6

Основи інформаційної та кібербезпеки

Лекція 6.
Захист даних і програмного забезпечення

1. **Шкідливі програмні засоби**
2. **Захист даних і програм**
3. **Методи захисту програмного забезпечення**
4. **Ідентифікація програм та захист авторських прав**

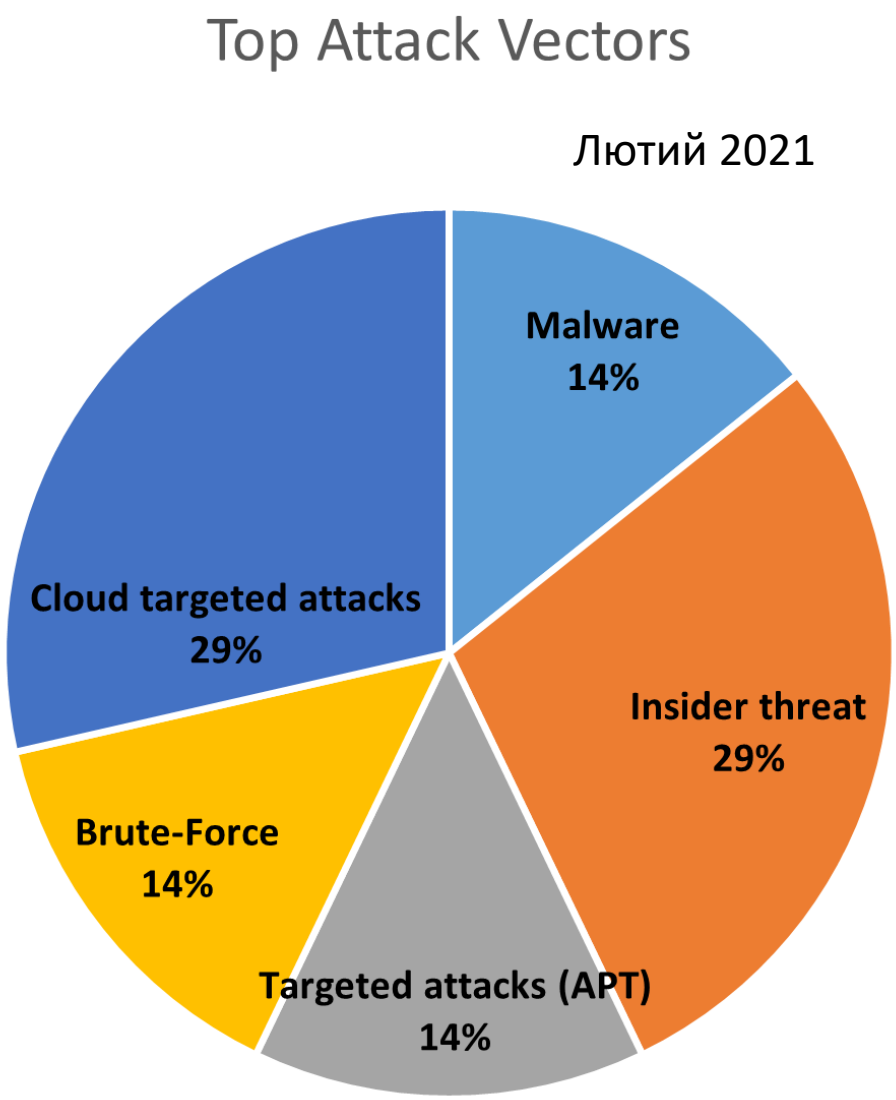
1 Шкідливі програмні засоби

Шкідливий програмний засіб, шкідливе програмне забезпечення (*англ*. malware — скорочення від **mal**icious — зловмисний і soft**ware** — програмне забезпечення) — програмне забезпечення, яке перешкоджає роботі комп'ютера, збирає конфіденційну інформацію або отримує доступ до приватних комп'ютерних систем.

Може проявлятися у вигляді коду, скрипту, активного контенту, і іншого програмного забезпечення.

Шкідливий — це загальний термін, який використовується для позначення різних форм ворожого або непроханого програмного забезпечення.

1 Шкідливі програмні засоби



The 10 types of **malware** :

- Virus
- Worms
- Trojans
- Polymorphic malware
- Ransomware
- Rootkits
- Keyloggers
- Bots
- Spyware and adware
- Mobile malware

1 Шкідливі програмні засоби

1 Шкідливі програмні засоби

Програма-вимагач, програма-шантажист (англ. *ransomware* - ransom — викуп і software — програмне забезпечення) — це тип шкідливої програми, який злочинці встановлюють на Ваших комп'ютерах.

Програми, які вимагають викуп, надають злочинцям можливість віддалено заблокувати Ваш комп'ютер.

1 Типи програмного забезпечення

Системне

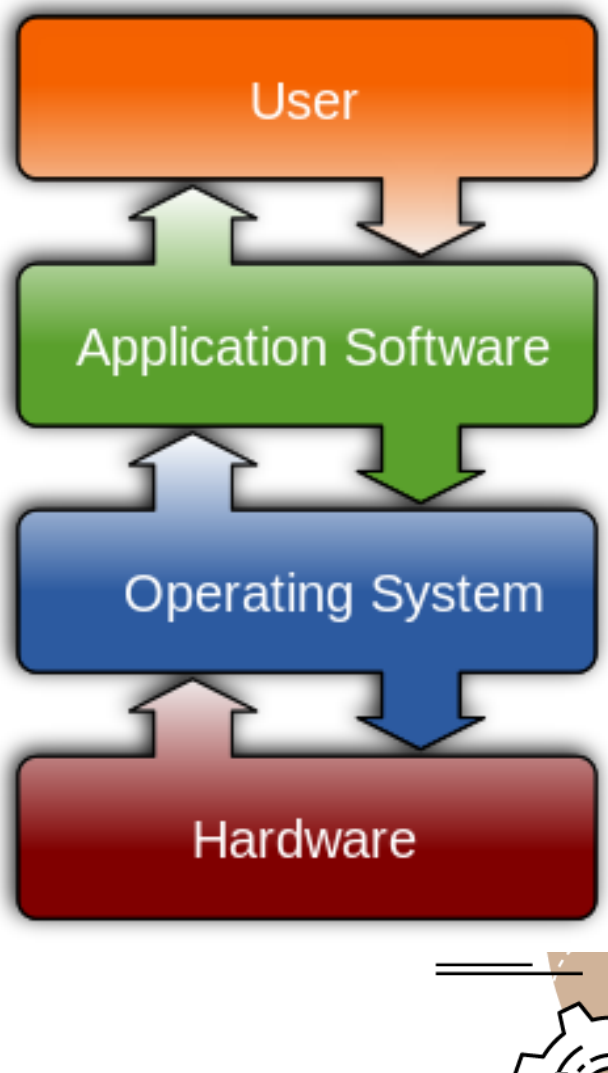
- Операційні системи, драйвери пристроїв, системні утиліти тощо

Прикладне

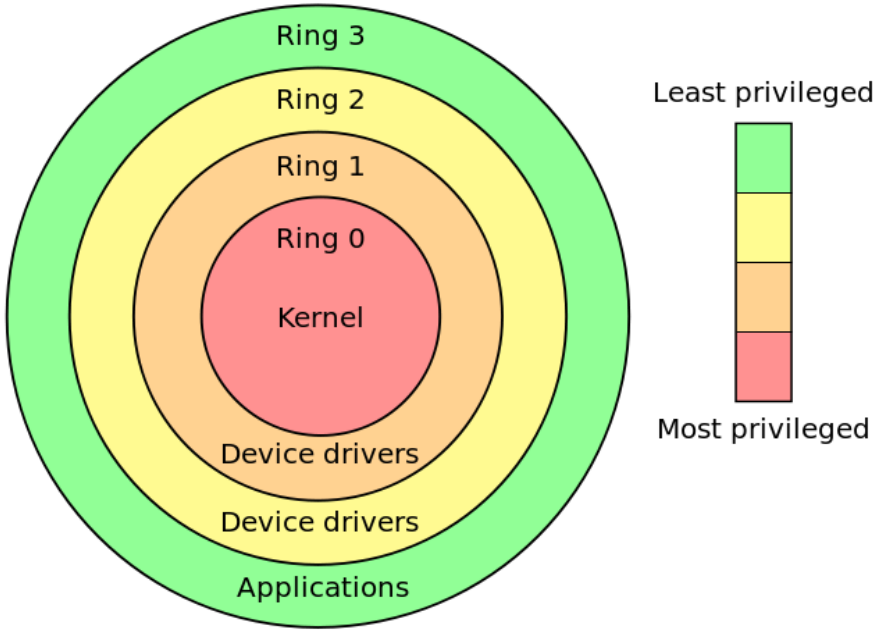
- Будь-які програми, які виконуються користувачем
- Веб-сервери, веб-сервіси, сервери баз даних тощо

