AP CYBERSECURITY • EXERCISE SHEET

5.6 Detecting Attacks on Data and Applications

Name: _____ Class: _____ Date: _____ **Total: 44 marks**

KEY WORDS honeypot 蜜罐 • accounting 记账审计 • file integrity 文件完整性 • buffer overflow 缓冲区溢出 • directory traversal 目录遍历

Objective

Build the skills to answer exam questions on detecting attacks against **applications and data**: **accounting** 记账审计 logs, **honeypots** 蜜罐, **file integrity** 文件完整性 by hash, and reading logs for application attacks.

You must be able to:

- explain what accounting is and what its logs reveal
- describe a honeypot and say what makes it a near-instant detector
- verify a file's integrity by comparing hashes, and state the property that makes this work
- choose detective controls against cost, criticality and data classification
- explain the false negative a hash check cannot catch
- name the application attack from log evidence

1 Worked examples

Study these first. Each one shows the method for a task used later.

■ Accounting: the log that says who touched the data

Devices track and log **when data are accessed and by whom**. Recording and monitoring user activity is called **accounting**, and analysing those logs is how an unauthorised access is discovered after the fact.

The limitation is scale: read by hand, log analysis is far too slow to be a detection method, so it has to be **augmented with automation** before it operates at a useful speed.

■ The honeypot

A **honeypot** is a file that *looks* valuable - credit card numbers, PII, passwords - but whose contents are **fake**. Nobody with a legitimate job has any reason to open it.

That is the whole design: because there is no innocent explanation, an access to it is **near-instantaneous, essentially false-positive-free detection**. It is also **inex-**

pensive, which is why it appears in the cost criterion as the cheap option alongside hash checking.

Verifying a file by its hash

The property in play is **repeatability**: the same input always produces the same output for a given hash function.

The method is three steps:

1. Hash the file and **record** the output.
2. Later, hash the file again.
3. **Compare** the two outputs. Identical means unchanged; different means the file has been altered.

Hashes can be produced from the command line, a website, or specialised software.

Example. A configuration file's recorded digest is `a91f...c3`. Today it hashes to `7d20...b8`. The file has changed. The hash does not say *what* changed, *who* changed it, or *when* - only that it is no longer the file that was recorded.

The false negative that matters

A hash check detects **alteration only**. An adversary who **views or copies** a file changes nothing, so the hash still matches and the theft is invisible to this control.

Say this explicitly whenever a question asks you to evaluate hashing: it protects **integrity**, not **confidentiality**. Detecting the copy needs accounting logs or a honeypot instead.

Choosing detective controls

Criterion	How it decides
Cost	honeypots and hash checking are inexpensive; full DLP tooling is not
Sensitivity or criticality	more sensitive or critical data is more likely to be attacked, so it justifies more
Classification	data classified private, educational, healthcare or financial usually carries a legal requirement for stronger protection

Naming the application attack from a log

Evidence in the log	Attack
SQL control words and symbols in user input	**SQL injection**
A `<script>...</script>` tag in user input	**XSS**
A request field far larger than expected	**buffer overflow**
A GET request whose path contains `../` sequences	**directory traversal**

2 Practice

2.1 Define **accounting** and state what its logs can reveal. [2]

2.2 Explain why raw log analysis alone is too slow to be an effective detection method. [2]

2.3 Describe a **honeypot** and explain why an access to one is almost never a false positive. [3]

2.4 A team records the hash of a configuration file, then re-hashes it a month later.

(a) State which property of a hash function makes this comparison meaningful. [1]

(b) State what a different output tells them. [1]

(c) State **two** things the comparison does **not** tell them. [2]

2.5 Explain the false negative that a hash-based integrity check cannot avoid. [2]

2.6 Name the attack indicated by each log entry.

(a) `GET /files/../../../etc/shadow` [1]

(b) A comment field containing `<script>alert(1)</script>` [1]

(c) A username field containing `' OR 1=1 --` [1]

(d) A 12-character field receiving 9000 characters. [1]

2.7 For each organisation, recommend a detective control and justify it against a named criterion.

(a) A small charity with almost no budget storing donor names and addresses. [2]

(b) A hospital's patient record system. [2]

2.8 Explain why data classified as healthcare or financial usually receives stronger detective controls. [2]

3 Exam-style questions

3.1 A honeypot contains [1]

A B
C D

A real credit card data used as bait
B fake data in a file that appears valuable
C an encrypted copy of the production database
D a list of known malware hashes

3.2 Which property of a hash function makes file integrity checking possible? [1]

A	B
C	D

A collision resistance
B pre-image resistance
C repeatability
D fixed length

3.3 An adversary copies a sensitive file without modifying it. A hash-based integrity check will [1]

A	B
C	D

A detect it, because the file was accessed
B detect it, because the hash changes on read
C fail to detect it, producing a false negative
D fail to detect it, producing a false positive

3.4 Which detection method offers near-instantaneous detection at low cost? [1]

A	B
C	D

A manual log review
B a honeypot
C a full data loss prevention deployment
D a quarterly vulnerability scan

3.5 A university research office stores three things on one server: a public list of seminar dates, a set of student grade records classified as educational, and a grant financial ledger. Detective controls are limited to a monthly manual review of access logs. The office adds a file named `student_bank_details.xlsx` containing fabricated entries, and records a hash of the grade-record file each Friday.

(a) Name the control the fabricated file represents and explain how the office will know it has been touched. [3]

(b) On one Friday the grade file's hash differs from the previous week's. State what this proves and what it does not prove. [3]

(c) A month later the office learns the grant ledger was copied and sold. Explain why neither the hash check nor the monthly log review caught this in time. [4]

(d) Recommend a detective control for each of the three data sets, justifying each against a different criterion. [4]

(e) The office proposes replacing everything with a full DLP deployment. Evaluate this proposal, referring to cost and to what the existing cheap controls already achieve. [3]