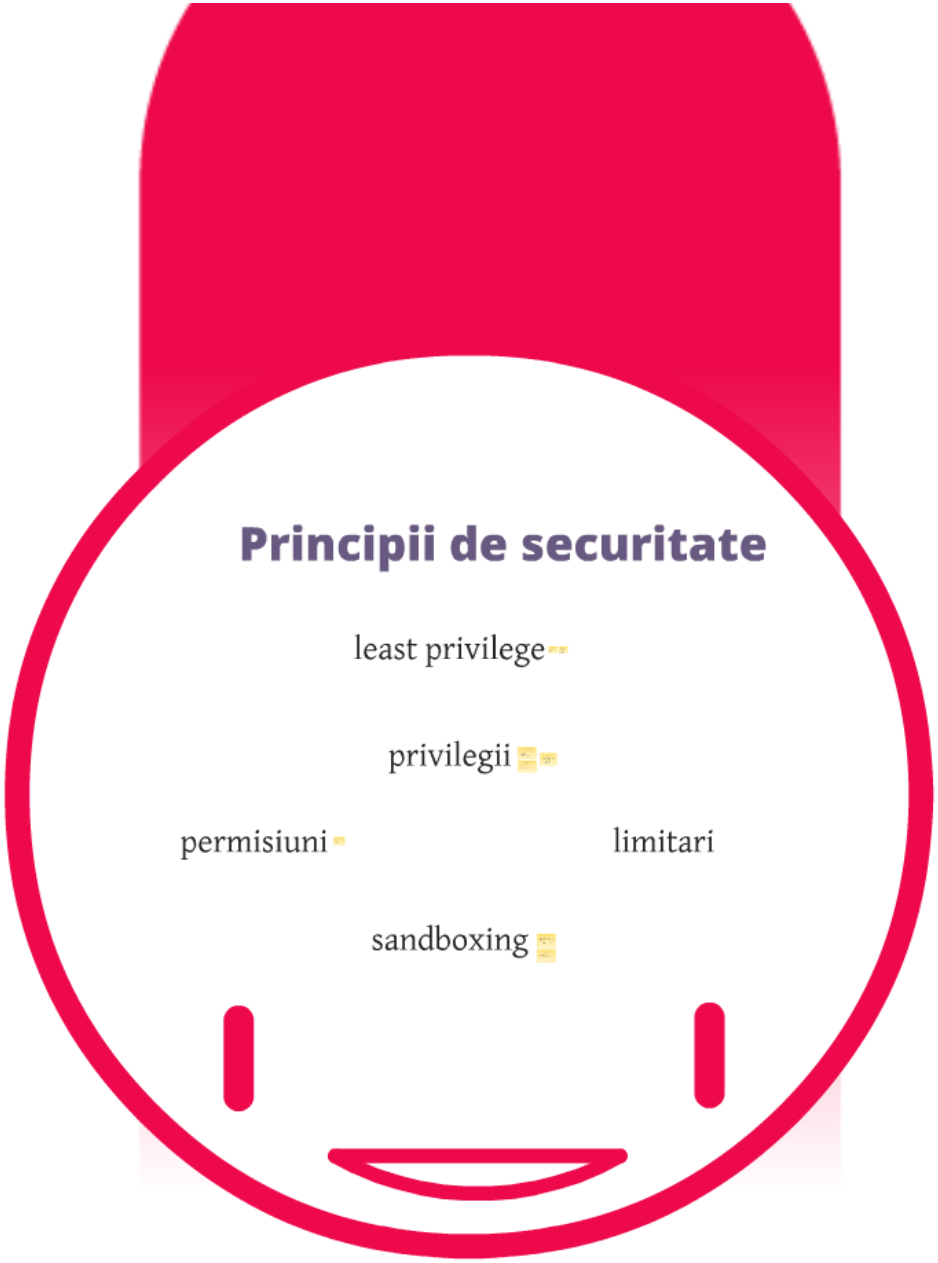
setuid

setuid, seteuid, setreuid, setresuid
- apeluri folosite pentru actualizarea EUID, UID

downgrade temporar al EUID
- proces neprivilegiat
- poate reface EUID

downgrade permanent al EUID
- procesul pierde privilegiile
- recomandat atunci cand nu mai e necesar

procesele create din executabile cu bitul set-user-ID activ sunt principala tinta de atacuri

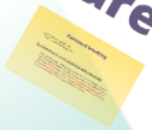

Principii de securitate
least privilege
privilegii
permisiuni
limitari
sandboxing

Securitatea fișierelor și utilizatorilor

drepturi pe fisiere

autentificarea utilizatorilor

De ce securitate la nivel de sistem de operare?



