Logging - CustomerID

Introduction

Ubisecure CustomerID has two log files at the application level. The log files are:

- `customerid_audit.log` – This log file contains the audit log.
- `customerid_diag.log` – This log file contains additional technical information, such as errors.

These log files are by default written to the following location inside the WildFly installation: *`standalone/log/`* . If you use the default configuration both log files will be renamed every day there is activity in the system. The date will be added to the end of the log file names in the following format: *`yyyy-mm-dd`* . You must manually purge the log files. You can also set up your own logging configuration and for example collect all logs to a separate logging server. One log entry is always on one line except when it includes an exception stack trace. The separator character between fields is ";". Audit logs don't include exception information.

Depending on configuration, additional log files are generated by the application server inside the WildFly installation: *`standalone/log/`* . These logs include

- `server.log`
- `service.log`
- `wildfly-stderr.log`
- `wildfly-stdout.log`
- `access.log` (disabled by default)

After the current day, the date will be added to the end of the log file names in the following format: *`yyyy-mm-dd`* . You must manually purge the log files. You can also set up your own logging configuration and for example collect all logs to a separate logging server.

CustomerID logging configuration instructions can be found from Logging configuration - CustomerID.

Audit Logging

Audit log is a chronological sequence of audit records. The audit records contain data pertaining to and resulting from activities such as transactions or communications by individual people, systems, accounts or other entities. Practically speaking, an audit log shows who has accessed a computer system and what operations he or she has performed during a given period of time.

Audit logging can be used in conjunction with different liability issues. In case of dispute, audit records can be used to reconstruct and examine sequences of events. In this way, they work as pieces of evidence for the incidents in question.
Ubisecure CustomerID collects an audit on user actions in the system.

Default Audit Log Format

- Logger format:
  - timestamp (23 characters) (based on ISO8061 standard)
    - Format: yyyy-MM-dd HH:mm:ss,SSS
  - application format (log message from application)

- Application format:
  - event (30 characters)
    - Is a very short title of the operation that is being performed.
  - effect (11 characters)
    - Possible values: IN_PROGRESS, SUCCESS, FAIL.
  - executor (36 characters)
    - In most cases contains the database ID (=User ID) of the executor of the operation. It is in UUID format.
    - Secondarily may also contain some other identification information that is no longer than 36 characters.
  - target (36 characters)
    - In most cases contains the database ID (usually User ID) of the target of the operation. It is in UUID format.
    - Secondarily may also contain some other identification information that is no longer than 36 characters.
  - log message text
    - Can contain any variable length information needed to clarify the logging situation.
  - ipaddress
    - IP address where the operation originates from.

Diagnostic Logging

Diagnostic log is a chronological sequence of system actions. The diagnostic records contain data used to debug the system in case of errors or to verify the system actions.

Default Diagnostic Log Format

- Logger format:
    - timestamp (23 characters) (based on ISO8061 standard)
        - Format: yyyy-MM-dd HH:mm:ss,SSS
    - log level (5 characters)
        - Possible values: DEBUG, INFO, WARN, ERROR.
    - node
        - Node name of the node on which the application runs.
    - thread
        - Thread name of the thread that is currently executing when the log is written.
    - category
        - This is the logging category. You can limit logging based on the category.
    - application format (log message from application)
    - exception stack trace

- Application format:
    - event (30 characters)
        - Is a very short title of the operation that is being performed.
    - effect (11 characters)
        - Possible values: IN_PROGRESS, SUCCESS, FAIL.
    - executor (36 characters)
        - In most cases contains the database ID (usually User ID) of the executor of the operation. It is in UUID format.
        - Secondarily may also contain some other identification information that is no longer than 36 characters.
    - target (36 characters)
        - In most cases contains the database ID (usually User ID) of the target of the operation. It is in UUID format.
        - Secondarily may also contain some other identification information that is no longer than 36 characters.
    - log message text
        - Can contain any variable length information needed to clarify the logging situation.
    - ipaddress
        - IP address where the operation originates from.
    - sessionid

