At a Glance

This document outlines the deployment scenarios, component architecture, technical feature descriptions and integration capabilities of ObserveIT Enterprise.

Contents

1 Product Architecture ................................................................. 2
  1.1 Overall Architecture ................................................................. 2
  1.2 Windows Agent ...................................................................... 2
  1.3 UNIX / Linux Agent ................................................................. 3
  1.4 Management Server ............................................................... 5
  1.5 Web Console ........................................................................ 5
  1.6 Database .............................................................................. 5

2 Deployment Scenarios ................................................................. 6
  2.1 Standard Agent-based deployment (Servers and Desktops) .......... 6
  2.2 Jump Server Gateway .............................................................. 7
  2.3 Outbound Jump Server Gateway .............................................. 8
  2.4 Citrix Server for Published Applications .................................... 8
  2.5 Hybrid Deployment: Agent-based + Gateway ......................... 9

3 Sizing and System Requirements ............................................... 10
  3.1 Small Deployment ................................................................ 10
  3.2 Medium Deployment ............................................................. 10
  3.3 Large Deployment with High-Availability ............................... 11

4 Security Infrastructure ............................................................... 13
  4.1 Windows Agent .................................................................... 13
  4.2 UNIX / Linux Agent .............................................................. 13
  4.3 AppServer-Agnt Health Check ............................................... 13
  4.4 Data Security (in Storage) ....................................................... 14
  4.5 Communication between ObserveIT Components .................... 14
  4.6 Privacy Protection ................................................................. 14
  4.7 Installation Security .............................................................. 14
  4.8 Identity Theft Detection ......................................................... 14
  4.9 Locking of user session ......................................................... 14

5 Data Management .................................................................... 15
  5.1 Database Structure ............................................................... 15
  5.2 File System Storage ............................................................... 15
  5.3 Archiving ............................................................................. 15
  5.4 Backup ................................................................................ 15

6 Installation Overview ............................................................... 16
  6.1 “One Click” installation .......................................................... 16
  6.2 Custom Installation ............................................................... 16
  6.3 Windows Agent Installation ................................................... 16
  6.4 UNIX / Linux Agent Installation .............................................. 16

7 Key Feature Overview .............................................................. 17
  7.1 ObserveIT Key Logging ........................................................... 17
  7.2 Ticketing System Integration .................................................. 17
  7.3 Audit Investigation and Reporting Viewing ................................ 18
  7.4 Policy Messaging and User Awareness of Auditing .................... 18
  7.5 Privileged Identity Management .............................................. 18

8 Key Configuration Settings ....................................................... 19
  8.1 Console Users (ObserveIT Administrator users) ......................... 19
  8.2 Server Configuration Policies ................................................ 19
  8.3 SMTP, LDAP, Active Directory ............................................. 20

9 SIEM and Log Management Integration ................................... 22
  9.1 Metadata Integration ............................................................. 22
  9.2 Video Replay Integration ..................................................... 24

10 Network Management (Alerting) Integration ......................... 24

11 Agent API for Process-Oriented Integration ............................. 25
1 Product Architecture

1.1 Overall Architecture

ObserveIT is a software-based user auditing platform, with no fixed hardware components:

![ObserveIT Architecture Diagram](image1)

**Figure 1 - The ObserveIT Architecture**

1.2 Windows Agent

The ObserveIT Windows Agent is a software component that is installed on any Windows-based operating system (server or desktop) that you wish to record. It can be installed on any version of Windows starting from XP Pro through Windows 7, Windows Server 2008 R2 (32 / 64 bit), and Windows 2012.

The Windows Agent is a user-mode executable that binds to every user session. As soon as a user logs in to a monitored server, the Agent is started and begins recording (based on a pre-determined recording policy).

When there is no active user session, the Agent is dormant and consumes no memory/CPU resources.

Once a user session is opened (user logs on), the Agent is triggered by user activities such as keyboard and mouse events. When triggered, the Agent performs a screen capture, and at the same time it captures textual metadata of what is seen on the screen (window title, executable name, file name, date, time, user name, etc.) The screen capture and textual descriptive metadata are packaged up and sent to the ObserveIT Management Server for processing and storage.

![Windows Agent Architecture Diagram](image2)

**Figure 2 - Windows Agent Architecture**
During the active user session, the Agent only performs actions when actual user activity is detected at the keyboard or mouse. During idle time (when the user is not actively doing something on the machine), the Agent does not record, and no repetitive data is captured.

