Workload Security with OSSEC

Mike Shinn, Founder & CEO
mike@atomicorp.com
703.403.6935
Workload Security is Forever Changed

Yesterday: A Secure Network of Servers

Secure the Network to Protect the Workload

Tomorrow: A Network of Secure Workloads

Workloads Themselves Must be Secured
But I Thought My Cloud Provider Did That...

Cloud Provider: Security OF the Cloud

Customer: Security IN the Cloud

“Through 2023 at least 99% of cloud security failures will be the customer’s fault.”
Security must be Baked in to Every Workload

AV
Deception
HIPS with Vulnerability shielding
Server workload EDR & behavioral monitoring
IaaS Data-at-rest Encryption
Exploit prevention/memory protection
Application control/whitelisting
System integrity monitoring/management
Network firewalling, segmentation and visibility
Hardening, configuration and vulnerability management

Market Guide for Cloud Workload Protection Platforms
Critical Requirements for Workload Security

Comprehensive
Address key security and compliance requirements without requiring a suite of point solutions.

Cross Platform
Protect workloads across physical and virtual machines, containers, and multiple public clouds with a single management interface.

Cost Effective
Scale cost effectively in dynamic public cloud environments where costs are driven by resource utilization.
Active response

Three way we find out about and respond to events in OSSEC

Logs

*See Something, Say Something*

External integration

*Let someone else handle it, but be informed*

Examples:

Atomic WAF, ClamAV, Integrator

Internal Integration

*Do it yourself*

Examples:

syscheck, rootcheck
Pros and Cons of internal integration

**Pros**

- Possibly more secure (isolation and separation)
- Possibly faster
- Better data

**Cons**

Harder/Higher LOE:
- you have to write code
- you have to maintain it

Possibly less secure (monolithic)
## Pros and Cons of logs

### Pros
- Quick, really really quick to integrate
- Support for almost anything
- No need to write code (easier)
- Big Picture

### Cons
- Security of stream, users injecting logs
- Decoders
- Obfuscation of attacks (eg web attacks)
- Reliant on the ability of the application to “detect” an attack
- Need to write a response mechanism
- lack of logs :-) (FIM)
Pros and Cons of external integration

Pros

If they can generate logs/readable out (eg API), really quick integration!

No need to write code

Specialization (eg WAF, kernel)

Possibly more secure (isolation and separation)

Integration with other platforms/projects

Cons

If you’re relying on logs, the same problems as logs

Synchronization (eg rule ids between projects, snort to ossec, modsecurity to ossec)

Overhead
To Repeat Myself: Security must be Baked in to Every Workload

AV
Deception
HIPS with Vulnerability shielding
Server workload EDR & behavioral monitoring
IaaS Data-at-rest Encryption
Exploit prevention/memory protection
Application control/whitelisting
System integrity monitoring/management
Network firewalling, segmentation and visibility
Hardening, configuration and vulnerability management
Atomicorp OSSEC Platform

Workload Security Modules
- Hardening
- Vulnerability Shielding (WAF + NIPS)
- Memory Protection
- Application Control and Visibility
- Server EDR / Behavioral Monitoring
- Micro-Segmentation
- Deception
- Antivirus / Antimalware

Data Presentation and Management Modules
- Management GUI
- Advanced File Integrity Management
- Roll Back
- Compliance Reporting
- API Integration with Slack, PagerDuty, Jira
- Threat Hunting

Community Threat Intelligence
- Log Analysis
- Registry Monitoring
- Integrity Checking
- Alerting & Active Response

Atomicorp Machine Learning Engine

Orchestration and Automation Engine
Questions?