State-full risk assessment & automated security incident policy enforcement

v0.3.0

Institute For Security & Open Methodologies
(isecom.org)

Rob J Meijer <rmeijer@xs4all.nl>
&
Rick Tucker <krutec@easystreet.com>

Mar 3 2003

Abstract

This document addresses the main problems in current incident response handling procedures. It outlines how a tighter integration of risk assessment, technical security measures, and incident response policies can lead to more effective incident handling. The case is also made to expand the definition of *incidents* to include any event that might affect the security state of a software product or an infra-structural component. Finally, the document advocates implementation and standardization of an information exchange format that integrates security event, security policy, and event handling tracing information in a way that respects the needs posed by the required stochastic modeling of risk.

1 Status

This document is a work in progress, this current version is the first draft version that is being submitted for peer review within ISECOM. For this reason any misinformation that might be contained in this document currently only reflects the opinions and/or level of knowledge of the main authors, and should in no way be considered to be in sync with the quality of peer reviewed ISECOM documents.

2 License

3 Introduction

This section introduces current issues regarding security policies and their enforcement possibilities. It illustrates factors needed to create a security incident policy enforcement system (SIPES), and presents the subsequent benefit of latency reduction to the field of IT security. It demonstrates how a SIPES could elevate the level of IT security world-wide, while keeping company incident costs down to an acceptable and predictable level.

3.1 Conflicts in the definition of failure

One of the most important issues in IT infrastructure is the prevention of so called *single points of failure*. This is a valid and important issue, but there are issues with how failure is perceived. Its interpretation has become an obstacle in the implementation of effective infra-structural security measures. To expand on this idea, the two core types of failure must be defined and discussed:

- Failure of the *authorized* availability of resources.
- Failure to prevent *unauthorized* availability of resources.

All too often, legitimate availability of resources is associated with a risk, or even acceptance, of unauthorized availability of those same resources. Conversely, denying unauthorized availability can lead to legitimate needs being unduly restricted. The conflict is fundamental. The desire to make resources available while preventing them from being compromised is the main issue in security measure implementation. The most important aspect to realize is neither type of failure can be completely *prevented*. The resolution of this issue lies in choosing the *least undesirable* form of failure within every conceivable situation, and doing so swiftly enough to be effective.

3.2 State-full risk assessment

The key in determining which way to fail is based on a *risk assessment*. Risk assessment, in its 'common practice' application on IT infrastructures, is mostly done in a rather stateless and static way. In other words, dynamic technical factors are de-emphasized. Hence, many diverse conditions, or states, in a technical environment are not considered when performing the risk assessment. Risk, however, is dependent on many components that are not stateless in nature, but are dependent on the security dynamics of all subsystems involved in the risk assessment. Every involved subsystem has a dynamic *security state* that changes in time as a result of both internal and external events. Examples of these events are:

- Publication of a vulnerability in the subsystem
- Installation of a patch or newer version of the subsystem
- Detection of unauthorized access to resources by an IDS (Intrusion Detection System)
- State changes in the security dynamics of connected or related systems

Dependent on the state of a subsystem, the outcome of the risk assessment will vary. The balance between the risk of the two types of failure, and costs that may arise from favoring one or the other, can change based on the state of a given subsystem. This dynamic environment presents a major issue that a stateless risk assessment does not address in security incident procedures:

Is a particular problem severe enough to warrant disabling access to the entire system?

The answer is elusive. In circumstances that might require swift action the wrong choice could have major repercussions. Acquiring the correct information is paramount to averting two potentially disastrous scenarios:

- The technical staff will take a 'shoot first' approach and will, based on purely technical considerations, decide to disable access to the entire system.
- The technical staff will keep the system intact and opt to formally send the issue through many delaying layers of management.

Both decisions, though made with good intentions, are inappropriate responses to the risks involved. The technical staff does not have the needed information to form a suitable response. Without instructions to the contrary, the technical staff will typically opt for more formal escalation procedures. Though not always wise, this is often perceived as a course of action that will prevent them from losing their jobs. If the technical staff is responsible for deciding how to respond to a failure, but has no clear directive from management, an organization's future can potentially be secondary to protection of an individual's career. This, in turn, can place both parties at significant risk. Entrusting the technical staff with such responsibility is clearly not desirable.

A solution must be devised to move the responsibility to a management level, but also reduce the latency involved with escalation. This would remove the high impact decision making from the technical staff and provide them with a viable framework allowing them to respond quickly and appropriately given the risks involved.