The Windows Agent can maintain an offline-mode buffer to temporarily collect data when network connectivity is lost. The buffer size is customizable. Once connectivity is restored, the data is delivered as normal.

The Windows Agent also includes a watchdog mechanism to prevent user tampering, including alerts that are sent directly to the security officer.

### 1.2.1 Windows Agent Overhead

The ObserveIT Agent is a user-mode process, which only runs when a user session is active. When active, the average utilization is 10MB of RAM. The typical CPU utilization is 1%-2%, only at the moment of data capture. During idle time, CPU utilization is negligible. These values are “per session”, and should be multiplied for concurrent sessions (for example, on a Citrix Server or Terminal Server).

![Windows CPU and memory overhead](image)

Each captured screenshot is between 5-50 KB (depending on the screen resolution and number of changes since the previous screen). The Agent is configured to record in grayscale by default, but can also capture in full color if required.

### 1.2.2 Supported Platforms for Windows Agent


For a full list of supported Windows platforms, please refer to [http://www.observeit-sys.com/Products/documentation/index.htm#supported_platforms.htm](http://www.observeit-sys.com/Products/documentation/index.htm#supported_platforms.htm).

### 1.3 UNIX / Linux Agent

The ObserveIT UNIX / Linux Agent is a software component that uses library/function interposition in order to hook itself into processes. It remains inactive until the moment it detects creation of the interactive session (by creation of a new pseudo tty device).

When activated, it spawns an auxiliary process (logger) that receives metadata ("interesting" system calls and library functions) sent by the Agent that is hooked into the child processes. The logger process also collects the entire interactive data (keyboard I/O) passing through the original pseudo tty device. When the interactive session terminates, the logger also exits, after first making sure that all the data was sent to the server.
When a user logs in on a UNIX / Linux machine, the Agent is started and begins recording, based on a pre-determined recording policy. The Agent is triggered by Command Line Interface (CLI) events. When a user is inactive, the Agent does not record and no repetitive data is captured. The ObserveIT UNIX/Linux Agent captures all the internal actions and the names of files and resources that are affected by command line operations, as follows:

<table>
<thead>
<tr>
<th>Actions to be audited</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User session launched</td>
<td>Existence of an actual user session</td>
<td>User name john opened an SSH session via PuTTY</td>
</tr>
<tr>
<td>TTY command line entry</td>
<td>All screen I/O: user input and screen output</td>
<td>John runs an alias script called innocentScript</td>
</tr>
<tr>
<td>Processes launched</td>
<td>Any process spawned from within a user I/O command</td>
<td>The script innocentScript contains the line: rm -rf /export/home/john</td>
</tr>
<tr>
<td>System calls spawned</td>
<td>Any file create, delete, open, permission change, or link creation</td>
<td>The rm process deletes any file found in that directory, thus spawning the system call: UNLINK samplefile</td>
</tr>
</tbody>
</table>

The UNIX / Linux Agent can maintain an offline-mode buffer to temporarily collect data when network connectivity is lost. The buffer size is customizable. Once connectivity is restored, the data is delivered as normal.

The UNIX / Linux Agent is a user-mode application that is bound to the secure shell, which means that if a user stops the Agent, the entire user session is killed.

1.3.1 UNIX / Linux Agent Overhead

A typical CLI event is considered from the moment <Enter> is pressed until 1KB of data is cumulated, or after a maximum of 5 seconds from the last event. A session that has a high CLI activity usage and intensity will produce more data, therefore, more packets will be sent from the Agent to the Application server. Data of a typical user event including metadata may consist of 10–20 KB. Since the Agent only captures user actions and trims idle time, bandwidth usage is negligible. Client-side or server-side compression can be used to reduce the size of the traffic transmitted by the Agents to the Application Server, but will incur additional CPU resource usage on the client-side.

The ObserveIT Agent uses an average of 5-20 MB of RAM, about 0.1% CPU utilization during idle time within a user session and 0.7% CPU utilization when actively recording. The ObserveIT Agent only consumes resources when a user is logged on to the monitored server(s).
1.3.2 Supported Platforms for UNIX / Linux Agent

- Solaris 10 (updates 4-10)
- RHEL/CentOS 5.0-5.8, and RHEL/CentOS 6.0-6.3
- Ubuntu 10.04, 12.23, and 12.31
- Oracle Linux 5.0-5.8, and Oracle Linux 6.0-6.3
- SLES SuSE 10 SP2-SP4, and SLES SuSE 11 initial release-SP2
- AIX 5.3 (TL10 or higher), and AIX 6.1

For a full list of supported platforms for UNIX/Linux Agents, please refer to [http://www.observeit-sys.com/Products/documentation/index.htm#supported_platforms.htm](http://www.observeit-sys.com/Products/documentation/index.htm#supported_platforms.htm).