Securitatea fișierelor și utilizatorilor

drepturi pe fisiere

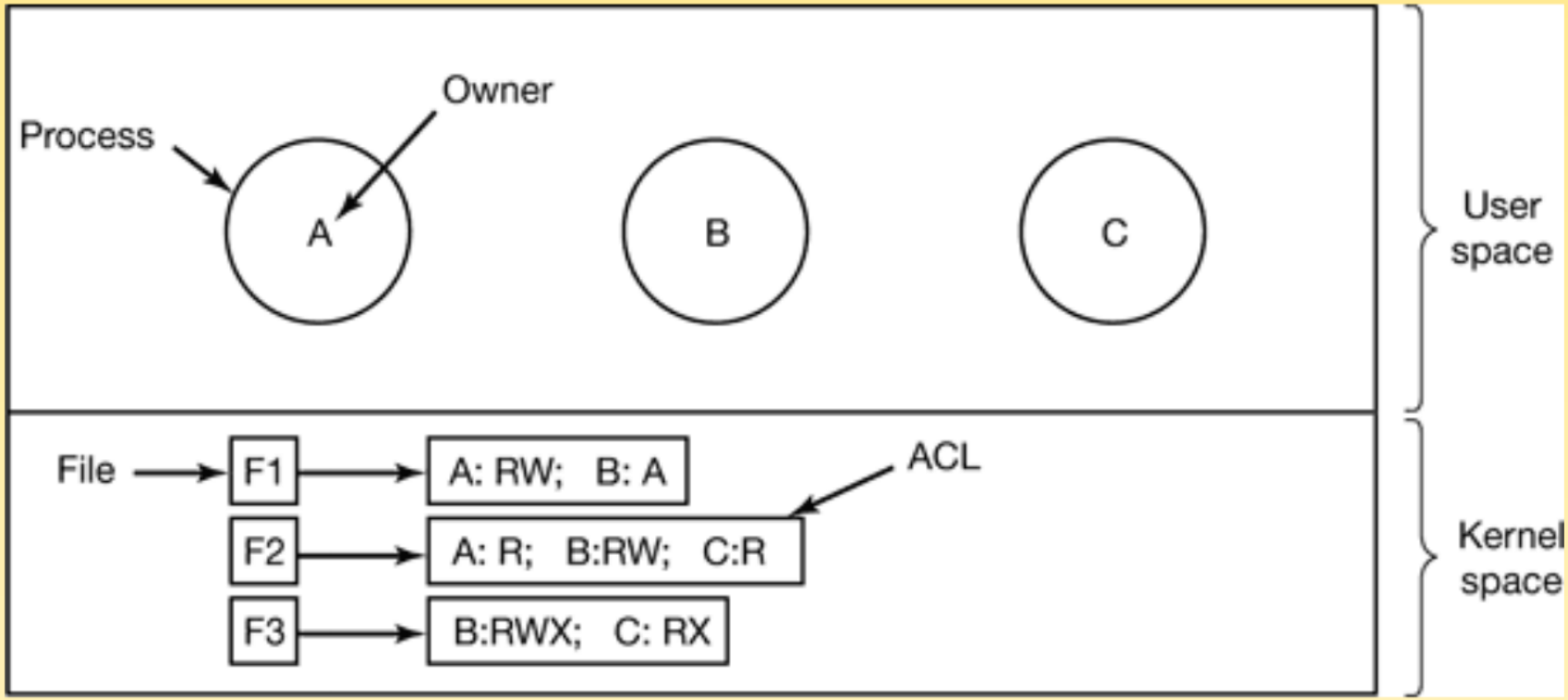
autentificarea utilizatorilor

Drepturi pe fisiere

ce actiuni poate face un utilizator cu un fisier
asociere de forma (subiect, obiect, permisiuni)

pe fisier: citire, scriere, stergere, executie
pe director: creare fisier, listare, stergere fisier, parcurgere

ACL



POSIX ACL

- implementate pe sisteme de fisiere Linux cu extended attributes
- getfacl, setfacl

Drepturi pe fisiere in Windows

- ACL pe NTFS
- read, write, list, read and execute, modify, full control

Securitatea fișierelor și utilizatorilor

drepturi pe fisiere

autentificarea utilizatorilor

/etc/passwd si /etc/shadow

/etc/passwd

- user:password_hash:uid:gid:...

problema

- accesul utilizatorilor (nevoie de informatii diferite de password_hash)

