Here’s the **most comprehensive, sentence-by-sentence breakdown** of the document titled **"97. System File Checker Notes"**, rewritten into **clear bullet-point study notes** to help you prepare for the **CompTIA A+ 220-1102 exam**.

**🧠 1. Concept Overview: System File Checker (SFC)**

**System File Checker** (SFC) is a built-in command-line utility in Windows that scans and restores **corrupted or damaged system files** using cached copies. Its main purpose is to **protect the integrity of the Windows operating system** and prevent malicious changes to important files or registry entries.

**📚 2. Exam Relevance – CompTIA A+ 220-1102**

**✅ Covered Under:**

* **Objective 1.5**: Use Microsoft command-line tools
* **Objective 2.4**: Troubleshoot Microsoft Windows OS problems

SFC is essential knowledge for diagnosing system integrity issues, repairing installations, and maintaining OS stability.

**✍️ 3. Detailed Study Notes – Sentence-by-Sentence**

**🔸 What Is SFC?**

* **SFC = System File Checker**
* It is a **manual interface** for verifying protected system files.
* If files are found to be **corrupt or damaged**, SFC will restore them from a **cached copy** stored locally.

**🔸 Purpose and Importance**

* Helps **protect Windows** from system file damage and **unauthorized modifications**.
* Also protects the **registry keys** and important configuration files.
* Acts as a **first line of defense** against malware that attempts to modify system files.

**🔸 Running SFC**

* To run the tool:
* sfc /scannow
* This performs a **one-time immediate scan**.
* Takes a few minutes to **scan all system files**.

**🔸 Admin Rights Are Required**

* SFC **must be run from an administrative Command Prompt**.
* If run from a normal prompt, you'll receive an error asking for **elevated privileges**.

**🔸 How SFC Works Internally**

* It checks **every Windows system file** one by one.
* For each file, SFC:
  1. Calculates a **hash value**
  2. Compares it to a list of **known good hashes** from Microsoft
* If hashes **match**, the file is considered safe and unchanged.
* If hashes **don’t match**:
  1. The file is flagged as compromised
  2. SFC **replaces it** with a **clean copy from the local cache**

**🔸 Protection Against Malware or Tampering**

* SFC helps prevent damage from:
  + Malware infections
  + File corruption
  + Unauthorized changes to system files

**🔸 Output Example**

* After running SFC, you may see:
* Windows Resource Protection found corrupt files and successfully repaired them.
* This means SFC **identified** and **repaired** system files that were altered.

**🔸 Additional SFC Options**

* To see all command-line options:
* sfc /?
* This shows:
  + /scannow – scan all files
  + /scanfile – scan a **specific file** (faster)
    - Must specify the **full file path**
    - Example:
    - sfc /scanfile=C:\Windows\System32\notepad.exe

**🔸 When to Use /scanfile**

* More efficient when:
  + You know **exactly which file** is problematic
  + You don’t want to scan the **entire system**

**🔸 Summary: Why Use SFC?**

* Verifies integrity of **protected system files**
* Replaces **incorrect versions** with **genuine Microsoft copies**
* Ensures OS is not damaged or compromised
* A go-to utility for troubleshooting weird system behavior or instability

**💻 4. Real-Life Use Cases**

**🛠 Technician Fixing Corrupt System Behavior**

* After a failed Windows Update, system behaves oddly.
* Runs sfc /scannow → finds and repairs corrupt DLL files.

**👨‍💻 Administrator Diagnosing Malware Damage**

* After removing malware, runs sfc /scannow to make sure no core system files were tampered with.

**💾 Support Staff Repairing Specific Executable**

* Notepad won’t launch correctly.
* Uses:
* sfc /scanfile=C:\Windows\System32\notepad.exe

**✅ 5. Exam Inclusion Notification**

✅ **Definitely included in CompTIA A+ 220-1102**

**Why:**

* Part of **Windows command-line tools** (Objective 1.5)
* Often appears in **performance-based questions** (simulations)
* Important for **repairing OS-level issues** and **maintaining file integrity**