Endpoints and Systems Course Summary

ملخص كورس الأجهزة/النقاط الطرفية والأنظمة

إعداد: إسماعيل محمد قطان

Prepared by: Ismail Muhammed Kattan

التاريخ: أكتوبر/ تشرين الأول 2025

Date: October, 2025



تمت كتابة هذا الملخص بناء على كورس Endpoints and Systems المقدم من CISCO على منصة Coursera

هذا الملخص يجمع ما قد يجهله أو ينساه خريج علوم الحاسوب وقد تم تجاهل بعض الدروس البسيطة والتي قد لا يحتاجها إلا من ليس له أدنى إطلاع على نظام التشغيل Windows أو Linux

أمثلة على الدروس التي تم تجاهلها:

* واجهة المستخدم الجرافيكية GUI
* تشغيل بوضع administrator

## نظام ويندوز Windows

### تاريخ نظام التشغيل Windows

* Microsoft Windows هو عائلة من أنظمة التشغيل لأجهزة الحاسوب الشخصية، ظهر أول إصدار منها سنة 1985.
* تم تصميمه في البداية ليعمل كواجهة رسومية شاملة لنظام MS-DOS.
* كان الهدف منه تسهيل استخدام الحاسوب عبر واجهة رسومية (GUI) بدلاً من الأوامر النصية فقط.

#### تطور نظام ويندوز

* تطوّر النظام تدريجيًا ليشمل الاتصال بالشبكات والإنترنت.
* أصبح قادرًا على العمل على أنواع مختلفة من الأجهزة، وليس فقط الحواسيب المكتبية.
* أصدرت مايكروسوفت أنظمة تشغيل خاصة بـ:
  + الخوادم (Servers), الهواتف المحمولة (Phones) والأجهزة الذكية والشخصية الأخرى (Mobile Devices)

#### أهمية نظام ويندوز

* ما يزال النظام الأكثر انتشارًا على أجهزة:
  + الحواسيب المكتبية
  + الحواسيب المحمولة (Laptops)
  + وهو كذلك أكثر الأنظمة استخدامًا كعميل (Client) للتفاعل مع منصات الحوسبة السحابية (Cloud Computing Platforms).

#### الموثوقية والأمان

* تطوّر ويندوز ليصبح نظامًا قويًا واحترافيًا للمؤسسات (Enterprise Platform).
* يستخدم في:
  + بيئات المستخدم اليومية
  + مراكز البيانات (Data Centers)
  + أنظمة الخوادم الكبيرة (Server Platforms)
* بنيته المعمارية أصبحت تتضمن خصائص تعزز:
  + استقرار النظام (System Stability)
  + أمان النظام (Security)

#### مجتمع المستخدمين والخبراء

* يتمتع ويندوز بقاعدة ضخمة من الخبراء التقنيين الذين يديرون البنية التحتية المعتمدة عليه. هذا الانتشار يجعله الخيار المفضل للشركات الكبرى والمؤسسات.

#### أهمية معرفة نظام ويندوز للأمن السيبراني

* يجب على محللي الأمن (Security Analysts) فهم طريقة عمل ويندوز بعمق.
* من الضروري أيضًا معرفة كيفية البحث عن الثغرات الأمنية المعروفة (Vulnerabilities).
* يمكن ذلك عبر قاعدة بيانات الثغرات الوطنية NIST – National Vulnerability Database (NVD)
* التي تحتوي على معلومات عن الثغرات الشائعة (CVEs) في أنظمة التشغيل.

### بنية نظام ويندوز Windows Operating System Architecture

تنقسم البنية إلى مكوّنين رئيسيين:

🔸 User Mode

🔸 Kernel Mode

#### وضع المستخدم (User Mode)

هو الجزء المسؤول عن إدارة العمليات التي ينفذها المستخدمون (User Processes).

يعمل بصلاحيات محدودة (Low Privilege) بحيث لا يستطيع المستخدم التأثير المباشر على العمليات الحساسة للنظام.

**الهدف من وضع المستخدم:**

* عزل كل عملية (Process) في مساحة ذاكرة مستقلة (Virtual Address Space).

➤ هذا يمنع أي برنامج من الوصول إلى ذاكرة برنامج آخر.

* يقلل احتمالية تلف النظام (System Corruption).
* يسمح بالوصول إلى موارد النظام بشكل غير مباشر عبر واجهات برمجة التطبيقات (APIs) التي يتحكم بها النظام.

##### أنواع العمليات التي تعمل في User Mode:

1. Non-service Processes (عمليات غير خدمية):
   * مثل خدمات تسجيل الدخول (Logon Services) وSession Manager.
   * تبدأ هذه العمليات نتيجة نشاط المستخدم وليس النظام.
2. Service Processes (العمليات الخدمية):
   * تعمل لدعم متطلبات المستخدم أو النظام.
   * يتم تشغيلها بواسطة Service Control Manager حتى قبل تفاعل المستخدم.
   * مثال: Print Spooler (إدارة الطباعة) أو Task Scheduler.
3. User Applications (تطبيقات المستخدم):
   * مثل البرامج التي يفتحها المستخدم (Word، Browser... إلخ).
   * تعمل ضمن الذاكرة الخاصة بها.
4. Operating System Environment Support (دعم بيئة التشغيل):
   * لتشغيل تطبيقات مصممة لأنظمة أخرى (مثل Subsystem for UNIX-based Applications – SUA).
   * سابقًا كانت تدعم OS/2 وPOSIX، لكن الآن تركز على دعم UNIX فقط.

##### التبديل بين User Mode وKernel Mode:

* أحيانًا تحتاج تطبيقات المستخدم إلى استخدام موارد يديرها النظام (مثل الكتابة إلى القرص أو الاتصال بالشبكة).
* في هذه الحالة، يسمح ويندوز بالانتقال المؤقت من User Mode إلى Kernel Mode عبر ما يسمى:
  + Kernel Interface Layer
* بعد انتهاء العملية، يعود النظام مجددًا إلى User Mode.
* هذه العملية تسمى Context Switching وتتم لحماية النظام من أي وصول غير مصرح به.

#### وضع النواة (Kernel Mode)

هو المكوّن الذي تعمل فيه العمليات الأساسية للنظام.

يمتلك أعلى مستوى من الصلاحيات (Highest Privilege Level) ويدير:

* وحدة المعالجة المركزية (CPU)
* الذاكرة (Memory)
* الأجهزة (Hardware)

**ملاحظات مهمة:**

* جميع عمليات Kernel Mode تشترك في نفس مساحة الذاكرة.
  + هذا يجعل الأداء أسرع ولكن أكثر خطورة في حال حدوث خلل.
* أي خطأ في Driver أو Kernel Process ممكن يؤدي إلى:
  + انهيار كامل للنظام (System Crash / Blue Screen).

##### الأمان في Kernel Mode:

* منذ Windows Vista، فرضت مايكروسوفت سياسة **Kernel Mode Code Signing**
* أي ملف أو تعريف (Driver) يعمل في Kernel Mode يجب أن يكون:
  + موقّع رقميًا (Digitally Signed)
  + بشهادة من سلطة موثوقة (Public CA) معتمدة من مايكروسوفت

**الهدف:** ضمان أن كل كود في النواة يمكن تتبع مصدره ومطوره.

##### المكونات الأساسية في Kernel Mode

| **المكوّن** | **الوظيفة** |
| --- | --- |
| ***Kernel Executive*** | يدير الخدمات الأساسية مثل الإدخال/الإخراج (I/O)، الشبكات، وإدارة الذاكرة. |
| ***Kernel*** | مسؤول عن العمليات منخفضة المستوى مثل جدولة الخيوط (Thread Scheduling) وإدارة المقاطعات (Interrupts). |
| ***Device Drivers*** | يدير التفاعل مع الأجهزة المتصلة (طابعة، ماوس، قرص...). |
| ***HAL – Hardware Abstraction Layer*** | طبقة وسيطة بين النظام والأجهزة لتوحيد التعامل مع العتاد المختلف. |
| ***Window Management & Graphics Subsystem*** | مسؤول عن واجهة المستخدم الرسومية (GUI) وإدارة العرض والإدخال. |

##### 

##### ملفات النظام المرتبطة بـ Kernel Mode

| **الملف** | **الوظيفة** |
| --- | --- |
| *Ntoskrnl.exe* | نواة نظام ويندوز الرئيسية. |
| *Hal.dll* | مكتبة HAL للتفاعل مع العتاد. |
| *Win32k.sys* | درايفر لإدارة العرض والرسومات ومدخلات لوحة المفاتيح والفأرة. |
| *Ntdll.dll* | مسؤول عن ربط خدمات النظام بالمستخدم. |
| *Kernel32.dll* | واجهة برمجية أساسية للنظام. |
| *Advapi32.dll* | يحتوي على وظائف أمنية وخدمات النظام. |
| *User32.dll* | مسؤول عن إدارة النوافذ والمكونات الرسومية. |
| *Gdi32.dll* | يدير العمليات الرسومية (Graphics). |

#### Windows Processes, Threads, and Handles

#### Process (العملية)

العملية (Process) هي ببساطة برنامج قيد التنفيذ — أي تطبيق يتم تشغيله على النظام.

* عندما يقوم المستخدم بفتح تطبيق مثل Microsoft Edge أو Word، يقوم النظام بإنشاء عملية مستقلة له.
* كل عملية تملك بيئة تشغيل خاصة بها (Virtual Address Space) تمنعها من التداخل مع العمليات الأخرى.

##### خصائص العملية (Process Characteristics):

كل عملية في ويندوز تحتوي على مجموعة من الخصائص:

| **الخاصية** | **الوصف** |
| --- | --- |
| **Virtual Address Space** | مساحة الذاكرة الافتراضية الخاصة بالعملية. |
| **Executable Code** | الكود القابل للتنفيذ الخاص بالتطبيق. |
| **Handles to System Objects** | مقابض (Handles) لربط العملية بالموارد التي تستخدمها. |
| **Security Context** | يحدد صلاحيات وأذونات المستخدم الذي شغّل العملية. |
| **Unique Process Identifier (PID)** | رقم فريد يُميز كل عملية. |
| **Environment Variables** | المتغيرات البيئية التي تؤثر على كيفية تنفيذ العملية. |
| **Priority Class** | أولوية العملية عند الجدولة (Scheduling). |
| **Working Set Size** | الحد الأدنى والأقصى من الذاكرة التي يمكن أن تستخدمها العملية. |
| **Threads of Execution** | كل عملية تحتوي على خيط واحد على الأقل للتنفيذ (Primary Thread). |

#### 

#### Thread (الخيط)

الخيط (Thread) هو أصغر وحدة عمل يخصّص لها النظام وقتًا من وحدة المعالجة المركزية (CPU).

يمكن أن تحتوي العملية على عدة خيوط تعمل في نفس الوقت، وكل منها يمكنه تنفيذ جزء مختلف من كود البرنامج.

##### مميزات الخيوط:

جميع الخيوط ضمن نفس العملية تشارك نفس مساحة الذاكرة والموارد الخاصة بها.

كل خيط يمتلك:

* معالجات استثناءات (Exception Handlers)
* أولوية جدولة (Scheduling Priority)
* تخزين محلي للخيط (Thread Local Storage)
* معرف فريد (Thread ID)
* سياق الخيط (Thread Context)، ويشمل:
  + سجلات المعالج (Registers)
  + مكدس النواة (Kernel Stack)
  + مكدس المستخدم (User Stack)
  + Thread Environment Block (TEB)

يمكن لكل خيط أن يملك سياق أمني خاص (Security Context) — مما يسمح له أحيانًا بـ تقمّص صلاحيات مستخدم آخر (Impersonation)، وهي ميزة تُستخدم كثيرًا في الخوادم.

#### تعدد المهام في ويندوز (Preemptive Multitasking)

* نظام ويندوز يدعم تعدد المهام المسبق (Preemptive Multitasking).
  + أي يمكنه تشغيل عدة خيوط من عدة عمليات في نفس الوقت.
* في أنظمة متعددة المعالجات (Multiprocessor Systems)، يمكن تنفيذ عدد خيوط مساوٍ لعدد المعالجات فعليًا في نفس اللحظة.

هذا هو السبب في أن الأنظمة الحديثة (Multicore CPUs) يمكنها تشغيل عشرات التطبيقات في وقت واحد بدون بطء ملحوظ.

#### الـ Object والـ Handle

الكائن (Object) في ويندوز هو بنية بيانات (Data Structure) تمثل موردًا من موارد النظام، مثل:

* ملف (File)
* خيط (Thread)
* صورة رسومية (Graphic Image)
* أو أي مورد آخر يمكن للنظام التعامل معه

لكن:

لا يمكن للتطبيق الوصول مباشرة إلى الكائن نفسه.

يجب أن يستخدم مقبض (Handle) للوصول إليه.

##### وظيفة الـ Handle:

* الـ Handle هو معرّف (Reference) يمنحه النظام للتطبيق ليتمكن من التعامل مع الكائن.
* يحدد الـ Handle:
  + نوع المورد (ملف، عملية، خيط...)
  + موقع المورد في الذاكرة أو النظام

النظام يحتفظ داخليًا بـ جدول مقابض (Handle Table)، يربط كل Handle بالعنوان والخصائص الخاصة بالمورد.

### Windows Virtual Memory Address Space

الذاكرة الافتراضية (Virtual Memory) هي آلية لإدارة الذاكرة في أنظمة التشغيل تساعد الجهاز على تشغيل البرامج حتى عندما تكون ذاكرة الـRAM غير كافية.

عندما تمتلئ الـRAM، يقوم النظام بنقل جزء من البيانات مؤقتًا إلى القرص الصلب (Hard Drive)، ليستخدمه كما لو كان ذاكرة RAM إضافية.

هذا الجزء من القرص يسمى ملف الترحيل (Paging File).

#### كيفية عمل الذاكرة الافتراضية

* كل عملية (Process) تحصل على مساحة ذاكرة افتراضية خاصة بها، لا يمكن لأي عملية أخرى الوصول إليها إلا بإذن.
* النظام يستخدم جدول الصفحات (Page Table) لترجمة العناوين الافتراضية إلى عناوين فيزيائية حقيقية داخل القرص أو RAM.
* هذه الترجمة تتم في كل مرة يحاول فيها البرنامج الوصول إلى موقع في الذاكرة.

#### الوصول إلى الموارد

* البرامج (Processes) لا يمكنها الوصول مباشرة إلى موارد النظام الحساسة (مثل المنافذ، والمقابس TCP/IP).
* للوصول إلى هذه الموارد، تحتاج إلى مقابض (Handles) — وهي مثل “مفاتيح” تتيح مؤقتًا التفاعل مع مورد معين.
* بعد الانتهاء، يجب على العملية إغلاق المقبض لتتيح لباقي العمليات استخدام نفس المورد.

#### المخاطر الأمنية

بعض البرمجيات الخبيثة (Malware) تستغل آلية الذاكرة الافتراضية لتعديل أو قراءة محتوى ذاكرة عمليات أخرى.

يتم ذلك عبر دوال مثل:

* ReadProcessMemory (للقراءة)
* WriteProcessMemory (للكتابة)

لكن هذه العمليات تتطلب صلاحية خاصة تُعرف بـ SeDebugPrivilege.

#### سعة العناوين الافتراضية

* في أنظمة Windows 32-bit: يمكن لكل عملية استخدام مساحة عناوين افتراضية تصل إلى 4 جيجابايت.
* في أنظمة Windows 64-bit: يمكنها الوصول حتى 8 تيرابايت.

### ملفات نظام ويندوز Windows File System Overview

نظام الملفات (File System) هو الطريقة التي ينظم بها نظام التشغيل البيانات على وسائط التخزين (مثل الأقراص الصلبة والـ SSD والفلاش USB).

يحدد كيف يتم تخزين الملفات، تسميتها، إدارتها، والوصول إليها.

#### أنواع أنظمة الملفات

1. FAT (File Allocation Table)
   * نظام ملفات عام تستخدمه أنظمة Windows وMac وLinux.
   * غالبًا يُستخدم في وسائط USB.
   * يوجد منه نسختان رئيسيتان:
     + FAT16
     + FAT32 (الأكثر شيوعًا لأنه يدعم أحجام ملفات أكبر)
   * ❌ يملك قيودًا على حجم الملفات والتقسيمات لذلك نادر الاستخدام في الأقراص الحديثة.
2. exFAT (Extended FAT)
   * نسخة مطورة من FAT أزالت القيود على حجم الملفات.
   * ✅ مناسبة للأقراص الكبيرة ووسائط التخزين الحديثة.
   * ❌ ليست مدعومة بشكل واسع في الأنظمة الأخرى غير Windows وmacOS الحديث.
3. HFS+ (Hierarchical File System Plus)
   * النظام الافتراضي في macOS.
   * يسمح بأسماء ملفات تصل حتى 255 حرفًا.
   * ❌ يحتاج Windows إلى برنامج إضافي لقراءة هذا النظام.
4. EXT (Extended File System)
   * النظام الافتراضي في أنظمة Linux، وأحدث إصدار هو EXT4.
   * يدعم التوافق مع الإصدارات السابقة (EXT2 وEXT3).
   * ❌ غير مدعوم أصليًا في Windows إلا عبر برامج مساعدة.
5. NTFS (New Technology File System)
   * النظام الافتراضي والأكثر استخدامًا في Windows.
   * ✅ مدعوم أصليًا في جميع نسخ Windows وLinux (قراءة وكتابة).
   * في macOS يمكن القراءة فقط إلا إذا تم تثبيت برنامج مخصص للكتابة عليه.

#### مميزات NTFS

* أداء واستقرار عالي.
* متوافق مع أحجام ملفات وتقسيمات ضخمة.
* يدعم أنظمة الأمان (Permissions & Ownership).
* يدعم تشفير الملفات لحمايتها أثناء التخزين.
* يحتوي على آليات استرداد (Recovery Features) عند حدوث أخطاء.
* يستخدم الـ Security Descriptors لتحديد من يمكنه الوصول إلى كل ملف.