/etc/shadow

- user:password_hash:...
- security enforcing
- număr de zile între schimbat parola
- număr de zile după care contul este dezactivat

Password hash

cryptographic hash function

- one way
- pentru verificare se aplica functia peste parola
  - se verifica hash-ul obtinut cu cel stocat

stocare (/etc/shadow):

- idsalt$encrypted
- id: ID: 1 (MD5), 2a (Blowfish), 5 (SHA-256), 6 (SHA-512)

salt

se perturba parola cu un string aleator

se aplica hash function-ul pe concatenarea dintre salt si parola furnizata
salt-ul este stocat in fiserul de parole

cand ai un fisier cu mai multe parole e mai dificil de spart

- se concateneaza salt-ul fiecareia cu parola presupusa

One time passwords (OTP)

time-synchronized OTP
RSA SecurID

algoritm matematic
- s – initial seed
- f – one-way function

cryptographic hash function
- it is easy to compute the hash value for any given message,
- it is infeasible to find a message that has a given hash,
- it is infeasible to modify a message without changing its hash,
- it is infeasible to find two different messages with the same hash.

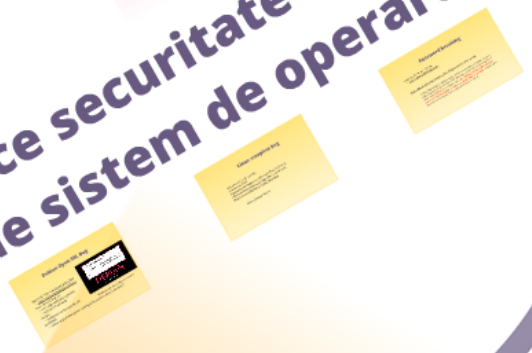
parolele sunt transmise in ordinea: f(f(f(f(...f(s)...))), ... f(f(f(s))), f(f(s)), f(s)

Securitatea fișierelor și utilizatorilor

drepturi pe fisiere

autentificarea utilizatorilor

De ce securitate la nivel de sistem de operare?



Vulnerabilitățile aplicațiilor

- integer overflow
- buffer overflow
- shellcode
- string management
- mitigation