Шкідливе

- Віруси, троянці, хробаки, здирники, шпигуни тощо



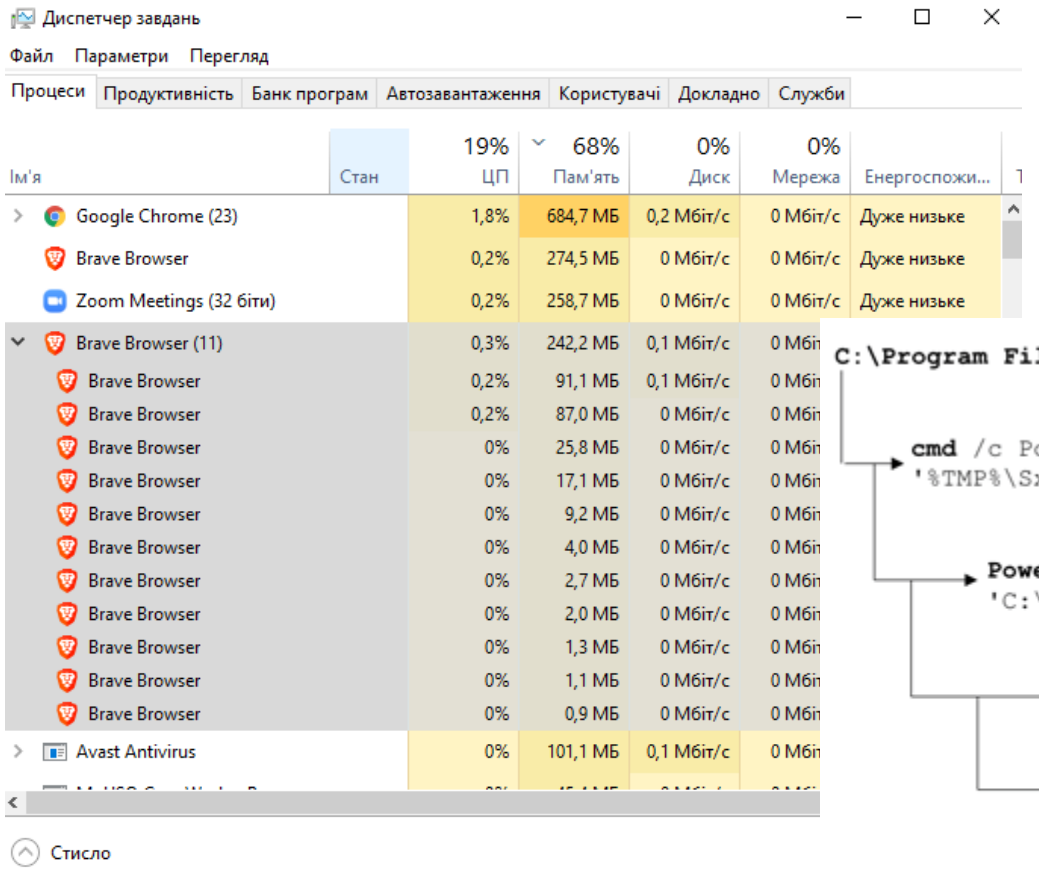
1 Кільцева модель безпеки ОС

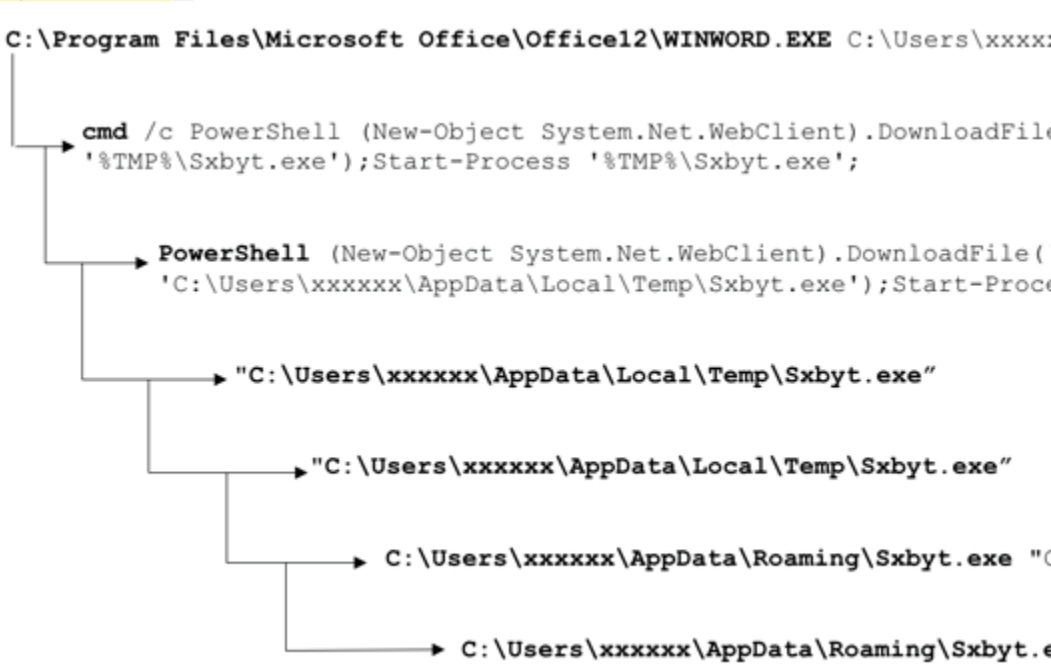


- В Ring3 виконуються прикладні програми, у інших – системні
- Кожна програма виконується з правами свого «творця» – користувача або іншого процесу
- Залежно від привілеїв, програми мають різні можливості
  - Ring3 має доступ до даних користувача
  - Ring0 до усіх даних, процесів тощо

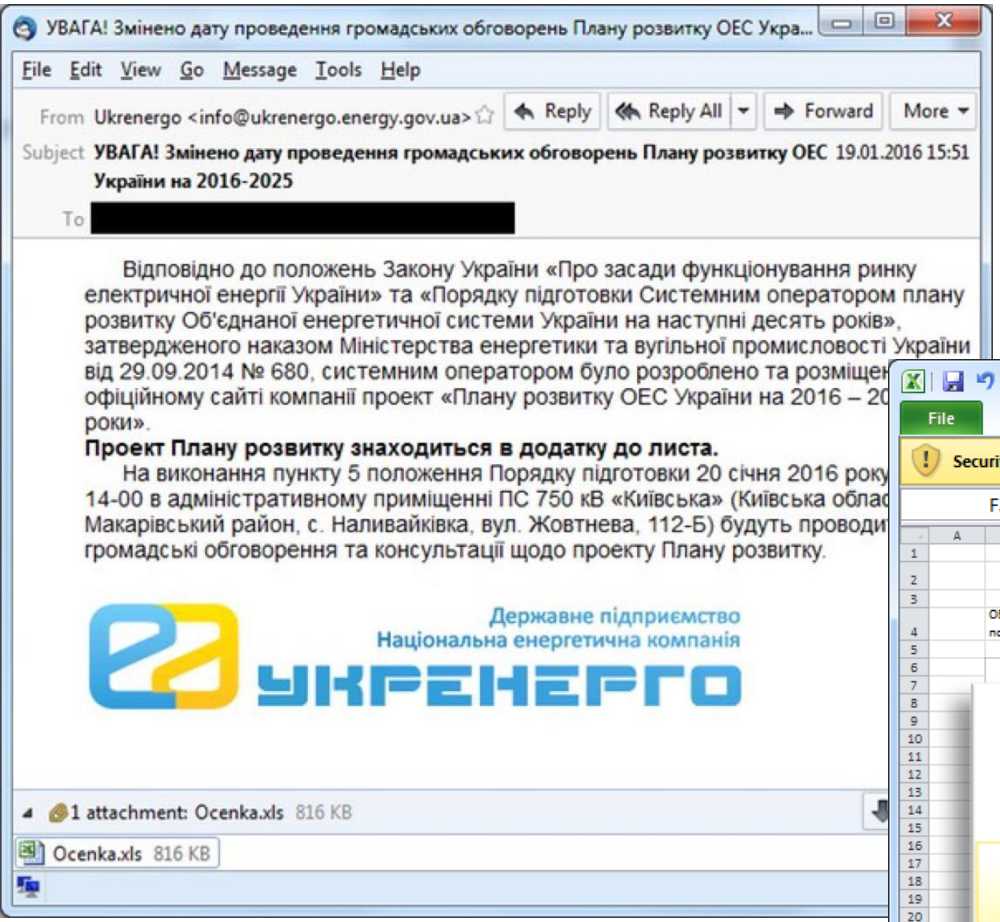
Процесор завжди знає, в якому кільці виконується код

