OWASP AppSensor

IMPLEMENT REAL-TIME
INTRUSION DETECTION
WITHIN YOUR SOFTWARE

ATTACK DETECTION

RESPOND AUTOMATICALLY

Free, open source, DevOps friendly and cloud compatible

Turn over to find out more

What Is AppSensor?

AppSensor provides real-time application-layer attack detection and response. AppSensor:

- Detects attackers, not vulnerabilities
- Is application-specific, not generic
- Does not use signatures, or try to predict anything
- Allows applications to adapt and respond in real-time to an identified attacker
- Stops and/or reduces the impact of an attack
- Provides visibility and security intelligence into your applications.

A single instance of AppSensor can support multiple client applications at the same time, and could aggregate attacker knowledge by, for example, correlation using SSO.

Why Should I Use It?

There are many security protections available to applications today. Many are at the host or network layer, and are not directly accessible to, or even known by, the application itself. Application level protections are generally focused around secure software development processes.

The AppSensor approach is implemented in the place where the most information is available to make the best security decisions: within the application itself. It also is implemented by the people with the most application context: developers and architects.

This leads to far greater accuracy and flexibility than many other application defence approaches.

How Do I Use It?

Policy Configuration

There are few steps to get setup using AppSensor. The first step is to configure your detection and response policy. You will build a configuration that has a number of business-justified descriptions such as:

```
3 Insufficient Authorization events in 5 minutes for an individual user
represents an attack. I want to respond by blocking the user account.
```

Collectively, these descriptions will define your policy for automated attack analysis and real-time response.

Application Instrumentation

Once the policy is created, you must place “detection points” that notify AppSensor of suspicious events. These might be done individually or using AOP, or even done with an external tool or process of some kind. Some example pseudo-code is below:

```
if ( isUserAuthorized( account ) ) {
      // present/view account
} else {
     //new code for appsensor
    appSensor.addEvent( logged_in_user, “INSUFFICIENT_AUTHORIZATION” )
}
```

Now, when a user attempts to access an account for which he is not authorized, the application notifies AppSensor and the event is tracked. If AppSensor determines the defined policy (e.g. 3 events in a span of 5 minutes) has been crossed, it is considered to be an attack. At that point, AppSensor executes the response, in this case an account lockout for the user.

Runtime Monitoring

Once the configuration is complete and the application is instrumented to signal events to AppSensor, the last step is to monitor the state of the running application. While AppSensor does provide an automated means of detection and response, monitoring the activity gives great visibility into the runtime state of the system. The intelligence you obtain from monitoring will lead to policy changes, new detection points, and new requirements for your system.

Code, documentation and tips www.appsensor.org

OWASP
Open Web Application Security Project