SPARKL®

Solution Brief

A Security Incident Management Solution with the SPARKL® Sequencing Engine

Inside This Brief

About SPARKL

Bring Machines Together

All enterprises suffer from the black box swamp. Systems that work fine on their own, but won't play nicely with others.

It's hard to describe how a system should work - let alone how or why different systems interact.

SPARKL® is powerful technology for managing the behaviour of distributed systems. The lightning fast, distributed SPARKL Sequencing Engine drives events between machines, applications and things.

It provides Distributed Intelligence for true fog computing, allowing edge devices to interact with or without the cloud.

It introduces Reasoned Provisioning which spins up secure, on-demand infrastructure to meet the need of actual business logic.

Secured by blockchain, SPARKL logs every single event in a clean, connected Audit Trail to solve compliance and regulatory reporting across machines and systems, old and new.

SPARKL designs and develops the SPARKL® Sequencing Engine in London, UK. We work with partners including Cisco and Intel to market the product to innovators and customers worldwide.

Executive Summary

SPARKL provides an autonomics-based orchestration capability for automating the functions of a Security Operations Centre (SOC), particularly in relation to Security Incident Management.

SPARKL autonomics implements a closed loop of:

- **M**onitoring the state of an enterprise environment
- **A**nalysing it
- **P**lanning change actions
- and **E**xecuting those actions.

- SPARKL has a wide range of technology integrations. These integrations allow for multi-purpose and highly functional runbooks, for example. Whereas SPARKL implements the **A** and **P** phases of the autonomic cycle, with **M**, it defers to tools such as Apache Storm for other **M** functions, and Configuration Management tools such as Ansible for **E** functions.

- The ability to combine human and automated interventions into an orchestration is a key feature of the SPARKL approach.

- Automating SOC functions with SPARKL not only enables wider automation of Security Incident Management processes comprising the overall enterprise security posture, but it also allows for immediate transparency of security activities to auditors regarding matters of risk and compliance thanks to the logging and analytics capabilities of SPARKL.

- SPARKL also has a rich web-based authoring and management tool for the administration of orchestrations used to automate SOC functions.

Background

Cyber security attacks are diverse and increasing in number and type. It is just as important for enterprises to have in place well-defined strategies and plans for handling an attack as it is to have strategies for preventing them in the first place. An effective strategy for Security Incident Management is an absolute must in an enterprise context.

We define incident management as the "***capability to effectively manage unexpected disruptive events with the objective of minimising impacts and maintaining or restoring normal operations within time limits***." Whenever an incident occurs, a number of people and processes need to come together to contain the security breach and ensure business continuity.

The NIST SP-800-61 guidelines are considered to be definitive in providing a reference with dealing with security incidents. The guidelines define an incident response to have several phases, from initial preparation through to post-incident analysis. These phases characterise the **Security Incident Management Lifecycle**, and may be summarised as follows:

1. **Preparation**: Development of a formal incident response capability.

2. **Identification**: Determines when and if an incident has occurred.

3. **Containment**: How to respond to security incidents: informing management, preventing further damage and preserving evidence.

4. **Eradication**: Removing the cause of the incident.

5. **Recovery**: Restoration of Business-As-Usual service provision.

6. **Lessons Learned**: Reflection leading to improvements in strategies for preventing and dealing with incidents.

An effective approach to the handling of security incidents is essential for continued business operations, as well as many other enterprise governance, risk and compliance concerns.

For instance, in the context of compliance, there are ever-increasing demands for greater transparency into the activities of enterprise systems.

These demands come from regulators who audit access to key financial data and personal consumer information, as well as from risk management auditors who may require security event and incident information as part as an overall evaluation of conduct risk posture.

Opportunities

SPARKL is orchestration middleware that is applicable in a variety of contexts, such as the Internet of Things, cloud and enterprise services orchestration and integration, and so on.

One of the principal capabilities of SPARKL lies in its support for autonomics. This focuses on continually seeking to move a system towards its desired state. An autonomics solution will repeatedly execute a closed loop of: monitoring the state of the environment, analysing it (e.g. with respect to its divergence from the desired system state), plan a course of mitigated actions, and then execute those actions.