Log Events

```
LIST_ROLES
LIST_ROLE_INVITATIONS
LIST_USERS
LIST_ORGANIZATIONS
LIST_ORGANIZATION_USERS
LIST_ORGANIZATION_APPROVALS
LIST_SELECTED_AUTHMETHODS
LIST_PENDING_APPROVALS
LIST_REGISTRATIONS
LIST_MANDATES
LIST_MANDATE_TEMPLATES
LIST_APPROVALS
LIST_DELEGATIONS
SEARCH_ORGANIZATION_USERS
SEARCH_ORGANIZATIONS
SEARCH_USERS
QUERY_ROLE
QUERY_ROLE_INVITATION
QUERY_USER
QUERY_ORGANIZATION
QUERY_APPROVAL
QUERY_REGISTRATION
QUERY_MANDATE
QUERY_MANDATE_TEMPLATE
QUERY_DELEGATION
CREATE_ROLE
CREATE_ROLE_INVITATION
CREATE_USER
CREATE_ORGANIZATION
CREATE_APPROVAL
CREATE_ASSIGNMENT
CREATE_REGISTRATION
CREATE_FEDERATION_LINK
CREATE_MANDATE_TEMPLATE
```

```
CREATE_MANDATE
CREATE_MANDATE_DELEGATION
CREATE_MANDATE_ROLE_DELEGATION
ASSIGN_AUTH_METHOD
ASSIGN_GROUP
ASSIGN_ROLE
ASSIGN_MANDATE_TEMPLATE
UPDATE_ROLE
UPDATE_USER
UPDATE_ORGANIZATION
UPDATE_APPROVAL
CHANGE_PASSWORD
UPDATE_USER_ORGANIZATION
UPDATE_REGISTRATION
UPDATE_BACKEND
UPDATE_MANDATE
UPDATE_MANDATE_TEMPLATE
REMOVE_ROLE
REMOVE_USER
REMOVE_ORGANIZATION
REMOVE_APPROVAL
REMOVE_MANDATES
REMOVE_REGISTRATION
REMOVE_MANDATE
REMOVE_MANDATE_DELEGATION
REMOVE_MANDATE_ROLE_DELEGATION
REMOVE_MANDATE_TEMPLATE
REMOVE_ASSIGNMENT
DEASSIGN_ROLE
DEASSIGN_GROUP
DEASSING_AUTH_METHOD
ROLE_INVITE_WIZARD
MANDATE_WIZARD
USER_ADD_ROLE_WIZARD
USER_REMOVE_ROLE_WIZARD
REGISTRATION_WIZARD
PASSWORD_RECOVERY_WIZARD
APPROVE_INVITATION
APPROVE_MANDATE
APPROVE_REGISTRATION
DENY_INVITATION
DENY_MANDATE
DENY_REGISTRATION
SYSTEM_OPERATION_NOT_SUPPORTED
SYSTEM_SERVICE_ACCESS
SYSTEM_IMPORTER
SYSTEM_INITIALIZATION
SYSTEM_SELF_SERVICE_UI
SYSTEM_ADMIN_UI
SYSTEM_INTERNAL_PROCESSING
SYSTEM_LOGOUT
SYSTEM_AUTHENTICATION
SYSTEM_AUTHORIZATION
SYSTEM_VALIDATION
SYSTEM_GENERATE_OTP_LIST
SYSTEM_DISABLE_USER
SYSTEM_ACTIVATE_USER
SYSTEM_SMS_SENDING
SYSTEM_EMAIL_SENDING
SYSTEM_EMAIL_CONFIRMATION
SYSTEM_MOBILE_CONFIRMATION
```

Access Logs

The CustomerID Application Server, Wildfly, can record an Access Log of all HTTP traffic received.

Recorded data include source IP, time, date, timezone, HTTP request with all query string parameters, HTTP status code returned and User Agent (browser used).

This log is disabled by default. To enable the access log, edit WildFly `standalone/configuration/standalone.xml` and add the element `access-log` to the `host` element as shown below.

A file `access_log.log` will then the written to the `standalone\log` directory in the format specified.

CustomerID Access Log Configuration

```
<host name="default-host" alias="localhost, www.example.com, localhost">
                    <location name="/" handler="welcome-content"/>
                    <filter-ref name="server-header"/>
                    <filter-ref name="x-powered-by-header"/>
                    <access-log pattern="%h %l %u [%t] &quot;%r&quot; %s %b &quot;%{i,Referer}&quot; &quot;%{i,
User-Agent}&quot;"/>
</host>
```

If a reverse proxy is used and the proxy passes the original source IP as X-Forwarded-For header, following configuration can be used:

CustomerID Access Log Configuration with reverse proxy

```
<host name="default-host" alias="localhost, www.example.com, localhost">
                    <location name="/" handler="welcome-content"/>
                    <filter-ref name="server-header"/>
                    <filter-ref name="x-powered-by-header"/>
                    <access-log pattern="%{i,X-Forwarded-For} %l %u [%t] &quot;%r&quot; %s %b &quot;%{i,Referer}
&quot; &quot;%{i,User-Agent}&quot;"/>
</host>
```

After making this change, the WildFly application server must be restarted.

Windows

Restarting WildFly Application Server on Windows

```
net stop wildfly
net start wildfly
```

Linux

Restarting WildFly Application Server on Linux

```
systemctl restart wildfly.service
```

An example of the `access_log.log` contents

access_log.log contents

```
127.0.0.1 - - [[20/Jul/2017:11:15:39 +0000]] "GET /eidm2/wf/admin?tab=overview HTTP/1.1" 302 - "https://www.
example.com/eidm2/wf/admin?10&tab=users" "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox
/54.0"
127.0.0.1 - - [[20/Jul/2017:11:15:39 +0000]] "GET /eidm2/wf/admin?11&tab=overview HTTP/1.1" 200 9374
"https://www.example.com/eidm2/wf/admin?10&tab=users" "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko
/20100101 Firefox/54.0"
127.0.0.1 - - [[20/Jul/2017:11:15:39 +0000]] "GET /eidm2/res/style.css HTTP/1.1" 200 46700 "https://www.example.
com/eidm2/wf/admin?11&tab=overview" "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0"
127.0.0.1 - - [[20/Jul/2017:11:15:39 +0000]] "GET /eidm2/wf/admin?11-1.IBehaviorListener.0-body-workflowPanel-
panel-organizationTabContent-resultPanel&tab=overview&_=1500549339407 HTTP/1.1" 400 96 "https://www.example.com
/eidm2/wf/admin?11&tab=overview" "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0"
127.0.0.1 - - [[20/Jul/2017:11:15:39 +0000]] "GET /eidm2/res/logo_fi HTTP/1.1" 200 7245 "https://www.example.com
/eidm2/wf/admin?11&tab=overview" "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0"
```