Ultimately, a security incident policy (SIP) based on *state-full* risk assessment (SFRA) needs to be created. This type of policy takes into account all the possible security states of all involved subsystems, combines these with the information from the stateless risk assessment, and produces a set of security states where the risk is more accurately identified.

3.3 State-full Threat Assessment

The foundation needed to create a State-full Risk Assessment, and also needed to extract usable configuration data for a SIPES, is a *State-full Threat Assessment*(SFTA). In a SFTA, the infrastructure's resources are not seen as targets. All resources in the infrastructure are interpreted as being only *Service Points* or *Proxy Points*.

A Service point is any point in the infrastructure that delivers some service to external and/or internal parties, surrounding components, and most importantly, to definable proxy points. A proxy point is defined as an environment that could become available through a service point by the use of some vulnerability.

Between service and proxy points there exists a barrier. The barrier inherently *resists* any intrusion or attempt to compromise it. Its resistance will degrade if efforts are made to overcome it. This degradation is quantified as degrading state-full resistance (DSFR). Its value is dependent on the 'state' of the service providing subsystem, and will degrade over time with a specific speed for that state.

The DSFR is the fundamental component of the SFTA. It is synonymous with the original idea of Risk Assessment Values (RAVs) outlined in The Open Source Security Testing Methodology Manual (OSSTMM) [1], but is limited to the threat assessment phase of a SIPES. The SFTA incorporates the following tasks:

- Inventory of all service points in the infrastructure
- Inventory of all possible proxy points in the infrastructure
- Inventory of all DSFRs and identification of duplicate components (duplicate components will have security behavior within the State-full Resistance network that is specific and difficult to model)
- Determination of all possible states to which State-full Resistances could be exposed, and the initial resistance and degradation factor of these states.
- The determination of timeout sub-states defined by pre-determined degradation levels within a state.

Although the itemization of tasks is straight forward, the practice of using duplicate components in different areas of the infrastructure is a complicating factor. The collective resistance of those duplicate components is equal to the resistance of just one component, so only one exploit is needed to compromise all of them. This significantly complicates the mapping and assessment of all the paths an adversary can utilize. One method of addressing this issue is to introduce somewhat crude calculable correction factors. Although this method is far from scientifically correct, it does deliver a reasonable result.

3.4 From state-full risk to incident policy enforcement

The risk to resources changes dynamically due to different events that affect the security state of IT infrastructure components. Different risk levels will warrant different decisions regarding the availability of resources. Using a SFRA provides the foundation for creating a strategic availability policy. This policy can be used to determine security dynamics that may warrant disabling an entire system or re-enabling a disabled system. In a partially automated system such a policy could substantially aid in enforcement of resource availability. Since determination of *disabling a system* is mostly done in the context of incidence response, such an automated system can be called a "Security Incident Policy Enforcement System" (SIPES).

[1] http://www.osstmm.org/

3.5 Threat to external resources

Enforcement of SFRAs and SIPES may lead to an increased liability for actions/inactions. This enforcement is straight forward within an organization. Any necessary calculations or communication regarding a breach of protocol can be dealt with relatively quickly and efficiently. However, the possibility of an external party experiencing fallout from an internal breach of protocol, such as a proxy attack, is a definite reality. Subsequently, any potential damage to external systems due to policy-based availability of compromised internal resources needs to be accounted for in the risk assessment.

Ultimately, responding appropriately to a compromised system, or just a vulnerability, will become an action based on a strategic management decision. Management will cease to be an *unwitting accomplice* when a compromised resource is used to attack external systems. It will, in many cases, become a *policy-based accomplice* that is partially accountable for any damage to external systems.

3.6 Extending the definition of incident

For most companies a security incident begins when there is a reasonable assumption that a compromise is in progress, or has already happened. The security state of a subsystem, however, can transition to a state that is 'likely vulnerable', or even an 'exploit publicly available' state, before the subsystem is actually compromised. Given this reality, the incident response policy derived from a SFRA should be initiated well before an actual compromise takes place. This can even go so far as to trigger an incident response because of a mere rumor of vulnerability.

In addition to security incidents that occur as a result of external threats and/or internal vulnerabilities, there are events not traditionally considered 'security' events that may affect resource availability. Such events might include, but are not limited to:

- Thresholds being reached in disk space
- CPU usage on a system that is responsible for availability of particular resources.

In order to create a partially automated system that can enforce a SIP, the definition of an incident needs to include any event that might negatively influence any of the two forms of resource availability.

Security events that positively influence resource availability, or might or (temporarily) alter the way incidents are evaluated must also be recognized. An example of the latter could be a so called *maintenance window*. Within a maintenance window of a subsystem the unavailability of a resource will be interpreted in a very different light than this same event outside of the maintenance window.