This is the so-called MAPE cycle of autonomic computing, as shown in **Figure 1**.

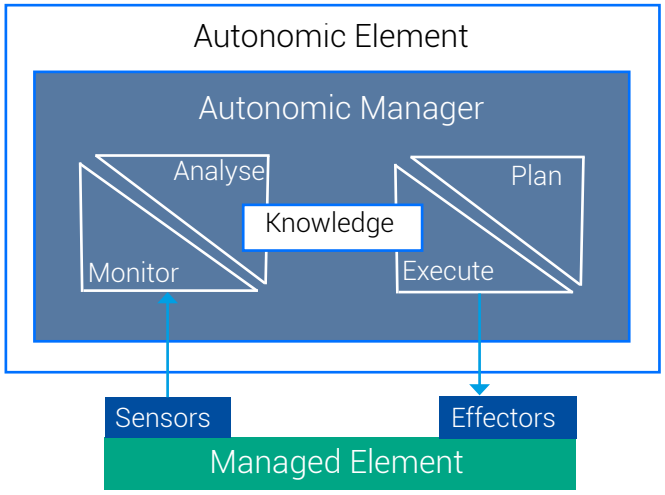

Figure 1: the Monitor, Analyse, Plan, Execute (MAPE) cycle of an Autonomic Element

Through its autonomic capabilities, SPARKL can respond to events in its environment, such as security incidents, and orchestrate the necessary steps (both human and machine-based) in response.

This is an ideal way to automating much of phases 2-5 of the Security Incident Management Lifecycle (Identification, Containment, Eradication and Recovery).

Typically, an enterprise will use a runbook, which is a document of steps that should be followed in order to deal with an incident. The document may instruct that certain support tools should be used, such as decision support systems; yet the overall orchestration of handling incidents remains largely manual.

SPARKL is capable of automating such a runbook as part of its wide-range of autonomics-based capabilities that are concerned with maintaining desired system states or invariants. Maintaining the health of an enterprise system from a security perspective is an example of a desired system invariant and an automated runbook is a means of ensuring it.

SPARKL has a number of technical integrations which allow for multi-purpose and highly functional runbooks such as **Ansible**, for pushing changes to several machines, and in parallel and **XMPP** for interacting with an administrator over **Jabber** and so on.

The ability to combine human and automated interventions into an orchestration is a key feature of the SPARKL approach.

At its core, SPARKL is an orchestration approach which focuses principally on the **A** and **P** phases of the MAPE cycle, namely **analysis** and **plannning**. SPARKL defers to tools such as **Ansible** to carry out some of the functions of the other phases.

We would recommend placing SPARKL at the heart of the systems providing the functions of a Security Operations Centre. By doing so, it is able to record events relating to security incidents and other SOC concerns as well as orchestrating SOC functions.

SPARKL offers analytics capabilities over these SOC event streams, which may include raw and aggregated data from security sensors, running on host and network devices.

SPARKL also has a rich web-based authoring and management tool for the adminstration of orchestrations used to automate SOC functions.

Capturing Security Incident Management runbooks as SPARKL orchestrations as part of the automation of SOC functions with SPARKL not only enables wider automation of the incident handling processes comprising the overall enterprise security posture, but it also allows for immediate transparency of security activities to auditors regarding matters of risk and compliance thanks to the logging and analytics capabilities of SPARKL.

Distributed & Centralized Deployment

The use of SPARKL for Cyber Security Incident Response allows an enterprise to make better use of the resources it has available, enabling a pragmatic approach to Security Operations, and meaning that we recognise that not every threat can be blocked. Whilst there will always be a place for incident prevention, we need to ensure effective mechanisms for incident detection and response.

SPARKL facilitates effective incident detection and response. It does so by providing support for the definition and execution of action sequences, or workflows, for the detection, investigation and mitigation of incidents.