1.4 Management Server

The Management Server listens for content delivered by each of the Agents. It then analyzes and categorizes the data and sends it to the Database Server for storage. The Management Server also actively communicates with Agents to deliver configuration updates and for system health monitoring. The Management Server is an ASP.NET application running in IIS.

1.5 Web Console

The Web Console application is an ASP.NET application that serves as the primary interface for audit review, video replay and reporting, as well as for admin / configuration tasks. The Web Console includes granular policy rules for limiting access to sensitive data.

1.6 Database

By default, ObserveIT uses Microsoft SQL Server for data storage. This storage includes configuration data, textual audit metadata and the actual screenshots for video replay. ObserveIT also can be configured to store the video replay screenshots in file-system storage instead. In such a case, the MS SQL Server database is still used for metadata and configuration data. ObserveIT can work with SQL Server Express, but it is not recommended due to its size limitations. Connectivity with the database is on standard TCP port 1433.
2 Deployment Scenarios

ObserveIT can be deployed in a number of different ways, as shown below. The different methods are not mutually-exclusive, allowing for a hybrid deployment when required.

2.1 Standard Agent-based deployment (Servers and Desktops)

The standard method of deployment involves deploying the ObserveIT Agent on each machine to be monitored.

An Agent is installed on each machine that is being monitored, which captures activity on the machine and feeds the video / log data to the management server.

![Figure 6 - Agent-Based Deployment](image-url)
2.2 Jump Server Gateway

In this scenario, the ObserveIT Agent is only deployed on a gateway machine. Users are routed via this gateway, and thus ObserveIT still records all user sessions in which the user connects to another target machine via RDP, SSH or other protocol.

ObserveIT does not record any user session in which a user logs on directly to the target machine (via local console login, or via a direct RDP/SSH/etc. window that isn’t routed via a gateway.) Also, the amount of textual metadata captured is less than for the full Agent deployment, due to the fact that the ObserveIT Agent on the gateway does not have access to OS specific information on the target machine (for example, it cannot see the name of a file opened within an RDP window).
2.3 Outbound Jump Server Gateway

The Jump Server Gateway architecture described above can also be used for environments in which remote users need to access multiple external resources (for example, a Managed Services Provider that needs to support multiple customers and wants to record and audit all the actions performed by the support employees).

The architecture is essentially the same as above, with the only difference being the location of each resource (i.e., the Terminal Server would not be on same network as the target machines).

![Outbound Jump Server Gateway Diagram]

2.4 Citrix Server for Published Applications

The ObserveIT Agent can also be deployed on a Citrix Server, in order to record all activities that take place within Published Applications served by the Citrix machine.

![Citrix Server Deployment Diagram]
2.5 Hybrid Deployment: Agent-based + Gateway

ObserveIT allows you to deploy any combination of these architectures simultaneously. A gateway can be used for full network coverage, providing an audit of all activities for the majority of users who are routed via the gateway. Then, Agents can also be deployed on specific sensitive servers that require a more detailed audit, including any logins performed by highly-privileged users who have direct access to the machine.

![Diagram of Hybrid Deployment: Gateway + Agent](image-url)
3 Sizing and System Requirements

3.1 Small Deployment

For installations with low user activity (less than 100 monitored servers), an “All in One” installation can be utilized, which means that the Application Server, Web Management Console and Database Server are all installed on the same platform. This platform can be a physical server, or it can be a virtual machine running in a typical virtualization solution.

3.1.1 System Requirements and Data Sizing for Small Deployment

- Physical Server with 2 Quad Core CPU 2.4 GHZ
- 16 GB of RAM
- 500 GB of High Speed IO Disk

3.2 Medium Deployment

For medium-sized implementations of ObserveIT, consisting of 100-1000 monitored servers, it is preferable for MS SQL Server to be installed separately from the AppServer/Web Console. If required, an existing SQL Server can be used, or a new instance can be created.