3.7 Default actions and timeouts

Failing to take (timely) action is perceived by many as more acceptable than prematurely taking an explicit action. Subsequently, there is a tendency toward

inaction. This can ultimately result in unneeded escalations, even in cases where policy dictated clear preemptory action.

To keep this issue from surfacing at critical points in the execution of the incident response policy, it is necessary to prevent the "do nothing" action. To enforce a security policy, inaction needs to held accountable. This could be implemented by creating an automated default "start timer" when action by a person in a specific role is needed. If the person takes no action, or does not respond within a specified time frame, an automated response occurs as a result of the *accountable* inaction.

3.8 Organizational entities and role actions

There are two general types of entities that fulfill specific roles in an event driven security incident policy (SIP): Organizational and Technical entities. Many of the entities may implement their own event driven SIPs. Those entities, or their semi-automated SIPES, will need to exchange information with related entities regarding role-specific actions.

Organizational entities can be divided in to two sub-types. The first sub-type fulfills different roles on behalf of a company from either inside or outside a company. Some of the roles can include, but are not limited to:

- Software end-user
- Software vendor
- Computer Emergency Response Team (CERT)
- Security research institute
- Independent hacker

The second sub-type is purely internal to a company. It fills a specific role in the incident response procedure. For example:

- Reverse-engineering specialist
- System Maintenance
- Screener of all outgoing role-actions

Many of the events relevant to the SIPES are generated not by organizational entities, but by the technical infrastructure itself. Some of these roles could be:

- IDS systems
- Related state-machines within a SIPES (local and remote)
- System and network monitoring systems

In order for the SIPES to be able to have any effect, it will need to generate events to be handled by systems (for example, fire-walls) that could play a role in taking appropriate action based on status changes.

Given the number of entities involved, the first requirement in building a usable SIPES is to define a 'Role-Action' exchange format that can share and distribute information for all types of defined entities and their corresponding role-actions. In order to arrive at such a *standard*, it is essential to make a thorough inventory of:

- All possible *roles* that can generate events which can trigger state transitions within SIPES state machines.
- The types of Role-Actions each of these roles can generate.
- What types of Role-Actions can be sent between which roles.
- The parameters of each Role-Action that is relevant to its evaluation.

Since information will be transferred between both internal and external entities, it is essential to integrate a Public Key Infrastructure (PKI) into the Role-Action exchange format. This will provide necessary authentication of the sender of the role-action.

To allow for the greatest flexibility in infra-structural implementation, the exchange format should probably make use of XML. The format used should be able to contain the required PKI signing and identity information.

3.9 Approaching Reality with State Machines

To make the right decisions at the right time it is important to know, as accurately as possible, the external reality. Although 100 percent accuracy is desirable, it is not feasible goal. Relying on the representations that role-actions give about external realities, enough internal information is maintained about the external realities needed to create a SIPES.

The DSFRs as described in the *State-full Threat Assessment* section are a good representation of an approximation of the external reality of a subsystem in between points in time where the external reality and the internal representation are more closely *synchronized* by role-actions. The DSFRs can be seen as state-machines that have non-discrete behavior within their states. With this non-discrete behavior being more difficult to model, the approximation of DSFRs by a discrete state-machine that introduces sub-states with timeouts and discrete states with static state resistance values becomes the more viable implementation method.