Vulnerabilitățile aplicațiilor

integer overflow

buffer overflow

shellcode

string management

mitigation

Integer overflow

o operatie aritmetica depaseste spatiul alocat unui tip de date

- 8 biti – 255
- 16 biti – 65535

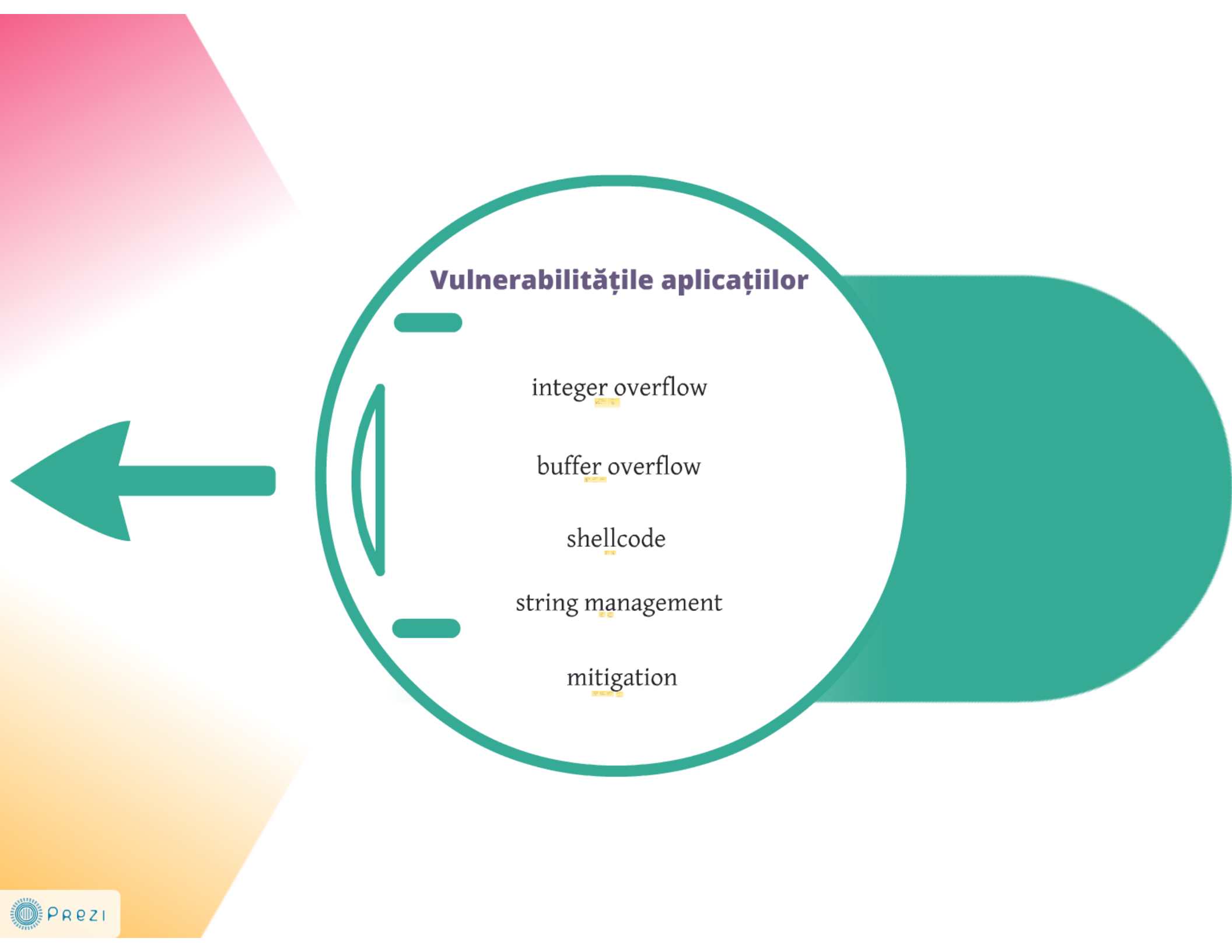
unexpected behavior

pentru întregi cu semn

- poate lua valoare negativă (nu mai poate fi folosit ca index)

Integer comparison

```
int a = -1;
unsigned int b = 20;
if (a < b) {
        /* expected behavior */
}
else {
        /* unexpected behavior */
}
```


Vulnerabilitățile aplicațiilor
integer overflow
buffer overflow
shellcode
string management
mitigation

Buffer overflow

suprascrierea unei adrese dincolo de limita unui buffer
buffer: array sau string

ce se poate suprascrie?

- un intreg (comportament nedorit)
- un pointer de functie
- adresa de retur de pe stiva

unde are loc overflow-ul?

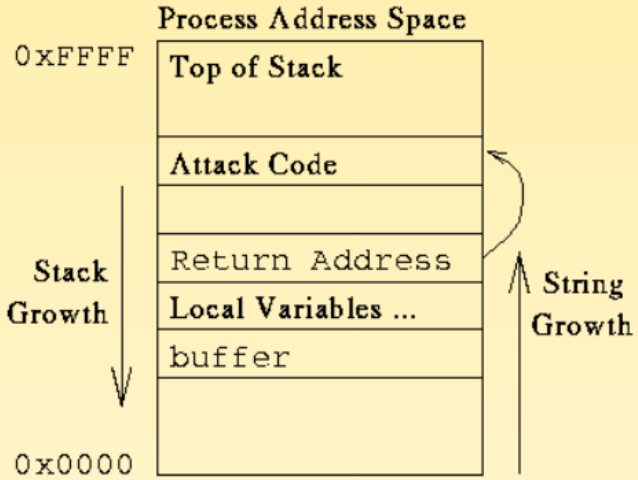
- pe stiva
- pe heap
- in zona de date

Stack buffer overflow

se suprascrie un buffer aflat pe stiva
se urmareste suprascrierea adresei de retur a functiei