SPARKL then closes the loop in making sense of the wealth of data that comes from enterprise infrastructure, often via a range of tools, and taking appropriate actions as a result. Current solutions do little to relieve the metaphorical problem of seeing the wood for the trees in security operations.

There are two principal deployment options, as depicted in **Figure 2**, of which one or both may be used for a particular enterprise setting, as part of a complete solution to Security Operations.

In broad terms, these options are distributed or centralised.

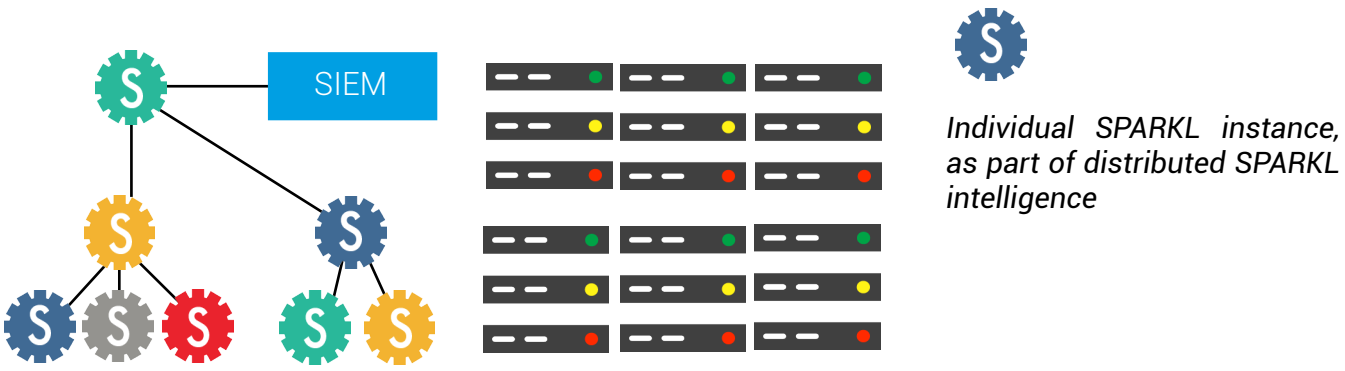

Figure 2: Hybrid Distributed and Centralized SPARKL Deployment

For the distributed option, multiple instances of SPARKL are deployed throughout the enterprise infrastructure, where events are handled as locally as possible to specific SPARKL instances.

SPARKL autonomics occurs locally to respond to a particular incident in order to effect an appropriate response. This removes the necessity for reliance on a central orchestration point and improves the resilience of the overall solutions. It also reduces complexity in that event streams can be tapped locally and do not necessarily require recording and aggregation in a centralized store.

This approach is based on SPARKL's local and hierarchical distributed intelligence approach, which effects autonomics (for detection, investigation and mitigation) as close as possible to the source of events.

For the centralised option, it would be common to deploy to SPARKL alongside a SIEM (Security Information and Event Monitoring) solution. The benefit is that the SIEM can be used for event aggregation and correlation across a number of event sources, across the breadth of the enterprise, which would provide alarms to SPARKL concerning the occurrence of multiple events that together may be indicative of a security incident.

This correlation may only be possible when the centralized option for SPARKL deployment is used. SPARKL would then be used to oversee appropriate investigative and mitigation actions that are captured as workflows and should be followed. SPARKL may also feed its own events, resulting from execution of these workflows back into the SIEM for presentation in the SIEM dashboard and SIEM reports.

Role of SPARKL in SIEM

Quick Incident Response

The use of SPARKL for Security Incident Management improves typical response times for handling incidents. Security analysts or autonomics will execute pre-defined detection, investigative and mitigation (DIM) workflows for certain attack types, rather than an analyst needing to manually instruct individual steps.

As a consequence, incident response is a process that may take seconds rather than minutes or hours whilst an analyst decides an appropriate response. Currently, the execution of these workflows with disparate types of both manual and automated steps is an aspect of security operations that is not effectively automated.