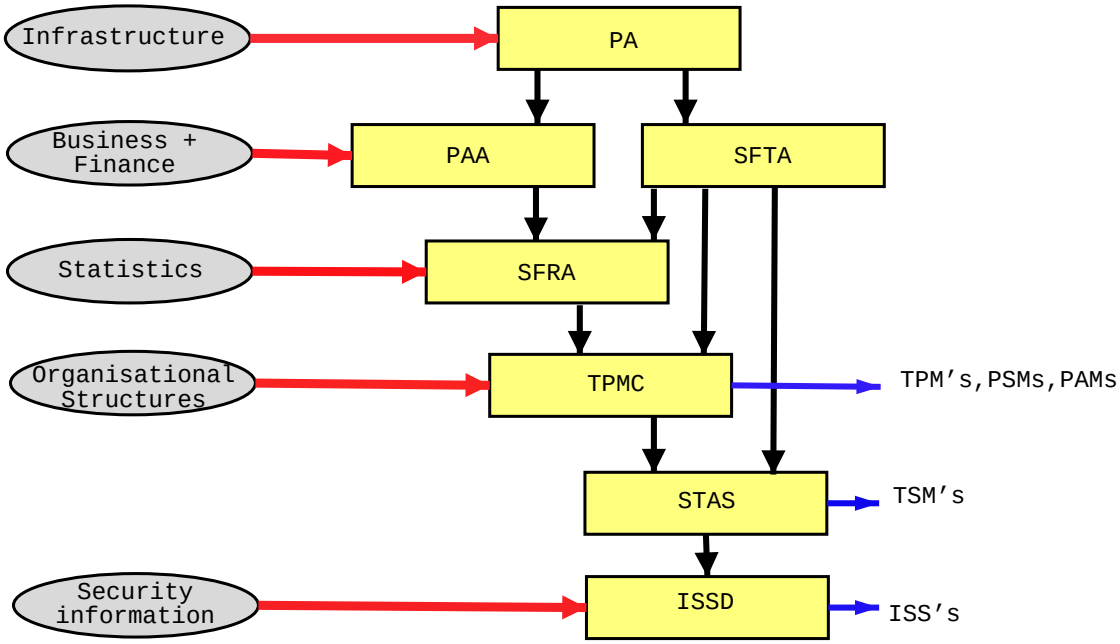
A *State Machine* defines a finite number of states for specific functional components within a SIPES. The state machine will change state as a result of input events. A SIPES defines two distinct types of state machines.

The first type is the *Threat* State Machine (TSM). It represents either a concrete subsystem or a virtual one that describes the software or hardware being used as a generic component. The combined TSMs try to form an up-to-date representation of the threats that are, at any specific time, present for the infrastructure.

The second type is the *Policy* State machine (PSM). It tries to be an up- to-date representation of the execution of the security policies by the organizational entities involved in its execution.

By combining a good implementation of TSMs and PSMs in an automated system, it will become possible to build an effective SIPES from this concept.

4 Creating enforceable policy



To create enforceable policy a number of actions are needed. This chapter outlines the steps to obtain the data needed for creation and configuration of a SIPES. The creation process is comprised of 7 steps:

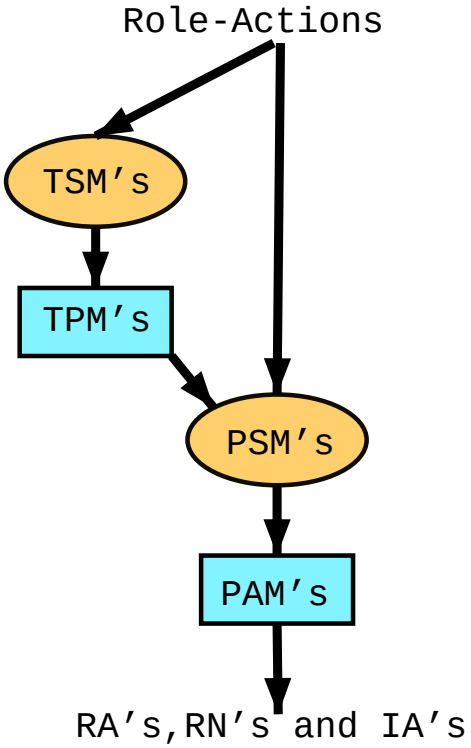
- *Point Assessment*(PA): Identify service and proxy points.
- *Point Asset Assessment*(PAA): The PA points are viewed from a business and financial perspective. All the points of economic worth are identified and valued.
- *State-full Threat Assessment*(SFTA): All the possible links between service points and proxy points are reviewed in detail, and DSFRs are identified. For all DSFRs the state machine variables are determined.
- *State-full Risk Assessment*(SFRA): A mapping sheet is created from PAA and SFTA data. It is combined with statistical information and used to calculate the risks of every potential state of the SFTA.
- *Threat/Policy Map creation*(TPMC): To create the SIPs, the SFRA and SFTA data is combined with information about the companies organizational structure. This will produce the following output needed by the SIPES:
  - *Policy Action Maps*(PAM): Maps policy state changes to corresponding actions. These actions include notification of roles, generation of role-actions to other SIPES, and control over plugs in the infrastructure.
  - *Plug locations*: The State-full Resistances that can be actively used by a SIPES in the usage of PAMs.

  - *Policy State Machines*(PSMs)
  - *Threat Policy Maps*(TPM): Maps SFTA to PSMs. Determines how the state of the PSM changes based on the state changes of SFTA components.

- *State-full Thread Assessment Simplification*(SFTAS): A process that uses the SFTA and TPMs to locate states, and even state machines, in the SFTA that are not having any effect on the TPM output. It will use this information to create a simplified set of Threat State Machines (TSMs) that only have the combined states that actually influence the TPMs.