#### بنية NTFS

عند تهيئة القرص بـ NTFS يتم إنشاء:

1. قطاع الإقلاع (Partition Boot Sector)
   * أول 16 قطاع من القرص.
   * يحتوي على تعليمات تشغيل النظام وموقع جدول الملفات الرئيسي (MFT).
2. MFT (Master File Table)
   * قاعدة بيانات تحتوي على موقع وخصائص كل ملف ومجلد على القسم.
   * يوجد منها نسخة احتياطية (Mirror) لحماية البيانات في حال تلف النسخة الأصلية.
3. System Files
   * ملفات نظامية مخفية لتخزين البيانات الوصفية (Metadata) الخاصة بالملفات وخصائصها.
4. File Area
   * المنطقة التي يتم فيها تخزين الملفات والمجلدات الفعلية.

#### NTFS Alternate Data Streams (ADS)

* ميزة في NTFS تسمح بإضافة تيارات بيانات إضافية (Alternate Data Streams) إلى ملف واحد دون أن تظهر في الحجم أو في القائمة العادية.
* تُستخدم لتخزين معلومات إضافية أو إخفاء بيانات داخل ملف موجود.

**شكل الملف:**

myFile.txt:SecretData

🔸 البيانات المخفية **لا تظهر عند استخدام dir العادي**،  
 لكن يمكن الوصول إليها عبر:

more < myFile.txt:SecretData

⚠️ هذه التقنية قد تُستغل لإخفاء **بيانات ضارة أو برمجيات خبيثة (Malware)**،  
 لذا تُعتبر من النقاط الحساسة في تحليل أمان الأنظمة (خاصة في بيئات الـSOC).

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://docs.google.com/document/d/1O3gd3l18fuzfZVdMHIGejTbamwzv2xc9Uu7-yCt8MY0/edit?usp=sharing)

### هيكل نظام الملفات (File System Structure) في Windows

نظام الملفات في ويندوز منظم بطريقة هرمية (Hierarchical).

في الأعلى يوجد الجذر (Root)، وغالبًا يكون هو القرص C:، وتحت هذا الجذر توجد مجلدات (Directories) وداخلها ملفات (Files) أو مجلدات أخرى.

#### أهم المجلدات في نظام ويندوز

* C:\ (الجذر الرئيسي)

يحتوي على المجلدات الأساسية التالية:

| **المجلد** | **الوصف** |
| --- | --- |
| Boot | يحتوي ملفات الإقلاع، وغالبًا يكون مخفيًا. |
| Logs | لتخزين سجلات النظام والأحداث (Event Logs). |
| PerfLogs | لتخزين سجلات الأداء إذا تم تفعيلها. |
| Program Files | لتثبيت التطبيقات **64-bit**. |
| Program Files (x86) | لتثبيت التطبيقات **32-bit**. |
| ProgramData | لتخزين بيانات التطبيقات العامة. |
| Users | يحتوي مجلدات المستخدمين (كل مستخدم له مجلد خاص). |
| Windows | يحتوي ملفات النظام، ومن ضمنه: |

* System (ملفات DLL القديمة 16-bit)
* System32 (ملفات DLL لـ 64-bit)
* SysWOW64 (ملفات DLL لـ 32-bit)
* WinSxS (نُسخ احتياطية لملفات النظام أثناء التشغيل).

#### مسار الملفات (File System Path)

* المسار الكامل (Fully Qualified Path):

هو الطريق الكامل من الجذر حتى الملف.

مثال:

| C:\Users\employee\Downloads\sample.txt |
| --- |

* المسار النسبي (Relative Path):

يُستخدم عندما تكون بالفعل داخل مجلد معين، وتريد الإشارة إلى ملف داخل نفس المجلد أو مجلد فرعي.

مثال:

| copy Downloads\myDoc.pdf Documents\myDoc.pdf |
| --- |

| **الامتداد** | **نوع الملف** |
| --- | --- |
| .exe | برنامج تنفيذي |
| .txt | ملف نصي |
| .pdf | ملف PDF |
| .rtf | نص منسق |
| .bmp | صورة نقطية |

#### أجزاء اسم الملف (File Naming Conventions)

كل ملف يتكون من جزأين:

1. اسم أساسي (Base Name)
2. امتداد (Extension) يحدد نوع الملف والبرنامج الذي يفتحه.

#### طول المسار (Path Length)

* الحد الأقصى الافتراضي للمسار هو 260 حرفًا (بما في ذلك حرف محرك الأقراص والنهاية).
* في Windows 10 يمكن تفعيله ليصل إلى حوالي 32,767 حرفًا باستخدام النظام الموسّع (Unicode Path).
* بعض الأدوات القديمة (مثل cmd) لا تدعم المسارات الطويلة جدًا.

#### 

#### خصائص الملفات (File Properties)

كل ملف يحتوي على خصائص (Properties) مثل:

* الاسم، الحجم، تاريخ الإنشاء والتعديل.
* نوع الملف والبرنامج الذي يفتحه.
* الصلاحيات (Permissions) والملكية (Ownership).

الملكية والصلاحيات (Ownership & Permissions):

| **الصلاحية** | **المعنى** |
| --- | --- |
| Read | عرض الملفات والمجلدات |
| Write | تعديل الملفات أو إنشاء مجلدات |
| Read & Execute | تشغيل الملفات التنفيذية |
| Modify | تعديل وقراءة الملفات |
| Full Control | تحكم كامل في القراءة، الكتابة، التعديل، والحذف |

* مالك الملف (عادةً المستخدم الذي أنشأه) والمشرف (Administrator) يمكنهما التحكم في من يملك صلاحية الوصول.
* نظام الصلاحيات يعتمد على نظام NTFS.

إذا كان هناك تعارض بين السماح والرفض، فإن الرفض (**Deny**) له الأولوية دائمًا.

| **السمة** | **الوظيفة** |
| --- | --- |
| R (Read-only) | يمنع التعديل أو الحذف. |
| A (Archive) | يعلّم الملف كجاهز للنسخ الاحتياطي. |
| S (System) | يُستخدم بواسطة النظام فقط. |
| H (Hidden) | يخفي الملف من القوائم العادية. |
| I (Not Indexed) | يمنع فهرسة الملف للبحث. |
| E (Encrypted) | يرمّز محتوى الملف. |
| C (Compressed) | يقلل من حجمه على القرص. |
| T (Temporary) | ملف مؤقت يتم حذفه بعد الاستخدام. |

#### سمات الملفات (File Attributes)

هي بيانات إضافية تخبر النظام كيف يتعامل مع الملف.

يمكن تعديل السمات من خلال:

واجهة الخصائص (Properties → Advanced)

أو من سطر الأوامر (CMD) عبر الأمر:

| attrib +R sample.txt  attrib -A sample.txt |
| --- |

### الحسابات المحلية وحسابات النطاق Windows Domains and Local User Accounts

#### 1. Windows Domains (نطاقات ويندوز)

* النطاق (Domain) هو مجموعة من أجهزة الكمبيوتر والمستخدمين تُدار مركزيًا من قبل خادم يسمى DC أو (Domain Controller).
* عند الدخول (login) إلى أي جهاز ضمن النطاق، يتم التحقق من هوية المستخدم عبر الـ DC باستخدام AD أو (Aktive Directory)
* النطاق يوفر:
  + إدارة مركزية للمستخدمين والصلاحيات.
  + مشاركة الموارد بسهولة (كالطابعات والمجلدات).
  + سياسات موحدة عبر Group Policy لتحديد إعدادات الأمان وسلوك الأجهزة.
* (Active Directory AD):
  + يعتمد على LDAP لتخزين معلومات عن جميع الكائنات (Objects) مثل المستخدمين، الأجهزة، والطابعات.
  + ينظم الكائنات داخل حاويات (Containers) ومجالات (Domains) بهيكل هرمي.
* التكرار (Redundancy):
  + المؤسسات الصغيرة: عادة تحتاج Domain Controller واحد أو اثنين.
  + المؤسسات الكبيرة: تحتاج عدة Domain Controllers في كل موقع لضمان توفر الخدمة دائمًا.

#### 2. Local Users and Groups (المستخدمون والمجموعات المحلية)

* في حال لم يكن النظام جزءًا من نطاق، يمكنه العمل كـ جهاز مستقل (Standalone).
* الحسابات المحلية تُدار على الجهاز نفسه فقط.

**الحسابات الافتراضية في ويندوز**:

1. Administrator (المسؤول):
   * يمتلك صلاحيات عالية جدًا.
   * معطّل بشكل افتراضي في الإصدارات الحديثة لأسباب أمنية.
   * يُفضل إنشاء مستخدم جديد بصلاحيات إدارية بدلاً من استخدامه مباشرة.
   * يمكن تنفيذ المهام الإدارية عبر خيار “Run as Administrator” بدون الحاجة لتسجيل الدخول كمسؤول دائمًا.
2. Guest (الضيف):
   * مخصص للاستخدام المؤقت من قبل أشخاص ليس لديهم حسابات.
   * مُعطّل افتراضيًا وينبغي أن يبقى كذلك لأسباب أمنية.
   * لا يحتاج كلمة مرور.

لا تُحفظ التغييرات بعد تسجيل الخروج.

#### 3. User Groups (مجموعات المستخدمين)

* المجموعات تحدد ما الذي يمكن للمستخدم فعله على النظام.
* أمثلة:
  + Administrators group ➜ وصول كامل للنظام.
  + Backup Operators group ➜ صلاحيات النسخ الاحتياطي والاستعادة.
  + Users group ➜ المستخدمون العاديون، يمكنهم استخدام البرامج والطابعات وقفل النظام.

#### 4. Microsoft Accounts (حسابات مايكروسوفت السحابية)

* منذ Windows 8، أصبح بإمكان المستخدم تسجيل الدخول باستخدام حساب Microsoft (مثل Outlook أو Hotmail).
* المزايا:
  + مزامنة الإعدادات والملفات تلقائيًا عبر الأجهزة.
  + الوصول إلى ملفاتك على OneDrive مباشرة بعد تسجيل الدخول.
  + إذا استخدمت حسابًا محليًا بدلاً منه، ستفقد المزامنة السحابية ما لم تسجّل دخولك يدويًا إلى الخدمات السحابية.

### واجهة سطر الأوامر

#### 1) أفكار أساسية

* الـCLI في ويندوز قوي ومهم لأتمتة التحقيقات، تشغيل أدوات، واستخلاص دلائل سريعة.
* أوامر ومسارات غير حسّاسة لحالة الأحرف (case-insensitive) بشكل افتراضي، لكن النظام يحتفظ بالحالة عند الإنشاء.
* إذا حذفت ملفاً من الـCLI فإنه يُحذف نهائيًا — لا يذهب إلى سلة المحذوفات. (احذر عند التحقيق الجنائي).
* إذا كان اسم ملف/مجلد يحتوي مسافة، ضع المسار بين علامتي اقتباس "...".
* استخدم / ? بعد أي أمر لعرض المساعدة مثال

ping /?

#### 2) أوامر ونصائح عملية مهمة (أمثلة)

* استعراض المجلدات:
  + dir — عرض الملفات
  + dir /p — (عرض صفحة صفحة اختصار للتفتيش الطويل)
* التنقل في النظام:
  + cd foldername — ادخل مجلد
  + cd .. — اصعد مستوى واحد
  + mkdir MyDirectory — إنشاء مجلد
  + cls — مسح الشاشة
* تشغيل تطبيق من السطر:
  + notepad.exe — يفتح Notepad
  + Ipconfig — عرض إعدادات الشبكة
* التعامل مع الملفات التي بها فراغات:
  + copy "My File.txt" D:\
* التكملة التلقائية:
  + استخدم Tab لِـ autocomplete للأسماء (يوفر وقت ويقلل الأخطاء).
* سجل الأوامر والتاريخ:
  + استدعاء سابق: مفاتيح السهم ↑ ↓
  + doskey /history — عرض سجل الجلسة الحالي
  + F7 — نافذة تاريخ الأوامر للتحديد والإنفاذ
  + F8 — بحث بالتسلسل في التاريخ بناءً على البداية المكتوبة
* التنقل بين محركات أقراص:
  + D: — يتحول إلى محرك D ويتم تغيير الموجه إليه

#### 

#### 3) ملاحظات عملية/أمنية (لمختص الأمن)

* لا تمسح أي دليل عبر CLI أثناء التحقيق؛ del يحذف دون المرور بسلة المحذوفات — استنسخ الصورة (image) أولاً.
* سجل الأوامر (history) يمكن أن يحتوي أدلة على أوامر نفذت؛ لكن التاريخ يختفي عند إغلاق الجلسة — اجعل التوثيق جزءًا من إجراءات الـIR.
* الأتمتة عبر سكربتات (batch/PowerShell) مفيدة جداً، لكن احذر من تشغيل سكربت مجهول المصدر بصلاحيات المستخدم العادي أو المسؤول.
* التكملة التلقائية والاقتباسات تمنع أخطاء مسارات/ملفات حساسة؛ استخدمها دائماً عند التعامل مع ملفات نظام أو أدلة التحقيق.
* command /? مفيدة جداً لاكتشاف خيارات قد تؤثر على السلوك (مثل ping /n, dir /s…).

#### 4) أمثلة سريعة للذاكرة

* dir /p — صفحة صفحة
* cd .. — اصعد مستوى
* mkdir Test → cd Test
* copy "Class Files.zip" D:
* doskey /history ، مفاتيح F7 / F8

### Windows Powershell

**هذا الدرس عبارة عن تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://docs.google.com/document/d/1rBVEeXUKuHmofWOdhLBievfadlL1r8626Fqmn-dsi7s/edit?usp=sharing)

### أوامر net في ويندوز Windows net Command

#### ما هي أوامر net؟

* + هي مجموعة أوامر خاصة بنظام Windows تُستخدم لإدارة الحسابات، الشبكات، الخدمات، والمشاركات.
  + جميعها تبدأ بكلمة net مثل:
    - net accounts
    - net user
    - net use
    - net start / net stop

#### عرض جميع أوامر net:

| net help |
| --- |

* + يعرض قائمة كاملة بالأوامر المتاحة مثل:
    - net user – إدارة حسابات المستخدمين
    - net use – ربط وحدات تخزين شبكية
    - net start / net stop – تشغيل وإيقاف الخدمات
    - net accounts – سياسات كلمات المرور للمستخدمين

#### إدارة سياسة كلمات المرور:

| net accounts |
| --- |

* + يُظهر إعدادات كلمات المرور (مثل الطول الأدنى، مدة صلاحية الكلمة، إلخ).
  + يمكن تعديلها باستخدام:

| net help accounts |
| --- |

#### ربط مجلد شبكي (Network Drive):

| net use z: \\192.168.7.88\users /user:emendez |
| --- |

* + يربط مجلد “users” من جهاز آخر إلى القرص المحلي Z:
  + مفيد للوصول إلى ملفات شبكية، مثل ملفات الأدلة الجنائية.

#### إدارة الخدمات (Services):

| net stop fileinfo net start fileinfo |
| --- |

* + إيقاف وتشغيل خدمة معينة — مهارة مطلوبة أثناء اختبار الاختراق أو تحليل نشاطات مشبوهة في النظام.

#### إدارة المستخدمين:

| net user USER1 password /add |
| --- |

* + إنشاء مستخدم جديد أو تعديل الصلاحيات بدون واجهة رسومية.
  + هذا مهم في حالات الطوارئ أو التحقيقات الأمنية.

### Controlling Startup Services and Executing System Shutdown

#### 1. التحكم في خدمات بدء التشغيل (Startup Services)

* الريجستري (Registry) يحتوي على إعدادات تشغيل النظام، ويمكن التحكم بالخدمات التي تبدأ مع النظام من خلاله، لكن التعديل اليدوي خطير جدًا وقد يؤدي إلى تلف النظام.
* الأداة Msconfig.exe هي الطريقة الآمنة لتعديل إعدادات بدء التشغيل دون المساس بالريجستري مباشرة.
* للوصول إلى الأداة، اكتب msconfig في موجه الأوامر أو مربع البحث.

أهم التبويبات في الأداة:

* General:
  + Normal startup → الوضع العادي.
  + Diagnostic startup → لتشغيل النظام بأدنى الخدمات (لإصلاح الأخطاء).
  + Selective startup → لتحديد ما يتم تشغيله يدويًا.
* Boot:
  + لاختيار نظام التشغيل عند وجود أكثر من نسخة.
  + التحكم في Safe Boot (الوضع الآمن) لأغراض الصيانة.
* Services:
  + عرض حالة الخدمات (Running أو Stopped).
  + يمكن تفعيل أو تعطيل الخدمات لتبدأ أو لا تبدأ عند الإقلاع.
* Startup:
  + عرض التطبيقات التي تعمل تلقائيًا عند بدء التشغيل.
  + يمكن تعطيل التطبيقات المشبوهة — مفيد جدًا لمحللي الأمن عند البحث عن برمجيات ضارة تعمل تلقائيًا.
* Tools:
  + يحتوي على مجموعة أدوات النظام يمكن تشغيلها مباشرة.

📌 ملاحظة أمنية:

من خلال تبويب Startup وServices يمكن التعرف على التطبيقات والخدمات غير المعروفة أو المريبة، وهي نقطة انطلاق مهمة لتحليل برمجيات خبيثة (malware) تعمل عند الإقلاع.

#### 

#### 2. إيقاف النظام (System Shutdown)

* إيقاف النظام بشكل صحيح ضروري لتجنب تلف الملفات المفتوحة أو الإعدادات غير المحفوظة.
* عملية الإيقاف تمر بعدة مراحل:

1. النظام يبلغ التطبيقات والخدمات بوجود أمر إيقاف.
2. يتم إنهاء العمليات المفتوحة وتنظيف الملفات.
3. إذا لم تستجب خدمة ما، يظهر تنبيه للمستخدم لاختيار انتظار أو إغلاق إجباري.

طرق الإيقاف:

* من قائمة Start: الخيار الافتراضي لإيقاف النظام.
* Ctrl + Alt + Delete: تظهر قائمة فيها خيارات منها الإيقاف (قد تختلف حسب نسخة ويندوز).
* من سطر الأوامر:
  + الأمر: shutdown
  + يمكن إضافة معاملات لتحديد نوع الإيقاف، مثل:
    - shutdown /s → إيقاف النظام.
    - shutdown /r → إعادة التشغيل.
    - shutdown /h → وضع السكون (hibernate).