se face jump la codul atactorului

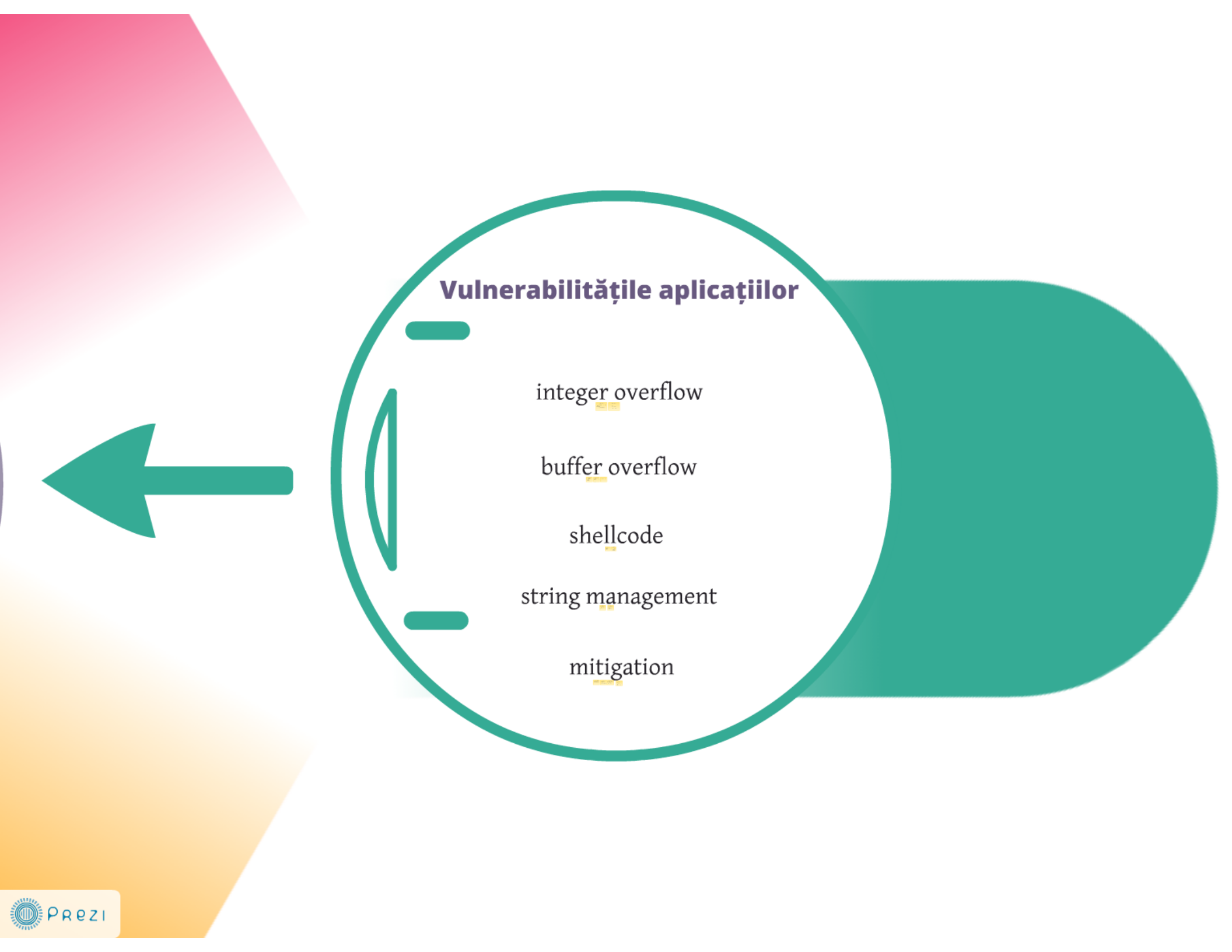
- de multe ori aflat pe stiva, in buffer, in forma unui shellcode



http://static.usenix.org/publications/library/proceedings/sec98/full_papers/cowan/cowan_html/node3.html

Return-to-libc Attack

se face jump la o functie din libc
nu este nevoie de shellcode, de scrierea de cod pe stiva
de obicei se face jump la system


Vulnerabilitățile aplicațiilor
integer overflow
buffer overflow
shellcode
string management
mitigation

Shellcode

in general se face jump chiar pe stiva
- in buffer
- intr-o variabila de mediu

se completeaza cu instructiuni de atac (shellcode)
- de obicei se executa exec("/bin/sh")