- *Initial system state determination*(ISSD): A process that identifies a reliable starting point for a SIPES and determines the exact state information when the SIPES is activated.

5 Working with enforceable policy



A Security Incident Policy Enforcement System (SIPES) consists of 4 types of subsystems:

- *Threat State Machines*(TSM) : State machines that maintain state about the security of DSFRs as defined in the SFTA. The state of TSMs will change as a result of Role-Actions, or as a result of timeouts that are a fundamental part of each state with a security level exceeding absolute zero.
- *Threat Policy Map*(TPM): Maps TSM states to input events for the PSMs.
- *Policy State Machines*(PSM): The PSMs are the factual core of the SIPES. These state machines change state based on TPM events, Role-Actions, and timeouts.
- *Policy Action Maps*(PAM): Maps PSM state changes to corresponding actions that have external consequences. Actions that can be triggered by PAMs are:
  - Role-Actions to be sent to external SIPES.
  - Role Notifications: For example, e-mails to people with a specific role that notify them of the current state of a PSM, and prompt them to generate new Role-Actions.
  - Infra-structural Actions: Notifications to systems with the physical components that comprise crucial DSFRs, and that are able to take so called 'plug' actions.

5.1 Policy publication

To determine the appropriate action for a specific entity's Role-Action, entities may want to exchange parts of their policies. This would necessitate a standardized Role-Action exchange format and SIPES that are widely and effectively implemented by the involved entities.

For Vulnerability publications, there have been efforts to standardize policies for so called *responsible* disclosure by christey & wysopal [2]. Although the intent of this publication is admirable, the concept of forcing all entities to use a policy with limited flexibility seems impractical. A policy may be unenforceable, but those using a policy can be held accountable for it execution. By using part of the policy for a SIPES as indication of what other parties might expect their peer's actions to be, both parties can interact. Even if their policies do not match, they are able to evaluate their peer's policy and generate an appropriate Role-Action. If the implementation of a policy causes damages, there may be legal ramifications. Since an incident response policy is based on the risks involved, a review and/or alteration of the risk assessment will be required.

In addition to direct use in vulnerability reporting, end-users selecting and buying software may use published vendor policies as a factor in their product selection. Users of network services could, for example, use the information to determine if they trust sending their credit card information to such a service to purchase a product over the Internet.

The Role-Action and security incident policies should in their exchange format be closely related and probably even integrated into one single XML+PKI based standard.

5.2 Incident Traces

With both the policy and the Role-Action defined in XML, it should become possible for the state machines to maintain incident traces in a related XML format. Collectively, traces could potentially be used in a court of law, and sections of traces could be included in status reports to entities involved in the reporting or handling of incidents.

5.3 Selective obscurity

A final point of concern for many entities is the concept of *selective obscurity*. Integrating policies (at least the technical translation of the policies), Role-Actions, incident tracing, and reporting into a single standard, requires a degree of caution. Parameters should be defined to determine which entities should have access to certain portions of policies and incident traces. This will ensure that only authorized entities can access sensitive policy and incident trace information.

[2] http://sipes.sourceforge.net/draft-christey-wysopal/

6 Step one: creating a SIPES Description Language

For a SFRA and a SIPES to become become a usable way of handling infrastructural security, there are 3 things that need to be realized:

- A *SIPES Description Language* (SIPESDL)
- A *SIPES* implementation.
- A *SFRA & Policy CASE tool* implementation.

The SIPES project at sourceforge [3] is developing an open source implementation of a SIPES. At this point no initiatives exist for the creation of a SFRA&Policy CASE tool (open source or other). An SFRA&Policy CASE tool (for lack of a better name) is a graphical tool for creating enforceable policy using a SFRA methodology. The existence of such a tool could greatly help in the acceptance of SFRA, and thus in the usability of SIPES on a global scale.

The first step, however, is the creation of a SIPESDL, which should incorporate the possible requirements for both open and closed source implementations of the SIPES and the SFRA&P CASE tool. In addition, the data exchange between SIPES systems and/or SFRA&P CASE tools should be problem free.

The SIPESDL should, as described in previous chapters, be an XML-based format that incorporates PKI information. The PKI key and signature information is an essential part of the SIPESDL. Existing standards for PKI usage in XML should be of primary consideration.

The SIPESDL will encompass each step and every component in the SFRA process, every component used by a SIPES, and all communications between a SIPES and:

- a SFRA&P CASE tool
- other SIPES systems
- persons or organizations
- Network/system monitoring systems and IDSs
- systems capable of doing administrative tasks (like disabling an entire system)

Considering the many fields of expertise that a SIPESDL will touch, a SIPESDL document must present a standard that will bring consensus from many experts in the involved fields and sub-fields. This chapter outlines some of the essential points to be considered in the process of creating a SIPESDL document.