أوضاع الإيقاف:

* Shutdown: إيقاف النظام تمامًا.
* Restart: إيقاف وإعادة تشغيل.
* Sleep / Hibernate: حفظ حالة النظام على القرص ثم الدخول في وضع طاقة منخفضة (مفيد للحواسيب المحمولة).

### التحكم بالخدمات والعمليات Controlling Services and Processes

هذا الدرس يشرح كيف يمكن **إدارة العمليات (Processes)** و**الخدمات (Services)** في نظام **Windows** — وهي من المهارات الأساسية لأي شخص يعمل في الأمن السيبراني أو إدارة الأنظمة، لأنها تساعد في **مراقبة أداء النظام** واكتشاف **الأنشطة المشبوهة** مثل البرمجيات الخبيثة أو العمليات غير المصرح بها.

#### الأدوات الأساسية

##### 1. Task Manager (مدير المهام)

🔹 أداة رئيسية لمراقبة وإدارة العمليات والخدمات قيد التشغيل.  
 🔹 يمكن فتحها عبر **Ctrl + Alt + Delete → Task Manager** أو **الزر الأيمن على شريط المهام**.  
 🔹 تُستخدم لتحديد العمليات التي تستهلك موارد غير طبيعية أو قد تكون ضارة.

أهم التبويبات:

* **Processes:** يعرض جميع العمليات والتطبيقات العاملة، ويتيح فرزها حسب الاستخدام (CPU, RAM, Disk, Network).  
   مفيد لاكتشاف البرامج الخبيثة التي تستهلك موارد غير طبيعية.  
   ⮕ يمكن إنهاء أي عملية مشبوهة يدويًا.
* **Performance:** يعرض أداء النظام في الوقت الحقيقي (CPU, Memory, Disk, Network).
* **App History:** يُظهر التاريخ الزمني لاستهلاك التطبيقات للموارد.
* **Startup:** يتيح تعطيل البرامج التي تبدأ تلقائيًا مع تشغيل النظام، مما يساعد في تسريع الإقلاع وتقليل المخاطر الأمنية.
* **Users:** يعرض المستخدمين النشطين على النظام والعمليات المرتبطة بكل مستخدم.
* **Details:** يوفر معلومات متقدمة حول كل عملية (مثل أولوية التنفيذ أو تخصيص نواة CPU).  
   يتضمن ميزة **Analyze Wait Chain** لمعرفة سبب توقف تطبيق عن الاستجابة.
* **Services:** يعرض الخدمات النشطة والموقوفة، مع إمكانية بدء أو إيقاف أو البحث عن معلومات الخدمة عبر الإنترنت.

##### 

##### 2. MSCONFIG Utility

🔹 أداة قديمة نوعًا ما لإدارة إعدادات النظام والخدمات عند الإقلاع.  
 🔹 مفيدة فقط لتمكين أو تعطيل **الخدمات** عند إعادة التشغيل.  
 🔹 لم تعد تُستخدم كثيرًا في الإصدارات الحديثة من Windows لأنها محدودة مقارنة بـ Task Manager.

##### 3. Computer Management

🔹 يمكن تشغيلها بالأمر:

compmgmt.msc

🔹 تتيح الوصول إلى **قسم Services and Applications** الذي يسمح بإدارة الخدمات بشكل متقدم (بدء، إيقاف، تغيير نوع التشغيل).

#### الجانب الأمني

في سياق الأمن السيبراني، تعتبر مراقبة العمليات والخدمات خطوة أساسية في:

* **تحليل السلوك الغريب** للنظام (مثل عمليات تستهلك CPU عاليًا أو تعمل دون مبرر).
* **الكشف عن البرمجيات الخبيثة** التي تحاول التخفي كخدمات نظام.
* **تحليل الأداء** لتحديد ما إذا كان هناك نشاط مريب أثناء الاختراق أو الإصابة.

ميزة **Search Online** المدمجة في Task Manager الحديث تساعد في التحقق بسرعة من هوية أي عملية مشبوهة.

### مراقبة موارد النظام Monitoring System Resources

هذا الدرس يشرح كيفية **مراقبة موارد النظام (System Resources)** في أنظمة **Windows** — وهي مهارة أساسية لأي مختص في **الأمن السيبراني** لفهم أداء النظام واكتشاف **الأنشطة غير الطبيعية** مثل استهلاك زائد للموارد من قبل برمجيات خبيثة أو عمليات مشبوهة.

#### مراقبة الموارد عبر Task Manager

يُعد **Task Manager** الأداة الأساسية في ويندوز لمراقبة الأداء في الوقت الحقيقي.  
 يوجد فيه عدة تبويبات مهمة تتعلق بالأداء:

1. Performance

* يعرض **نسبة استخدام CPU** ومعدل استخدام **الذاكرة (RAM)** بشكل رسومي.
* يمكن من خلاله معرفة استهلاك الموارد العام للنظام.
* يفيد في تحديد ما إذا كان النظام تحت ضغط مفرط أو في حالة طبيعية.

2. Networking

* يعرض معدل استخدام الشبكة لكل بطاقة شبكة (Network Adapter).
* يساعد في **تتبع العمليات أو البرامج التي تستهلك نطاقًا عريضًا (Bandwidth)** بشكل مريب، وهو أمر مهم لاكتشاف **نقل بيانات غير مصرح به**.

3. Users

* يعرض المستخدمين النشطين على النظام.
* يمكن من خلاله **قطع الاتصال** عن المستخدمين أو **تسجيل خروجهم** — مفيد في **التحكم في الجلسات النشطة** خلال تحقيقات الأمان أو أثناء التحكم بالخوادم.

#### 

#### مراقبة أعمق عبر Resource Monitor

عندما تحتاج إلى تحليل أعمق من Task Manager، يمكنك استخدام **Resource Monitor** (يُفتح بالأمر: resmon.exe أو من تبويب Performance في Task Manager).

🔸 المهام الرئيسية للأداة:

* مراقبة موارد النظام الأربعة الأساسية:

1. **CPU**
2. **Memory**
3. **Disk**
4. **Network**

* يمكن **تصفية النتائج حسب العملية (Process)**، مثل تحديد Firefox أو Wireshark لمعرفة كيف يستخدم كل مورد على حدة.  
  بعد تحديد عملية معينة، يمكنك:
  + معرفة الملفات التي تقرأها أو تكتبها العملية.
  + معرفة حجم البيانات المرسلة والمستقبلة عبر الشبكة.
  + تحليل استهلاك المعالج والذاكرة بدقة عالية.

#### الاستخدام الأمني:

* يساعد في **تحليل السلوك الغريب** لعملية معينة.
* يمكن من خلاله تحديد **عمليات التجسس أو البرامج الضارة** التي تستهلك الشبكة أو القرص بشكل غير طبيعي.
* يُستخدم أثناء **تحقيقات الحوادث الأمنية (Incident Response)** لتحديد البرامج المسببة لمشكلات الأداء أو النشاط المريب.

### عملية الإقلاع في ويندوز Windows Boot Process

#### المراحل الثلاث لعملية الإقلاع (Boot Process)

1. BIOS Initialization Phase
   * يبدأ النظام من الـ BIOS أو UEFI firmware.
   * يقوم الـ firmware بتهيئة العتاد (hardware) وتشغيل Power-On Self-Test) POST) للتأكد من سلامة المكونات.
   * يبحث عن قرص إقلاع صالح، ثم يقرأ الـ Master Boot Record) MBR) ويشغّل Bootmgr.exe.
2. Operating System Loader Phase
   * يبدأ Bootmgr.exe بتشغيل Winload.exe المسؤول عن تحميل التعريفات الأساسية للنظام (system drivers) الضرورية لتشغيل النظام.
   * يقوم Winload.exe بتحميل إعدادات الإقلاع من Boot Configuration Database) BCD).
   * إذا كان النظام في وضع السبات (hibernate)، يُستخدم Winresume.exe لإعادة الحالة السابقة من ملف hiberfil.sys.
   * بعد ذلك يبدأ Winload.exe بتحميل نواة النظام (kernel) وتهيئة Hardware Abstraction Layer) (HAL.
3. Operating System Initialization Phase
   * يبدأ النظام بتشغيل الخدمات (services) وفقاً لإعداداتها في Windows Registry.
   * يتم تشغيل Session Manager Subsystem) SMSS) الذي يجهز بيئة المستخدم ويشغّل Winlogon لبدء عملية تسجيل الدخول.
   * بعد تسجيل الدخول، يتم تحميل سطح المكتب (Explorer).

#### الجوانب الأمنية الهامة

* النواة (Kernel) هي أهم جزء أمني في النظام، وأي تعديل أو اختراق فيها يُعتبر كارثيًا.
* برامج الإقلاع مثل Bootmgr وWinload قد تكون هدفًا لهجمات Bootkits التي تحقن تعليمات خبيثة في هذه المرحلة المبكرة.
* Kernel Mode Code Signing) KMCS) يمنع تشغيل أي تعريف أو كود غير موقّع رقمياً، وهو خط دفاع أساسي أثناء الإقلاع.
* في الأنظمة الحديثة UEFI، عملية الإقلاع تتم بطريقة أكثر أمانًا من BIOS لأنها لا تستخدم "real mode" القديم وتدعم ميزات مثل Secure Boot.

#### بدء الخدمات عند الإقلاع (Auto-start Services)

* الخدمات التي تعمل تلقائيًا تُحدد في Windows Registry، خصوصًا في المسارات التالية:

| HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run  HKEY\_LOCAL\_MACHINE\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\Run |
| --- |

* وجود برامج غير معروفة في هذه المسارات قد يدل على برمجيات خبيثة (malware) تعمل عند الإقلاع.
* لذلك، فحص هذه المفاتيح يُعتبر خطوة أساسية في تحليل البرمجيات الخبيثة أو التحقيقات الجنائية الرقمية.

#### ملخص أمني تطبيقي

كمحلل أمني، من الضروري:

* معرفة تسلسل الإقلاع لفهم أين يمكن أن يزرع المهاجمون برمجياتهم.
* استخدام أدوات مثل Autoruns أو regedit لمراجعة البرامج التي تعمل تلقائياً.
* التأكد من تفعيل UEFI + Secure Boot لتقليل فرص إصابة النظام ببرمجيات boot-level malware.
* مراقبة سلامة ملفات الإقلاع مثل bootmgr.exe وwinload.exe عبر التواقيع الرقمية.

### الشبكات في نظام ويندوز Windows Networking

الدرس يوضح أساسيات إعداد الشبكات في أنظمة Windows وبعض الأدوات المهمة لاختبار الاتصال بالشبكة وتحليلها، وهي أمور ضرورية لفهم حركة المرور (Network Traffic) وكشف الأنشطة المشبوهة في الأنظمة المصابة.

#### الأفكار الأساسية

1. إعداد الشبكة (Network Configuration):
   * يتم ضبط إعدادات الشبكة عبر واجهة النظام (Network and Sharing Center → Change adapter settings).
   * يمكن إعداد عنوان IP بشكل يدوي (Static) أو ديناميكي (DHCP).
   * أهم القيم التي تُضبط يدويًا:
     + IP Address: عنوان الجهاز على الشبكة.
     + Subnet Mask: لتحديد نطاق الشبكة المحلية.
     + Default Gateway: العنوان الذي يوجه الحركة خارج الشبكة المحلية.
     + DNS Servers: تُستخدم لترجمة أسماء النطاقات إلى عناوين IP.
2. أداة netsh.exe:
   * أداة قوية لإدارة الشبكات من سطر الأوامر، تُستخدم لتعديل إعدادات الشبكة أو حفظها كسكريبتات — مفيدة في التحقيقات أو الأتمتة.

#### أدوات مهمة لاختبار وتشخيص الشبكات

1. Ping
   * ترسل طلبات ICMP Echo لاختبار الاتصال مع جهاز آخر.
   * نجاح الرد يعني وجود اتصال فعّال بالشبكة.
   * مفيد للتحقق من سلامة الاتصال الفيزيائي والمنطقي.
   * كمحلل أمني، يمكن استخدامه لاختبار توفر الأجهزة أو الجدران النارية (firewalls).
2. Nslookup
   * أداة لاختبار عمل خوادم DNS.
   * تُستخدم لمعرفة ما إذا كانت الأسماء تُترجم إلى عناوين IP بشكل صحيح.
   * مفيدة في كشف عمليات إعادة التوجيه الخبيثة (DNS Hijacking).
3. Netstat
   * أداة تحليل اتصالات النظام.
   * تُظهر:
     + الاتصالات المفتوحة (Inbound/Outbound).
     + المنافذ التي تعمل عليها الخدمات.
     + المعرفات (PIDs) التي تملك الاتصالات.

أوامر مفيدة:

netstat -abno → يعرض المنافذ والبرامج المالكة لها.

أداة مهمة جدًا لتحليل الاتصالات المشبوهة أو الباك دورز (Backdoors) أو اتصال البرمجيات الخبيثة بخوادم تحكم (C2 Servers).

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://docs.google.com/document/d/1snzPkgctLw9jMouCKhsCTpaV6KOobMl0IEKv08cJgbE/edit?usp=sharing)

### أمر netstat في ويندوز Windows netstat Command

#### أهمية الأمر:

netstat هو أداة قوية لتحليل الشبكة من خلال سطر الأوامر في نظام Windows، تُستخدم لفحص **الاتصالات النشطة**، **المنافذ المفتوحة**، **جدول التوجيه (routing table)**، و**إحصاءات البروتوكولات** مثل TCP وUDP وIPv4 وIPv6.

في مجال **الأمن السيبراني**، يعد netstat أداة تحليل أساسية لاكتشاف **الاتصالات المشبوهة أو نشاط البرامج الضارة (malware)** التي تفتح منافذ خلفية أو تتصل بخوادم تحكم (C2 servers).

#### أوامر netstat الأساسية والمفيدة للأمن السيبراني:

| **الأمر** | **الوظيفة** | **الاستخدام الأمني** |
| --- | --- | --- |
| netstat | يعرض الاتصالات النشطة فقط | معرفة الاتصالات المفتوحة حالياً |
| netstat -r | يعرض جدول التوجيه | تحليل مسار الشبكة وتأكد من الإعدادات الصحيحة |
| netstat -an | يعرض كل الاتصالات بالأرقام فقط | تحديد المنافذ والعناوين بدقة رقمية دون أسماء |
| netstat -s | يعرض إحصائيات البروتوكولات | مراقبة حركة مرور الشبكة عبر البروتوكولات |
| netstat -e | يعرض إحصاءات Ethernet | التحقق من حجم البيانات المرسلة والمستلمة |
| netstat -p TCP | يعرض فقط اتصالات بروتوكول TCP | حصر تحليل الاتصالات في بروتوكول معين |
| netstat -abno | **الأهم في التحليل الأمني**: يعرض البرامج (executables) المالكة للاتصالات مع PID | تحديد العملية (process) التي أنشأت الاتصال ومعرفة إن كانت مشبوهة |

**ملاحظة: استخدم الأوامر التي في الجدول كتمرين عملي**

ملاحظات تقنية:

* netstat مفيد جداً في تحليل حركة المرور المحلية قبل استخدام أدوات أعمق مثل Wireshark.
* يمكن تشغيل netstat بشكل دوري في التحقيقات (Incident Response) لرصد الأنشطة غير المعتادة.
* الأمر ما زال مستخدماً وفعالاً حتى اليوم رغم وجود أدوات أحدث، لذلك يُعتبر من **الأساسيات المهمة في الأمن السيبراني**.

### الوصول إلى موارد الشبكة باستخدام ويندوز Accessing Network Resources with Windows

في أنظمة Windows، الوصول إلى الموارد الشبكية يتم عادةً عبر **بروتوكول SMB (Server Message Block)**، وهو البروتوكول الذي يسمح بمشاركة الملفات والطابعات بين الأجهزة على الشبكة. تم تطوير SMB في الأصل من قبل IBM، ثم تبنته Microsoft وأصبح جزءًا أساسيًا من أنظمة Windows. هناك أيضًا نسخة مفتوحة المصدر تُعرف باسم **Samba** تتيح لأنظمة تشغيل أخرى التفاعل مع أجهزة Windows.

يُستخدم **SMB** بشكل رئيسي للوصول إلى الملفات أو أنظمة الملفات البعيدة. يتم تحديد الموارد باستخدام صيغة   
تسمى (**UNC (Universal Naming Convention** بالشكل التالي:

| \\server\_name\share\_name\file |
| --- |

* server\_name: هو اسم الخادم أو عنوان الـIP.
* share\_name: اسم المجلد أو القسم الذي تمت مشاركته.
* file: الملف أو المجلد المطلوب الوصول إليه.

بعض المشاركات تكون إدارية (Administrative Shares) ومخفية عن المستخدمين العاديين، وتُستخدم من قبل المسؤولين فقط. من أهمها:

* C$ للوصول إلى محرك الأقراص C بالكامل.
* ADMIN$ يشير إلى مجلد النظام C:\Windows.
* IPC$ يُستخدم للاتصال بين العمليات (Interprocess Communication).

يمكنك رؤية المشاركات المتاحة على النظام باستخدام الأمر:

| net share |
| --- |

وللوصول إلى مجلد مشترك على جهاز بعيد، نستخدم الأمر:

| net use z: \\192.168.7.90\Users /user:joe |
| --- |

* حيث يتم ربط المشاركة بمحرك أقراص محلي (مثل Z:).
* إذا كانت بيانات الاعتماد مخزنة مسبقًا، لن يُطلب إدخال كلمة مرور مجددًا.

يمكن أيضًا الوصول إلى المشاركات من واجهة File Explorer بكتابة المسار UNC في شريط العنوان.

**كمتخصص أمن سيبراني:**

يجب أن تكون على دراية بهذه المفاهيم لأن SMB يعد هدفًا شائعًا للهجمات (مثل WannaCry الذي استغل ثغرة في SMBv1).

معرفة كيفية التعرف على المشاركات المخفية أو المفتوحة يساعدك في تحليل الثغرات، التحقق من صلاحيات الوصول، واكتشاف النشاطات غير الطبيعية على الشبكة | **ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://docs.google.com/document/d/1YkkN9Cp954Aqm6Y-0qy1iS_zC84rrQ3Wg-kh1dPhebk/edit?usp=sharing)

### سجل ويندوز Windows Registry

#### الفكرة الأساسية

Windows Registry هو قاعدة بيانات مركزية تُخزّن فيها إعدادات النظام، المستخدمين، البرامج، والأجهزة في نظام ويندوز. أي تعديل أو تغيير في النظام تقريبًا ينعكس في هذا السجل، مما يجعله **عنصرًا حساسًا جدًا من الناحية الأمنية**.