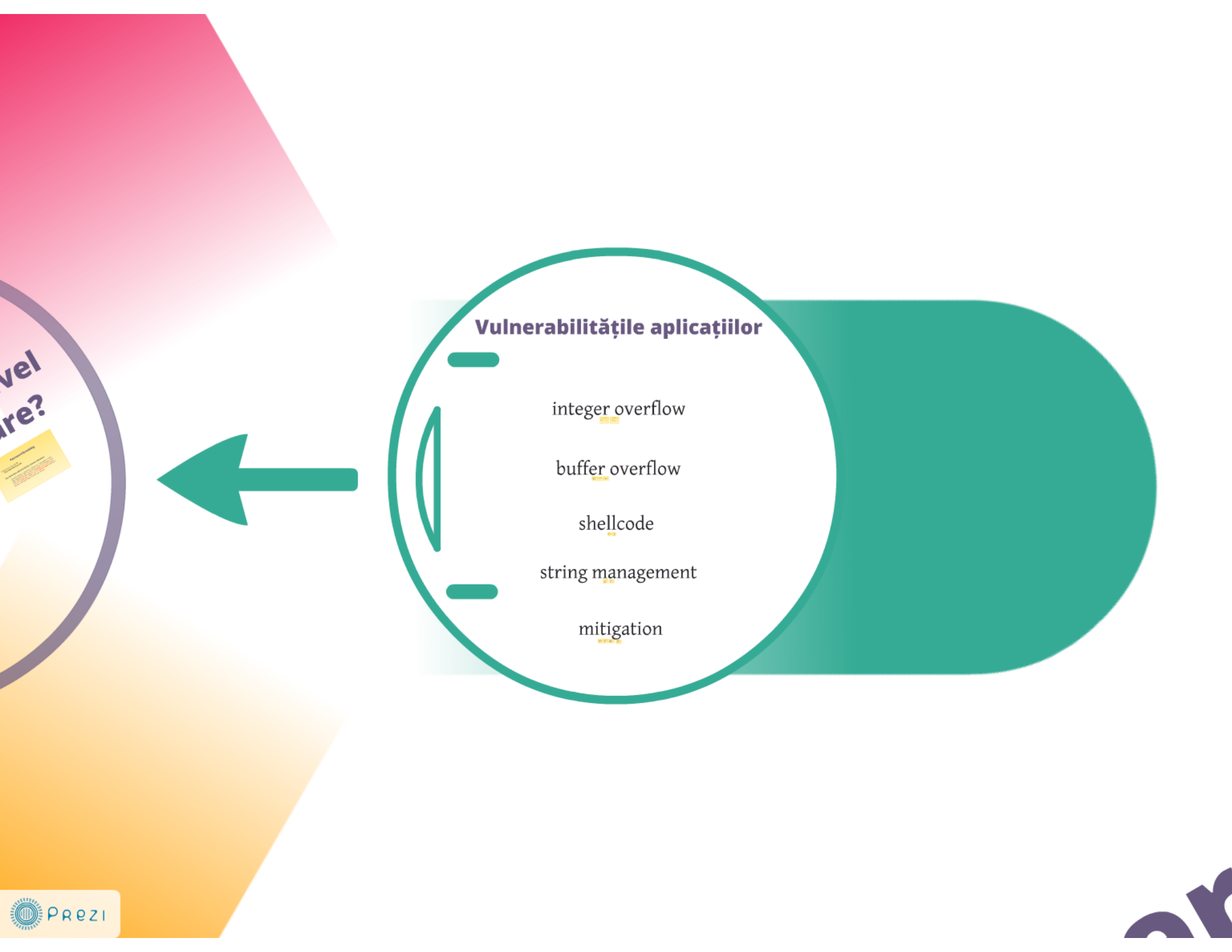
probleme posibile:
- bufferul este prea mic
  - shellcode mic sau din doua parti
- exista caractere NUL in sir
  - se foloseste 'xorl eax, eax' in loc de 'movl 0, eax'

Exemplu de shellcode

```
char shellcode[] =

                // setuid(0);
    "\x31\xdb"        // xorl %ebx,%ebx
    "\x8d\x43\x17"    // leal 0x17(%ebx),%eax
    "\xcd\x80"        // int $0x80

                // exec('/bin/sh');
    "\x31\xd2"        // xorl %edx,%edx
    "\x52"            // pushl %edx
    "\x68\x6e\x2f\x73\x68" // pushl $0x68732f6e
    "\x68\x2f\x2f\x62\x69" // pushl $0x69622f2f
    "\x89\xe3"        // movl %esp,%ebx
    "\x52"            // pushl %edx
    "\x53"            // pushl %ebx
    "\x89\xe1"        // movl %esp,%ecx
    "\xb0\x0b"        // movb $0xb,%al
    "\xcd\x80";       // int $0x80
```


Vulnerabilitățile aplicațiilor
integer overflow
buffer overflow
shellcode
string management
mitigation

Siruri

un vector de caractere NUL-terminat

probleme cu siruri

- suprascriere dincolo de limita (overflow)
- nu sunt NUL terminate
- trunchiere siruri
- caractere nevalide (sanitization)

reguli:

- trebuie sa stii permanent dimensiunea sirului
- sirulire trebuie sa fie NUL-terminate