Some standards may have already been created in the various fields that a SIPESDL is attempting to encompass. The SIPESDL may be, in large part, nothing more than an extension of one or more of these existing standards, or a meta language that incorporates the existing (pre) standards.

An important point to keep in mind is that while existing standards may be fully valid for the concerning subfield of info-sec, the ultimate goal is to

[3] http://sourceforge.net/projects/sipes/

contribute to the effectiveness of a SIPES. If that goal is not achieved through complete inclusion of an existing standard, a modified approach will have to be implemented. Once a SIPES has been created from a SFRA process, it will be doing the actual SFRA calculations constantly, with changing parameters, and consuming large amounts of CPU time to keep the infra-structural safety at an optimum. The effectiveness of a SIPES relies heavily on the synchronization between the internal state-machine and reality to keep latency as low as possible. Therefore, the information received by a SIPES must be in an efficient, streamlined format so as to avoid prohibitively intensive resource allocation and/or possibly human intervention. The indiscriminate use of existing standards may not serve this end and should be thoroughly considered.

6.1 Essential point on RA Math

Risk assessment includes a substantial amount of mathematics that influence the SIPESDL definition. Discussion of the mathematical aspects will be largely conceptual to keep the document readable and comprehensible to the intended (wide) audience. The SIPESDL document will eventually need to include some specific examples of the RA math, but this falls outside of the scope of the initial document.

In (State-full) Risk Assessment many properties can seemingly be represented as some number, when in fact, they can only be correctly represented as a stochastic variable (SV). A SV describes the probability (Pr) that the true value of the variable falls within a specific range of possible values. The following equation illustrates this description: $\int_a^b f(x) = Pr(a < x < b)$, and $\int_{-\infty}^{\infty} f(x) = 1$. Rarely will a variable have an *exact number*, or be within a small percentage of that exact number. It is not unusual for a value many times higher, or even many tens or hundreds of times higher, than the mean value of a function to become a crucial factor in the RA process. Further, an important calculation in RA is the determination of the probability that the value of a SV is higher or lower than some other constant or variable. Without maintaining information about the stochastic properties of the variable, these calculations will become impossible or meaningless.

In addition, determining if events are dependent, and to what extent, is essential to the SIPESDL. The usability of the SIPESDL relies greatly on defining events as either completely independent or fully dependent on other events. If there is a partial dependency that is too large to safely ignore, it should be defined in some form of cluster entity that holds the dependency between its nodes that are fully dependent on the cluster but independent of each other.

The math of the SFRA is mostly built around a simple principle; there is a diverse group of external entities, of which some should have access to a particular asset, and others should not. Each group will have some amount of potential interest in an asset, will ascribe some worth to accessing it, and will invest some maximum amount of effort to reach that goal. Between the external entity and the asset there will be an amount of resistance put up by DSFRs. This requires a hostile external entity to invest some amount of effort. The risk to the asset for particular hostile groups of external entities is primarily determined by *the probability that the effort a single entity is willing to invest is equal or greater than the combined resistance value of the DSFRs between the*

external entity and the asset. Additional determinations of risk are:

- Cost or income (where income is defined only as cost) associated with an external entity accessing an asset.
- The dependence, or independence, of this cost on external entity profiles in repeated (failed or successful) access attempt events.
- The numbers of entities described by the different external entity profiles.

The first of these items might need a little clarification. Income will become cost when resources are unavailable at the time an authorized EE tries to access the asset and fails.

6.2 Important issues for data-modeling for the SIPESDL

To elaborate on the concepts in the previous section, the following sections describe some of the most fundamental data classes that should be defined for a SIPESDL. Detailing a complete set of data classes will be an extensive task within the SIPESDL creation process and fall beyond the scope of this document.

Two essential pieces of data needed by the SFRA are the *External Entity Profile* (EEP) and the *Asset Profile* (AP). While these profiles are independent of a SIPESDL, they augment its effectiveness greatly.

The EEP describes the characteristics of an external entity belonging to a group of external entities that can be uniquely identified by these characteristics.

The AP describes a set of characteristics of a group of assets that can be uniquely identified by these characteristics. An asset will possess certain characteristics that make it interesting to an external entity. Many, but not all, of these characteristics can be described as having some *worth*. Please note that the *worth* of an asset is explicitly defined as not being part of the asset profile.