#### مكونات السجلّ (Hives)

السجلّ منظم بشكل هرمي، وأعلى مستوى في هذا الهيكل يُسمّى **Hive**. كل Hive مرتبط بملف في النظام يحتوي على مفاتيح (Keys) وقيم (Values).  
 أهم الـ Hives هي:

| Hive | الوظيفة |
| --- | --- |
| HKEY\_CURRENT\_USER (HKCU) | يحتوي إعدادات المستخدم الحالي (ألوان، إعدادات سطح المكتب، التطبيقات...) |
| HKEY\_USERS (HKU) | معلومات كل المستخدمين في النظام |
| HKEY\_CLASSES\_ROOT (HKCR) | روابط الملفات والـ OLE (Object Linking and Embedding) |
| HKEY\_LOCAL\_MACHINE (HKLM) | إعدادات النظام والعتاد (Hardware & System settings) |
| HKEY\_CURRENT\_CONFIG (HKCC) | إعدادات العتاد الحالية (Current hardware profile) |

#### المفاتيح والقيم (Keys & Values)

كل **Key** يشبه مجلد في نظام الملفات، ويمكن أن يحتوي على:

* **Subkeys** (مفاتيح فرعية)
* **Values** (قيم بيانات)

🔹 **أنواع القيم المهمة:**

* **REG\_BINARY:** قيم ثنائية (0 أو 1) — تُستخدم لتفعيل/تعطيل شيء.
* **REG\_DWORD/QWORD:** أرقام 32 أو 64 بت (تخزين إعدادات رقمية).
* **REG\_SZ:** نصوص (Strings) بتشفير Unicode — تُستخدم لمسارات أو أسماء.

مثال على المسار في السجلّ:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\USBSTOR

#### أداة التحرير (Registry Editor – regedit)

* يمكن فتح الأداة بكتابة:  
   🔹 regedit.exe من قائمة Start.
* فقط **المستخدم الإداري (Administrator)** يستطيع تعديل السجلّ.
* التصفح يتم بطريقة مشابهة لـ File Explorer، حيث يُعرض الهرم على اليسار والقيم في اليمين.

#### 

#### المخاطر الأمنية والتحليل الجنائي

* السجلّ يمكن أن يكون **نقطة استغلال رئيسية**:  
  + بعض البرمجيات الخبيثة تضيف نفسها في المفاتيح التي تعمل عند الإقلاع (مثل مفتاح Run) لتعمل تلقائيًا بعد إعادة التشغيل.
  + لذلك، مراجعة المفاتيح مثل

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run

مهمة جدًا لاكتشاف البرامج غير المرغوبة.

* من الناحية الجنائية (Forensics):  
  + يمكن استخدام السجلّ لتتبع نشاط المستخدمين والأجهزة.
  + مثال:

HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR

يحتوي على سجل لجميع أجهزة USB التي تم توصيلها بالنظام، مع بيانات البائع والرقم التسلسلي.

#### 

#### تطبيق عملي (Lab Suggestion)

**Lab: فحص سجلّ ويندوز للعثور على آثار أمان أو نشاط مشبوه**

1. افتح أداة **Registry Editor** كمسؤول (regedit.exe).
2. انتقل إلى المسار:

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run

1. راقب البرامج التي تبدأ تلقائيًا مع النظام — تأكد من عدم وجود تطبيقات غريبة.
2. انتقل إلى:

HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR

* + استعرض قائمة أجهزة USB التي تم توصيلها بالنظام.
  + لاحظ **VendorID** و **SerialNumber**.

1. دوّن أي إدخالات مشبوهة لمزيد من الفحص.

#### نصائح أمنية مهمة

* لا تُعدّل السجلّ إلا عند الضرورة وبعلم ما تفعله.
* احذر من السكربتات أو الملفات التي تُعدّل السجلّ تلقائيًا.
* في مجال الأمن السيبراني، فهم السجلّ يُساعد في:  
  + تحليل البرمجيات الخبيثة.
  + إجراء التحقيقات الجنائية الرقمية (Digital Forensics).
  + إصلاح الأنظمة المصابة.

### Windows Management Instrumentation

الـ (WMI (Windows Management Instrumentation هي بنية تحتية (Framework) من مايكروسوفت تُستخدم لإدارة الأنظمة والتطبيقات والأجهزة في بيئة Windows — سواء كانت محلية أو عن بُعد.

تسمح للمسؤولين بقراءة بيانات النظام وتنفيذ أوامر إدارية من خلال سكربتات أو أدوات مثل PowerShell.

#### ما الذي يمكن فعله باستخدام WMI

* جمع معلومات عن النظام مثل:
  + رقم الجهاز التسلسلي، نوع الـ BIOS، الشركة المصنعة.
  + حالة مكونات العتاد (Hardware).
* تنفيذ أوامر عن بُعد مثل ping أو مراقبة حالة جهاز آخر على الشبكة.
* الحصول على معلومات عن بطاقات الشبكة (Network Adapters).
* تعديل خصائص النظام أو إعدادات الأمان.
* مراقبة سجلات الدخول والخروج للمستخدمين (Login Events) في الشبكة.

#### WMI والأمن السيبراني

✅ الاستخدام الشرعي:

* أداة قوية جدًا للمراقبة والإدارة المركزية في الشبكات الكبيرة.
* تُستخدم في منتجات الأمان مثل Cisco Discovery Agent (CDA) لاكتشاف من قام بتسجيل الدخول على جهاز معين، مما يساعد في التحكم بالهويات (Identity-Based Firewalling).

⚠️ الاستخدام الخبيث:

* المهاجمون يستخدمون WMI لتنفيذ أوامر عن بُعد بصمت تقريبًا، مثل:
  + تعديل السجلّ (Registry).
  + الوصول إلى سجلات الأحداث (Event Logs).
  + تشغيل سكربتات أو برمجيات خبيثة بدون إنشاء ملفات جديدة على النظام.
* ميزة WMI في كونها "مخفية" نسبيًا تجعلها وسيلة شائعة في الهجمات المستمرة (Advanced Persistent Threats - APTs).

#### الأمان والتحكم بالوصول (Access Control)

* WMI يمتلك طبقة أمان مبنية فوق أمان نظام ويندوز نفسه.
* فقط حساب المسؤول (Administrator) يملك صلاحيات كاملة افتراضيًا.
* يمكن منح صلاحيات إضافية لمستخدمين محددين عبر:
  + إضافتهم إلى مجموعة Administrators.
  + أو ضبط أذوناتهم مباشرة في إعدادات WMI

#### النقاط التي يجب أن يتقنها محلل الأمن السيبراني

* معرفة كيفية جمع وتحليل البيانات عبر WMI.
* فهم كيفية استغلال المهاجمين لها لتجنب أو كشف الهجمات.
* تطبيق ضوابط أمان وصلاحيات دقيقة لمنع الوصول غير المصرح به.
* القدرة على تتبّع الأوامر الصادرة عبر WMI أثناء التحقيق الجنائي (Forensic Investigation).

### وظائف خوادم ويندوز الشائعة (Common Windows Server Functions)

يُستخدم Windows Server بشكل واسع في المؤسسات لتقديم خدمات الشبكة وإدارة البنية التحتية. فهم وظائفه ضروري لأي مختص في الأمن السيبراني لأن كل خدمة فيه تمثّل سطح هجوم محتمل (Attack Surface) يجب تأمينه ومراقبته.

#### أهم الوظائف والخدمات في Windows Server

1. (DHCP (Dynamic Host Configuration Protocol
   * يقوم بتوزيع عناوين IP للأجهزة في الشبكة تلقائياً.
   * يدير الشبكات الفرعية (Subnets) من خلال ما يسمى بـ Scopes.
   * يمكن إعداد سياسات (Policies) وحجوزات (Reservations).

🔒 من منظور أمني:

* + يجب حماية خادم DHCP لأنه يمكن أن يُستغل لتوزيع عناوين IP مزيفة (DHCP spoofing).
  + مراقبة DHCP logs ضرورية لتتبع الأجهزة المتصلة.

1. (DNS (Domain Name System
   * يحوّل الأسماء مثل example.com إلى عناوين IP.
   * يحتوي على مناطق (Zones) أمامية وخلفية، وسجلات مثل A, NS, CNAME.

🔒 من منظور أمني:

* + استهداف DNS ممكن من خلال هجمات مثل DNS poisoning أو DNS tunneling.
  + يُنصح بمراقبة استجابات DNS غير الطبيعية.

1. (IIS (Internet Information Services
   * خادم ويب داخلي في Windows لتشغيل مواقع الويب وواجهات الـ APIs وFTP.
   * يدعم ASP.NET وFTP وغيرها من التطبيقات.

🔒 من منظور أمني:

* + من أكثر الخدمات تعرضاً للهجمات مثل SQL Injection، Directory Traversal، أو Privilege Escalation.
  + يجب تعطيل الميزات غير الضرورية وتحديث الخادم باستمرار.

1. (Active Directory (AD
   * يدير المستخدمين، المجموعات، والأجهزة داخل النطاق (Domain).
   * يُستخدم للمصادقة (Authentication) والتفويض (Authorization).

🔒 من منظور أمني:

* + أكثر الأهداف حساسية في الشبكة.
  + اختراقه يمنح المهاجم صلاحيات واسعة (Domain Admin).
  + يجب تفعيل المراقبة، التشفير، ومبدأ Least Privilege.

#### رؤية عامة لمختص الأمن السيبراني

عليك كخبير أمن سيبراني أن:

* تعرف ما الخدمات المفعّلة على كل خادم في الشبكة.
* تقيّم المخاطر المحتملة لكل خدمة (DHCP, DNS, IIS, AD).
* تراقب السجلات Logs بشكل دوري.
* تفصل الأدوار الحساسة على خوادم مختلفة لتقليل المخاطر (Role Separation).

### أدوات الطرف الثالث الشائعة (Common Third-Party Tools)

يُركّز هذا الدرس على أدوات مجانية ومفتوحة المصدر تُستخدم في **إدارة أنظمة ويندوز ومراقبتها**، وغالباً ما يعتمد عليها مختصو الأمن السيبراني في التحليل والاستكشاف لأنها توفر قدرات تفوق الأدوات المدمجة في النظام.

#### ⚙️ 1. PuTTY

* **الوظيفة:** عميل SSH يُستخدم للاتصال بالخوادم عن بُعد بطريقة آمنة ومشفرة.
* **أهمية أمنيّة:**
  + يُستخدم للاتصال الآمن بخوادم لينكس أو أجهزة الشبكة (مثل الراوتر والسويتش).
  + يدعم حفظ الجلسات لتسهيل إعادة الاتصال.
* **من منظور مختص أمن:**
  + أداة أساسية لاختبار الاتصال وتأمين قنوات الإدارة البعيدة.
  + يمكن مراقبة استخدام SSH للكشف عن أي وصول غير مصرح به.

#### ⚙️ 2. WinSCP

* **الوظيفة:** أداة لنقل الملفات بشكل آمن عبر بروتوكولات مثل SFTP وSCP وFTP.
* **أهمية أمنيّة:**
  + تُستخدم لتبادل الملفات بين الأجهزة المحلية والخوادم البعيدة بطريقة مشفّرة.
  + واجهتها ثنائية اللوحة (Local/Remote) تسهّل الفحص اليدوي للملفات أثناء التحقيق.
* **من منظور مختص أمن:**
  + مفيدة في **تحليل عمليات نقل البيانات المشبوهة** أو استعادة الملفات من خوادم بعد الاختراق.

#### ⚙️ 3. Wireshark

* **الوظيفة:** التقاط وتحليل حزم الشبكة (Packet Capture).
* **أهمية أمنيّة كبيرة جداً:**
  + تُستخدم لتحليل الهجمات الشبكية (مثل ARP spoofing, DNS poisoning, MITM).
  + تُمكّنك من **قراءة محتوى الحزم** وتحليل البروتوكولات المستخدمة.
  + يمكن فتح ملفات **pcap** في التحقيقات الرقمية (forensics).
* **من منظور مختص أمن:**
  + أداة أساسية في تحليل الحوادث الأمنية (Incident Analysis).
  + يجب استخدامها بحذر في بيئات الإنتاج لأنها تلتقط كل البيانات بما فيها الحساسة.

#### ⚙️ 4. Sysinternals Suite (من Microsoft)

مجموعة ضخمة (70+ أداة) لتشخيص وإدارة أنظمة ويندوز — تُعتبر **من أهم الأدوات لأي خبير أمن سيبراني**.

#### **🔹 أدوات مهمة داخل الحزمة:**

* **Process Explorer:** بديل متقدم لـ Task Manager
  + يُظهر العمليات الجارية وعلاقاتها (Parent/Child).
  + يسمح بفحص الملفات عبر **VirusTotal** مباشرة.
  + يُظهر النصوص (Strings) داخل الملفات، مما يساعد في كشف **عناوين C2 أو روابط خبيثة داخل البرمجيات**.
* **Process Monitor (ProcMon):**
  + يراقب نشاط النظام والملفات والسجل (Registry).
  + يُستخدم لتحديد ما تفعله عملية مشبوهة على النظام.
  + مهم جداً في تحليل **البرمجيات الخبيثة وسلوكها**.
* **TCPView:**
  + يعرض العمليات التي تجري اتصالات عبر الشبكة.
  + يُستخدم لكشف **الاتصالات غير المصرح بها أو الغريبة** (مثل تطبيق حاسبة يقوم باتصال خارجي!).
  + يمكن إنهاء العملية أو غلق الاتصال مباشرة.

## نظام لينكس Linux Operating System

### تاريخ وفوائد نظام لينكس (History and Benefits of Linux)

#### تاريخ لينكس (History)

* الأصل:
  + الجذور تعود إلى نظام UNIX الذي طوّرته مختبرات AT&T عام 1969–1970 بلغة C من أجل القدرة على النقل (portability) بين الأجهزة.
* التفرعات المهمة:
  + (BSD (Berkeley Software Distribution عام 1977 – نظام مبني على يونكس.
  + مشروع GNU عام 1987 – هدفه إنشاء نسخة حرة من يونكس.
* بداية لينكس:
  + عام 1991 بدأ Linus Torvalds مشروعه لإنشاء نظام شبيه بـ UNIX يمكن تشغيله على جهازه الشخصي.
  + كتب النواة (Kernel) باستخدام لغة C و GNU Compiler.
  + تم إصدار النظام تحت رخصة (GPL (GNU Public License التي تسمح بتوزيع وتعديل الكود بحرية طالما يبقى الكود مفتوح المصدر.

#### فوائد لينكس (Benefits)

1. مفتوح المصدر ومجاني (Free & Open Source)
   * لا توجد تكاليف ترخيص.
   * يمكن لأي شخص فحص الكود أو تعديله أو توزيعه.
   * معظم البرامج عليه مجانية أيضًا.
2. مجتمع دعم ضخم (Large Community Support)
   * موارد تعليمية كثيرة: منتديات، مدونات، دورات، توثيقات.
   * المساعدة متاحة بسهولة لمستخدمي ومطوري النظام.
3. الاستقرار والموثوقية (Stability & Reliability)
   * لينكس معروف بقدرته العالية على العمل لفترات طويلة دون إعادة تشغيل.
   * يستخدم في بيئات الشركات والخوادم بكثرة بسبب استقراره.
4. أهمية في الأمن السيبراني (Cybersecurity Relevance)
   * معظم أدوات الاختبار والاختراق والتحليل الجنائي تعمل على لينكس.
   * يُستخدم من قبل المدافعين والمهاجمين على حد سواء، لذلك من الضروري لمختص الأمن فهمه بعمق.

### معمارية لينُكس (Linux Architecture)

يُعد الـKernel (النواة) القلب الحقيقي لنظام لينُكس، فهو المسؤول عن إدارة كل شيء في النظام — من العمليات (Processes) والذاكرة إلى التعامل مع العتاد (Hardware).

لكن فوق النواة توجد طبقة المستخدم (User Space) التي تشمل التطبيقات، وواجهات الأوامر، والبيئة الرسومية، وكل ما يتعامل معه المستخدم مباشرة.

#### طبقات معمارية لينُكس الأساسية

1. User Space (فضاء المستخدم):
   * يحتوي على كل البرامج التي يشغلها المستخدم مثل التطبيقات الرسومية (GUI) أو الطرفية (Terminal).
   * يعمل في مساحة ذاكرة منفصلة عن النواة لحماية النظام من أخطاء المستخدم أو البرامج.
   * لا يمكنه التواصل مباشرة مع العتاد، بل يتواصل مع النواة عبر system calls وواجهات برمجية (APIs).
   * هذه العزلة (Isolation) تزيد من استقرار النظام وأمانه — ميزة مهمة في الأمن السيبراني.
2. Kernel Space (فضاء النواة):
   * هو الجزء المسؤول عن إدارة موارد النظام، ويتحكم في:
     + الوصول إلى المعالج (CPU).
     + إدارة الذاكرة (Memory Management).
     + التعامل مع التعريفات (Device Drivers).
     + الجدولة (Scheduling) بين العمليات.
   * يقوم بعملية تسمى Context Switching لفصل المهام بين العمليات المختلفة.
   * النواة تتعامل بشكل مباشر مع الأجهزة لكنها لا تسمح للمستخدمين بذلك مباشرة، ما يقلل من المخاطر الأمنية.
3. Hardware (العتاد):
   * يمثل طبقة الأجهزة المادية (CPU، الذاكرة، وحدات التخزين...).
   * يتواصل فقط مع النواة عبر التعريفات.

#### كيف تعمل الطبقات معًا

عندما يقوم المستخدم بعملية (مثل الطباعة أو الاتصال بالشبكة):

→ التطبيق يرسل الطلب من User Space

→ إلى Kernel Space

→ الذي بدوره يتعامل مع Hardware عبر drivers.