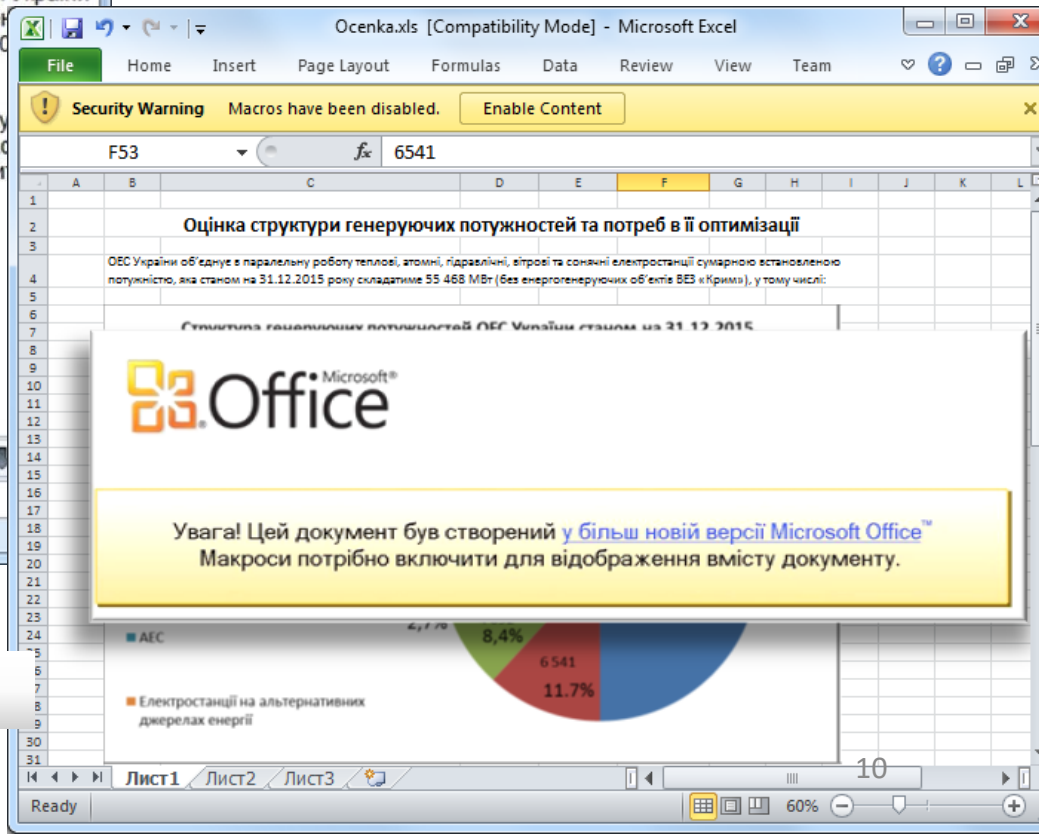
1 Дерево або ієрархія процесів





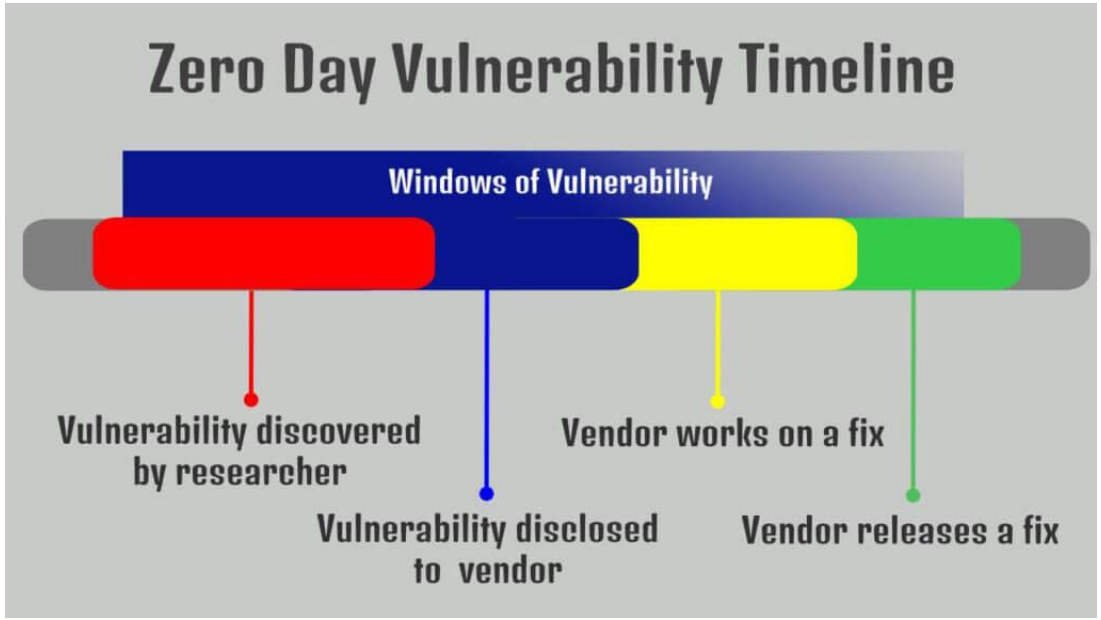
1 Фішинг

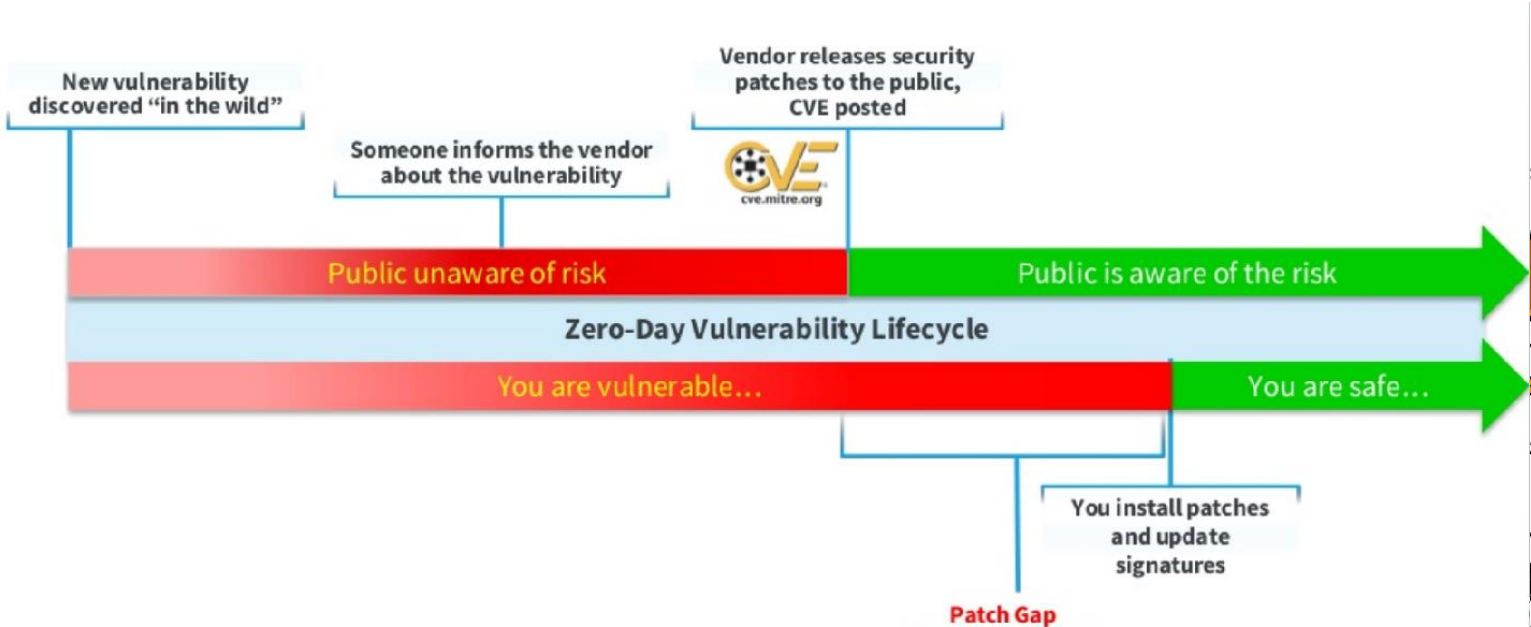




Дай людині рибу, і вона один раз наїсться.