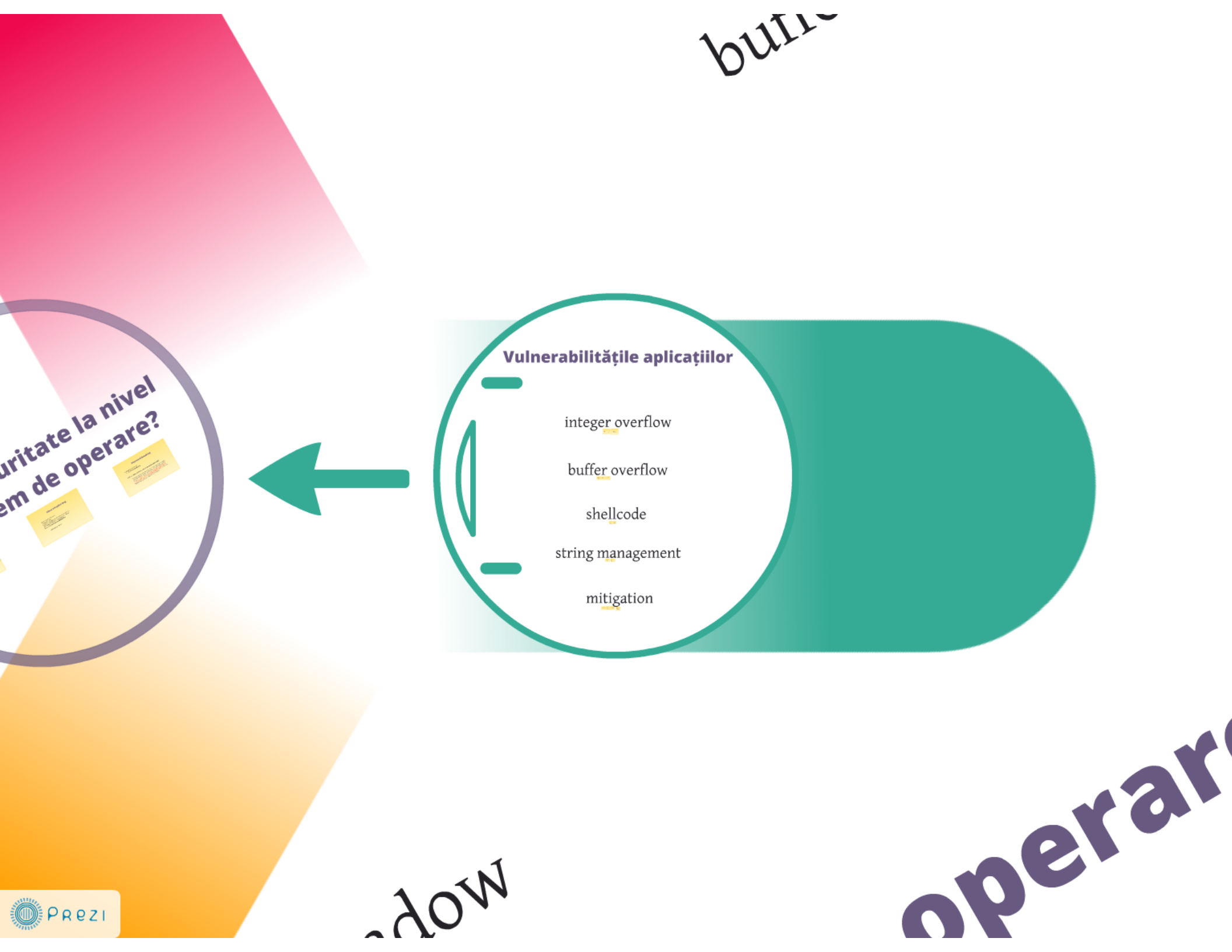
Functii de lucru cu siruri

gets
- man pages: “Never use gets”
- nu se opreste la capacitatea bufferului
- alternativa fgets

strcpy, strcat
- pot conduce la buffer overflow
- trebuie stiuta lungimea sirului

strncpy, strncat
- ineficiente (dacă se cunoaste dimensiunea sirului)
- nu se adaugă automat NUL terminatorul

strcpy_s, strcat_s
- se transmite dimensiunea sirului destinatie
- esueaza dacă sirul destinatie nu este suficient de mare
- pot trunchia sirul


Vulnerabilitățile aplicațiilor
integer overflow
buffer overflow
shellcode
string management
mitigation

Non-executable stack

implementare in software (W^X, PaX, ExecShield)
implementare hardware: flag-ul NX

zone de memorie (stiva) nu sunt executabile
nu se poate executa cod de pe stiva (shellcode)

dar ...