Workflow for Incident Response

With SPARKL, workflows for Incident Response may be authored using a convenient GUI. Workflows will consist of steps of disparate types which involve potentially many different types of (primarily) IT stakeholders. These workflows are crafted for the specific needs of the enterprise in question, although they may be based on generic templates.

An enterprise will typically have several policy documents which need to be enforced in DIM workflows to ensure that any violation of policies by an attack is swiftly dealt with. Overall, requirements will be driven through a combination of industry standards, government regulations, accepted best practices, and organizational policies that reflect these.

Although some policies will be common to most enterprises, others will be specific to the particular sector of the enterprise. For instance:

- In **healthcare**, privacy requirements regarding protection of patient information are in play;
- In **financial services**, credit card information may need to be particularly protected, and banks may have controls to safeguard against fraudulent transactions.

Workflow steps may include:

- Queries to run against systems when events of particular types occur;
- Automatically making changes to a managed system, in mitigation;
- Gathering input from an analyst or some other role in the organization such as escalation to higher-level management or change boards;
- Feeding information to a ticketing system etc.

By explicitly capturing DIM workflows, we record the wisdom of senior analysts in a form that can be automated, where, if human input is needed, often it can be provided by junior analysts, which then relieves the burden on senior team members.

The execution of a DIM workflow may, in part, resemble traditional case handling, where the security analyst or knowledge worker is guided to the detection, investigative, or mitigation goal, rather than being prescribed absolute steps for getting there, thus offering additional flexibility regarding the resolution of an incident.

Appropriate Incident Response workflows are derived from answering core questions, such as:

1. What are we trying to protect? What are the risks to my business?
2. What are the threats to my assets, based on the assessed risks?
3. How do we detect incidents based on these threats?
4. How should we respond to incidents?

SPARKL can help with the automation of procedures developed as a result of answering the last two, but it also can be used in helping to answer some of these questions, as described below.

It is intended that SPARKL supports process mining. Often in legacy enterprise integration solutions, it is unclear which machines and system processes running on them, are of key business importance.

SPARKL can be deployed to mine the typical interactions between system end-points, that is to realise mining of business processes, in order that:

- the infrastructure of key importance is identified; that is, we answer the question of what we are trying to protect (question 1);
- we identify what constitutes as anomalous system behaviour, which lends itself to answer the question of how we detect incidents (question 3) - the answer to which leads to steps in a detection workflow that is also automated by SPARKL.

The larger and more complex an enterprise, the more overhead is required for the necessary exercise of capturing assets and processes, in order to understand where threats to security are likely to lie. It would be an advantage to automate this preparatory phase, as much as possible. Process mining is scheduled on the SPARKL technical roadmap for development.

Security runbooks have been in place in enterprise IT for many years. However, there is typically next to no automation in enforcing them. In SPARKL-based Security Incident Management, an enterprise security runbook is principally comprised of plays, each of which describes the DIM workflows that are appropriate to a threat that has been identified in answering the core questions, given previously.

In having explicitly captured security plays, we are able to show readily how incidents are handled if and when they occur. This will go a long way toward showing due diligence, when the auditors come calling, as well as reassuring the business that their assets are being protected in mitigating the risks that have been identified.

SPARKL Provenance for Data Compromises

When enterprise data is compromised, effective mitigation can only be carried out by understanding both the owner of the data and other parties who hold a vested interest in the data (such as those who have contributed to the data, in its current form).

SPARKL Data Provenance mechanisms allow for the tagging of data with information (meta-data) concerning the evolution of the data over its lifetime (its provenance), in terms of the system and people who contributed to or manipulated the data since its inception. Other pieces of meta-data may include the owner, and also other stakeholders who hold some vested interest or privilege over the data.

Confidence Scoring Example

Consider the following example where SPARKL is being used with other tools to perform confidence scoring of security threats based on host and network fingerprints, such as duplicate system processes running on a host or traffic on network, and to take appropriate actions. Security Management processes and controls may define very specific attack scenarios as well as general procedures for incident handling.

