



Product description



2012

D-Man Distributed monitoring for e-Business Environments

Traditional monitoring systems don't fit very well e-Business environments. Because they were initially designed to monitor internal processes and applications mainly from a technical point of view, they don't pay too much attention to infrastructural security and the end-user's perspective.

The first aspect is reflected by the heavy frameworks on which many of these systems rely. These frameworks tend to use protocols that are not supported within a DMZ or across firewalls. The latter aspect might result in e-Business environments of which the isolated tiers are in perfect health, but where the service as a whole doesn't behave as expected.

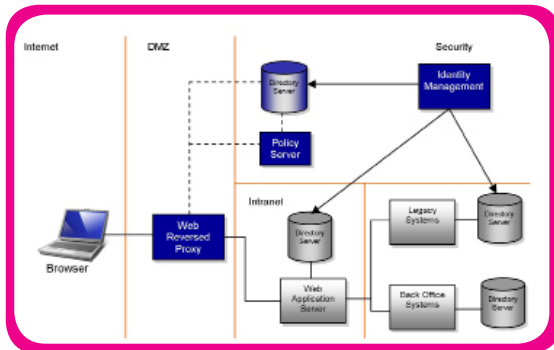
SecurIT has designed and built a monitoring solution for Distributed e-Business Environments, from the ground up, with exactly these aspects in mind. It addresses both, helping IT-Administrators isolating and solving IT problems appropriately as well as end-to-end Monitoring of business applications to identify service outage before it effects the end-users.

This solution was released by SecurIT under the name SecurIT D-Man in 2004.

Product description

Distributed e-Business Infrastructures

This paragraph briefly sets the scene by describing the high-level architecture of a typical e-Business environment. The following figure illustrates such an environment.



It should be noted that this figure only serves as an example and that SecurIT D-Man is by no means restricted to only this e-Business architecture.

An e-Business environment is characterised by four main aspects: Applications, Security Zones, Security Components and End-users.

With Applications we refer to legacy based systems (e.g. mainframes) and back office systems (e.g. SAP, Siebel). In an e-Business environment these systems are usually made accessible to the end-user by hooking them up to application server based portals. These could either be J2EE or .Net based (or both). While there clearly is a tendency to standardise applications on LDAP for user management, there will always be distinct and proprietary repositories and as such we will continue to find a heterogeneous collection of directories in e-Business environments.

It is common practise to apply layered security within e-Business environments by dividing the infrastructure into different Security Zones. Often, outer zones are said to provide more security than inner zones. This is however a misconception. Each of the zones provides a different level of security, protecting against the risks that this zone is vulnerable for. The idea of outer zones is to reduce the number of risks that may apply to inner zones. End-to-end security can only be guaranteed if each of these zones take full responsibility over security for its part of the e-Business infrastructure.

E-Business environments also host a number of Security Components. These mainly fall into two categories: prevention components and enabling components. The first category consists of anti-virus and intrusion detection systems. In the second category we find Identity and Access Management systems. While all these components are very important in any e-Business Environment, enabling components are crucial as their

failure could cause the whole e-Business environment to come to a stop.

Last, but not least, there are the End-users. E-Business environments are mostly targeted at large communities of external (internet) users. Because these communities are so big, and often also spread over different time-zones, any hiccup of the E-Business environment will usually have an impact on active users.

This high-level overview of a traditional e-Business environment clearly shows that such a system consists of several interacting components (legacy systems, repositories, application servers, security zones, security component) that each could cause the overall system to fail. As such, a monitoring system is only suitable for e-Business environments if it is able to check the health and status of each of these components individually as well as their interaction.

On top of this, it should be able to do this pro-actively and from the perspective of the end-user so that any problem is noticed before it can harm the end-user. And finally, it should be in line with the security policies that are inherent to the security zones of e-Business environments.

The following paragraphs describe the major design criteria behind D-Man. It explains why D-Man suits the monitoring requirements expressed in this paragraph and what makes it the most suitable system for e-Business monitoring on the market.

The SecurIT D-Man Monitoring Framework

The D-Man monitoring framework consists of two main components:

- D-Man Application Manager
- D-Man Data Manager

The D-Man Application Manager (DAM) represents a traditional monitoring agent. It is used to gather information from any component in the infrastructure that needs to be monitored. Typically there will be one DAM residing on every single server in the e-Business environment. A DAM is responsible for monitoring any activity that is relevant to the server on which it resides. This could be local as well as remote activity.