هذه البنية تضمن أن المستخدم لا يملك وصولًا مباشرًا للعتاد، ما يمنع التلاعب بالنظام ويحافظ على أمانه واستقراره.

### نظرة عامة على نظام الملفات في Linux

نظام الملفات في لينكس منظم بطريقة قياسية تقريبًا في جميع التوزيعات، مما يجعل التنقل بينها سهلًا بعد فهم البنية الأساسية.

كل شيء في لينكس يبدأ من الجذر (Root) الذي يُرمز له بالشرطة المائلة /.

أي ملف أو مجلد أو جهاز سيكون موجودًا تحت هذا الجذر.

#### أهم المجلدات في نظام الملفات

| المجلد | الوظيفة |
| --- | --- |
| / | الجذر الرئيسي لكل الملفات والمجلدات. |
| /tmp | لتخزين الملفات المؤقتة. يُمسح غالبًا عند إعادة التشغيل. يستخدمه النظام والتطبيقات كمساحة مؤقتة. |
| /home | يحتوي على المجلدات الخاصة بكل مستخدم. مثلًا  /home/ismail.  تُخزّن هنا ملفات المستخدم الشخصية والإعدادات الخاصة به. |
| /usr | يحتوي على برامج المستخدم (user-space) ومكتباتها. يشمل مجلدات فرعية مثل:  • /usr/local لتثبيت البرامج يدويًا بواسطة المدير.  • /usr/share لمشاركة الملفات (استخدام قديم).  • /usr/lib مكتبات تشغيل البرامج.  • /usr/bin و /usr/sbin للملفات التنفيذية (binaries). |
| /dev | يحتوي على ملفات تمثل الأجهزة (devices) مثل الأقراص والطابعات. على سبيل المثال  /dev/sda  يمثل قرص صلب، و  /dev/null  جهاز يُهمل البيانات المرسلة إليه. |
| /etc | ملفات إعداد النظام والخدمات (مثل كلمات المرور، مفاتيح SSH، إعدادات الشبكة). يجب تأمينه جيدًا. |
| /var | لتخزين بيانات التشغيل والسجلات (logs). أشهر المجلدات الفرعية:  • /var/log ملفات السجلات.  • /var/tmp بيانات مؤقتة أثناء التشغيل. |
| /bin و /sbin | ملفات النظام التنفيذية الأساسية (ضرورية للإقلاع وتشغيل النظام). |
| /lib | مكتبات النظام المطلوبة لتشغيل البرامج الموجودة في  /bin و/sbin. |

#### تراكيب خاصة داخل نظام الملفات

| **الرمز** | **المعنى** |
| --- | --- |
| **.** | يشير إلى المجلد الحالي. يستخدم لتشغيل ملف موجود بنفس المجلد مثلًا:  ./new-app. |
| **..** | يشير إلى المجلد الأب (الذي يحتوي المجلد الحالي). مثال:  cd ..  تعيدك خطوة للخلف. |
| **~** | اختصار للدليل الرئيسي للمستخدم (Home directory). مثال:  cd ~  يأخذك إلى  /home/username. |

لاحظة:

عند تشغيل أمر أو برنامج، يبحث لينكس في مجلدات محددة معرفة في متغير البيئة $PATH.

لعرضها استخدم:

| echo $PATH |
| --- |

لذلك لتشغيل ملف موجود في مجلدك الحالي، يجب أن تسبق اسمه بـ ./.

#### المراجع المطلقة (Fully Qualified) والمراجع النسبية (Relative Paths)

🔹 المراجع المطلقة

تبدأ دائمًا بالشرطة المائلة /.

تشير إلى موقع الملف من الجذر مباشرة.

مثال:

| cp /usr/local/snort/snort.sysconfig /etc/sysconfig/snort.sysconfig |
| --- |

تعمل من أي مكان في النظام، لكنها طويلة.

🔹 المراجع النسبية

لا تبدأ بـ /.

تشير إلى موقع الملف نسبةً إلى مكانك الحالي.

مثال:

| cp test.pcap ../caps |
| --- |

هنا .. تعني "ارجع مجلدًا واحدًا للأعلى"، ثم ادخل مجلد caps.

يمكن أيضًا استخدام ~ بدلاً من المسار الكامل إلى مجلدك الرئيسي:

| cp test.pcap ~/caps |
| --- |

#### تمثيل الأجهزة كملفات

في لينكس، الأجهزة تمثل كملفات داخل

/dev

أنواعها الأساسية:

| النوع | الوظيفة | المثال |
| --- | --- | --- |
| Block | أجهزة تخزن البيانات على شكل كتل (مثل الأقراص). | /dev/sda |
| Character | أجهزة تتعامل مع تدفقات بيانات. | /dev/null, الطابعات |
| Pipe (Named Pipe) | تمرر البيانات بين العمليات. |  |
| Socket | تُستخدم للتواصل بين العمليات أو عبر الشبكة. |  |

يمكن معرفة نوع كل ملف داخل /dev بالأمر:

| ls -la /dev |
| --- |

الحرف الأول من كل سطر يدل على النوع:

b = block device,

c = character device.

### الأوامر الرئيسية لإدارة الملفات Basic File System Navigation and Management Commands

1. التنقل داخل نظام الملفات (cd, pwd)

cd <directory> → الانتقال إلى مجلد محدد.

cd /home/<username>/Downloads → باستخدام **المسار الكامل (absolute path)**.

cd Downloads → باستخدام **مسار نسبي (relative path)**.

cd ~/Downloads → باستخدام الرمز ~ للوصول إلى المجلد الشخصي من أي مكان.

pwd → عرض المجلد الحالي (present working directory).

cd .. → العودة إلى المجلد الأب.

cd / → الانتقال إلى الجذر root.

2. إدارة الملفات (cp, mv)

cp <source> <destination> → نسخ الملفات

mv <source> <destination> → نقل أو إعادة تسمية الملفات

3. استخدام أوامر المساعدة

man <command> → عرض دليل الأوامر (manual).

تطبيق عملي (مهم)

pwd # معرفة موقعك الحالي

cd Downloads # الانتقال إلى مجلد التنزيلات

cd .. # العودة خطوة للخلف

cd / # الذهاب إلى الجذر

cd ~/Documents # الانتقال إلى المستندات من أي مكان

cp http.pcapng http2.pcapng # نسخ ملف

mv http2.pcapng new\_app/http.pcap # نقل الملف وإعادة تسميته

### خصائص الملفات والصلاحيات File Properties and Permissions

1. عرض خصائص الملفات

الأمر الأساسي هو:

| ls -l |
| --- |

ويعرض معلومات تفصيلية عن الملفات والمجلدات، منها:

* نوع الملف:
  + - ملف عادي
  + d مجلد (directory)
  + b أو c لأجهزة block/character
* الصلاحيات (Permissions): 9 خانات تمثل صلاحيات:
  + المالك (User)
  + المجموعة (Group)
  + الآخرون (Others)
* عدد الروابط (Links)
* المالك (Owner)
* المجموعة (Group)
* حجم الملف
* تاريخ آخر تعديل
* اسم الملف

2. الصلاحيات (Permission}s)

لكل ملف أو مجلد ثلاث أنواع من الصلاحيات:

* **(r (read:** قراءة الملف أو محتوى المجلد.
* **(w (write:** تعديل أو حذف المحتوى.
* **(x (execute:** تشغيل الملف (في حالة البرامج) أو الدخول إلى المجلد.

مثلاً في الإخراج التالي:

-rw-rw-r--

* أول 3 أحرف (rw-): صلاحيات **المالك** (قراءة وكتابة)
* ثاني 3 (rw-): صلاحيات **المجموعة**
* ثالث 3 (r--): صلاحيات **الآخرين**

في الأمن السيبراني، هذه الصلاحيات مهمة لتقييد وصول المستخدمين غير المصرح لهم إلى ملفات النظام الحساسة.

### تعديل خصائص الملفات Editing File Properties

1. الطريقة العددية (Absolute / Octal Mode):

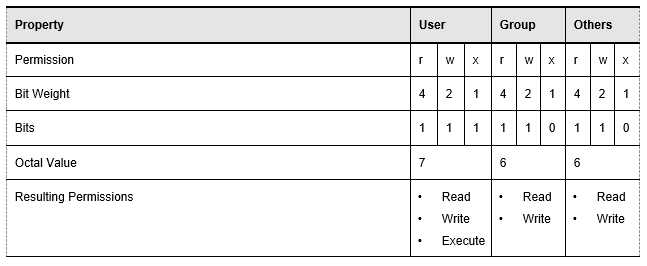
* تُستخدم الأرقام لتحديد الصلاحيات لكل من **المستخدم (user)** و **المجموعة (group)** و **الآخرين (others)**.
* كل نوع من الصلاحيات يُمثّل بقيمة ثنائية (bit value):
  + **r (read)** = 4
  + **w (write)** = 2
  + **x (execute)** = 1
* مجموع هذه القيم يحدد مستوى الصلاحية.
  + **7 = 4 + 2 + 1 → rwx**
  + **6 = 4 + 2 → rw-**
  + **5 = 4 + 1 → r-x**
  + **4 = 4 → r--**

الصيغة العامة:

chmod [user][group][others] filename

**مثال:**

chmod 766 myfile.txt

* النتيجة:
  + المستخدم: **rwx**
  + المجموعة: **rw-**
  + الآخرون: **rw-**

2. الطريقة الرمزية (Symbolic Mode):

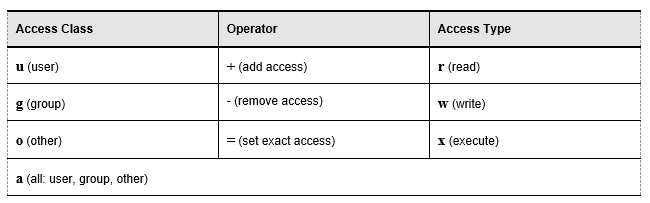
* **تُستخدم رموز لتحديد الفئة (class) ونوع الصلاحية (permission) والعملية (operator).**
* **الفئات (Classes):**
  + **u → user**
  + **g → group**
  + **o → others**
  + **a → all**
* **العمليات (Operators):**
  + **+ → إضافة صلاحية**
  + **- → إزالة صلاحية**
  + **= → تعيين صلاحيات محددة (يستبدل القيم السابقة)**
* **أنواع الصلاحيات (Permissions):**
  + **r → read**
  + **w → write**
  + **x → execute**

**أمثلة:**

**chmod u+r MyFile.txt # إضافة صلاحية القراءة للمستخدم فقط**

**chmod ug+rw MyFile.txt # إضافة القراءة والكتابة للمستخدم والمجموعة**

**chmod go+r-w MyFile.txt # السماح بالقراءة للمجموعة والآخرين وإزالة الكتابة**

**chmod a=r MyFile.txt # جعل كل المستخدمين لديهم صلاحية القراءة فقط**

### الجذر و sudo

في أنظمة Linux، المستخدم root (أو superuser) هو الحساب الذي يمتلك صلاحيات مطلقة على النظام — يمكنه الوصول والتعديل على أي ملف أو إعداد.

لكن بسبب خطورة هذا المستوى من الوصول (إمكانية التسبب بأضرار كبيرة أو سيطرة المهاجم الكاملة إذا تم اختراقه)، فإن العديد من توزيعات Linux الحديثة تقوم بتقييد استخدام حساب root أو حتى تعطيله افتراضيًا لأسباب أمنية.

بدلًا من تسجيل الدخول كـ root، يتم استخدام أمر sudo (اختصار لـ superuser do) الذي يسمح للمستخدم العادي بتنفيذ أوامر معينة بصلاحيات root بشكل مؤقت وآمن.

* يمكن تقييد الصلاحيات بحيث يُسمح لمستخدمين محددين بتنفيذ أوامر معينة فقط.
* يقوم sudo أيضًا بإنشاء سجل تدقيق (audit trail) يمكن من خلاله معرفة من نفذ أوامر إدارية ومتى.
* بعد إدخال كلمة المرور، يكون لدى المستخدم نافذة زمنية (عادة 5 دقائق) يمكنه خلالها تنفيذ أوامر أخرى دون إعادة إدخالها.

هذا النظام يُعتبر أفضل ممارسة أمنية (security best practice) لأنه يقلل مخاطر الاختراق ويُبقي النظام قابلًا للإدارة بأمان.

### الأقراص وأنظمة الملفات Disks and File Systems

في لينكس، كل شيء يبدأ من الجذر / ويُعتبر نظام الملفات (File System) هو الأساس. حتى الأقراص الفيزيائية (مثل الأقراص الصلبة وUSB) تُعرض على شكل ملفات داخل /dev وتُربط بنظام الملفات عبر ما يسمى **mount points**.

#### 2. الأقراص والتقسيمات (Partitions)

* كل قرص يُعرف عادةً باسم يبدأ بـ sd، مثل:
  + /dev/sda → أول قرص
  + /dev/sdb → الثاني
  + التقسيمات تُرقّم: /dev/sda1, /dev/sda2, إلخ.
* يوجد **جدول تقسيم (Partition Table)** على كل قرص يحدد أين تبدأ وتنتهي كل مساحة تخزين.

🧠 **أنواع جداول التقسيم:**

* **MBR (Master Boot Record):**
  + الأقدم والأكثر توافقًا.
  + يدعم حتى **4 تقسيمات فقط**.
  + أقصى حجم قرص: **2 تيرابايت**.
  + في حال تلف الـMBR، قد يصبح القرص غير قابل للقراءة.
* **GPT (GUID Partition Table):**
  + الأحدث ويُستخدم مع **UEFI**.
  + يدعم أقراص أكبر من 2 تيرابايت وعدد غير محدود تقريبًا من الأقسام.
  + يحتوي على **نسخ احتياطية** لجدول التقسيم في أماكن مختلفة على القرص، لزيادة الأمان.

📌 **أوامر إدارة الأقسام:**

* fdisk, gdisk, parted, gparted

⚠️ تُستخدم بحذر لأنها قد تمسح محتوى القرص بالكامل.

تطبيق عملي مهم:

🔍 عرض الأقسام باستخدام parted

sudo parted -l

يُظهر تفاصيل القرص مثل:

* نوع الجدول (MBR أو GPT)
* نوع نظام الملفات (ext4، swap، إلخ)
* حجم الأقسام

#### 3. أنواع أنظمة الملفات (File System Types)

* **ext2 / ext3 / ext4:** أنظمة لينكس الأصلية، وext4 هو الأحدث والأكثر استخدامًا.
* **NTFS:** خاص بويندوز، يدعم الصلاحيات المتقدمة والأقراص الكبيرة.
* **FAT16 / FAT32 / exFAT:** قديمة وبسيطة، لا تدعم الصلاحيات، تُستخدم غالبًا في الـUSBs.
* **ISO9660 / Joliet:** لأقراص CD وDVD.
* **HFS+:** خاص بنظام macOS.
* **Swap:** ليس نظام ملفات فعلي، بل مساحة يستخدمها النظام كذاكرة مؤقتة عندما تمتلئ الـRAM.

⚙️ **ملاحظة مهمة للأمن السيبراني:** معرفة أنواع الملفات وأنظمة الملفات مفيد عند تحليل الأدلة في **التحقيقات الجنائية الرقمية (Digital Forensics)**، حيث يمكن أن تختلف بنية البيانات وطرق الاسترجاع من نظام لآخر.

#### 4. تركيب الأقراص (Mounting Devices)

عند الإقلاع، يقوم لينكس بتركيب الأقراص تلقائيًا حسب إعدادات ملف:

/etc/fstab

كل سطر في هذا الملف يحتوي على:

* **UUID:** رقم فريد للقرص (يُستخدم بدل اسم /dev/sda لتجنب التغيير عند إعادة التشغيل).
* **Mount point:** المكان الذي يُربط فيه القرص (مثل /, /home, /media...).
* **File system type:** نوع نظام الملفات (مثل ext4).
* **Options:** إعدادات مثل errors=remount-ro لجعل القرص للقراءة فقط في حال وجود خطأ.
* **Order:** ترتيب التحميل (1 للجذر، 2 لبقية الأنظمة).

🔹 **أمر التركيب اليدوي:**

mount -t <type> <device> <mount\_point>

مثال:

mount -t ext4 /dev/sdb1 /media/usb

### تهيئة النظام في لينكس (System Initialization)

#### 1. مفهوم عام

عند تشغيل نظام لينكس، تتم عملية الإقلاع (Boot Process) التي تبدأ من الـBIOS أو UEFI وتنتهي ببدء العمليات الخاصة بالمستخدم.

الهدف من هذه العملية هو تحميل النواة (Kernel) وإعداد النظام ليصبح جاهزًا للعمل.

#### 2. مراحل الإقلاع الأساسية

* Hardware Checks: اختبار مكونات الجهاز (لوحة المفاتيح، الذاكرة، الأقراص...).
* Device Bus Discovery: التعرف على ناقلات الأجهزة مثل PCI وUSB.
* Device Discovery: اكتشاف الأجهزة المتصلة فعليًا.
* Kernel Initialization: تحميل وتشغيل نواة لينكس (Linux Kernel).
* Mount Root File System: تركيب نظام الملفات الجذري /.
* Start User Processes: بدء تشغيل العمليات والخدمات التي يستخدمها المستخدم.

#### 3. الـ Bootloader ودوره

* هو المكوّن المسؤول عن تحميل **النواة (kernel)** من القرص إلى الذاكرة.
* يتفاعل مع **BIOS أو UEFI** للوصول إلى الأجهزة في المراحل الأولى.
* أمثلة عليه: **GRUB** (الأكثر استخدامًا)، و **LILO** (قديم جدًا ونادر الاستخدام اليوم).

📁 **ملف الإعدادات المهم:**

/proc/cmdline → يحتوي على **وسائط الإقلاع (boot parameters)** مثل:

BOOT\_IMAGE=/boot/vmlinuz-4.4.0 root=UUID=xxxx ro quiet splash

* **BOOT\_IMAGE:** مسار ملف النواة.
* **root:** يحدد القرص الذي يحتوي على النظام (عادة عبر UUID).

#### 4. أنظمة إدارة الإقلاع (Boot Management Systems)

هناك طريقتان أساسيتان لإدارة بدء التشغيل:

🔹 System V init (الطريقة التقليدية)

* تعتمد على تشغيل سكربتات نصية داخل مجلدات **run levels**.
* ملف التكوين الرئيسي:

/etc/inittab.