In Figure 3, we see that an event concerning the detection of a duplicate system process has been defined as `RaiseSystemProcessDuplicate`. Duplicate system processes on a host can be indicative of malware. Perhaps this is a common fingerprint at the time in question for a particular type of attack.

As a result, the SOC defines an orchestration, or what SPARKL calls a 'mix', specifically for this:

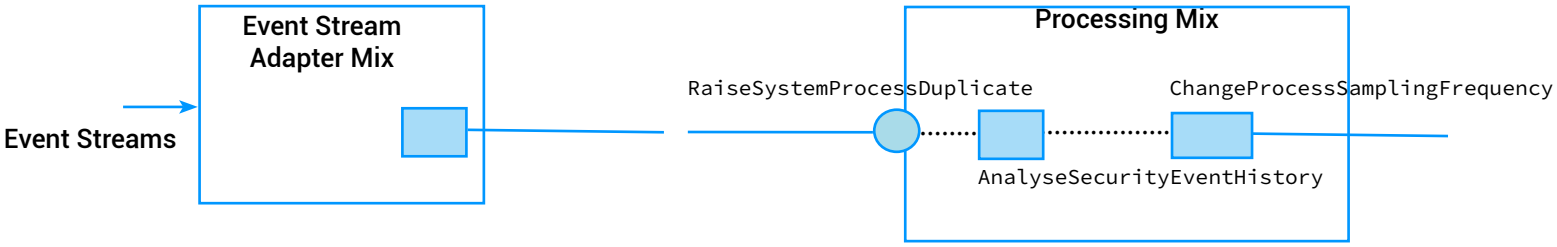

Figure 3: Handling duplicate process anomalies, using SPARKL and Ansible

When the `RaiseSystemProcessDuplicate` event occurs, SPARKL will perform planning (shown by the dotted lines) and ultimately may call `AnalyseSecurityEventHistory`. This operation will determine what course of action should be taken and through some confidence scoring, by example, may decide that all hosts in a group of webservers should report their process dumps at ten times the frequency.

This could be done by using the Configuration Management tool Ansible for instance, which is well suited to pushing machine changes in parallel.

This change is carried out by the `ChangeProcessSamplingFrequency` operation, as shown in Figure 2. Another side-effect may be to send a message to an administrator over Jabber/XMPP, say, in order that the administrator should decide an additional mitigation.

SPARKL's support for XMPP interactions as part of orchestrations is evident in the following screenshot, where an administrator is informed by SPARKL of a problem.

Confidence Scoring Example

SPARKL's support for XMPP interactions as part of orchestrations is evident in the following screenshot, where an administrator is informed by SPARKL of a problem.

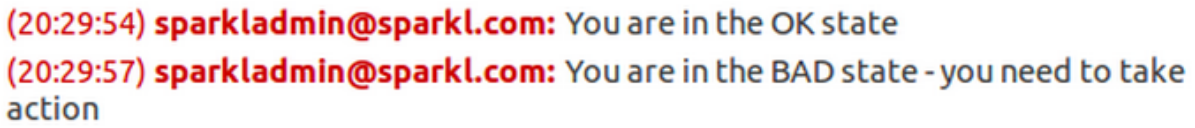

Figure 4: SPARKL sending state updates over XMPP - more details would follow; the user could invoke mitigations over XMPP via web etc

In orchestrating service artefacts, SPARKL records its own action events as well as environmental events to an analytics database. This can be queried using a SPLUNK-like web-based or CLI-based tool in order to elicit insights from these event streams.

This capability thus brings an analytics capability to SOC operations, out of the box. For example, the following query (Figure 5) would get the values of the `frequency` data field (in times per hour) that system processes are sampled, as per the previous example:

```
                                                    2016-06-07 14:17:51
andrew@sparkl.com/desktop:
query frequency=* op=ChangeProcessSamplingFrequency | table frequency sparkl__
timestamp as timestamp
(14:17:52) sparkladmin@sparkl.com:
+-----------+--------------------+
| frequency |          timestamp |
+===========+====================+
| 60        | 1441580462.841065  |
+-----------+--------------------+
| 600       | 1441580612.353648  |
+-----------+--------------------+
```

Figure 5: SPARKL sending state updates via an XMPP session

SPARKL Developer Console

SPARKL orchestrations can be dynamically edited directly through a web-based console. To get a feel of how SPARKL works, you can download and run pre-built mixes (SPARKL-speak for configuration) on the Developer Console.