1 Вразливість нульового дня (0-day)





1 Вразливість нульового дня (0-day)

Квітень 2021

У Zoom знайшли критичну вразливість, яка дозволяє легко зламати комп'ютер без відома користувача

Вразливість була виявлена в рамках змагання в сфері кібербезпеки Pwn2Own, організованого **Zero Day Initiative**. Дослідники Computest продемонстрували ланцюжок атак з трьома помилками, яка дозволяла віддалено виконати код на цільовій машині. Важливо зауважити, що все це не вимагало ніякого взаємодії з боку користувача. Тобто, атаку можна провести абсолютно без відома жертви.

1 Вразливість нульового дня (0-day)

Для виявлення вразливостей хакери використовують різні техніки, наприклад:

Дизасемблювання (*disassembler*) програмного коду і подальший пошук помилок безпосередньо в коді програмного забезпечення;

Зворотна розробка (*reverse engineering*) і подальший пошук помилок в алгоритмах роботи програмного забезпечення;

Fuzz-тестування (*fuzzing,fuzz testing*) - свого роду стрес-тест для програмного забезпечення, суть якого полягає в обробці програмним забезпеченням великого обсягу інформації, що містить завідомо неправильні параметри.

2 Захист програмного забезпечення

- Клієнтські програми не повинні запускатися з правами адміністратора

- Інтерактивний вхід в систему з правами адміністратора – лише у випадку “аварії”

- Вхід “адміна” в систему – це інцидент безпеки

2 Безпечні налаштування

Windows

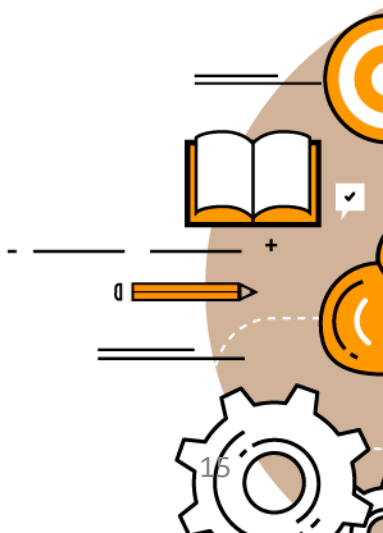
Microsoft Baseline Security Analyzer
A guide to Windows 10 security settings

Сканери вразливостей

OpenVAS
Nessus
Nexpose

macOS

25 Mac security tips and settings
A practical guide to securing macOS
Best Mac security settings



2 Оновлення ОС та програм

Flexera PSI

Patch My PC

SUMo

Homebrew

Менеджер пакетів для macOS

2 Захист від шкідливих програм

Avast Free Antivirus AVG AntiVirus FREE

Avira Antivirus Bitdefender Antivirus Free Edition

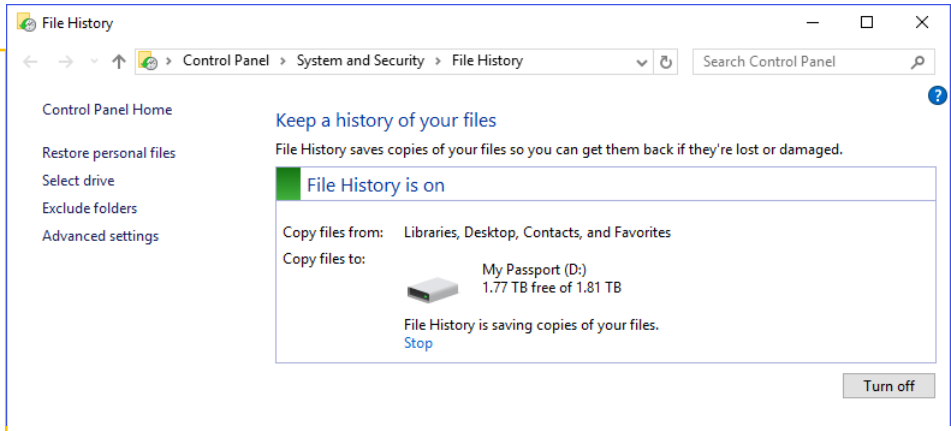
Kaspersky Security Cloud Free

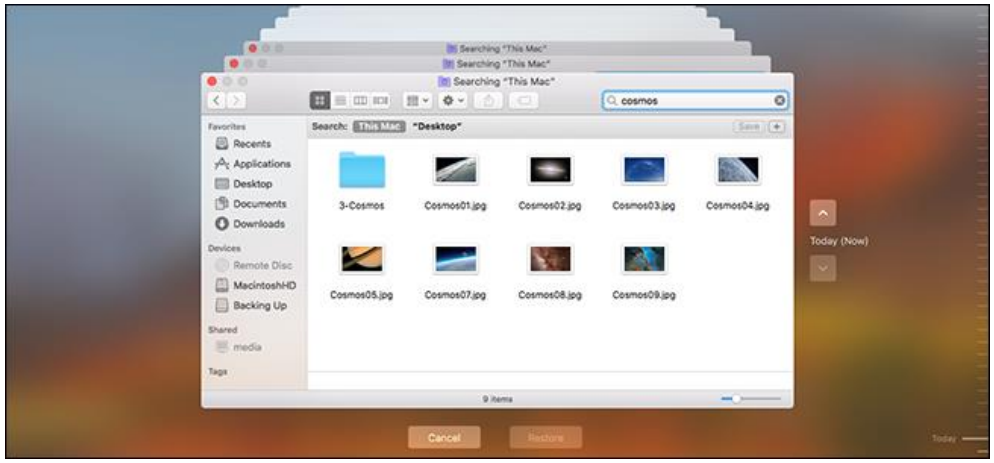
Microsoft Windows Defender

2 Створюйте резервні копії

- Windows FileHistory
- Windows OneDrive sync

- MacOS TimeMachine
- MacOS iCloud sync





OneDrive

2 Страхування кібер-ризиків

Секція 1. Відповідальність за дані.

Покриваються:

Збитки від втрати персональних та/або корпоративних даних.

компенсація внаслідок події, помилки чи упущення, результатом чого стало:

а) зараження даних третіх осіб вірусом,

б) пошкодження даних або ПЗ,

в) втрата паролів та доступу до даних,

г) спотворення даних,

д) фізичне пошкодження чи знищення даних,

е) розкриття працівником даних третіх осіб.

Секція 2. Витрати на розслідування та штрафи.

Секція 3. Витрати на відновлення репутації та реагування на кібер інцидент.

Програмно-технічну експертизу.

Відновлення репутації Страхувальника:

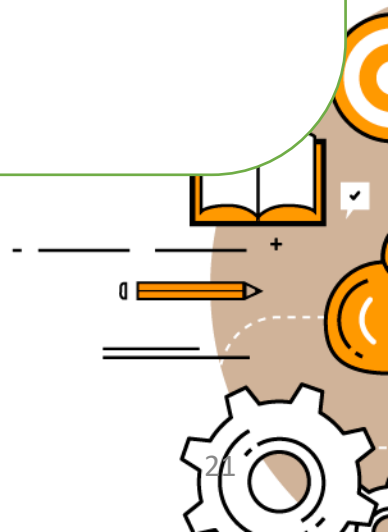
3 Методи захисту програмного забезпечення

Захист програмного забезпечення — комплекс заходів, спрямованих на захист програм від несанкціонованого придбання, використання, розповсюдження, модифікування, вивчення і відтворення аналогів.