Provide an easy-to-use, flexible and comprehensive monitoring solution for highly distributed e-Business infrastructures. The monitoring solution should not be part of the problem, but should provide a fast track approach to problem solving without overloading the administrators with false alarms.

Product description

The following example illustrates some aspects of an e-Business environment that can be monitored by a DAM residing on a server within the DMZ

- Availability of the process running a Secure Reversed Proxy
- Response time of a remote Directory Server (e.g. LDAP) when validating a username/password authentication by the Secure Reversed Proxy

The D-Man Data Manager (DDM) sits at the core of the D-Man monitoring framework. On one side it is used for the administration of the D-Man environment while on the other side it is responsible for collecting and analyzing the data that is gathered by all DAMs in the infrastructure.

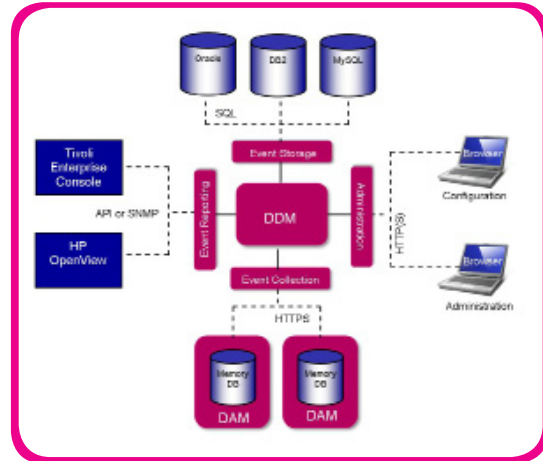


Figure 1 illustrates how the D-Man monitoring framework can be integrated within the above-shown e-Business environment. Typically there will be multiple DAMs within a D-Man environment but only a single DDM. A stand-by DDM running on it.

Cloud Monitoring

D-Man also provides a centralized way of monitoring multiple local D-Man environments. These could be entities within a large corporation or Cloud, but can also be used to monitor multiple customer environments from the Cloud. D-man provides a number of important features in order to fulfill the specific requirements in a cloud-based environment, like Scalability, Multi-domain and Multi-tenant support, Security and Deployment.

For more information about these specific environments visit our website and download the Solution Briefs (www.securit.biz/DMan).

The D-Man Monitoring Concept

The main responsibility of the DAM is running D-Man monitors and caching the data that is retrieved by these monitors. On its own the DAM contains no in-

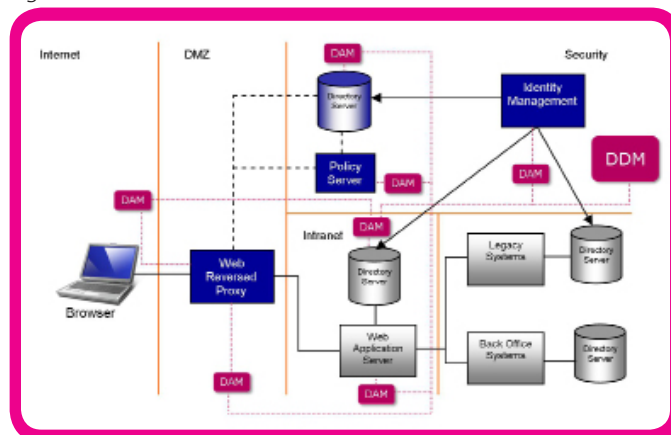
formation about the environment. All the knowledge it needs to run its monitors and to retrieve their output is sent to it on a regular basis by the DDM. The DAM caches all monitors and retrieved information in a memory-based database. This means there is no need to keep any configuration or event data on disk. The major advantage of this approach is that the DAM can be installed on a target system without any information about the system or the components running on it. The configuration will be pushed by the DDM to the DAM at runtime.

As the central component of a D-Man environment, the DDM exposes four interfaces: Administration, Event Collection, Event Storage and Event Reporting. These interfaces are described in some more detail hereafter.

Administration

It is the responsibility of the DDM to make sure that all DAMs in the environment behave as expected. More precisely this means that it is up to the DDM to provide any DAM with all the monitors it needs to observe the status of the e-Business components that are running on the same server as that DAM. E.g. if a server is running an LDAP server, the DDM will make sure that the DAM residing on that server will get the latest versions of all relevant LDAP monitors.