To learn more, click here to go to SPARKL's User Guide, where you can read how-to guides, tutorials and solution briefs.

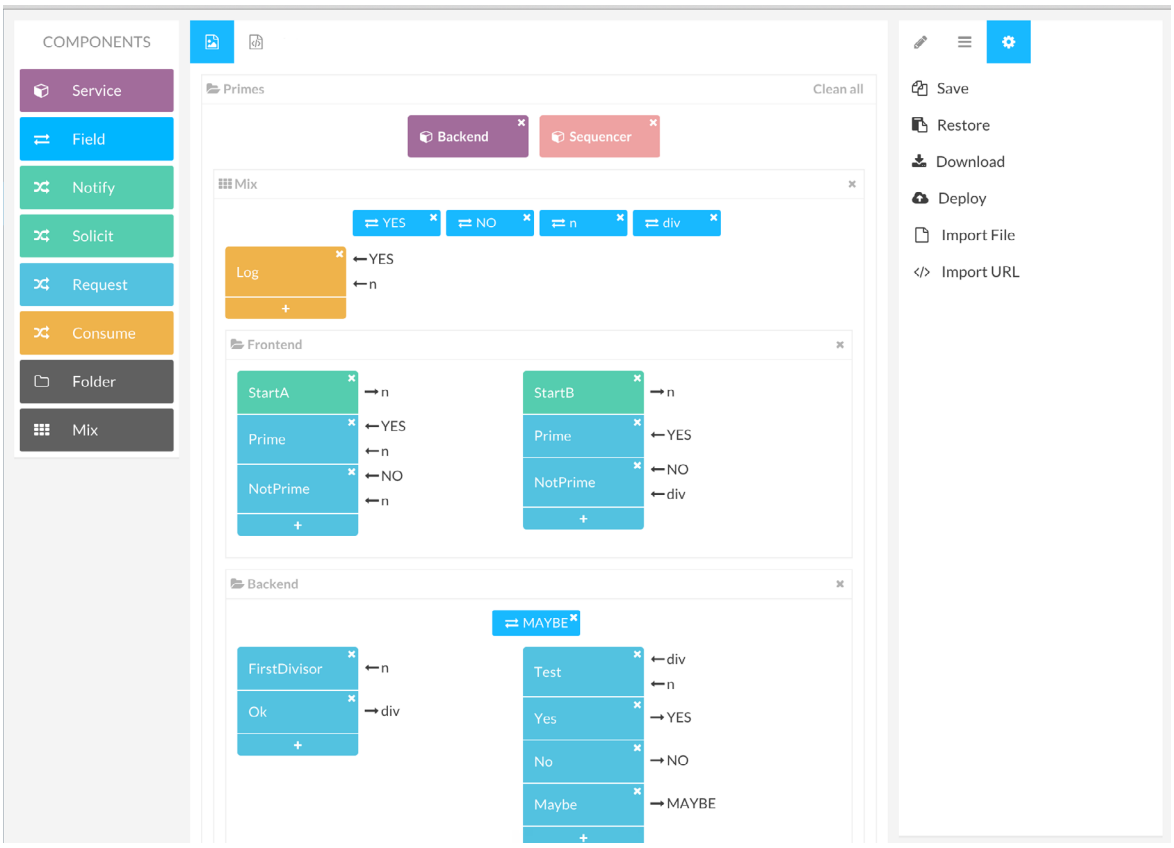

Figure 6: Browser-based editing of mixes

Find Your Use Case

Mark Dawber
Head of Business Development
mark@sparkl.com

sparkl.com
@sparkl

See SPARKL tutorials and demos at
sparkl.com/docs/web