* يبدأ العملية الأساسية init التي تتحكم في تشغيل أو إيقاف الخدمات.

| **رقم المستوى** | **الوظيفة الرئيسية** |
| --- | --- |
| 0 | إيقاف النظام (halt) |
| 1 | وضع الصيانة (Single-user mode) |
| 2 | وضع نصي بدون شبكة (نادر الاستخدام) |
| 3 | وضع نصي مع الشبكة (بدون واجهة رسومية) |
| 4 | غير مستخدم عادة (احتياطي) |
| 5 | تشغيل بواجهة رسومية (GUI) |
| 6 | إعادة التشغيل (reboot) |

* بطيئة نسبيًا لأن السكربتات تُنفّذ **تسلسليًا (واحدة تلو الأخرى)**.

🔹 systemd (الطريقة الحديثة والمستخدمة حاليًا)

* نظام حديث موجود في أغلب توزيعات لينكس الحالية.
* يشغل العمليات **بشكل متوازي (parallel)** → يجعل الإقلاع أسرع.
* يتحكم أيضًا في **تركيب أنظمة الملفات، والخدمات، والجلسات**.
* مجلد التعريف:

/etc/systemd/

* الأداة الأساسية: systemctl

#### 5. مستويات التشغيل (Run Levels)

هي أوضاع تشغيل تحدد الخدمات التي تُفعّل أثناء الإقلاع 👈.

توجد هذه المجلدات عادة في

/etc/rc

مثل

/etc/rc3.d, /etc/rc5.d.

* الملفات التي تبدأ بـ **S** = Start (تشغيل الخدمة).
* الملفات التي تبدأ بـ **K** = Kill (إيقاف الخدمة).
* الأرقام بعد الحرف تحدد ترتيب التنفيذ.

### الطرق البديلة لتشغيل نظام لينكس Emergency/Alternate Startup Options

#### 1. Single User Mode (وضع المستخدم الواحد)

يتم تشغيل النظام في runlevel 1 حيث تُشغّل فقط الخدمات الأساسية.

لا يتم تحميل الواجهة الرسومية أو الشبكة، ويتم الدخول إلى سطر الأوامر فقط.

يُستخدم عادةً في حل المشاكل أو إصلاح النظام.

⚠️ تحذير:

لا يُنصح باستخدام هذا الوضع في حالات التحليل الجنائي (forensics) لأن النظام يعتمد على الملفات الأصلية أثناء الإقلاع، مما يؤدي إلى تعديل الطوابع الزمنية (timestamps) وسجلات النظام (logs)، وبالتالي فقدان سلامة الأدلة.

#### 2. Live CD (الإقلاع من وسيط خارجي)

* يتم الإقلاع من وسيط خارجي (مثل USB أو DVD) يحتوي على نظام تشغيل جاهز.
* هذه الطريقة لا تستخدم الملفات الموجودة على القرص الداخلي، مما يجعلها مثالية في تحليل الحوادث الأمنية أو الفحص الجنائي.
* بعض التوزيعات مصممة خصيصاً للتحليل الأمني أو الإصلاح (مثل Kali Linux أو SystemRescue).
* يمكنك اختيار ما إذا كنت تريد إتاحة الكتابة على النظام أو إبقائه في وضع القراءة فقط (read-only) لحماية البيانات من التعديل.
* الوسائط الضوئية (CD/DVD) أكثر موثوقية لأنها غير قابلة للكتابة، بينما USB مفضل إذا أردت تعديل البيئة بسهولة.

#### تطبيق عملي مهم: الدخول إلى Single User Mode باستخدام GRUB

1. عند الإقلاع، اضغط على زر Shift للوصول إلى قائمة GRUB.
2. اختر Advanced Options.
3. اختر النواة (kernel) التي تود الإقلاع بها ثم حدد خيار (recovery mode).
4. من قائمة الاستعادة، اختر root للدخول إلى سطر الأوامر كـ superuser.
5. النظام سيكون في وضع read-only، ويمكنك تغييره إلى read/write عند الحاجة.
6. بعد الانتهاء، اكتب الأمر reboot لإعادة تشغيل النظام بشكل طبيعي.

### إيقاف تشغيل النظام Shutting Down the System‎

أهمية الإيقاف السليم

* الإيقاف غير الصحيح أو فقدان الطاقة المفاجئ يمكن أن يؤدي إلى **تلف الملفات (file corruption)** أو **خلل في النظام**.
* الهدف هو **إغلاق العمليات والملفات بشكل منظم وآمن** قبل إيقاف التشغيل.

**🔹 الأمر الرئيسي: shutdown**

الصيغة الأساسية:

shutdown [options] [time] [message]

أهم الخيارات:

| **الخيار** | **الوظيفة** |
| --- | --- |
| -h | إيقاف (halt) النظام تماماً |
| -r | إعادة تشغيل النظام (reboot) |
| -c | إلغاء عملية إيقاف أو إعادة تشغيل كانت مجدولة |
| -k | إرسال تحذير للمستخدمين دون تنفيذ الإيقاف |

الوقت:

* now → تنفيذ فوري
* +m → بعد عدد دقائق محدد
* hh:mm → في وقت محدد بصيغة 24 ساعة

أمثلة:

shutdown -h now # إيقاف فوري

shutdown -r now # إعادة تشغيل فورية

shutdown +30 "Rebooting soon" # بعد 30 دقيقة مع رسالة تحذير

shutdown -c # إلغاء عملية إيقاف مجدولة

🔹 أوامر أخرى مفيدة

* reboot → لإعادة تشغيل النظام مباشرة.
* systemctl poweroff → لإيقاف النظام في توزيعات تستخدم **systemd**.
* systemctl reboot → لإعادة التشغيل باستخدام **systemd**.

### عمليات النظام System Processes

#### ما هي العملية (Process)

* كل أمر يُنفَّذ في النظام يصبح **عملية process**.
* العملية تحتاج إلى **ذاكرة (Memory)** و**مدخلات/مخرجات (I/O)** و**موارد CPU**.
* لكل عملية رقم مميز اسمه **Process ID) PID)**.
* النظام (kernel) هو المسؤول عن إدارة كل العمليات.

#### ما هو الـ Thread

* **Thread** = خيط تنفيذي داخل العملية.
* يمكن أن يكون:
  + **Single-threaded** (خيط واحد فقط)
  + **Multi-threaded** (عدة خيوط تعمل بالتوازي)
* لكل Thread رقم مميز اسمه **Thread ID)TID)**.

#### إنشاء العمليات في Linux

يتم إنشاء العمليات عبر آليتين:

##### 1. ()fork

* تنشئ نسخة جديدة من العملية الأصلية.
* العملية الأصلية = **Parent**
* العملية الجديدة = **Child**
* الـ Parent يحصل على PID الابن، والـ Child يحصل على القيمة 0.

##### 2. ()exec

* تستبدل العملية الجديدة نفسها ببرنامج آخر (تنفذ أمر جديدًا).
* مثال: عندما تكتب أمرًا في الـ shell، يتم تنفيذ fork ثم exec.

أحيانًا الـ fork لا يتبعه exec — مثلًا في السيرفرات (daemons) التي تنشئ نسخًا متعددة للتعامل مع اتصالات مختلفة.

#### 

#### مراقبة العمليات (Monitoring)

من المهم مراقبة العمليات لمعرفة:

* الموارد التي تستهلكها
* عددها
* حالتها (نشطة، نائمة، متوقفة، إلخ)

#### أمر top

يعرض العمليات في الزمن الحقيقي (real-time).

الأقسام:

##### القسم العلوي

* **Uptime:** مدة تشغيل النظام
* **Load Average:** متوسط الحمل خلال آخر 1، 5، و15 دقيقة

| **العمود** | **المعنى** |
| --- | --- |
| PID | رقم العملية |
| USER | المستخدم الذي بدأ العملية |
| PR | أولوية الجدولة |
| NI | قيمة الـ nice (الأولوية القابلة للتعديل) |
| VIRT | الذاكرة الافتراضية |
| RES | الذاكرة الفعلية المستخدمة |
| SHR | الذاكرة المشتركة |
| S | حالة العملية (status) |
| %CPU | نسبة استخدام المعالج |
| %MEM | نسبة استخدام الذاكرة |
| TIME+ | الوقت الذي استهلكته العملية |
| COMMAND | اسم العملية |

* **CPU Usage:**
  + us = عمليات المستخدم
  + sy = عمليات النظام
  + id = الوقت الخامل
* **Memory/Swap usage**

##### القسم السفلي

يعرض العمليات الجارية في أعمدة 👈:

**رموز حالة العملية (S column) 👇**

| **الرمز** | **المعنى** |
| --- | --- |
| D | انتظار I/O (نوم غير قابل للمقاطعة) |
| R | قيد التشغيل |
| S | نائمة |
| T | متوقفة من kernel أو job control |
| t | متوقفة بسبب debugger |
| Z | Zombie (منتهية ولكن لم تُحذف بعد) |

#### Nice Value

* تحدد أولوية العملية.
* **الرقم الأعلى = أولوية أقل**.
* النطاق من **–20 (أعلى أولوية)** إلى **19 (أقل أولوية)**.
* مفيدة لتقليل تأثير عملية ثقيلة على النظام.

#### أمر ps

يُظهر العمليات مرة واحدة (ليس في الزمن الحقيقي مثل top).

الاستخدام الأساسي:

ps

يعرض العمليات الخاصة بالمستخدم الحالي.

خيارات مفيدة:

| **الخيار** | **الوظيفة** |
| --- | --- |
| -f | عرض مفصل (full format) |
| -e | عرض جميع العمليات لجميع المستخدمين |
| -ef | دمج الاثنين (كامل + كل العمليات) |
| aux | النمط BSD لنفس النتيجة |
| -C name | عرض العمليات باسم معين |
| ps -ef | grep ssh |

## 

#### أمر lsof

يعرض الملفات المفتوحة على النظام (بما في ذلك الأجهزة والاتصالات).

مثال:

sudo lsof /var/log/syslog

### **الأعمدة المهمة:**

| **العمود** | **المعنى** |
| --- | --- |
| COMMAND | اسم العملية |
| PID | رقم العملية |
| USER | المستخدم |
| FD | واصف الملف (File Descriptor) |
| TYPE | نوع الملف (REG, DIR, BLK, etc.) |
| DEVICE | رقم الجهاز |
| SIZE/OFF | حجم الملف أو الإزاحة |
| NODE | رقم العقدة (inode) |
| NAME | اسم الملف |

**ملاحظة**: تحتاج صلاحيات root لتشغيل lsof بشكل كامل.

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

### مفاهيم أساسية في سطر أوامر لينُكس (Linux Command Shell‎)

هذا الدرس عبارة عن تطبيق عملي في هذا الملف 👈 [📃](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### Piping Command Output & Running Multiple Commands

#### 1. Piping (التوجيه عبر الأنابيب |)

* **الفكرة:** تمرير مخرجات أمر (output) إلى أمر آخر لمعالجتها بدلًا من عرضها فقط.
* **الرمز المستخدم:** | (pipe character).
* **الاستخدام الشائع:** تصفية النتائج الطويلة أو المعقدة باستخدام أوامر مثل grep.

**مثال عملي:**

ps -ef | grep ssh

* 🔹 يعرض جميع العمليات التي تحتوي على كلمة "ssh".  
   🔹 يتم تمرير ناتج أمر ps -ef إلى grep ليتم تصفيته.

يمكن توجيه المخرجات إلى أكثر من أمر متتالي:

ps -ef | grep ssh | sort -d

* 🔹 هنا يتم تصفية النتائج أولًا بـ grep، ثم ترتيبها أبجديًا باستخدام sort.

#### 2. Running Multiple Commands (تشغيل أوامر متعددة في سطر واحد)

* **الرمز المستخدم:** &&
* **الفكرة:** تنفيذ عدة أوامر بالتسلسل، بحيث لا يُنفذ الأمر التالي إلا إذا نجح الأمر السابق.

**مثال شائع عند ترجمة البرامج من المصدر:**

./configure && make && make install

🔹 يتم تنفيذ make فقط إذا نجح

./configure.

🔹 يتم تنفيذ

make install

فقط إذا نجح make.

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### أدوات سطر أوامر أخرى مفيدة Other Useful Command-Line Tools

هذا الدرس عبارة عن تطبيق عملي في هذا الملف 👈 [📃](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### Secure Shell (SSH) Protocol

يُستخدم **SSH (Secure Shell)** كأداة أساسية للوصول الآمن إلى الأنظمة عن بُعد عبر الشبكات، وهو بديل حديث وآمن لبروتوكول **Telnet** الذي ينقل البيانات وكلمات المرور بنص واضح (Cleartext).

#### المفاهيم الأساسية

* **SSH** يوفر اتصالًا مشفّرًا بين **العميل (client)** و**الخادم (server)**، مما يمنع التنصت أو اعتراض البيانات أثناء الاتصال.
* يستخدم **التشفير بالمفتاح العام (Public Key Cryptography)**:
  + الخادم يعرّف نفسه للعميل بمفتاحه العام.
  + يمكن للمستخدمين أيضًا إنشاء **زوج من المفاتيح (عام/خاص)** للدخول إلى الخوادم دون كلمة مرور، بشرط أن يكون المفتاح العام مسجلًا في الخادم.
* الملف

**~/.ssh/known\_hosts**

يخزن المفاتيح العامة للخوادم التي تم الاتصال بها سابقًا، لمنع انتحال الهوية (man-in-the-middle attack).

#### التحقق من الهوية والأمان

* عند الاتصال لأول مرة بخادم جديد، سيطلب SSH تأكيد الثقة بالمفتاح العام (fingerprint).
* يجب على المستخدم التحقق من هذا المفتاح يدويًا (عبر مدير النظام أو قناة موثوقة) لتجنب هجمات MITM.
* في حال تغيّر الخادم أو تم استبداله، سيظهر تحذير **"REMOTE HOST IDENTIFICATION HAS CHANGED"**.

يمكن حذف المفتاح القديم بالأمر:

ssh-keygen -f "~/.ssh/known\_hosts" -R <IP address>

* + ثم إعادة الاتصال لتسجيل المفتاح الجديد.

### **الاستخدام الأساسي**

**الاتصال بخادم:** ssh <username>@<remote\_host>

* ثم إدخال كلمة المرور الخاصة بالمستخدم في الخادم.
* بعد تسجيل الدخول، يمكنك تنفيذ الأوامر وكأنك تعمل مباشرة على الجهاز البعيد.

**إنهاء الجلسة:**

exit

#### نسخ الملفات بأمان باستخدام SCP

* **(SCP (Secure Copy Protocol** يستخدم SSH لتأمين نقل الملفات بين الأنظمة.

**من الجهاز المحلي إلى خادم بعيد:**

scp <source\_path> <user>@<remote\_host>:<destination\_path>

**من الخادم البعيد إلى الجهاز المحلي:**

scp <user>@<remote\_host>:<remote\_file\_path> <local\_destination>

* أول مرة يتصل فيها المستخدم بخادم جديد، سيُطلب منه تأكيد المفتاح العام كما في SSH.
* النقل يكون مشفرًا بالكامل، مع عرض نسبة التقدم والسرعة.

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://docs.google.com/document/d/10bCHrhaM0tLbC5PLVYWFyOmWuGSJVBdHDortokQ_n-Y/edit?usp=sharing)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### إدارة الشبكات في لينكس Networking

**لينُكس** يحتوي على نظام قوي جدًا لإدارة الشبكات تم تطويره منذ البداية لدعم الاتصال وإعدادات الشبكة.

يمكن إدارة إعدادات الشبكة من خلال **الواجهة الرسومية (GUI)** أو من **سطر الأوامر (CLI)**.

* التوصية: استخدم الواجهة الرسومية إن كانت متاحة، لأن التعديلات من سطر الأوامر قد تُستبدل تلقائيًا عند إعادة التشغيل.
* على الخوادم (server distributions) غالبًا لا توجد واجهة رسومية، لذا يتم الاعتماد على الأوامر.

#### عرض إعدادات الشبكة

الأمر الأساسي:

ifconfig -a

* يعرض جميع الواجهات (حتى غير المفعّلة) مع تفاصيل مثل:
  + عنوان MAC
  + عنوان IP (IPv4 و IPv6)
  + عنوان البث (Broadcast)
  + قناع الشبكة (Subnet Mask)
  + حالة الواجهة (UP/DOWN) وإحصاءات النقل والاستقبال

لتحديد واجهة معينة فقط:

ifconfig eth0

💡 **ملاحظة حديثة:** الأمر **ifconfig** قديم، والأداة الحديثة البديلة هي **ip** (من حزمة iproute2):

ip addr show

ip link show

وهي أكثر قوة وشمولاً من ifconfig.

#### إعداد خصائص الشبكة يدويًا

لتعيين عنوان IP يدويًا:

sudo ifconfig eth0 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255

* التغييرات تحتاج صلاحيات الجذر (root/sudo).
* هذه الإعدادات مؤقتة وتفقد بعد إعادة التشغيل إلا إذا تم حفظها في ملفات الإعدادات أو أدوات الإدارة الدائمة.

#### إعداد المسارات (Routing)

عرض جدول التوجيه:

netstat -rn

* يظهر المسارات المحلية (مثل الشبكة المحلية) والمسار الافتراضي (default gateway).

إضافة بوابة افتراضية (gateway):

sudo route add default gw 192.168.1.1

إضافة مسار ثابت (static route):

sudo route add -net 192.168.133.0 netmask 255.255.255.0 gw 192.168.1.200

حذف مسار:

sudo route del -net 192.168.133.0 netmask 255.255.255.0 gw 192.168.1.200

⚠️ كما في ifconfig، هناك بديل أحدث لأوامر التوجيه هو:

ip route show

ip route add default via 192.168.1.1

#### اختبار الاتصال (Connectivity Testing)

لاختبار الاتصال بالشبكة:

ping 192.168.1.1

* يستخدم بروتوكول **ICMP** للتحقق من الوصول إلى المضيف.  
   يمكن أيضًا تجربة أسماء نطاقات (مثل ping google.com) لاختبار عمل DNS.