Figure 1



This is achieved by means of a memory based database that is integrated within the DDM. Within this database the DDM keeps an abstract representation of the monitored environment. At startup this information is loaded by the DDM from a configuration file. This information can be maintained either through a secure web-based interface (HTTPS) that is exposed by the DDM or directly within the configuration file. This latter feature is extremely interesting in larger environments where the configuration of D-Man is often controlled by batch processes. Finally, D-Man provides hooks that allow auto-discovery of new components within the e-Business infrastructure.

Product description

Event Collection

Event Collection is the key functionality of D-Man. It happens at three levels: Monitors, DAMs and DDM. The Monitors gather the events from the systems or application they are monitoring and hand over this information to the DAM that is operating the monitor. The DAM immediately analysis this information and marks the events according to their severity. Subsequently the DAM stores this information in its memory based database.

On a regular basis the DDM polls all registered DAMs and collects the stored events. At this stage the DAM can clear the events from its memory database. Before the DDM add these events to the persistent D-Man store, it applies the relevant correlation rules. D-Man correlation is described in more detail further on in this document.

Event Storage

The DDM stores all events it collects from the DAMs directly into an SQL based event database (also referred to as persistent storage). On top of this it runs a number of parallel correlation processes that further analyse this raw data. The result of these correlation processes is also stored within that same database. Out-of-the-box the DDM provides an embedded MySQL database, but through a pluggable interface this database can be replaced by most common SQL databases (e.g. Oracle, DB2, etc.).

Event Reporting

All the information that is stored within the above-mentioned event database can be made available in three different ways: Secure Web-based interface, Event database and Pluggable interface. These three interfaces are described in more detail hereafter.

All event data can be made available through a DDM embedded Secure Web-based interface. This interface can be accessed from any browser. Through a configurable hierarchical view it allows an administrator to catch any issue at a glance and drill-down on the root cause of the problem. It is also possible to customize these views (e.g. provide a view on the status of a cluster of reversed-proxies or LDAP servers).

Of course, all raw and analyzed event data can also be retrieved directly from the Event database by any SQL enabled application. This is especially interesting for batch driven reporting and statistics tools.

Finally, the DDM provides a Pluggable interface that can be used to integrate directly with external reporting tools. Out-of-the-box the DDM provides connectors that allow integration with Tivoli Enterprise Console (using the native TEC adapter or SNMP), HP OpenView (through SNMP) and mail systems (SMTP).

This interface is however published and as such other connectors can be provided both on a product end services basis.

D-Man embedded security

Because D-Man was designed to be deployed within sensitive e-Business environments where security is a key concern, it is obvious that it is based on a secure core as well.

By default the communication between the DDM and the DAMs runs over HTTPS using mutual authentication. For security reasons the communication between the DDM and the DAMs is always initiated by the DDM. Because the DAM may be located in a DMZ it is not recommended to open channels towards the secure Intranet. Furthermore the DDM will only fetch information from DAMs which it has configured and the DAM will only accept requests from the DDM that has configured it.

D-Man Abstract Data Model

As mentioned above D-Man creates an abstract model of the environment it will be monitoring. This model is represented as a hierarchical structured document in which each node represents a server, application or application aspect that needs to be monitored.

The D-Man data model recognizes the following types of nodes:

- D-Man Monitor
- D-Man Agent
- D-Man DAM
- D-Man DDM

A D-Man Monitor focuses on a particular aspect of an application. E.g. it can look at the availability of a Directory Server or the response time of a Secure Reversed Proxy. D-Man recognizes three types of monitors:

- Script Monitor
- Embedded Monitor
- Semi-embedded Monitor

The most common type of monitor is the Script Monitor. A Script Monitor is written using a scripting language (like Unix Shell, PERL or Windows Management Instrumentation) that is supported on the platform on which the monitor should run.

An Embedded Monitor is a monitor that is included in the DAM and which cannot be modified, only configured. A typical example of such a monitor is an HTTP client that requests a URL from a web server.

Product description

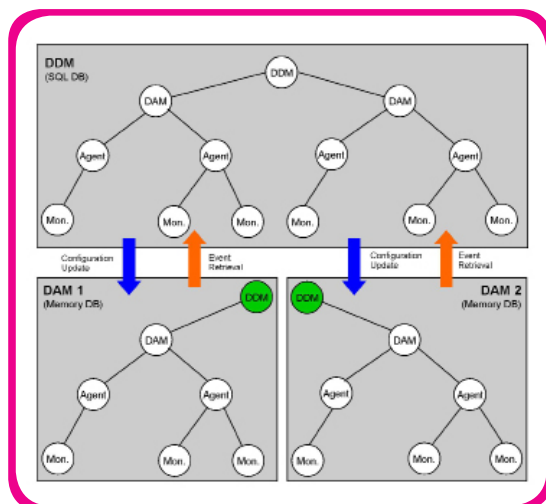
Finally Semi-embedded Monitors are scripted monitors that rely on embedded functionality. A very popular semi-embedded monitor is the LDAP monitor. It relies on an LDAP client that is embedded within the DAM. Through scripting, however, it allows to control the LDAP requests that will be sent and to dictate how the response should be interpreted.