Depending on the company’s data storage strategies, a file-system storage method for screen capture data might be considered for this size deployment.
3.2.1 System Requirements and Data Sizing for Medium Deployment

For the AppServer, the recommended requirements are:
- 1 Quad Core CPU 2.4 GHZ
- 4 GB RAM
- 36 GB free hard disk space

For the SQL Server, the recommended requirements are:
- Physical Server with 2 Quad Core CPU 2.4 GHZ
- 24 GB of RAM
- Actual data storage depends on volume of user activity. Typical customer environment of 1000 production servers and 600 admin users generates 550 GB per year

3.3 Large Deployment with High-Availability

Large enterprise implementations of ObserveIT consisting of more than 1000 monitored server will typically also be accompanied by load balancing (LB), high-availability (HA) and redundancy requirements. There are a few factors in deploying HA:

- Two or more servers running ObserveIT Application Server and Web Console.
- Cluster-based implementation of Microsoft SQL Server.
- SQL Server using a dedicated storage device, or alternatively, using ObserveIT’s file system storage mechanism for visual screen shot data storage.

3.3.1 Round Robin DNS Implementation

The simplest LB mechanism would use DNS Round Robin. Note that this does not provide true HA, just a simple LB mechanism based on DNS, which has no knowledge of the state of these machines.

![Round Robin DNS Implementation](image)

3.3.2 Load Balancer Implementation

When full LB and HA is required, you can use a software-based LB (such as NLB) or hardware-based (such as F5). Optionally, this can be further augmented by a failover cluster for the application server, with an active/passive cluster that has only one node operational at any given time. Also, more nodes can be added to the failover cluster, as needed.
3.3.3 **File System storage**

To improve performance of the MS SQL Server, it is sometimes recommended to use ObserveIT’s file-system storage capabilities. In this scenario, the SQL Server is still used for metadata and configuration data, but the actual screenshot images are stored in a file system directory structure, which is fully managed by ObserveIT.

3.3.4 **System Requirements and Data Sizing for Large Deployment**

For each AppServer, the recommended requirements are:
- 1 Quad Core CPU 2.4 GHZ
- 4 GB RAM
- 36 GB free hard disk space

For the SQL Server, the recommended requirements are:
- Physical Server with 2 Quad Core CPU 2.4, GHZ
- 24 GB of RAM
- Actual data storage depends on volume of user activity. Typical customer environment of 1000 production servers and 600 admin users generates 550 GB per year. (This is true for both SQL storage and file system storage options.)
4 Security Infrastructure

ObserveIT is a highly-secure product platform, designed for full reliability and non-repudiation.

4.1 Windows Agent

The Windows Agent is protected by a multi-layered Watchdog mechanism. The Agent itself consists of two separate processes that act as a watchdog for each other; each will restart the other process if ended. In addition, a local service watches both processes to restart them if they are somehow stopped simultaneously.

4.2 UNIX / Linux Agent

The UNIX / Linux Agent hooks to the terminal device and to the user shell. Thus, any attempt to stop / kill the logger will immediately result in killing or hanging the user shell.

4.3 AppServer-Agent Health Check

An additional system health check residing on the Application Server will alert the administrator of any actions involving improper modification or stopping of Agent processes or services. This includes stopping of any Agent process/service, file modification and registry modification.

4.4 Data Security (in Storage)

Data that is stored in MS SQL Server automatically inherits any data protection mechanisms already in place for the corporate database.

In addition, if the data integrity of the ObserveIT database storage is violated (for example, if a dba succeeds in deleting an incriminating screenshot from within the entire collection), ObserveIT will provide a warning indicator within the Web Console.

For privacy, all screen capture data (whether stored in the SQL database or in the file system) is encrypted by a synchronous Rijndael 256-bit key. To further protect this key, the key itself is encrypted by an asynchronous 1024-bit X509 certificate (with RSA encryption key). This encryption is also inherited in any exported offline sessions.

4.5 Communication between ObserveIT Components

Communication between ObserveIT components is handled over HTTP protocol. SSL is fully supported (optional feature) in order to encrypt all communication between the different components.

If required, an IPSec tunnel can also be used to protect the Agent to Server traffic.
4.6 Privacy Protection

**Granular access rights** - ObserveIT users can be limited to viewing the sessions of specific servers or specific users. For example, the Database group manager can view sessions by DBA’s on any computer, plus any user session that took place on the database server. These rules extend to all metadata summaries, reports, and video replay.