ملاحظة: بعض الأنظمة قد تحظر ICMP، لذا فشل الـ ping لا يعني دائمًا أن النظام غير متصل.

### Managing Services in SysV Environments & Systemd

**الخدمة (Service):** برنامج يعمل في الخلفية (background) مثل خادم FTP أو SSH.

عند تعديل إعدادات الخدمة، يجب **إعادة تشغيلها (bounce)** لتطبيق التغييرات، أي "إيقافها ثم تشغيلها مجددًا".

#### إدارة الخدمات في النظام القديم (SysV Init)

الأوامر كانت تعتمد على سكربتات موجودة في:

/etc/init.d/

الصيغة كانت كالتالي:

sudo service <service-name> start|stop|restart

مثال عملي لإدارة خدمة **vsftpd (خادم FTP)**:

ps -ef | grep ftp # التحقق من عمل الخدمة

sudo service vsftpd stop

sudo service vsftpd start

يمكن أيضًا تشغيل السكربت مباشرةً:

cd /etc/init.d

sudo ./cups restart

⚠️ هذه الطريقة قديمة، وما زالت موجودة فقط **للتوافق الخلفي (backward compatibility)**.

#### إدارة الخدمات في النظام الحديث (Systemd)

* **systemd** هو النظام الافتراضي حالياً في معظم توزيعات Linux (مثل Ubuntu وCentOS).
* يستخدم مفهوم **الوحدات (Units)** لإدارة المكونات، مثل الخدمات (service units).

ملفات الوحدات عادةً موجودة في:

/usr/lib/systemd/system/

* مثال:

Vsftpd.service

#### 

#### أوامر systemctl الأساسية

الصيغة العامة:

sudo systemctl <command> <service-name>

أمثلة:

sudo systemctl start vsftpd # تشغيل الخدمة

sudo systemctl stop vsftpd # إيقاف الخدمة

sudo systemctl restart vsftpd # إعادة تشغيل الخدمة

sudo systemctl reload vsftpd # إعادة تحميل الإعدادات دون إيقاف الخدمة

sudo systemctl daemon-reload # إعادة تحميل ملفات systemd نفسها

#### الفرق بين SysV و systemd

| **الجانب** | **SysV Init** | **systemd** |
| --- | --- | --- |
| نوع الإدارة | سكربتات Bash في /etc/init.d | وحدات (Units) في /usr/lib/systemd/system |
| الأداة المستخدمة | service | systemctl |
| دعم الإصدارات الحديثة | قديم – مستخدم للتوافق فقط | النظام القياسي في أغلب التوزيعات |
| إمكانية إعادة تحميل الإعداد دون توقف الخدمة | ❌ لا | ✅ نعم (reload) |

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### رؤية خدمات الشبكة التي تعمل Viewing Running Network Services

* **الخدمات الشبكية (Network Services)** هي **عمليات (daemons)** تعمل في الخلفية وتستمع على منافذ معينة (ports) في الجهاز لتلقي الاتصالات — مثل FTP, HTTP, SSH.
* كخبير أمني، من المهم معرفة **ما الخدمات النشطة** و**على أي المنافذ تعمل**، لتحديد أي نشاط مشبوه أو غير ضروري.

أداة netstat (مهمة جدًا)

* **الغرض:** عرض جميع الاتصالات والخدمات الشبكية الجارية.
* **يُفضل تشغيلها بـ sudo** للحصول على كل التفاصيل.

| **الأمر** | **الوظيفة** |
| --- | --- |
| sudo netstat -a46 | يعرض جميع الاتصالات في أي حالة (IPv4 و IPv6). |
| sudo netstat -lt | يعرض فقط الخدمات التي **تستمع (LISTEN)** عبر **TCP**. |
| sudo netstat -lun | يعرض الخدمات التي **تستمع (LISTEN)** عبر **UDP** مع عرض العناوين بالأرقام. |
| sudo netstat -atnp | يعرض جميع الاتصالات **(TCP, أي حالة)** مع **رقم العملية (PID)** واسم البرنامج. |

**أعمدة الإخراج المهمة:**

* **Proto:** البروتوكول (TCP/UDP).
* **Local Address:** عنوان الـ IP والمنفذ المحلي.
* **Foreign Address:** الطرف الآخر المتصل.
* **State:** حالة الاتصال (LISTEN, ESTABLISHED, CLOSE\_WAIT…).
* **PID/Program Name:** رقم العملية والبرنامج المرتبط بالمنفذ.

💡 **نصيحة أمنية:** أي منفذ في حالة **LISTEN** ولم تكن تعرف خدمته = **احتمال خطر أمني** (قد يكون backdoor أو خدمة غير ضرورية).

أداة (lsof (List Open Files

* **الغرض:** عرض العمليات المفتوحة بما في ذلك **الاتصالات الشبكية النشطة**.
* أيضًا مفيدة في **تتبع البرامج التي تستخدم منفذًا محددًا** أو **تفتح اتصالًا مشبوهًا**.

| الأمر | الوظيفة |
| --- | --- |
| sudo lsof -i | يعرض جميع الاتصالات الشبكية المفتوحة. |
| sudo lsof -i tcp | يعرض فقط اتصالات TCP. |
| sudo lsof -i tcp:80 | يعرض العمليات التي تستخدم المنفذ 80 (HTTP). |
| sudo lsof -i udp:53 -P | يعرض عمليات UDP/53 (DNS) بالأرقام بدل الأسماء. |
| sudo lsof -i @192.168.222.1 | يعرض العمليات المتصلة بعنوان IP معين. |
| sudo lsof -i @192.168.222.1:21 -P | يعرض العمليات المتصلة بعنوان ومنفذ محدد (مثل FTP/21). |

**أعمدة الإخراج الرئيسية:**

* **COMMAND:** اسم البرنامج أو العملية.
* **PID:** رقم العملية.
* **USER:** المستخدم الذي شغّل العملية.
* **TYPE:** نوع الاتصال (TCP/UDP).
* **NAME:** الطرفين المتصلين مع المنفذ وحالة الاتصال (مثل ESTABLISHED).

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### (Name Resolution (DNS في Linux

في أنظمة Linux، **نظام DNS** هو المسؤول عن تحويل أسماء النطاقات (مثل example.com) إلى **عناوين IP** يمكن للنظام التواصل من خلالها. فهم آلية عمله وتكوينه أمر ضروري، خصوصًا في مجال **الأمن السيبراني**، لأن أي تلاعب فيه قد يؤدي إلى **هجمات تصيد أو إعادة توجيه ضارة**.

#### الملفات الأساسية في تهيئة DNS

1. **/etc/resolv.conf**
   * يحتوي على إعدادات DNS **الرئيسية**.
   * أهم التوجيهات:
     + search: يحدد نطاقات يتم إضافتها تلقائيًا عند حل الأسماء القصيرة.
     + nameserver: يحدد عنوان خادم DNS المستخدم.

مثال:

search example.com local.lan

nameserver 172.16.1.254

nameserver 172.16.2.254

* + من منظور أمني:
    - احذر من وجود خوادم DNS عامة (مثل 8.8.8.8 أو 4.2.2.1) غير مصرح بها في بيئة العمل.
    - أجهزة المستخدمين المتنقلة قد تُظهر إعدادات مؤقتة من شبكات سابقة.

1. **/etc/hosts**
   * آلية قديمة لحل الأسماء، ما زالت مستخدمة محليًا.
   * تحتوي على تعيينات يدوية بين **عنوان IP** و**اسم مضيف**.

مثال:

127.0.0.1 localhost

::1 localhost

192.168.1.50 server1

* + ⚠️ **مخاطر أمنية:**
    - يمكن للبرمجيات الخبيثة تعديل هذا الملف لتوجيه المستخدم إلى موقع مزيف (مثل موقع بنك مزيف لجمع كلمات المرور).
    - لذلك، يجب مراقبة أي تغييرات على هذا الملف باستخدام أدوات كشف التلاعب (مثل FIM).

1. **/etc/nsswitch.conf**
   * يحدد **ترتيب مصادر المعلومات** التي يعتمد عليها النظام عند حل الأسماء.

الجزء المهم في سياق DNS هو السطر:

hosts: files dns

* + هذا يعني أن النظام يحاول أولًا الحل عبر ملف /etc/hosts، ثم عبر DNS.
  + 🧠 **تأثير أمني:**
    - إذا تمكن المهاجم من تغيير هذا الترتيب أو محتوى ملف /etc/hosts، يمكنه تجاوز الـ DNS الشرعي.
    - مثال: برمجية خبيثة تغيّر ترتيب البحث لتجعل النظام يستخدم /etc/hosts فقط، مما يؤدي إلى إعادة توجيه ضارة.

### اختبار محول الأسماء Testing Name Resolution

هذا الدرس عبارة عن تطبيق عملي في هذا الملف 👈 [📃](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### مشاهدة تحركات الشبكة Viewing Network Traffic **tcpdump** في لينكس، هي أداة قوية لتحليل الشبكة تُستخدم لالتقاط حزم البيانات (packets) المارة عبر واجهة الشبكة. تُعد tcpdump أداة أساسية في الأمن السيبراني لاكتشاف وتحليل الهجمات أو التحقيق في الأنشطة المشبوهة على الشبكة.

الأمر الأساسي لتشغيل الأداة:

sudo tcpdump <options> <filters>

* **options**: تحدد كيفية التقاط البيانات (مثل الواجهة أو حجم الحزمة).
* **filters**: تحدد نوع أو وجهة أو مصدر الحزم المطلوب التقاطها.
* غياب الفلاتر يعني التقاط كل حركة المرور على الواجهة المحددة.، تطبيق عملي في هذا الملف 👈 [📃](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

#### أهم الخيارات (Options)

| **الخيار** | **الوظيفة** |
| --- | --- |
| -i <interface> | تحديد واجهة الشبكة (مثل ens33). |
| -X | عرض البيانات (payload) بالهيكس والعادي (ASCII). |
| -nn | عرض العناوين والمنافذ بالأرقام دون ترجمة DNS. |
| -s <length> | تحديد حجم البايتات الملتقطة (0 = كل الحزمة). |
| host <IP> | التقاط الحزم المرسلة من/إلى عنوان محدد. |
| and port <number> | تقييد الالتقاط بمنفذ محدد (مثلاً 80 أو 21). |
| not port <number> | استثناء منفذ معين (مثلاً تجاهل SSH على 22). |
| -w <file.pcap> | حفظ الحزم في ملف بصيغة PCAP لعرضها لاحقاً بـ Wireshark. |
| -r <file.pcap> | قراءة وعرض ملف PCAP محفوظ مسبقاً. |

#### مخرجات tcpdump

* كل سطر يبدأ بوقت الالتقاط (timestamp).
* يظهر عنوان IP المصدر والوجهة والمنفذ.
* بعد الرأس (header) تظهر البيانات الداخلية (payload) بالهيكس والنص.
* عند إنهاء الالتقاط بـ **Ctrl + C** تظهر إحصائيات عن عدد الحزم الملتقطة والمفقودة.

### Configuring Remote Syslog

يشرح هذا الدرس كيفية إعداد **النظام لتصدير سجلات syslog إلى خادم بعيد (Remote Syslog Server)** — وهي خطوة أساسية في بيئات **SIEM (Security Information and Event Management)** لجمع وتحليل السجلات مركزياً من أجهزة وشبكات متعددة.

الفكرة هي أن **Linux يمكنه إرسال السجلات (logs)** الخاصة به إلى خادم مركزي باستخدام **بروتوكولات UDP أو TCP** عبر إعدادات بسيطة داخل ملفات rsyslog.

يمكن إرسال جميع الرسائل إلى الخادم البعيد عبر UDP باستخدام:

\*.\* @<remote\_host>:<port>

أو عبر TCP باستخدام:

\*.\* @@<remote\_host>:<port>

الافتراضي هو المنفذ **514** لكل من UDP وTCP إن لم يتم تحديده.

يمكن تحديد نوع الرسائل المرسلة عبر تحديد **facility** و**severity** مثل:

\*.emerg @192.168.222.1:10514

(يرسل فقط الرسائل من مستوى **emergency** إلى الخادم).

بعد أي تعديل في ملف إعدادات rsyslog يجب **إعادة تشغيل الخدمة** لتطبيق التغييرات.

**ملاحظة: هذا الدرس يحتوي على تطبيق عملي التدريب في هذا الملف 👈** [**📃**](https://drive.google.com/file/d/1dL_L5_Wf5OreJgWfZ6Fnj9kWhTA15htQ/view?usp=drive_link)

**ملاحظة:** قم بتنزيل الملف على جهازك الخاص وقم بتشغيله لأن الملف عبارة عن ملف .html

### تشغيل برمجيات على لينكس Running Software on Linux

#### 🧩 أولاً: المتطلبات لتشغيل برنامج في لينكس

لكي يعمل أي ملف في لينكس يجب:

1. أن يكون **قابلاً للتنفيذ** (structured to run directly أو عبر interpreter).
2. أن تكون لديه **صلاحية التنفيذ (execute bit)** في أذونات الملف.

#### ⚙️ ثانياً: فكرة التحويل من المصدر إلى برنامج

* الكود المصدري يُكتب بلغة مثل **C**.
* يتم تحويله إلى برنامج جاهز للتشغيل باستخدام **مُصرّف (compiler)** مثل:
  + **GCC (GNU Compiler Collection)** — وهو الأكثر استخداماً في لينكس.

🧱 أنواع الملفات أثناء عملية التحويل

1. **Source Files (.c)** → الكود المكتوب بلغة البرمجة.
2. **Object Files (.o)** → ناتجة من ترجمة أجزاء من الكود، يتم دمجها لاحقاً.
3. **Shared Object Files (.so)** → مكتبات ديناميكية تُستخدم أثناء التشغيل دون دمجها داخل البرنامج.
4. **Header Files (.h)** → تحتوي تعريفات ودوال تُستخدم أثناء الترجمة.

#### 💻 ثالثاً: الأوامر الأساسية في عملية الترجمة

الطريقة اليدوية لتجميع ملف بسيط:

gcc -o output\_file source\_file.c

* -o تحدد اسم الملف الناتج (البرنامج التنفيذي).
* source\_file.c هو ملف الكود المصدري.

### 

#### 🛠️ رابعاً: الأداة الأوتوماتيكية (Autotools)

عند التعامل مع برامج كبيرة، تُستخدم أدوات مثل **Autotools** لتسهيل العملية عبر سكربتات جاهزة:

1. ./configure  
   * يفحص النظام ويتأكد من وجود المتطلبات (dependencies).
   * يمكن تمرير خيارات لتفعيل ميزات معينة.
2. make  
   * يُشغّل المُصرّف **gcc** لإنشاء الملفات التنفيذية.
3. make install
   * ينسخ الملفات الناتجة إلى أماكنها في النظام (/bin, /usr/local/bin...).
   * عادة يُنفّذ بصلاحيات الجذر (superuser).

### **🧪 التطبيق العملي: تثبيت برنامج Snort**

tar zxvf src/snort-2.9.5.3.tar.gz # فك ضغط المصدر

cd snort-2.9.5.3 # الدخول إلى المجلد

./configure --enable-sourcefire # إعداد التهيئة

make # التجميع

make install # التثبيت النهائي

بعد هذه الخطوات يصبح البرنامج جاهزاً للتشغيل على النظام.

## تقنيات أمان الأجهزة الطرفية Endpoint Security Technologies

### جدار الحماية الشخصي Host-Based (Personal) Firewall

الـ **Host-Based Firewall** (أو **Personal Firewall**) هو برنامج يعمل على **جهاز واحد فقط** لحمايته من الاتصالات غير المصرح بها، بخلاف **الـ Network Firewall** التقليدي الذي يحمي الشبكة ككل.

يُعد هذا النوع ضروريًا خصوصًا للأجهزة **المتنقلة** (مثل الحواسيب المحمولة) التي تتصل أحيانًا خارج بيئة العمل الآمنة.

الفرق بين الجدار الناري التقليدي وHost-Based:

| **العنصر** | **الجدار الناري التقليدي** | **الجدار الناري الشخصي** |
| --- | --- | --- |
| الموقع | بين الشبكات (Gateway) | على الجهاز نفسه |
| الوظيفة | يراقب ويفرض سياسات مرور البيانات بين الشبكات | يراقب ويفرض سياسات مرور البيانات من وإلى الجهاز |
| الاستخدام | حماية الشبكات الداخلية | حماية الجهاز الفردي، خصوصًا عند اتصاله من أماكن عامة أو عبر VPN |

الأهمية في الأمن السيبراني:

* يحمي الجهاز عندما **يخرج من بيئة العمل** (مثلاً في المطار أو المقهى).
* عند استخدام **VPN مع Split Tunneling**، يحمي الاتصالات **الخارجة من النفق الآمن**.
* يمنع البرامج الضارة من فتح منافذ غير مصرح بها أو الاتصال بالإنترنت دون إذن المستخدم.

ميزات شائعة في الجدران النارية الشخصية:

1. **التحكم حسب البروتوكولات والمنافذ** (مثل TCP/UDP).
2. **التحكم حسب التطبيقات** (whitelist/blacklist).
3. **سياسات مختلفة حسب نوع الشبكة** (عمل / منزل / عامة).
4. **تنبيهات للمستخدم** عند محاولة تطبيق جديد استخدام الشبكة.
5. **تسجيل (Logging)** لجميع الأنشطة المسموح بها والمرفوضة.
6. **التقارير والتنبيهات** لمساعدة المحللين الأمنيين في تتبع النشاط المشبوه.

استخدام السجلات (Logs) في التحليل الأمني:

السجلات تتضمن معلومات مثل:

* عناوين IP المصدر والوجهة
* المنافذ والبروتوكولات
* الإجراء (سمح/رفض)
* التوقيت

يستخدم محللو الأمن هذه البيانات لاكتشاف أنشطة مثل:

* اتصالات صادرة غير مبررة (قد تشير إلى اختراق أو botnet)
* محاولات وصول متكررة من IP واحد

🔹 يمكن الاستعانة بأدوات مثل **WHOIS** لمعرفة هوية عناوين IP، مع الحذر من **IP spoofing**.