A D-Man Agent focuses on a particular application. It creates an abstraction of the application by grouping together a set of monitors that are relevant for that application. For example a Web Server Agent will typically hold monitors for checking the availability of the web server, response time, memory usage, etc..

Apart from the technical role of a D-Man DAM it also has a more logical function. Basically the D-Man DAM is an abstraction of a particular server in the environment. It groups together all D-Man Agents that are required to monitor the applications running on that server. However, the scope reaches beyond that of the server's boundaries as it also looks at external dependencies that can have an impact on these applications. For example a D-Man DAM running on a Secure Reversed Proxy in the DMZ might also hold D-Man Agents for monitoring the availability of an LDAP server on which it relies for authentication.

Finally the D-Man DDM, while being the technical root of the D-Man monitoring framework, logically groups together all D-Man DAMs of the environment. As such it is the entry-point for administration of all other nodes in the environment and for extracting and analysing monitored data.

The following figure shows a snapshot the D-Man abstract model and some example nodes.



The above model represents an environment consisting of 1 DDM and 2 DAMs. The DDM will send configuration update information (e.g. new Monitors) to the DAMs registered within the same domain.

On the other side, the DDM will retrieve information

gathered by monitors (events) from the DAMs on a regular basis. Both the DDM and the DAMs will store all information in their respective databases (DDM: disk-based SQL database, DAM: memory-based database).

Basically a DAM will only hold its own information in the local memory-based database, though it knows its location within the overall structure. For this reason a DAM also holds a virtual entry for the DDM to which it belongs (see greyed node).

D-Man Event Analysis

In the previous sections we highlighted the D-Man framework and data model that serves as the foundation of the D-Man monitoring solution. While these are very important aspects of the D-Man architecture, they only serve a higher goal.

The D-Man Event Analysis interface was designed with only this goal in mind. Instead of looking at monitoring from a technical angle only, D-Man primarily takes the perspective of the end-user.

Clearly, in an e-Business infrastructure, the perception of the end-user defines the success of the system. Long response times, unexpected results and interrupted sessions increase the end-user's frustration. However, on the other side, system hiccups of this kind are in most cases caused by technical problems. For this reason D-Man looks at an e-Business infrastructure both from an end-user and from a technical angle, where the technical aspects are however subordinate to the end-user aspect. D-Man achieves this goal by offering different views on the monitored environment.

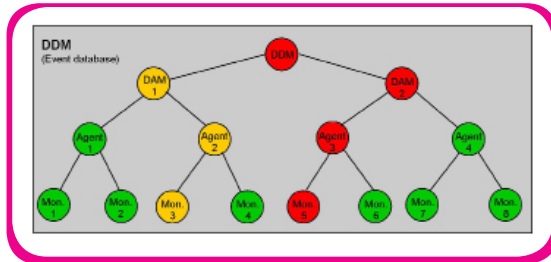
Prior to providing a more detailed description of these views, it is important to have a look at some of the underlying D-Man techniques: Consolidation & Correlation.

Consolidation

In complex distributed environments, consisting of many servers and applications, it is often very difficult to pinpoint the component that causes a problem. Consolidation is a technique that allows bringing the smallest detail to the surface. Consolidation allows a helpdesk or administrator to identify a problem without having to look through hundreds of event queues. Consolidation is very easy to achieve in a D-Man framework because of the hierarchical structure of the model whereby a problem at a lower level will raise the consolidated severity at the higher level.

The next figure shows how an event with a high priority at a monitor level propagates up the three, at each higher node overruling events with a lower severity.

Product description



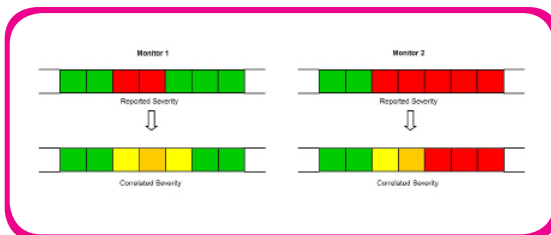
Correlation

Correlation is a technique that allows relating events to each other and avoiding unnecessary alarms. The D-Man monitoring solution supports two types of correlation: Time-based and Dependency Correlation. Time-based Correlation is a mechanism that allows tracking of a specific monitor over a certain period of time before raising an alarm.