Although EEPs and APs are not part of the SIPESDL, developing standards for these profiles is essential for a SIPESDL standard capable of describing reusable components for use in world wide security event information exchange. Without EEP and AP standards, evaluation of role-actions will continue to be a job that needs intensive human evaluation on the receiving side. The definition of a SIPESDL assumes the existence of standardized EEPs and APs, and builds on these to create a description language for security policy enforcement and security event exchange that will allow many of the human evaluation tasks to be done more efficiently. Furthermore, many tasks that now involve intensive human evaluation can be implemented by technical measures.

The following sections will try to describe some of the essential components the SIPESDL should provide description possibilities for, and specific concerns with respect to these components.

6.3 External Entity Stats (EES)

Where the EEP describes the common characteristics of single entities within a group of entities, the *External Entity Stats* (EES) will describe known (or extrapolated) statistical information about the entities as a group. The information from the EES will, in turn, yield information about single entities within the group.

The first essential set of information about an EEP is the probability density function (PDF) that describes the number of existing entities that adhere to a profile.

The SFRA process will require two important aspects of the profile also described in the EES:

- *Interest* that an EE might have in a particular asset.
- *Effort* that a hostile EE might invest to gain access to a particular asset.

Please note there is a distinction between authorized and unauthorized access. The *Effort* only applies to unauthorized profile instances.

Both *Interest* and *Effort* will be parameterized probability density functions, where the resulting SIPESDL components will likely be large multidimensional arrays (combined with some mathematical and notational tricks to limit the size of the description) that are able to generate the (PDF) for any given set of values for the implemented parameters. As Interest and Effort matrices can potentially become very big, and may be very similar for related profiles, it is essential that EEPs, and their statistics data sets, can be described as differentials from a common referenced object. The most important parameters or *dimensions* of an Interest or Effort PDF are (or might be, as they are all optional):

- The Asset profile ID
- The Asset worth
- The time (time of day/week/month etc)

6.4 Point Asset

In the Point Asset Assessment(PAA) an extensive list of asset objects is located and evaluated. The result of this assessment will be a small set of SVs, and some additional elementary information about the asset.

The Asset Object will hold two parameterized SVs:

- The *Worth* this asset might have to a particular EE.
- The *Cost* involved if a particular EE accessed this asset.

The *Worth* PDF is used as a parameter with the Interest and Effort functions of the EES. Worth might have the following parameters or dimensions itself:

- EE Profile ID.
- time (time of day/week/month etc)

The Cost function is a bit more involved and requires close attention to the dependence of cost events. For this reason, small dimension and profile parent/child relations should be used. The basic idea is to break up a single event into multiple events (those on profile instances and those on profiles), and to discard repeating events on profiles. It should then be possible to tackle the dependence of event costs to a large extent.

A further note on Cost is that it could be negative, and it might be involved with both a successful and a failed attempt. The *Cost* function will thus have the following parameters or dimensions:

- EE Profile ID
- Instance or Profile (the mentioned additional small dimension)
- Success or Failure
- time (time of day/week/month etc)
- time (Duration of (un)availability)

6.5 DSFRs Points and (Sub)System

The PAs and EES define the framework the SIPES uses to coordinate the staff and, to some extent, the infrastructure, to minimize costs resulting from different kinds of failure. To develop this framework, the combined resistance value of all components in between EEs and APs need to be determined. In the PA and SFTA, the infrastructure will be described in terms of Points and TSMs. Points and TSMs, and their respective graphs, should be objects definable in a SIPESDL.

(Sub)systems built from the atomic components of the SFTA should also be defined. Each SFTA of a particular (sub)system can potentially be re-used as a single components in an infra-structural SFTA. The definition of (Sub)Systems is essential in designing a SIPESDL that accommodates the process of reporting vulnerabilities and exploits.

The re-usability of subsystems in SFTA is essential for the usability of this otherwise extremely extensive task. Essential information a (Sub)System should be able to implement should be:

- Vendor identification.
- Software identification.
- Software version range identification.
- description (and identification) of the atomic and non atomic components in the (sub)system.
- description of the interconnection graphs of the components.
- Inclusion of other subsystems (for example libraries or modules)
- Usage of templates and overruling of default includes
- Usage of defines and ifdefs for configuration dependencies

6.6 Policy and organizational state-machines

A SIPES will not take direct action based only on TSMs, EES, and PAs. It will use this information as indirect input for *Policy* State-machines (PSM). These PSMs maintain state, with respect to the current SIPs, about the handling of security events by the responsible entities within the organization. There are no apparent points of concern about these state-machines and their descriptions in a SIPESDL that need to be addressed explicitly in this document.