#### الجدران النارية في أنظمة التشغيل:

🪟 Windows:

* الجدار الناري مفعّل **افتراضيًا** ويمنع جميع الاتصالات الواردة غير المطلوبة.
* يمكن إنشاء قواعد (Rules) عبر **Windows Defender Firewall with Advanced Security**:
  + **Program Rules:** للتحكم في البرامج.
  + **Port Rules:** للتحكم في المنافذ والبروتوكولات.
  + **Predefined Rules:** لخدمات النظام.
  + **Custom Rules:** تجمع أكثر من شرط.
* يمكن تفعيل **التسجيل (Logging)** لتتبع الاتصالات وإجراءات الحظر.

🐧 Linux:

* **IPtables:** الجدار الناري الأساسي في نواة النظام، للتحكم في المنافذ والبروتوكولات.
* **TCP Wrappers:** يعمل على مستوى التطبيقات (أقدم وأقل استخدامًا حاليًا).
* **UFW (Uncomplicated Firewall):** واجهة بسيطة للتحكم في iptables.

### مضاد فيروسات محلي Host-Based Antivirus

الـ **Host-Based Antivirus** هو برنامج أمني يُثبت على الجهاز (المضيف) لحمايته من الفيروسات والبرمجيات الخبيثة.

رغم أن فعاليته بدأت تقل أمام التطور السريع في أساليب الهجوم، إلا أنه لا يزال **مصدرًا مهمًا للبيانات (telemetry)** التي تساعد محللي الأمن في الكشف عن التهديدات على مستوى الجهاز.

تطور وظيفة مضاد الفيروسات:

في البداية صُمم للكشف عن **الفيروسات فقط**، وهي نوع من البرمجيات الخبيثة (malware) تنتشر عبر نسخ نفسها داخل برامج أخرى.  
 لكن مع الوقت ظهرت أنواع أخرى من البرمجيات الخبيثة، وأصبح مضاد الفيروسات يتعامل مع:

* **Keyloggers** (مسجلات المفاتيح)
* **Backdoors** (الأبواب الخلفية)
* **Rootkits**
* **Browser hijackers**
* **Trojan horses**
* **Ransomware**

ورغم توسع قدراته، فإن سرعة انتشار البرمجيات الخبيثة اليوم تفوق قدرة مضادات الفيروسات على مجاراتها.

تقنيات الكشف المستخدمة:

1. **Signature-based Detection (الكشف بالتواقيع):**
   * يعتمد على قاعدة بيانات تحتوي تواقيع لأنواع معروفة من البرمجيات الخبيثة.
   * فعال ضد الهجمات المعروفة فقط.
   * غير قادر على كشف **الهجمات الجديدة (Zero-day Attacks)**.
2. **Heuristic Detection (الكشف بالتحليل الاستدلالي):**
   * يكشف البرمجيات الخبيثة المتحورة أو التي تغير شكلها لتفادي التواقيع.
   * يساعد في التعرف على **عائلات أو أنماط** من البرمجيات الخبيثة.
3. **Behavioral-based Detection (الكشف بالسلوك):**
   * يراقب سلوك العمليات بدلاً من الشيفرة البرمجية.
   * إذا حاولت عملية القيام بسلوك خبيث (مثل تعديل ملفات تنفيذية أو سرقة ضغطات المفاتيح)، يتم كشفها.
   * فعال ضد **التهديدات غير المعروفة (zero-day)**.

خصائص التشغيل:

* يجب أن يتم **تحديث قاعدة بيانات التواقيع بانتظام** لضمان الفعالية.
* معظم مضادات الفيروسات الحديثة تقدم **حماية فورية (Real-time Protection)**:
  + تفحص البيانات أثناء تحميلها إلى الذاكرة، مثل:
    - عند تشغيل تطبيق من USB
    - عند فتح بريد إلكتروني
    - عند تنفيذ معاملة ويب

Spyware وAdware:

* **Spyware:** يجمع المعلومات دون علم المستخدم، مثل ملفات تتبع (tracking cookies) في المتصفحات.  
   قد لا يكون خطيرًا دائمًا، لكن يمكن أن يصبح **malware** إذا استُخدم لأغراض ضارة (مثل التجسس أو سرقة البيانات).
* **Adware:** يعرض إعلانات مقابل استخدام البرنامج.  
   إذا كان المستخدم **يعلم ويوافق** على هذا، فلا يُعد تهديدًا.  
   أما إن كان غير مدرك أو إذا أُسيء استخدام البيانات الإعلانية (مثل استهداف المستخدمين بهجمات تصيد)، فهو يُعتبر **spyware أو malware**.

Antispyware:

* برنامج يُثبت على الأجهزة لاكتشاف وإزالة **Spyware**.
* يوجد **تداخل وظيفي كبير** بينه وبين برامج **Antivirus** لأن الحدود بين البرمجيات الخبيثة وملفات التجسس أصبحت غير واضحة.

### (Host Intrusion Prevention System (HIPS

نظام **HIPS** هو برنامج يُثبت على المضيف (host) لحمايته من الهجمات عبر **الكشف والمنع**، ويجمع بين وظائف **مضاد الفيروسات (Antivirus)** و**مضاد التجسس (Antispyware)** و**الجدار الناري الشخصي (Personal Firewall)**.

يتميز HIPS بأنه يرى ما يحدث داخل النظام نفسه، ويمكنه تحليل البيانات بعد فك تشفيرها — وهو ما لا يستطيع نظام IPS الشبكي (NIPS) القيام به.

طرق الكشف التي يستخدمها HIPS:

1. **Signature-Based Detection (الكشف بالاعتماد على التواقيع):** يقارن النشاط أو البيانات بقواعد محددة (signatures) للهجمات المعروفة.
   * يحتاج إلى تحديث مستمر للتواقيع.
   * قد ينتج **إنذارات كاذبة (False Positives)** بسبب تشابه نشاط طبيعي مع نشاط ضار.
   * الحل: **التحسين (Tuning)** لتقليل الإنذارات الكاذبة.
2. **Anomaly-Based Detection (الكشف عبر الشذوذ):** يعتمد على مقارنة السلوك الحالي مع **خط الأساس (Baseline)** للنشاط الطبيعي.
   * إذا انحرف السلوك عن النمط المعتاد → يولّد إنذارًا.
   * فعال ضد الأنشطة الجديدة وغير المعروفة.
3. **Policy-Based Detection (الكشف بناءً على السياسات):** يراقب الأنشطة وفق **سياسات معدّة مسبقًا** (مثل منع تشغيل ملفات معينة أو تعديل النظام).
4. **مزيج من الطرق السابقة** لتحسين الدقة وتقليل الأخطاء.

أنواع التنبيهات (Alarms) في HIPS:

| **النوع** | **الشرح** | **النتيجة** |
| --- | --- | --- |
| **True Positive** | كشف تهديد فعلي وأطلق إنذارًا | ✅ جيد — يعني أن النظام فعّال |
| **True Negative** | لم يكتشف أي نشاط ضار ولم يُصدر إنذارًا | ✅ طبيعي وصحيح |
| **False Positive** | أطلق إنذارًا على نشاط طبيعي | ⚠️ مزعج — يجب **تعديله أو تجاهله عبر tuning** |
| **False Negative** | لم يصدر إنذارًا رغم وجود تهديد | 🚨 خطير — غالبًا بسبب نقص في التحديثات أو التواقيع |

### **أهمية الـ Tuning:**

* عملية **تحسين إعدادات الكشف** لتقليل الإنذارات الكاذبة وزيادة دقة النظام.
* تمر عادة بمرحلة مراقبة لعدة أيام بعد التثبيت حتى تستقر النتائج، ثم يتم **تثبيت السياسة النهائية (Lock Policy)**.

### Application Allowed Lists and Blocked Lists

قوائم السماح والمنع (**Allowlist / Blocklist**) هي أدوات تُستخدم للتحكم في **الوصول إلى الموارد** (مثل الشبكات أو التطبيقات أو المواقع).

يُعدّ الحفاظ على هذه القوائم من مسؤوليات محلل الأمن (Security Analyst).

الفرق بين القائمتين

| **القائمة** | **المبدأ** | **النتيجة** |
| --- | --- | --- |
| **Blocked List (Blacklist)** | أي كيان في القائمة **يُمنع** من الوصول | الكيانات غير الموجودة في القائمة مسموح بها |
| **Allowed List (Whitelist)** | أي كيان في القائمة **مسموح له** بالوصول | الكيانات غير الموجودة في القائمة تُمنع |

يمكن أن تحتوي القوائم على:

* **عناوين IP أو نطاقات IP**
* **تطبيقات أو ملفات تنفيذية**
* **نطاقات Domains أو روابط URLs**
* **عناوين بريد إلكتروني**

### **الاستخدام في منتجات الأمان:**

* يمكن تطبيق القوائم يدويًا على:  
   **أنظمة منع التسلل (IPS)**، **جدران الحماية (Firewalls)**، **أنظمة البريد الإلكتروني**، **أمن الويب**.
* يمكن الاستفادة من **القوائم الديناميكية (Dynamic Lists)** المقدمة من خدمات استخبارات التهديدات (Threat Intelligence Services) التي تُحدّث تلقائيًا.
* يمكن الدمج بين القوائم:
  + **الـ Allowlist تتفوق على الـ Blocklist** (في حال التعارض، السماح يتغلب على المنع).

### Host-Based Malware Protection & Next-Generation Endpoint Security

حتى أقوى برامج مكافحة الفيروسات لا تتجاوز فعاليتها الواقعية **40٪** في الكشف عن البرمجيات الخبيثة، لذلك يجب اعتبار أن **الاختراق سيحدث لا محالة**، والتركيز يكون على **الكشف والاستجابة بعد الإصابة** وليس فقط المنع المسبق.

الحماية القائمة على المضيف (**Host-Based Malware Protection**) مثل **Cisco AMP for Endpoints** توفر مستوى متقدم من الرؤية والتحكم لحماية الأنظمة **قبل وأثناء وبعد الهجوم**.

Cisco AMP for Endpoints:

هو نظام ذكي من Cisco يجمع بين **التحليل السحابي، البيانات الضخمة، والتحليلات المستمرة** لتتبع الأنشطة الضارة على الأجهزة.

أهم الخصائص:

* **حماية مستمرة** قبل وأثناء وبعد الهجوم.
* يعتمد على **الذكاء السحابي (Cisco Talos Threat Intelligence Cloud)** لتحديث التهديدات في الوقت الحقيقي.
* يستخدم **التعلّم الآلي (Machine Learning)** لتحليل الملفات ذات الوضع غير المعروف.
* يمكنه **تتبع مسار الملفات (File Trajectory)** و**مسار الأجهزة (Device Trajectory)** لمعرفة مصدر العدوى وسلوك الملف.
* يدعم **التحذير الرجعي (Retrospective Alerting / Cloud Recall)**: إذا تبيّن لاحقًا أن ملفًا كان خبيثًا، يقوم النظام بعزله من جميع الأجهزة التي تحتوي عليه.
* يسمح بإنشاء **سياسات مخصصة** أو **تواقيع مخصصة** (Custom Signatures) لأنواع محددة من التهديدات.
* يقدم **تقارير وتحليلات متقدمة** للأحداث الأمنية.

المكونات الأساسية:

1. **Cisco Talos Threat Intelligence Cloud:** القلب الرئيسي للنظام، يحتوي على محركات التحليل والتعلم الآلي، ويُحدث التواقيع الأمنية باستمرار.
2. **Client Connectors:** برمجيات تُثبت على الأجهزة، ترسل بيانات الملفات للسحابة وتستقبل قرارات التصنيف.
3. **AMP for Networks:** يتيح لأجهزة **Firepower** الاستعلام من السحابة عن الملفات المشبوهة.

### **Next-Generation Endpoint Security:**

يشمل الجيل الجديد من أمن النقاط الطرفية مزيجًا من **المنع + الكشف المستمر + الاستجابة السريعة**.  
 يتكون من عنصرين رئيسيين:

1. Endpoint Protection Platform (EPP)

* يُعنى **بمنع التهديدات المعروفة** (مثل مضاد الفيروسات التقليدي).
* يعتمد على **توقيعات التهديدات** وميزات مثل الجدار الناري ومكافحة البرمجيات الضارة.
* يعاني من ضعف في مواجهة التهديدات **بدون ملفات (Fileless)** أو **متعددة الأشكال (Polymorphic)**.

2. Endpoint Detection and Response (EDR)

* يكتشف **التهديدات المتقدمة** التي تجاوزت الطبقات الوقائية.
* يقدم **رؤية شاملة لدورة حياة الهجوم**: كيف دخل، ماذا فعل، وأين انتشر.
* يشمل مراحل:
  + **Detection:** الكشف عبر التحليل السلوكي المستمر.
  + **Containment:** عزل التهديد لمنع انتشاره.
  + **Investigation:** تحليل السلوك داخل بيئة معزولة (**Sandboxing**).
  + **Elimination:** إزالة التهديد وإعادة النظام لحالته السابقة.

💡 المبدأ الذهبي:

لا يمكن لأي نظام وقائي منع جميع الهجمات، لذلك يجب دمج **EPP + EDR** للحصول على حماية شاملة.

### ِSandboxing

تقنية **Sandboxing** تُستخدم لتحليل الملفات المشبوهة في **بيئة معزولة وآمنة** بهدف معرفة سلوكها قبل السماح بتشغيلها في النظام الفعلي. وهي تُعد مكملًا مهمًا لأنظمة الكشف التقليدية (مثل الكشف بالتواقيع Signature-based detection) التي تعجز عن اكتشاف البرمجيات الخبيثة الحديثة **متعددة الأشكال (Polymorphic Malware)**.

### **لماذا نحتاج إلى Sandboxing؟**

* البرمجيات الخبيثة المتغيرة الشكل (**Polymorphic malware**) تغيّر مظهرها باستمرار لتفادي أنظمة الكشف بالتواقيع.
* لذلك، بدلاً من الاعتماد على مظهر الملف، تقوم الـSandbox بتحليل **سلوك الملف أثناء التنفيذ** داخل بيئة افتراضية.
* هذا يسمح بكشف التهديدات الجديدة حتى لو لم تكن هناك **تواقيع مسبقة** لها.

### **🔹 كيف تعمل الـSandbox:**

1. **تُحاكي بيئة نظام تشغيل حقيقية** (افتراضية أو معزولة).
2. **تشغّل الملف المشبوه داخلها (detonation)** بدون أي خطر على النظام الفعلي.
3. **تراقب وتُسجّل كل سلوك الملف** مثل:
   * محاولات **تسجيل ضغطات المفاتيح (Keylogging)**
   * **الوصول إلى مفاتيح الريجستري الحساسة**
   * **تعديل ملفات النظام**
   * **تحميل مكتبات DLL** أو الاتصال بخوادم خارجية (Command & Control).
4. بناءً على هذا السلوك، يتم تحديد ما إذا كان الملف **ضارًا أو سليمًا**.

### **🔹 مثال عملي من Cisco:**

* منتج Cisco في هذا المجال هو **Cisco Threat Grid**.
* يمكن نشره **سحابيًا (Cloud)** أو **محليًا (On-premises appliance)**.
* عند تحليل ملف، يمكن للنظام كشف:
  + **الاتصالات الصادرة (Outbound HTTP Traffic)** التي قد تشير إلى **اتصال خبيث بخادم C2 (Command & Control)**.
  + تفاصيل مثل **المسار الكامل للـURI** المستخدم في النشاط الخبيث.
* هذه البيانات تساعد محللي الأمن في **حظر العناوين الخبيثة** أو تتبع مصادر العدوى.

### **🔹 نقاط ضعف Sandboxing:**

رغم قوتها، إلا أن التقنية ليست كافية لوحدها، ومن أبرز **القيود**:

1. **Inherent Efficacy** – ليس مضمونًا أن يُظهر الملف سلوكًا خبيثًا أثناء التحليل.
2. **Evasion Tactics** – بعض البرمجيات الخبيثة **تتعرف على بيئة Sandbox وتتوقف عن النشاط** لتفادي التحليل.
3. **Means to an End** – الـSandbox أداة مساعدة، وليست حلاً شاملاً، ويجب دمجها مع تقنيات أخرى مثل **EDR وThreat Intelligence** لتحقيق حماية متكاملة.

التحقق من سلامة الملفات

التحقق من سلامة الملفات (**File Integrity Checking - FIC**) هو أسلوب يُستخدم لاكتشاف **أي تعديل غير مصرح به** على الملفات أو أنظمة التشغيل بعد تثبيتها.

يُعد من الأدوات المهمة في **الكشف عن الاختراقات ومحاولات التلاعب بالنظام**.

### **آلية العمل:**

1. يقوم النظام أو الأداة بحساب **قيمة تجزئة (Hash)** لملفات النظام الهامة.
2. يتم حفظ هذه القيم في قاعدة بيانات آمنة.
3. لاحقًا، تتم مقارنة القيم الجديدة بالقيم الأصلية:
   * ✅ إذا كانت **القيم متطابقة** → الملف لم يتغير.
   * ⚠️ إذا كانت **مختلفة** → تم تعديل الملف أو التلاعب به.

### **🔹 متى يتم تنفيذ الفحص:**

* **بعد التثبيت الأول للنظام (Fresh Install)** لإنشاء baseline آمن.
* **بعد تثبيت التحديثات أو التصحيحات (Patches)** لأن الملفات قد تتغير رسميًا.
* **عند الاشتباه في اختراق أو تعديل غير مصرح به.**

### **🔹 أمثلة على الأدوات:**

* **Tripwire**
* **AIDE (Advanced Intrusion Detection Environment)**
* **AFICK (Another File Integrity Checker)**

ملاحظة: هذه الأدوات يمكن أن تعمل أيضًا كجزء من **Host-Based Intrusion Prevention System (HIPS)**.

### **🔹 التحقق من سلامة أجهزة الشبكة:**

لا يقتصر الأمر على أنظمة التشغيل فقط، بل يشمل أيضًا:

* **أجهزة الشبكة (مثل الموجّهات والجدران النارية)** التي تحتوي على **صور برمجية (software images)**.
* إذا تم التلاعب بهذه الصور، يمكن للمهاجم التحكم في الجهاز بالكامل.

لذلك يجب التحقق من **توقيع الصور الرقمية** باستخدام مفاتيح التحقق العامة للشركة المصنعة (مثل Cisco).