Typically in e-Business environments it does occasionally happen that some components experience peak loads, resulting in CPU usage to get close to 100%. In such a case a monitoring system should not raise an alarm immediately because by the time the administrator has tracked the problem, it may have resolved itself.

To deal with such situations, an administrator can optionally use time-based correlation on a monitor. With this option the severity of an event will not directly be raised to the measured level but will gradually be increased. The more critical a component is for the entire environment the higher this factor will be.

The following figure shows two examples of monitors. Monitor 1 detects a problem, but because of its transient character, the overall severity of the monitor never reaches a critical point. For Monitor 2 the problem is however persistent and as such eventually the monitor raises an alarm.



Note, however, that the initial severity is maintained, because this could be the basis for another monitor that measures the frequency of peaks over time and raise an alarm when the frequency gets too high.

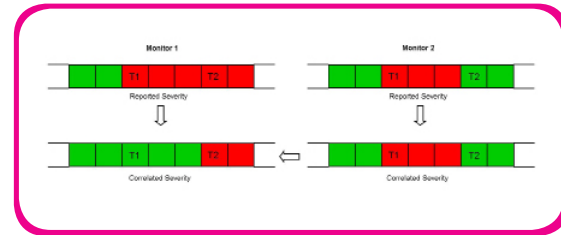
Dependency Correlation is a mechanism that allows creating relations between monitors.

For example, a user might experience an abnormal delay during authentication to a web site. Although it is the web server, hosting the site that interacts with the user it is not necessarily its responsibility. The web

server might rely on an authentication service that is causing the delay.

To deal with such situations, an administrator can make a monitor dependent on another monitor, or a set of monitors. Such a monitor will not raise an alarm if a monitor, on which it depends, is reporting an event with at least the same severity level.

The following figure shows two monitors, where Monitor 1 depends on Monitor 2.



At some point in time (T1) Monitor 1 detects a problem. However, because of its dependency, the status of Monitor 2 is checked. As Monitor 2 already generated an alarm, Monitor 1 will temporarily neglect the error. Some time later (T2) the problem detected by Monitor 2 was solved and as such its correlated severity drops. If at this stage the problem detected by Monitor 1 persists (as shown in the figure), this latter monitor will raise an alarm.

D-Man Event Viewers

As stated above, D-Man provides a number of views that makes the analysis and solving of problems reported by monitors easier. All these views show both Consolidated and Correlated events. The views are organized as hierarchies that provide a one-screen overview of the whole environment and that allow to easily zoom in on a problem.

D-Man offers the following event views:

- Current End-user View
- Current Technical View
- Historical End-user View
- Historical Technical View
- Timeline view

The Current End-user View (CEV) shows at any time the current status of the system from an end-user's perspective. It is based on the result of monitors that simulate the transactions initiated by the users of the system. This view will not generate an alarm as soon as a component would show unexpected behavior. It rather compares the service status against the optimal status and only raises an alarm when the service no longer meets an SLA.

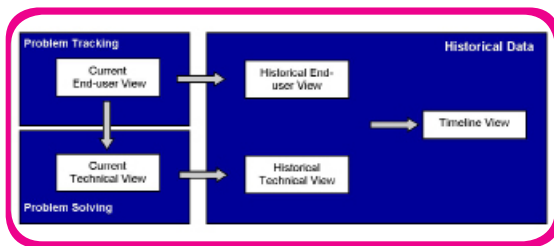
For example, while a failing web server in a high-availability cluster would generate an alarm during peak hours, it will only raise a warning when the load is low.

The CEV does not allow acknowledging or closing

Product description

of problems. Instead, if the CEV shows behavior that would diverge too much from the expected behavior, the helpdesk or administrator's attention will be attracted by the Current Technical View (CTV). This view is based on monitors that do look at the status of all the components of the environment. Because of the three structure of D-Man, this view allows the administrator to easily identify the cause of the problem. As soon as the problem is identified or solved, the event can be Acknowledged or Closed in the CTV. This will result in problems being cleared automatically in the CEV as well.