Захист від несанкціонованого використання програм — система заходів, спрямованих на протидію нелегальному використанню програмного забезпечення. При захисті можуть застосовуватися організаційні, юридичні, програмні та програмно-апаратні засоби.

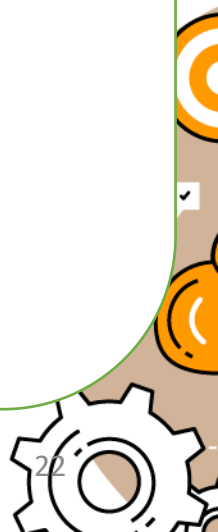
3 Моделі розповсюдження програмного забезпечення

- безкоштовне програмне забезпечення (**Freeware**);
- умовно безкоштовне програмне забезпечення (**Shareware**); ***"try before you buy“ “Adware”***
- комерційне програмне забезпечення (**payware**).
  Demoware Trialware Nagware



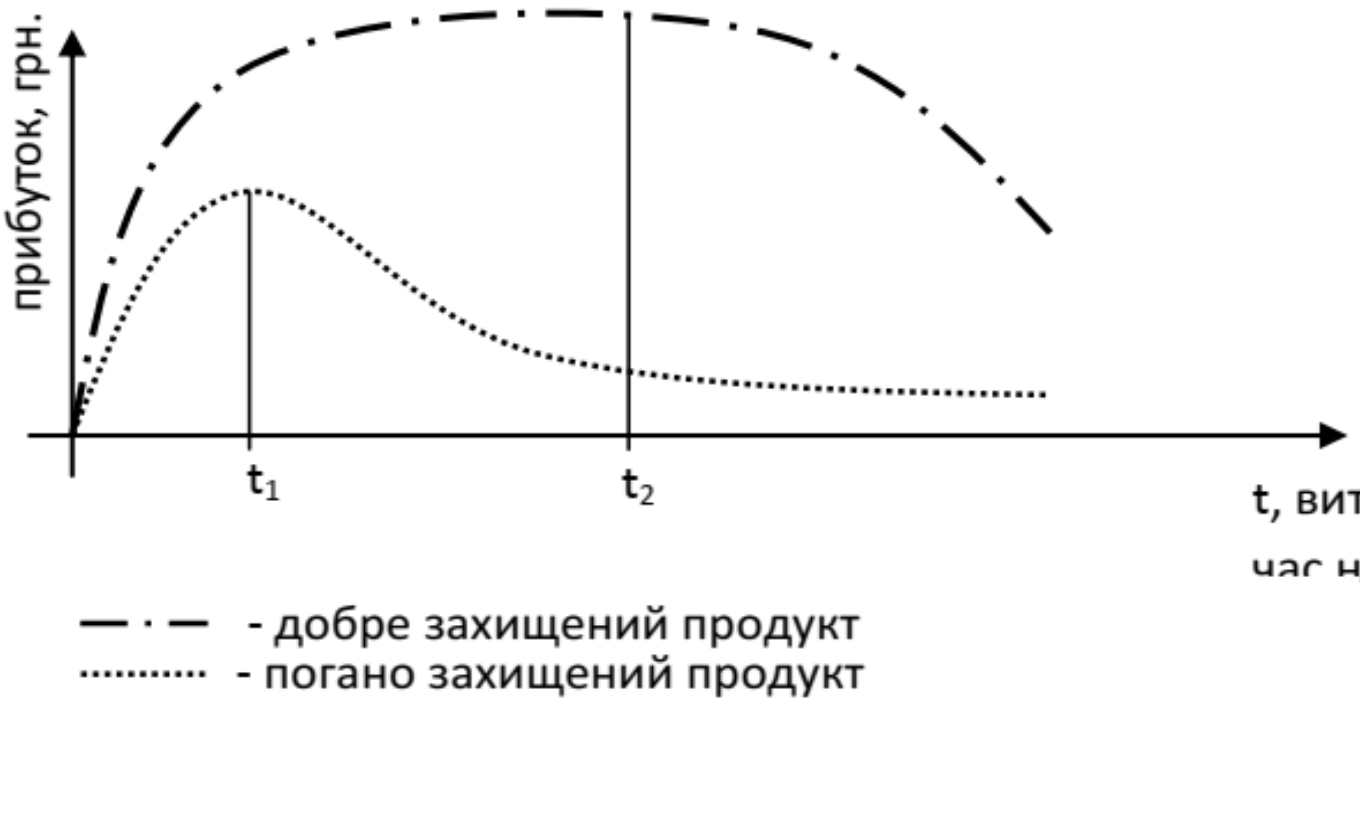
3 Проблеми захисту ПЗ

- ***промислове шпигунство*** - незаконне використання алгоритмів, що є інтелектуальною власністю автора, при написанні аналогів продукту;
- ***крадіжка і копіювання*** - несанкціоноване використання ПЗ ;
- ***несанкціонована модифікація*** *ПЗ* з метою впровадження програмних зловживань;
- ***піратство*** - незаконне поширення і збут ПЗ.



3 Методи захисту програмного забезпечення

Динамічна залежність прибутку від ступеня захищеності продукту



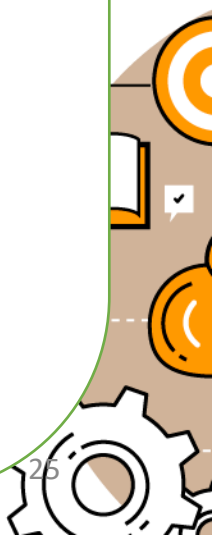
3 Методи захисту програмного забезпечення

Всі методи захисту можна розділити на **апаратні** та **програмні**

Апаратні засоби - електронні ключі, які підключаються до портів комп'ютера) або фізичні особливості носіїв інформації (компакт-дисків), щоб ідентифікувати оригінальну версію програми і захистити продукт від нелегального використання.

3 Методи захисту програмного забезпечення

Засоби дослідження роботи програмних продуктів:
Статичні засоби, які оперують початковим кодом програми як даними і будують її алгоритм без виконання.
Динамічні засоби, які вивчають програму, інтерпретуючи її в реальному або віртуальному обчислювальному середовищі.
Синтаксичні методи. До цієї групи відносяться методи, що ґрунтуються тільки на результатах лексичного, синтаксичного і семантичного аналізу програми
Статистичні методи. Статистичні методи використовують інформацію, зібрану в результаті значної кількості запусків програми на великій кількості наборів вхідних даних .



3 Методи захисту програмного забезпечення

Конкретні інструменти, що використовуються зламниками для несанкціонованого дослідження захищених програм:

Налагоджувачі і дизасемблери. Традиційно обидва ці типи інструментів використовуються в парі, оскільки дизасемблер видає лише "чистий код".

Декомпілятори і вузькоспеціалізовані налагоджувачі. не слід чекати повного відновлення початкового тексту програми.

Утиліти для **розпакування та дампінга процесів**. Дизасемблювати запаковану або зашифровану програму напряму неможливо. Але якщо дуже хочеться одержати хоч якийсь лістинг, можна спробувати витягнути з пам'яті комп'ютера знімок (дамп) програми у момент її роботи.

3 Методи захисту програмного забезпечення

Утиліти аналізу файлів. Дають змогу дізнатись, яким пакувальником або захисним програмним забезпеченням оброблена та або інша програма, знайти всі текстові рядки в дампі пам'яті, проглянути вміст файлу у вигляді таблиці записів, вивести у файл список функцій, що імпортуються і експортуються програмою, і багато іншого.

Шістнадцяткові редактори і редактори ресурсів.

Поруч із редакторами йдуть патчери, які дозволяють створити невеликий виконуваний файл, що автоматично вносить зміни в оригінальний файл програми або в код цієї програми безпосередньо в пам'яті.

API-шпигуни і інші утиліти моніторингу.

знати, які саме дії виконує та або інша програма, звідки читає і куди записує дані, які стандартні функції і з якими параметрами вона викликає.