**Dual Password Protection for Playback (4-Eyes Protection)** - ObserveIT allows you to specify a second password (not managed by the ObserveIT administrator) that is required in order to replay the video of a user session. This ensures both audit completeness and employee privacy. In the most typical usage, Management (via ObserveIT Administrator) holds the main ObserveIT password, and a union rep or legal council holds the second password. This then satisfies stringent privacy protection regulations, including BDSG (Germany), CNIL (France), DPD 95/46/EC (EU), Human Rights Act (UK). Granular deployment allows textual audit logs to be accessed by compliance officers (without the second password), but video replay requires employee council authorization (both passwords).

**ObserveIT self-auditing** - ObserveIT audits itself, capturing logs and video replay of every ObserveIT user who views recorded sessions.

4.7 Installation Security

The ObserveIT administrator can protect against improper or unauthorized Agent installation by enforcing the person installing any Agent to provide a password, which is registered on the Management Server.

4.8 Identity Theft Detection

ObserveIT’s Identity Theft Detection module brings a brand new approach to preventing and discovering incidents of stolen privileges. Today, security officers provide users with tools and education on how to protect their identity (such as, Two Factor Authentication, Password complexity and reset rules, etc.) But once an identity is stolen, no tool can clearly identify or track the incident, and the responsibility for detection lies entirely on the security officer. ObserveIT enables you to include users in the detection process, and thus make the user responsible for his/her identity. IT identity theft incidents can be detected and neutralized much quicker if users have a means to flag unauthorized logins.

For each monitored server, ObserveIT keeps track of authorized/confirmed pairings of User IDs and client machines. If a user logs in to a server from a client that s/he is not already paired with, an email is sent to the user. For example:

- A hacker steals a password and logs in from a remote machine. An email is sent to the user saying “The user ‘johnsmith’ just logged in to server WEBSRV-PROD from unauthorized IP address 11.22.33.44. Please confirm that it was you who performed this action.”

- An internal user steals the administrator’s password and logs in to a server from her own desktop, generating email “The user ‘johnsmith’ logged in to server DBPROD-4 from unauthorized desktop KATHY-DSKTP. Please confirm that it was you who performed this action.”

The user can either confirm or deny that this was his action. In parallel, an event is logged for the administrator to track and monitor unauthorized pairings. Granular security rules can be applied to specify how to manage each user confirmation.

4.9 Locking of user session

With ObserveIT, you have the ability to view live user sessions in real-time. If required, you can interact with the user of each session by sending messages (for example, “You should not be running SQL Queries on the production database”), and can also stop the user session entirely by locking the user’s desktop.
5 Data Management

5.1 Database Structure

By default, ObserveIT utilizes the following databases, which are created during installation:

- ObserveIT
- ObserveIT_Data
- ObserveIT_Archive_1
- ObserveIT_Archive_template

This storage includes configuration data, textual audit metadata and the actual screenshots for video replay. In addition, the database user “ObserveITUser” is also created.

5.2 File System Storage

The visual screenshots represent the largest portion of ObserveIT’s data storage needs. This portion of the data storage can be switched to file-system instead of SQL database, if required. This is most commonly used for large deployments or when MS SQL Server database performance issues arise.

When using file-system storage, there is still a need to maintain the MS SQL Server database, in order to store the textual metadata and the ObserveIT configuration data.

ObserveIT automatically manages the directory where you specify that screenshot data should be stored, including an auto-generated and archived subdirectory tree per date and per session.

5.3 Archiving

ObserveIT has built-in database archiving capabilities, to move data from the main ObserveIT database to a secondary database. This is in order to improve performance and to limit old unnecessary data when required. Archiving jobs can be launched manually or can be scheduled for automatic periodic archive rotation.

![Figure 19 - Archiving](image)

The archive process moves the visual screen captures, but maintains the metadata for searching. This ensures that the data that consumes the most storage is moved, while maintaining searchability of log info. In addition, actual video replay can be launched straight from an archived session.

5.4 Backup

All data stored in SQL databases can utilize existing backup solutions that are built-in to MS SQL Server, or 3rd party database backup solutions.
6  Installation Overview

6.1  "One Click" installation

A one-click installation is the easiest way to deploy ObserveIT in the most common environments. In the main installation screen, there are 3 separate sections; SQL Server settings, Web applications (Management console and Application server), and Licensing. The one-click installation will also install an Agent locally on the application server machine.