While the Current views are probably the most important views for tracking and solving problems, D-Man also provides Historical views: Historical End-user View, Historical Technical View and Timeline View. These views are merely informative. They have been provided to make sure that one can still keep track of issues that have occurred in the past. While D-Man provides a web based view on this historical data, it allows of course also filtered exporting of this data for report generation.



The previous figure shows how these views relate to each other and how they should be used by an administrator when tracking and solving problems or when generating reports.

Key features of D-Man

Platform independent.

D-Man is entirely written in Java and as such monitoring agents are supported on any platform supporting a JRE. Among the tested platforms are: AIX, HP-UX, RedHat, SuSE, Solaris and Windows.

Straightforward installation.

All D-Man components are provided as native platform packages. Alternatively, all components can also be provided as single TAR files that allow straightforward fully-automated installation and configuration. Furthermore, there are no external pre-requisites.

Embedded Security.

All internal D-Man transactions run over HTTPS using mutually authenticated SSL. Furthermore, connections are only established from secure to less-secure zones. This makes D-Man entirely compliant with firewall and DMZ policies, allowing it to even scale beyond the company boundaries. And last, but not least, moni-

tor scripts are never written to the file system. As such there is no risk of these scripts being overwritten by malicious users or programs.

Self Monitoring.

D-Man provides extensive self-monitoring. By means of embedded monitoring of the monitoring agents and a D-Man heartbeat, any internal problem can be detected and reported (e.g. to TEC or HP Openview) immediately.

Based on Open Standards. D-Man is entirely based on open standard. For its internal communication it relies on HTTP and SSL. For rendering and exporting events it uses HTML and XML. Integration with commonly used monitoring systems (TEC, HP Openview, etc.) can be accomplished using SNMP, SMTP, CLI or any native API plugin. And finally it also supports JMX for registering to event based systems.

Central Configuration and Administration.

Once installed, all configuration and administration of D-Man is done through a central D-Man component using only a browser. As such, there is no need to access the monitoring agents directly, not even when new monitoring targets (e.g. applications) are added.

User-centric and Technical Monitoring.

D-Man provides a balance between user-centric and technical monitoring. Where problems in any e-Business environment usually have a technical cause, because of the inherent redundancy of such environments, the impact of any technical problem is best measured from the end-user's perspective. D-Man correlates end-user's experience back to technical causes.

Ready for Cloud

SecurIT D-Man is a powerful monitoring solution that can easily be deployed in the Cloud for a large, distributed environment, where there is a need to have a centralised view on the status and health of the overall system and its individual parts.

Product description

D-Man Generic Monitors

This paragraph provides an overview of the generic monitor suite, which is provided by SecurIT D-Man out-of-the-box for general purpose usage. Additional monitor suites are available for specific target environments. For more information on other available monitors, please refer to the following guides:

- SecurIT D-Man. Monitor Reference Guide
- SecurIT D-Man. Distributed monitoring for Tivoli Access Manager (Solution Brief)
- SecurIT D-Man. Distributed monitoring for WebSphere Application Server (Solution Brief)
- SecurIT D-Man. Distributed Monitoring for Federated e-business environments (Solution Brief)

System Monitors

- File System usage/capacity
- Process Availability
- Process Memory Usage
- Process CPU Usage
- System Load
- System Memory Usage
- Swap Space Status
- Network Connection Status

D-Man Self Monitoring

- System Monitors
- DAM Status
- DDM Stats
- Logfile Parser

Generic D-Man monitors

HTTP(S) Monitors

- Page Content Validation
- Response Time
- Certificate Validation
- Certificate Lifetime

File Monitors

- Rotating Logfile Parser
- File Integrity Checker

LDAP(S) Monitors

- Search Result Parser
- Response Time

System Monitors

- File System usage/capacity
- Process Availability
- Process Memory Usage
- Process CPU Usage
- System Load
- System Memory Usage
- Swap Space Status
- Network Connection Status

D-Man Self Monitoring

- System Monitors
- DAM Status
- DDM Stats
- Logfile Parser

More information available on www.securit.biz.

Copyright © 2012 SecurIT bvba. All Rights Reserved

No part of this document may be copied, translated, or reduced to any electronic medium or machine readable form, in whole or in part, without the prior written consent of SecurIT bvba. SecurIT has made every effort to ensure that the information in this document is accurate and complete; however, SecurIT assumes no responsibility for any errors or omissions. Information in this document is subject to change without notice.

Product names mentioned in this paper may be trademarks or registered trademarks of their representative companies and are hereby acknowledged.