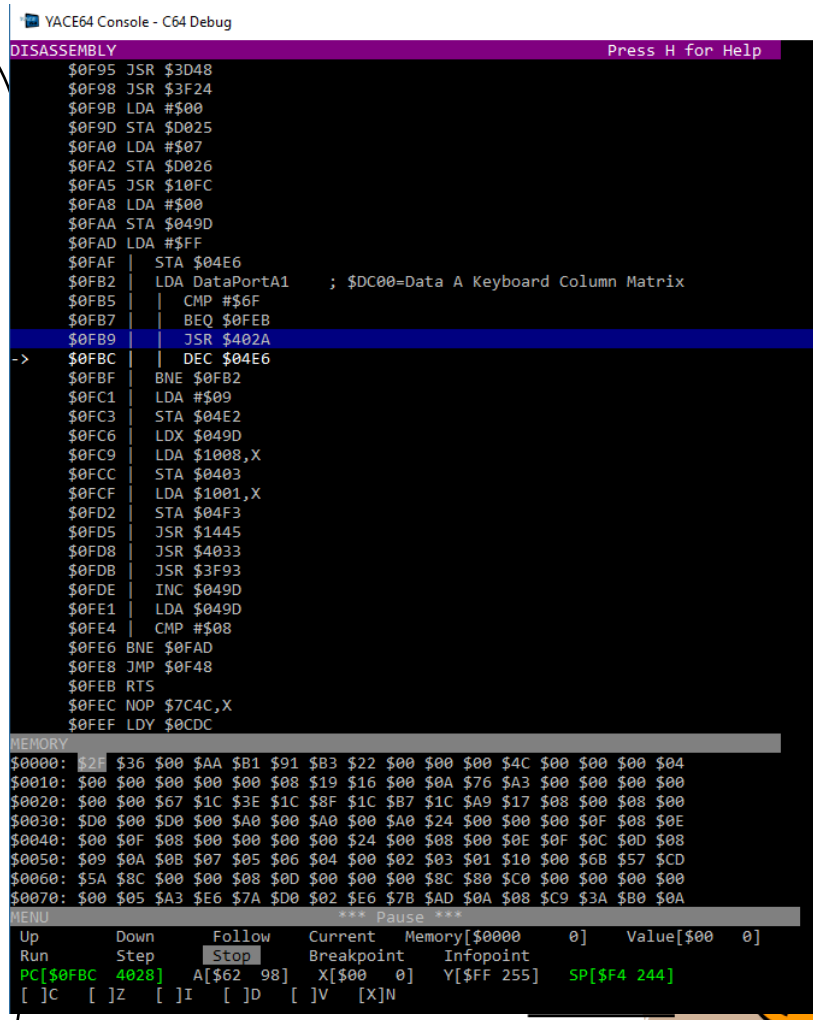
3 Методи захисту програмного забезпечення

3 Захист від дизасемблювання

Вхідними даними для дизасемблера є програма або окрема ділянка коду. Результатом його роботи є лістинг (вихідний текст програми мовою асемблера), який в ідеалі повинен бути максимально близький до оригіналу.

Методи протидії:

- Шифрування коду.
- Маніпулювання заголовками EXE-файлів.
- Обман дизасемблера.
- Методи емуляції.



3 Захист від дизасемблювання

Захист програм шляхом обфускації

заплутати програмний код і усунути більшість логічних зв'язків у ньому, тобто трансформувати його так, щоб він був дуже важкий для вивчення і модифікації сторонніми особами

Найпростіший приклад обфускованого HTML:

```
<b>Євро</b><b>па</b>
```

Лексична обфускація

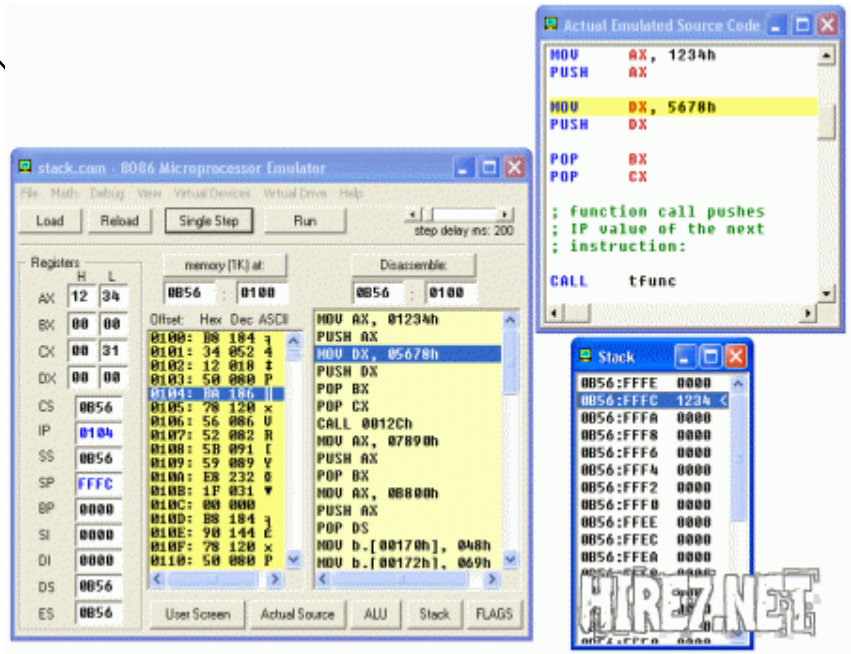
Обфускація даних

```
#include                                  <math.h>
#include                                <sys/time.h>
#include                                <X11/Xlib.h>
#include                               <X11/keysym.h>
                                       double L ,o ,P
                                      ,_=dt,T,Z,D=1,d,
                                      s[999],E,h= 8,I,
                                      J,K,w[999],M,m,O
                                     ,n[999],j=33e-3,i=
                                     1E3,r,t, u,v ,W,S=
                                     74.5,l=221,X=7.26,
                                     a,B,A=32.2,c, F,H;
                                     int N,q, C, y,p,U;
                                    Window z; char f[52]
                                  ; GC k; main(){ Display*e=
 XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
 ; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(O)){ struct timeval G={ 0,dt*1e6}
 ; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+=_*P; r=E*K; W=cos( O); m=K*W; H=K*T; O+=D*_*F/ K+d/K*E*_; B=
sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T*E+ D*B*W; j+=d*_*D-_*F*E; P=W*E*B-T*D; for (o+=(I=D*W+E
*T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
]== 0|K <fabs(W=T*r-I*E +D*P) |fabs(D=t *D+Z *T-a *E)> K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2/ K
 *D; N-1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_* (X*t +P*M+m*l); T=X*X+ l*l+M *M;
  XDrawString(e,z,k ,20,380,f,17); D=v/l*15; i+=(B *l-M*r -X*Z)*_; for(; XPending(e); u *=CS!=N){
                                  XEvent z; XNextEvent(e ,&z);
                                      ++*((N=XLookupKeysym
                                        (&z.xkey,0))-IT?
                                        N-LT? UP-N?& E:&
                                        J:& u: &h); --*(
                                        DN -N? N-DT ?N==
                                        RT?&u: & W:&h:&J
                                         ); } m=15*F/l;
                                         c+=(I=M/ l,l*H
                                         +I*M+a*X)*_; H
                                         =A*r+v*X-F*l+(
                                         E=.1+X*4.9/l,t
                                         =T*m/32-I*T/24
                                          )/S; K=F*M+(
                                          h* 1e4/l-(T+
                                          E*5*T*E)/3e2
                                          )/S-X*d-B*A;
                                          a=2.63 /l*d;
                                          X+=( d*l-T/S
                                           *(.19*E +a
                                           *.64+J/1e3
                                           )-M* v +A*
                                           Z)*_; l +=
                                           K *_; W=d;
                                           sprintf(f,
                                           "%5d  %3d"
                                           "%7d",p =l
                                          /1.7,(C=9E3+
   O*57.3)%0550,(int)i); d+=T*(.45-14/l*
  X-a*130-J* .14)*_/125e2+F*_*v; P=(T*(47
  *I-m* 52+E*94 *D-t*.38+u*.21*E) /1e2+W*
  179*v)/2312; select(p=0,0,0,0,&G); v-=(
   W*F-T*(.63*m-I*.086+m*E*19-D*25-.11*u
    )/107e2)*_; D=cos(o); E=sin(o); } }
```

3 Захист від дизасемблювання

Емуляція процесора та мультизадачності

частина коду або вся програма цілком повинна бути написана під "саморобний" процесор, інструкції якого виконуються емулятором на реальному процесорі x86.

