When researching attack signature analysis techniques, I have found many advertisements for different programs that offer looking for different attacks. Looking closer though, I believe the different analysis techniques can be grouped into eight techniques. These techniques are listed here:

* Pattern Matching
* Heuristic Analysis
* Anomaly Detection
* Behavioral Analysis
* Traffic Flow Analysis
* Machine Learning Clustering
* Code Analysis (Static and Dynamic)
* Honeypots

Below I have provided small definitions of what each technique is along with some strengths and weaknesses of each.

**Pattern Matching**

Pattern matching scans data for predefined strings or byte sequences that match known attacks. This type of detection is used in programs like Snort and Suricata. A simple example of a pattern that Snort can look for would be for a simple SQL injection ‘ OR 1=1 --.

This type of detection is considered fast and efficient with simple implementation but the weaknesses of this are it only looks for known therefore zero-day attacks will be missed. Another weakness is that it becomes a constant war between attacker and defender when payloads are obfuscated.

**Heuristic Analysis**

Norton Antivirus heavily promotes this analysis as a key detector. Instead of looking at predefined strings and other sequences. Heuristic analysis looks at the system’s behavior and traffic patterns to look for suspicious activity or known behavior outcomes. For instance, sudden encryption of multiple files in a short period of time.

This type of detection is good for looking for malware variants and even polymorphic malware. The processing overhead is considered medium. This also gives a medium amount of false positives. So refining is a must.

**Anomaly-Based Detection**

This type of detection looks for deviations from a network or system’s normal behavior. This can be accomplished by using either machine learning or statistical models. An example of this would be a user logging in at a weird hour for them.

The strength of this is that it can potentially detect a zero-day attack or an Advanced Persistent Threat. One example of a system like this would be Splunk. The challenge with this is the overhead of continuous learning for the system and the need for tuning because of the common false positives.

**Behavioral Analysis**

This is like Heuristic Analysis but focuses on detecting the attack behavior instead of signatures by looking at action sequences using AI or predefined models. An example of this type of behavior would be as soon as a person successfully logs in, Mimikatz is run.

This type of analysis is good for looking for fileless malware or living-off-the-land attacks. This type also involves a lot of resource overhead and can provide a lot of false positives. Also, it can be countered by using low-and-slow attacks. It can be effective though with the example of using CrowdStrike.

**Traffic Flow Analysis**

Traffic flow analysis watches network traffic flow patterns with something like Wireshark. Analysis of things like duration, volume, and frequency are the triggers.

An analysis like this can potentially detect a C2 beacon by finding small periodic connections to an external server. This makes it very effective when discovering communication patterns of malware but requires deep network visibility. The overhead for this is quite large. For instance, Wireshark can easily fill a hard drive with packet data in a very short time on a busy network. Therefore, it becomes necessary to periodically take samples in the hopes of catching things. It also can be evaded by things like HTTPS.

**Signature Clustering**

Signature clustering is another name for machine learning. The AI model is taught to look for group similarities with attacks. Darktrace is an example AI model of this.

Because of looking for cluster patterns AI can be taught to identify things like ransomware based on the style of encryption instead of needing to look at hashes. It can also look at scripts to detect malicious PowerShell activation. Because it is using AI it requires heavy overhead on the network. AI also needs update training to prevent fading.

**Code Analysis**

This requires examination of the source code or execution behavior to detect malware. The two types are static and dynamic. Static looks at the code without executing it and dynamic, where the program is run in a sandbox.

The benefits of this type of analysis is it can detect obfuscated malware and potentially zero-days. However, it is very resource-intensive sandboxing and analyzing. Static can be countered by encryption or packed binaries using a took like Themida. But it has low false positives. VirusTotal is an example of this.

**Honeypots**

These are decoy systems on the network that are not used but can lure an attacker to access it when they are doing network discovery. For instance, a fake web server. Without any other knowledge, the attacker on a new system might try to access this system and set off an alarm because there is no legitimate reason for access.

Because of this honeypots have relatively low false positives. But processing overhead is high. An example of a honeypot is T-Pot. There was a debate about using honeypots with the thought that when advertising a service it is attracting more potential malicious actors. And while they are looking at the decoy, they might actually find the real network.

**Conclusion example**

As with the first example of Snort. There becomes a back-and-forth war with defenders and attackers. When looking at Snort. It can be set to look for scanning more than 10 ports within 3 seconds for example. The counter would be adjusting the Nmap setting to have a scan delay. This becomes back and forth with the defenders increasing the detection window from 3 seconds to 5 minutes. Every system has some sort of back and forth like this. To be honest, this is one of the main reasons for job security!