Tema

Se concentreaza asupra stilul de codare defensiv, orientat spre securitate si identificarea unor posibile vulnerabilitati. Implementarile (in C si PHP) trebuie sa respecte indicatiile primite in acest sens la cursuri si laboratoare. Tema consta in:

(Code Review, Vulnerability search)

Identificarea a cat mai multe vulnerabilitati in sursele primite in arhiva (ClientServer). Fiecare descriere a unei vulnerabilitati va contine cel putin urmatoarele informatii prezentate in sablon. Descrierea vulnerabilitatilor gasite se va include in arhiva temei, intr-un fisier *CodeReview.txt*.

Sablon descriere vulnerabilitate:

Numar: numar curent vulnerabilitate identificata, porneste de la 1

Localizare: fisierul sursa in care a fost identificata, functia, linia de cod

Tip vulnerabilitate: de ce tip este vulnerabilitatea

Metoda: prin ce metoda ati identificat-o (ex. code review, fuzzing, debugging, etc)

Severitate: cat credeti ca e de grava vulnerabilitatea (1 – cel mai putin grava, 4 – cea mai grava)

Riscuri: descrieti pe scurt ce s-ar putea intampla daca este exploatata vulnerabilitatea

Remediere: descrieti pe scurt cum se poate remedia vulnerabilitatea

Exemplu de descriere vulnerabilitate:

Numar: 1

Localizare: main.c, functia main, linia 35

Tip vulnerabilitate: buffer overflow, *sprintf* fara verificarea dimensiunii maxime a string-ului

Metoda: debugging cu Ollydbg

Severitate: 2 (exploatarea e doar locala, neprivilegiata)

Riscuri: prin exploatare (locala) se poate executa cod arbitrar in interiorul acestui proces (neprivilegiat)

Remediere: verificarea dimensiunii maxime a string-ului inainte de a se folosi *sprintf*

(C/C++)

Implementarea unei comenzi „**createmsg** xyz” care va deschide / crea fisierul (xyz.txt) si a comenzii „**writemsg** string” care va scrie string-ul primit ca parametru (singur pe o linie noua), in fisierul (mesaj) anterior deschis / creat.

Implementarea unei comenzi „**encryptmsg** xyz” care va cripta mesajul (continutul fisierului) primit ca si parametru. Criptarea (de tip XOR -> ^) se va realiza din cel putin 2 threaduri paralele (folosind API-uri de Windows ca si: CreateThread, WaitForSingleObject, ...).

De asemenea, la un anumit numar de octeti criptati (de exemplu: 64) se va incrementa din fiecare thread o variabila globala care mentine numarul de octeti criptati pana in acel moment (va contoriza progresul).

(PHP + SQL)

Implementarea unei interfete WEB folosind PHP si o baza de date SQL. Prima pagina va fi una de login. In aceasta pagina un utilizator se va putea autentifica cu drepturi normale sau drepturi de administrator (se va specifica in pagina urmatoare ce tip de user este). In cazul in care autentificare se efectueaza cu succes (in urma interogarii bazei de date) se va trece la pagina 2.

In pagina 2 utilizatorul va putea adauga un comentariu intr-un text field, care la apasarea unui buton va fi salvat intr-un tabel din baza de date. Dupa acest pas se va actualiza si lista de comentarii afisata in pagina.

Tot in pagina 2 utilizatorul va putea da calea catre un fisier si apasand un buton se va afisa continutul acelui fisier.

De asemenea, in aceeasi pagina se va permite executarea unei comenzi (ping), cu parametri primiti prin interfata WEB (text field).

Deadline

Tema se va trimite prin email (Gheorghe.Hajmasan@cs.utcluj.ro) pana la data de: **7 ianuarie 2019, ora 08:00 AM**.

Atasati la email codul sursa intr-o arhiva **NumePrenume.zip** (continand **DOAR** fisierele sursa + lista de vulnerabilitati gasite).

Exemplu de subiect pentru mail: „[PSnCS][TEMA]NUME Prenume”.