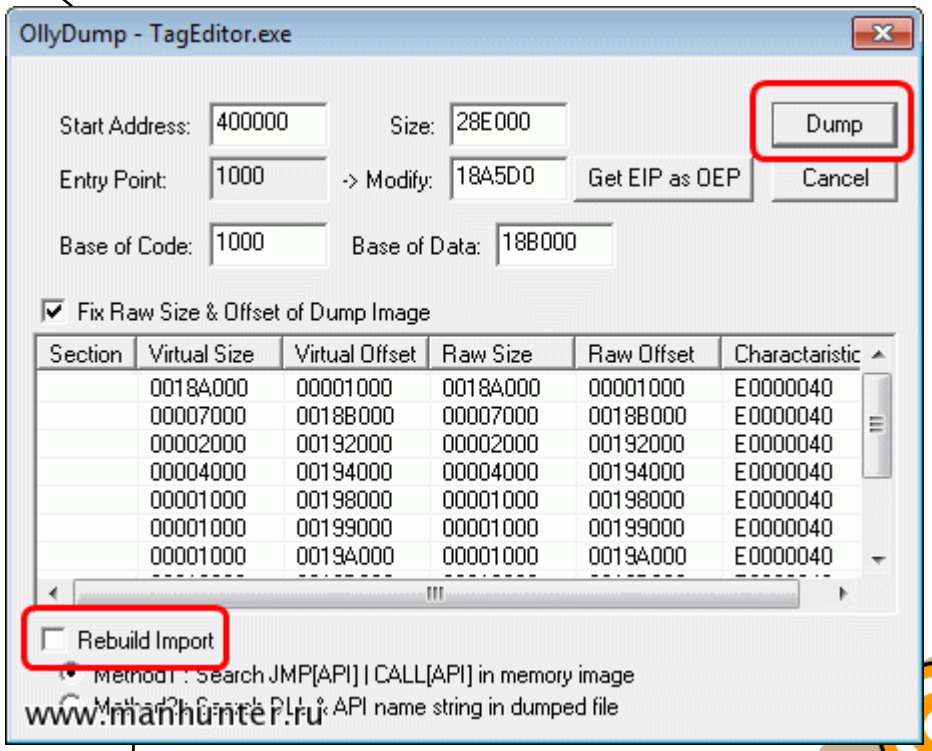


3 Захист від дампінгу

Навісний захист – ускладнити аналіз роботи програми за допомогою шифрування коду програми і розшифровування його безпосередньо перед виконанням.

Дамп пам'яті – це копія вмісту оперативної пам'яті, що знаходиться на жорсткому диску …

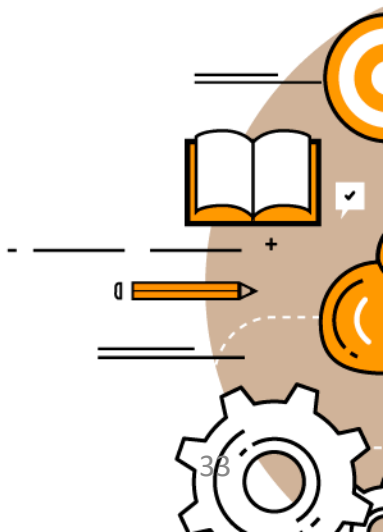
Антидампінг у нульовому кільці
Динамічне розпаковування



4 Ідентифікація програм та захист авторських прав

Способи захисту програмного продукту від неправомірного використання

1. охороняти вихідний код програми за допомогою авторського права;
2. охороняти алгоритми, які використовуються в програмах, як способи вирішення конкретних завдань;
3. охороняти назву програми.



4 Ідентифікація програм та захист авторських прав

авторське право на твір, а саме комп'ютерну програму, виникає з моменту його створення та матеріального вираження і обов'язкової реєстрації його не вимагається. В Україні існує ***презумпція авторства***, відповідно до якої, автором є особа, зазначена як автор на оригіналі або примірнику твору. Проте для більшої гарантії можна зареєструвати авторське право на програму і договорів, які стосуються права автора на цей продукт.

4 Ідентифікація програм та захист авторських прав

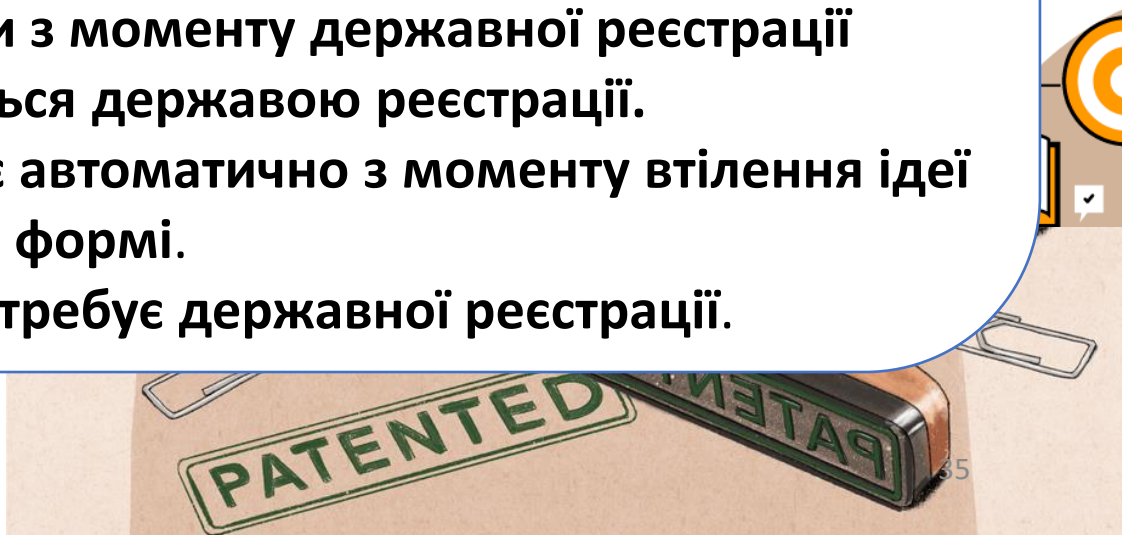
Авторське право != патент

авторське право захищає конкретне втілення певної ідеї (скажімо, алгоритму, записаного у вигляді коду), але не поширюється далі — на конкретне технічне рішення, що лежить в основі програмного забезпечення. Наприклад, власне програму чи її складову як код буде захищено, але інноваційне рішення, сам алгоритм — ні.

Патент починає діяти з моменту державної реєстрації та обмежується державою реєстрації.

Авторське право виникає автоматично з моменту втілення ідеї у більш-менш фіксованій формі.

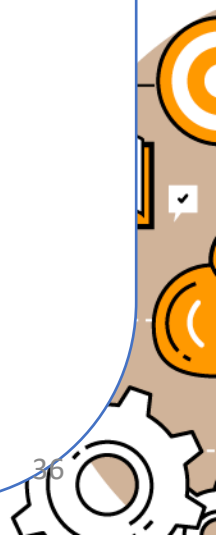
Автоматично, тобто **не потребує державної реєстрації**.



4 Ідентифікація програм та захист авторських прав

об’єкти, які **можна захистити як свою власність**:

- програма загалом як єдиний комплекс/продукт;
- окремі елементи програми: виражені власне командами або рядками коду без очевидного виконуваного ефекту (класична англійська справа *IBCOS Computers Ltd v Barclays Mercantile Highland Finance Ltd*, де суддя визнав рядки data division у COBOL-програмі частиною, захищеною копірайтом; іншим прикладом є апаратний ключ (донгл) у справі *Autodesk Inc v Dyason:* копіювання таблиці кодів з програми донгла було визнано порушенням авторських прав);
- специфікації;
- схеми (зокрема, блок-схеми);
- діаграми;
- макети меню та інтерфейсу користувача ;
- командні рядки;
- зображення на дисплеях екранів;
- бази даних;
- протоколи (але досі не всюди чітко визначено, чим вважати протокол — ідеєю, принципом чи захищеним втіленням ідеї);
- звіти тощо.



4 Ідентифікація програм та захист авторських прав

Мови програмування і функціонал, а також формат файлів з даними не підлягають захисту в межах авторського права. Принаймні у ЄС, що підтверджується рішенням у справі *SAS Institute Inc. v. World Programming Ltd.* (C-406/10).

