Securing Devices

AP Cybersecurity ▪ Unit 4

AP Cybersecurity

What we will cover

1. Devices & malware types
2. Hashing & salt
3. Password attacks
4. Authentication factors
5. Protecting devices
6. Detecting attacks

Devices, and the malware that hunts them

A device is any computer – server, laptop, phone, or an **embedded computer** 嵌入式计算机. Everyday embedded devices are **IoT** 物联网: water pumps to washing machines.

> The main threat is **malware** 恶意软件 – and each type is defined by one trait.

Fingerprint auth supports the idea on this slide

Fileless malware, and what adversaries exploit

Most malware is a file. **Fileless malware** 无文件恶意软件 is different: it lives only in **RAM** 内存 and abuses legitimate programs already on the device.

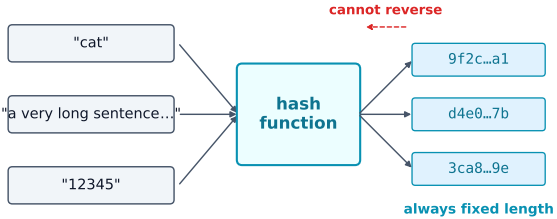
> It leaves no file for a scanner to find – which is exactly why signature scanning misses it.

Security key supports the idea on this slide

The cryptographic hash function

A **hash function** 密码散列函数 is one-way maths turning any input into a fixed-length **hash** 散列值. Three vital properties:

- **collision resistant** 抗碰撞 – hard to find two inputs with one output
- **pre-image resistance** 抗原像 – you cannot work backwards
- repeatable – the same input always gives the same hash



fixed length, and no way back

> A service never stores your password. It stores the hash, then hashes what you typed and compares.

Worked example – why salt matters

Two users both choose `sunshine`

❌ **No salt:** both stored hashes are identical – cracking one instantly cracks the other. ✅

With salt 盐值: give each a few random bits – `x7` and `q2`. The service hashes `sunshinex7` and `sunshineq2`, so the stored hashes look completely different.

> The adversary must now attack each account separately – which is why a stolen hash database is far less dangerous when salted.

Never say a service "stores the password". It stores the salted hash.

Password attacks – and their log signatures

Password spraying 密码喷洒
one common password against many accounts
log: many users failing from one IP

Credential stuffing 撞库
stolen or default credentials
log: a burst of default logins

Rainbow table 彩虹表
a precomputed table of passwords and hashes

Guessing
log: one user + many wrong passwords

Online attacks guess against a live login. Offline attacks steal the hash database and crack it on the adversary's own machine – so they cannot be detected. A favourite exam gotcha.

Authentication factors

something you **know**
a password

something you **have**
a token or phone

something you **are**
a **biometric** 生物特征

somewhere you **are**
a location factor

MFA 多因素身份验证 combines two or more categories – a fingerprint plus a password, not two passwords.

Protecting devices

Managerial sets the rules: an **acceptable use policy** 可接受使用政策, a password policy (length, reuse), a software installation policy.

Anti-malware 反恶意软件 – matches signatures, quarantines

each **patch** 补丁 closes a known hole first

a **host-based fire-wall** 主机防火墙 guards one device

> Patching closes known holes before adversaries can use them – most breaches use a hole with a patch already available.

Detecting attacks – indicators of compromise

Logs reveal an **indicator of compromise** 入侵指标 – evidence an adversary got in:

- **host-based** – unexpected processes, changed settings
- **file-based** – a file whose hash matches known malware
- **behaviour-based** – many failed logins, unusual hours

Choosing a method weighs performance (signature is lighter – better for weak devices), cost (**EDR** 端点检测与响应 is powerful but expensive), and how sensitive the device is.

Unit 4 – key takeaways

1 **One trait each**

worm self-spreads, virus needs a user, rootkit hides

2 **Salted hash**

one-way, fixed length – never "stores the password"

3 **Log signatures**

one user many passwords = guessing; many users one IP = spraying

4 **MFA**

two different categories – not two passwords

Check yourself

1. Worm or virus: which needs a user to open a file?
2. Why can an offline password attack not be detected?
3. Are a password and a PIN multifactor authentication?
4. What does salt stop two identical passwords from doing?

Answers **1.** virus **2.** it runs on the adversary's own machine **3.** no – both are "something you know" **4.** producing identical hashes