![Figure 20 - One-Click Install](image)

6.2  Custom Installation

If needed, each of the ObserveIT components can be installed separately as part of a custom installation. Each standalone installer allows you to distribute the components as needed and to use advanced configuration options.

Active Directory Domain membership is not mandatory, but ideally, all components should be placed on domain members. This enables usage of AD groups for Console Users; filtering of AD groups for Privileged Identity Management; DNS integration for Agent auto-configuration; and GPO-based installation.

6.3  Windows Agent Installation

Windows Agent installation is performed over a standard Windows installer package (.MSI) that is well supported by software distribution applications and Group Policy (GPO). Agents can be easily configured to automatically install itself with a simple batch file. Agents can be auto-configured by using DNS. A password can be used to prevent rouge Agent installations. No reboot is required after installation. An optional system tray icon can be displayed on the machine when the Agent is running.

6.4  UNIX / Linux Agent Installation

UNIX / Linux Agent installation is a one-step process. No reboot is required after installation.

- Solaris: ./observeit-agent-solaris10-i386-release-5.5.xx.run -- -I -s <ServerIP>:<Port>
- Linux: ./observeit-agent-linux-5.5.xx.run -- -I -s <ServerIP>:<Port>

For detailed instructions on installing the ObserveIT Agent on all supported UNIX/Linux systems, please refer to [http://www.observeit-sys.com/products/documentation/index.htm#unix_linux_agent_deployment.htm](http://www.observeit-sys.com/products/documentation/index.htm#unix_linux_agent_deployment.htm).
7 Key Feature Overview

7.1 ObserveIT Key Logging

Corporate key loggers track and record an employee’s computer activity for the purposes of monitoring, root cause analysis, forensic investigation and regulatory auditing. ObserveIT takes key logging into the next generation with some unique capabilities. ObserveIT Key Logging is supported on Windows and Unix-based operating systems.

ObserveIT’s key logger generates and replays video recordings of all on-screen user activity, including every key press and mouse click. Any portion of any recording is directly accessible via key word search in the ObserveIT Web Management Console. You can jump directly to relevant portions of recordings by searching for particular activities based on text entries, launched programs, opened windows, system commands executed, etc. ObserveIT’s key logger records and enables you to search for specific text entries made anywhere in the system, whether by typing, editing, keyboard shortcuts, auto-complete or even copy and paste via the Windows clipboard.

On Unix systems, the Unix Auditor records user activity in any interactive shell running on the machine, and transfers the data to the ObserveIT Management Server. Recording begins whenever a user starts any interactive session on the system, whether remotely (via Telnet, SSH, rlogin, etc.) or locally via a console login.

For more detailed information about ObserveIT’s key logger, please refer to [http://www.observeit.sys.com/Products/documentation/index.htm#observeit_text_logger.htm](http://www.observeit.sys.com/Products/documentation/index.htm#observeit_text_logger.htm).

7.2 Ticketing System Integration

The integration of ObserveIT’s session recording system with an IT ticketing system provides additional layers of security and monitoring.

Two types of ticketing systems can be integrated with ObserveIT: Built-in and Customized.

- **Built-in ticketing systems** are provided by ObserveIT as out-of-the-box integrations (for example, ServiceNow).
- **Customized ticketing systems** are implemented by customers according to their own requirements. ObserveIT provides API instructions to help customers build a Web Service that will enable them to implement the integration of ObserveIT with their own ticketing system.

When an administrator or remote vendor for whom this feature is enabled attempts to log in to a monitored server, a message is displayed asking the user to enter a valid ticket number from a ticketing system in order to log on to the server.

![Figure 21 - Ticket Window](image)

The ticket number entered is validated against the ticketing system database before the user is granted access to the system. The ticket associated with the session is linked to a video recording of the session. In addition, specific information about the login session is automatically saved by ObserveIT and included in the ticketing system.

Within the ticketing system itself, a direct link to the video recording of the particular session in which the administrator or remote vendor addressed the ticket, provides faster and easier auditing of the exact actions performed by administrators and remote vendors.
For more detailed information about integrating ObserveIT’s session recording system with an IT ticketing system, please refer to [http://www.observeit-sys.com/Products/documentation/index.htm#ticketing_system_integration.htm](http://www.observeit-sys.com/Products/documentation/index.htm#ticketing_system_integration.htm).

### 7.3 Audit Investigation and Reporting Viewing

ObserveIT’s Web console includes numerous ways to navigate, search, run reports and export the user activity log data.