Що дає спеціалісту авторське право:

майнові (tangible) та **немайнові**, інколи **моральні** (intangible, moral) права.

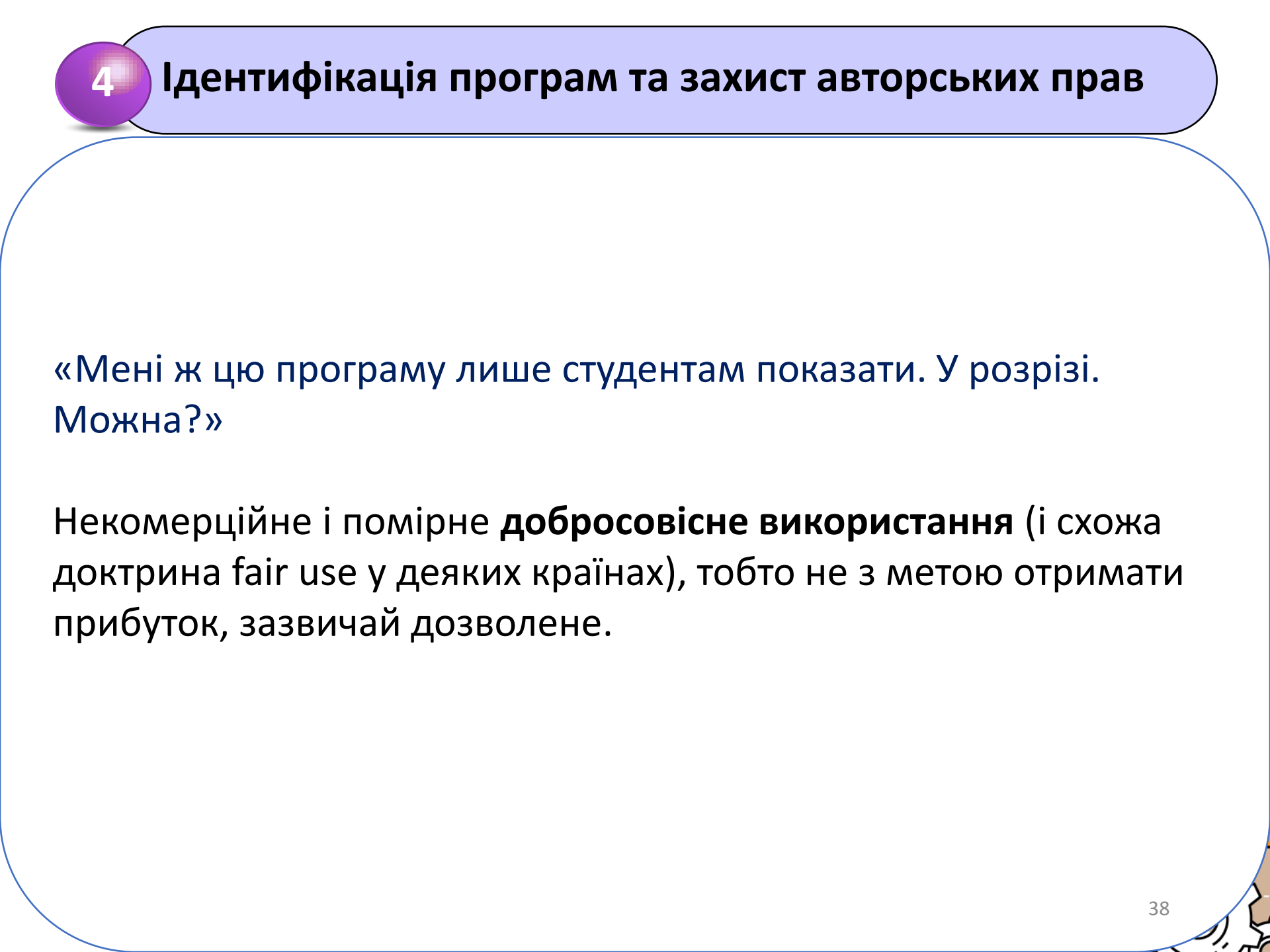
До **майнових прав** належать виключне право використання твору і заборони його використання іншими особами.

Немайнові права охоплюють право бути вказаним як автор програми (хоча залишитися анонімом — теж ваше право), бути згаданим під своїм псевдонімом замість справжнього імені, право заперечувати проти зміни твору…

4 Ідентифікація програм та захист авторських прав

«Мені ж цю програму лише студентам показати. У розрізі. Можна?»

Некомерційне і помірне **добросовісне використання** (і схожа доктрина fair use у деяких країнах), тобто не з метою отримати прибуток, зазвичай дозволене.



4 Ідентифікація програм та захист авторських прав

Чи можна код захистити від плагіату

Комп’ютерні програми підлягають захисту як літературні твори.

В англійському праві **актом копіювання може бути навіть завантаження комп’ютерної програми у volatile memory (RAM) комп’ютера**. А копіювання — виражатися у різних формах:
як **пряме запозичення коду** тією ж мовою програмування, при цьому з незначними змінами чи внесенням додаткових функцій, щоб приховати оригінальні авторські права на програму;
непряме копіювання: запозичується лише частина програми, наприклад, послідовність операцій, структура, функції, інтерфейси, методології тощо без копіювання оригінального коду тією ж чи іншою мовою.

В Україні неправомірне зберігання копії комп’ютерної програми в пам’яті комп’ютера є порушенням майнового авторського права.

Кому належать права на код

Цікаво, що **звичайною законодавчою практикою є обмеження права робити кілька копій програмного забезпечення для особистого використання**, у той час як це дозволено щодо копій книг чи картин.

Майнові права на твір можна передати кількома способами:

1. Договір про надання послуг або трудовий контракт.
2. Ліцензія. Їх можна поділити на кілька підвидів залежно від умов, на яких інші можуть користуватися програмою:
   - ті, що надають можливість користувачам вільно використовувати й далі поширювати твір;
   - комерційні ліцензії, зокрема умовно-безоплатні.

Немайнові права залишаються за автором, він не може бути позбавлений їх примусово чи передати ці права іншим особам.

4 Ліцензії на вільне програмне забезпечення

Термін **вільне програмне забезпечення** ввів Річард Столмен, засновник проєкту GNU.

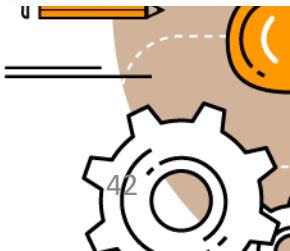
Щоб програмне забезпечення вважалось вільним, воно повинно поширюватись під однією з вільних ліцензій, котра закріплює за користувачем вищеописані права, та з вільно-доступними джерельними кодами. Найвідомішими з них є:

- Загальна публічна ліцензія GNU (GNU General Public License)
- Менша загальна публічна ліценція GNU (GNU Lesser General Public License)
- Ліцензія BSD (BSD License)
- Публічна ліцензія Mozilla (Mozilla Public License)
- Ліцензія MIT (MIT License)
- Ліцензія Apache (Apache License)

4 GNU General Public License

GPL надає одержувачам комп'ютерних програм такі права, або «свободи»:

- свободу запуску програми з будь-якою метою;
- свободу вивчення того, як програма працює, і її модифікації (попередньою умовою цього є гарантований доступ до сирцевого коду);
- свободу розповсюдження копій;
- свободу поліпшення програми й випуску поліпшень у публічний доступ (попередньою умовою цього є гарантований доступ до сирцевого коду).



4 Підсумки:

1. **Авторське право відокремлене від патенту**. В Україні програмне забезпечення захищається як літературний твір, права на програму виникають автоматично і не потребують реєстрації.
2. Автор програмного забезпечення може захистити лише свій власний творчий внесок.
3. Переважно автор передає тільки **майнові права**, які дозволяють правовласнику отримувати певний прибуток з продажу чи використання ПЗ. **Моральні права** — бути згаданим як автор, здебільшого невіддільні від особи автора. Водночас у законодавстві є винятки, за яких **роялті платити необов'язково**. Наприклад, під час використання ПЗ **для освітніх потреб**.
4. При розробці варто уважно читати умови ліцензій, якщо запозичується опенсорс-ПЗ. Цілком імовірно, що вони можуть бути непоєднуваними з вашими цілями та умовами договору, за яким ви створюєте нове ПЗ.
5. Pet-проекти краще робити у вільний час і на особистій техніці.