6.7 Role-Actions, Action-Maps and StateActionMaps

Role-Actions are the encapsulation of any information that is inserted into a SIPES, thus the definition of Role-Action objects is the biggest part of the data exchange portion of a SIPESDL.

A Role-Action will eventually need to influence relevant objects in the SIPES. It is therefore essential that the SIPES be able to determine to which internal objects the role-action is relevant, and, if possible, which parameters influence which parts of the internal object in what way. This can be done in 3 ways:

- Either as is preferred the Role-Action will include identifiers that are used in the same form in the internal components of the active SIPES.
- The Role-Action defines fields that are usable as input for Action-Maps, that in fact do pre-processing on incoming Role-Action in order to identify the relevant internal SIPES objects.
- The least desirable way of processing is having the Role-Action inserted into a PSM for the sole purpose of getting it evaluated for re-insertion by a human.

Some of the Role-Actions might further be generated internally by State-machines or StateActionMaps. A StateActionMap (of what a TPM could be an example) defines sets of combined states or state-machine parameters that will trigger the internal generation of a Role-Action. Both Action-Maps and StateActionMaps should be defined in the SIPESDL. Some Action-Maps and StateActionMap may even need to become static parts of each SIPES if the Role-Action definition requires it. This could be likely if adherence to existing, non-fitting standards is needed.

For TSMs to become fully synchronized with all threat information, it is crucial that Role-Actions either contain, or refer implicitly/explicitly to, all available information relevant to the evaluation of theRole-Action in the SFRA process. If this information is stochastic in nature, it needs to retain this characteristic when it's communicated.

6.8 Role-Notifications

Conceptually, Role-Actions are input for the SIPES and Role-Notifications are output, but some Role-Notifications might be Role-Actions for some other SIPES. It might be possible to combine the two in the SIPESDL as one. However, this will have to be subject of further study.

6.9 PKI info for authentications and tracing

A Role-Action or Role-Notification will always include a PKI signature of the role entity or SIPES that generated it, and will always define the role from the entity that claimed to send it. Further, the Role-Action will always include the public key of the sender. In instances where an entity signature is present, the public key of the signer will also be included. It is important to note that such a signature will sign a key/role combination.

The use of signing and public keys in role-actions and role-notifications is needed for two reasons:

First, to verify the authenticity and trustworthiness of the source. This will allow the assignment of resources, or the taking of infra-structural actions based on the content of Role-Actions.

Second, to trace all actions to an event-log. If an incident response is handled poorly resulting in legal action, or a negligent employee is considered for termination, having an event-log that includes signed role-actions will very likely (if care is taken in the SIPES implementation) be admissible in a court of law.

7 References

This section tries to give some references to freely available documents and resources relevant to this document. By lack of better suitable references this list currently contains a link to a risk management document not quite in sync with current insights that however should give a reasonable feel for the subject, especially when combined with the information found in the introduction to probability. Next to this it also contains an ISBN number of a subject that does not appear to have a freely available source of information that is usable.

Australian Communications Security Instruction 33
Handbook 3: Risk Management
http://www.dsd.gov.au/infosec/acsi33/HB3p.pdf

Introduction to probability
Charles M. Grinstead and J. Laurie Snell
http://www.cs.jmu.edu/common/coursedocs/CS685netsecnew/COMNETIII/Probability/introprobbook.htm

Engineering Uncertainty and Risk Analysis
Sergio E Serano
ISBN 0-9655643-8-X

Responsible Vulnerability Disclosure Process
Steve Christey and Chris Wysopal
http://sipes.sourceforge.net/draft-christey-wysopal/

Extended Incident Handling (inch)
http://www.ietf.org/html.charters/inch-charter.html

Common Vulnerabilities and Exposures
http://cve.mitre.org/

Open Source Security Testing Methodology Manual
http://www.osstmm.org/

8 Conclusion

Creating a standard for automated security incident policy enforcement data exchange is an ambitious project. It will need the help and dedication of many security specialists, including CERTs, to become a usable standard. Management will be responsible for creating clearly documented policy which will minimize escalations and avert unnecessary expenses. The end result will be swift, policy-based measures capable of being implemented for any security incident.

9 GNU Free Documentation License

Version 1.1, March 2000

Preamble

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

9.1 Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is

released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

9.2 Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

9.3 Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

9.4 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- Include an unaltered copy of this License.
- Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties – for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

9.5 Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

9.6 Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

9.7 Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

9.8 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections

with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9.9 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9.10 Future Revisions of This License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software

license, such as the GNU General Public License, to permit their use in free software.