- The report generator includes canned reports and customizable report rules for filtering by user/user group, server / server group, date, application, resources accessed and more.
- Reports can be run ad-hoc or delivered on a schedule by e-mail.
- Full-text Google-like searching allows pinpoint identification of user sessions.
- Session metadata drill-down allows each session to be viewed item-by-item, to see exactly which applications were run during that session.
- Video replay can be launched directly from any audit view or report.
- Specific audit video can be exported for delivery as a simple HTML file for forensic evidence delivery.

### 7.4 Policy Messaging and User Awareness of Auditing

Policy information can be delivered to users exactly as they log in to a server or desktop. This policy info can include awareness of auditing activity (e.g., “Please note that all activity is being recorded in this user session.”). Policy info can also relate to company or regulatory policies (e.g., “Please note that PCI requirements mandate that no database traces be implemented on this DB.”)

Policy messages can also be set to require user response. This is often used for capturing Ticket # info for service calls. Users can be prevented from doing anything on the computer until they provide confirmation and/or response.

![Policy Messaging delivered to the user on Windows and on UNIX/Linux](image)

### 7.5 Privileged Identity Management

When admin users log in using a shared ‘administrator’ account, ObserveIT presents users with a secondary challenge-response, forcing them to specify their named-user account ID. Secondary ID’s can be tied to an Active Directory repository, or can be managed locally in the ObserveIT admin console. ObserveIT’s User Identity mechanism allows you to managed shared-user access without requiring the overhead of password rotation.
8 Key Configuration Settings

8.1 Console Users (ObserveIT Administrator users)

Console Users can log on to the ObserveIT Web Console and view metadata logs and recorded sessions, as well as make configuration changes based upon their role. The default Console User is an "Admin" operator, which has the highest permissions for any configuration task. Different levels of access can be defined for specific users or user groups.

8.2 Server Configuration Policies

Server Policies are sets of configuration options that control aspects of how the monitored server is configured. By using Server Policies, the administrator can configure one set of recording settings, and apply these settings on many monitored servers simultaneously. Policy settings include:

- Enabling Identity Theft Detection
- Hiding the Agent Tray Icon
- Enabling Agent API
- Agent Recording Status
- Restricting Recording to RDP Only
- Enabling Hotkeys
- Enabling Key Logging
- Enabling Recording Notification
- Recording in Color or Grayscale
- Setting Session Timeout
- Keyboard Stroke Recording Frequency
- Offline Mode Configuration
- Identification Policy (Secondary User Identification / PIM)
- AD User / User Group Recording Policy
- Application Recording Policy

8.3 SMTP, LDAP, Active Directory

In order to allow ObserveIT to send messages and scheduled reports to Console Users, configure SMTP communication information.
LDAP integration is commonly used for secondary user authentication.

If during installation, the server which hosts the ObserveIT Application Server component is a member of an AD domain, this connector is created automatically. If the server is not a member of a domain during installation, but is made a member afterwards, the connector can still be created.
9 **SIEM and Log Management Integration**

ObserveIT can easily integrate with SIEM products to enable user metadata to be viewed and video replay to be launched straight from within the external SIEM dashboard or report environments.

Integration typically involves two main factors: Metadata integration and Video Replay integration.

9.1 **Metadata Integration**

Most SIEM platforms utilize a ‘data collector’ mechanism for importing log data. ObserveIT’s user activity metadata logs fit this model well. Any SIEM can access the ObserveIT metadata in one of two ways; either via direct SQL access or via real-time log file polling. Each of these methods use direct access to the data source, without the need to go via a Web service or API-call layer.

9.1.1 **SQL Integration**

A sample SQL query for polling data would be as follows:

```
"USE ObserveIT; SELECT ScreenshotTime, MachineName AS ServerName, LoginName, DomainName, ApplicationName, WindowTitle, UserName, ClientName, ClientAddress, SessionID, ScreenshotID, ApplicationServerName, 'WindowTitle' EventType FROM dbo.SessionWindowTitleInstances INNER JOIN ServerInventory on ServerInventory.SrvID=SessionWindowTitleInstances.SrvID WHERE ScreenshotTime > '%TRACKING%'
```