- se poate executa cod de altundeva
- se poate face return-to-libc attack

ASLR

Address Space Layout Randomization
se pornesc stiva, heap-ul de la valori aleatoare

util pe 64 de biti
relevanta scazuta pe 32 de biti

ajuta la prevenirea atacurilor de tip return-to-libc si altele

Stack Smashing Protection

se plaseaza o valoare intre variabilele locale si adresa de retur
canary value
stack buffer overflow suprascrie canary value

la iesirea din functie se verifica valoarea

- daca este schimbata a fost buffer overflow - se termina programul

optiunea '-fstack-protector' la gcc

CTF

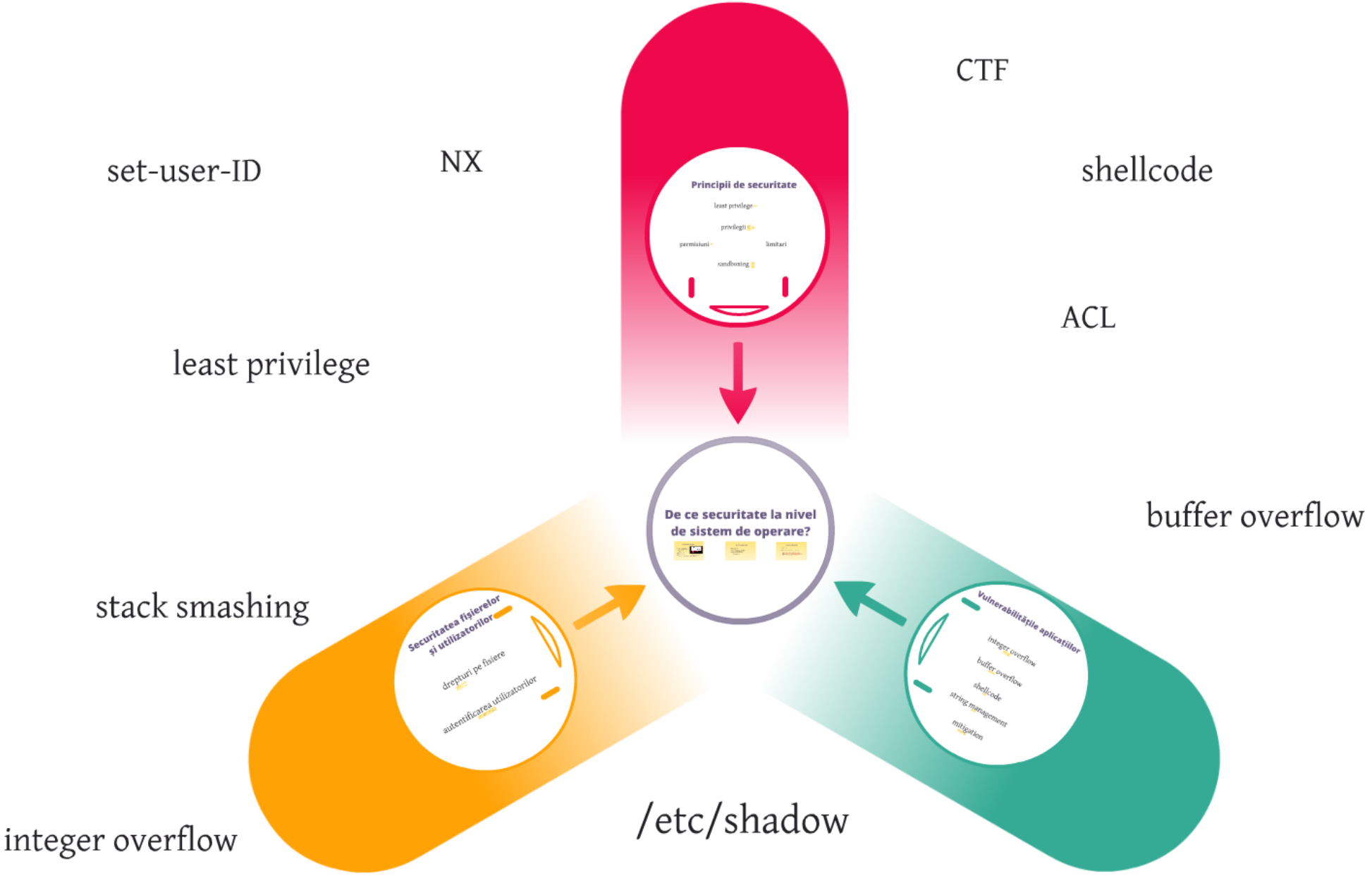
Capture the Flag
competitii de securitate

crypto, exploit, reverse, pwnable, stega, web

https://ctftime.org/

war games:

- http://www.smashthestack.org/ (IO)
- http://www.overthewire.org/wargames/



Securitatea sistemelor de operare

SO Curs 12