The query would produce the following output:

```
<table>
<thead>
<tr>
<th>ScreenshotTime</th>
<th>ServerName</th>
<th>LoginName</th>
<th>DomainName</th>
<th>ApplicationName</th>
<th>WindowTitle</th>
<th>UserName</th>
<th>ClientName</th>
<th>ClientAddress</th>
<th>SessionID</th>
<th>ScreenshotID</th>
<th>ApplicationServerName</th>
<th>EventType</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-12-10 12:21:17.729</td>
<td>gpt WARRANT</td>
<td>gpt warrant</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-12-10 12:21:17.729</td>
<td>gpt WARRANT</td>
<td>gpt warrant</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-12-10 12:21:17.729</td>
<td>gpt WARRANT</td>
<td>gpt warrant</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-12-10 12:21:17.729</td>
<td>gpt WARRANT</td>
<td>gpt warrant</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013-12-10 12:21:17.729</td>
<td>gpt WARRANT</td>
<td>gpt warrant</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Figure 28 - SQL Server data collector schematic**

**Figure 29 - Sample SQL Query Results for SIEM Integration**
9.1.2 Log File Integration

All ObserveIT metadata can be added to a real-time log file, by enabling this within the Configuration Settings.

The resulting log files will appear as follows (you can select to display them separately per platform or merged):

**Sample Windows Log**

```
"FirstScreenshotTime","SessionId","ClientName","ServerName","DomainName","LoginName","UserName","ApplicationName","WindowTitle"
2011-08-11T07:07:20,afd3fe2b-2243-4ccb-b4fe-b2ba39cdda08,OIT-BRAD,OITHostedDemo-S,OITHostedDemo-S,Administrator,brad,ObserveIT,ObserveIT Login (5.3.0.0)
```

**Sample UNIX / Linux Log**

```
"OperationDate","SessionId","ClientName","ServerName","DomainName","LoginName","UserName","CommandParam"
2011-08-11T08:57:29,d2526b82-8d37-4c35-b74e-26242a0f73e5,10.1.100.5,c56-32-3,observeit.com,dima,n/a,/bin/grep -q /usr/kerberos/bin
2011-08-11T08:57:30,d2526b82-8d37-4c35-b74e-26242a0f73e5,10.1.100.5,c56-32-3,observeit.com,dima,n/a,/bin/grep -q /usr/kerberos/bin
2011-08-11T08:57:31,d2526b82-8d37-4c35-b74e-26242a0f73e5,10.1.100.5,c56-32-3,observeit.com,dima,n/a,/sbin/consoletype stdout
2011-08-11T08:57:33,d2526b82-8d37-4c35-b74e-26242a0f73e5,10.1.100.5,c56-32-3,observeit.com,dima,n/a,/sbin/lsattr
2011-08-11T08:57:35,d2526b82-8d37-4c35-b74e-26242a0f73e5,10.1.100.5,c56-32-3,observeit.com,dima,n/a,/usr/bin/id -u
2011-08-11T08:57:36,d2526b82-8d37-4c35-b74e-26242a0f73e5,10.1.100.5,c56-32-3,observeit.com,dima,n/a,/bin/id -u
```
9.2 Video Replay Integration

Unlike the metadata log, the video replay data is typically maintained within the ObserveIT environment, enabling enhanced custom playback functionality and reducing the amount of data that would otherwise be continuously added to the SIEM.

Figure 32 - Video Replay Integration Schematic

The video replay is available as a single HTTP target even if the ObserveIT database is federated across multiple local installations. The custom application does not need to be aware of the actual video storage location.

Figure 33 - Video Replay Integration with Federated Databases

10 Network Management (Alerting) Integration

The same data integration highlighted above for SIEM integration can be utilized to implement a custom alerting method within any common Network Management Platform.
11 Agent API for Process-Oriented Integration

ObserveIT’s Agent API enables external applications to build custom logic for what and when to record. The Agent API exposes a set of classes that enables:

- Start, Stop, Pause, Resume, and End a recorded session
- Custom logic for when to start recording (based on process ID, process name, computer name, user, URL, etc.)
- Perform system health check
- View recorded sessions

Recording additional processes can be tied to existing sessions or to a new session (thus creating a separate sessions for each recorded process). The API is built-in to the Agent, but not enabled by default. It can be enabled from the ObserveIT Web Console.

*Controller Machine & Monitored Machine are loosely coupled. Can be (but does not need to be) same machine.*

*Figure 34 - Agent API Schematic*

This can be utilized in many ways. One example API implementation is as an ActiveX trigger.

*Figure 35 - Sample API Implementation using ActiveX*