SIEM Integration

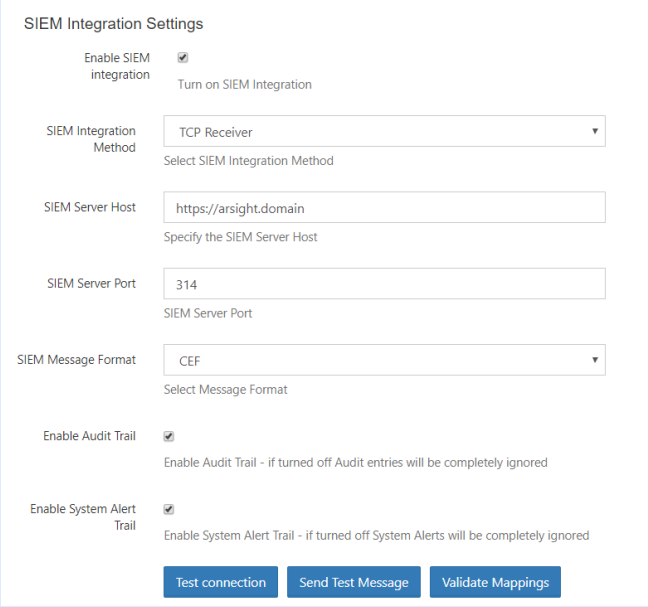
SIEM Integration

SIEM Integration is available from FileCloud 19.2

In the field of computer security, security information and event management (**SIEM**), software products and services combine security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by applications and network hardware.

Since version 19.2, FileCloud has allowed system administrators to integrate FileCloud's system alerts and auditing with external SIEM systems, enabling them to monitor all alerts and potential security issues in one place.

FileCloud SIEM Configuration



To configure SIEM Integration Settings

1. Open a browser and log into *Admin Portal*.
2. In the left navigation panel, under *SETTINGS*, select *Settings*.
3. On the *Manage Settings* screen, select the *Third Party Integrations* tab.
4. Select the *SIEM* tab.
5. Modify settings as needed.
6. Click *Save*.

The following options are available:

Option	Description
Enable SIEM Integration	Turns SIEM integration ON or OFF
SIEM Integration method	Specifies the SIEM Integration method. Following options are available: ▪ TCP Receiver - messages are sent to the specified SIEM server endpoint (host and port) via TCP socket connection ▪ UDP Receiver - messages are sent to the specified SIEM server endpoint (host and port) via UDP socket connection ▪ Syslog - messages are written directly to the Syslog, which can be imported by the SIEM server **Note:** SIEM software providers should specify supported integration methods in the SIEM documentation.
SIEM Server Host (TCP and UDP integration only)	URL or IP Address of the SIEM server.
SIEM Server Port (TCP and UDP integration only)	Port exposed by the SIEM Server for the given socket connection.

SIEM Message Format	Specifies the SIEM Message format. The following formats are available: ■ CEF - Common Event Format ■ LEEF - Log Event Extended Format NOTE: SIEM software provider should specify supported formats in the SIEM documentation.
LEEF Version (LEEF Format only)	Specifies the version of the LEEF format message. Available versions: ■ 1.0 ■ 2.0
Enable Audit Trail	Specifies whether Audit records should be processed and send to the SIEM Server. Please check the Managing SIEM mappings section for more details.
Enable System Alert Trail	Specifies whether System Alerts generated within FileCloud should be processed and send to the SIEM Server. Please check the Managing SIEM mappings section for more details.
Test Connection (TCP and UDP integration only)	Tests connection to the server specified by the Host and Port. **NOTE: All settings have to be saved first. Connection tests are based on the *currently* saved settings.**
Send Test Message	Sends a test message in the given format (CEF/LEEF) to the SIEM server specified by the Host and Port or saves a test message to the Syslog. **NOTE: All settings have to be saved first. Connection tests are based on the *currently* saved settings.**
Validate Mappings	Validates all defined mappings. Please check the Managing SIEM mappings section for more details.

Syslog Integration

In order to provide more flexibility, FileCloud allows admins to specify two important Syslog parameters - ident and facility. **Ident** specifies the name of the application logged in Syslog. **Facility** specifies where all FileCloud messages are sent and can be utilized by the system level Syslog configuration (e.g. in "rsyslog"). Both settings can be overridden in the *cloudconfig.php* configuration file by inputting the following settings:

- Ident - to specify ident value, add the following setting to *cloudconfig.php*

```
define('TONIDOCLOUD_SIEM_SYSLOG_IDENT', 'IDENT_VALUE');
```

  If no value is provided, by default it will be set to 'SIEM'.
- Facility -to specify ident value please add the following setting: to the *cloudconfig.php*

```
define('TONIDOCLOUD_SIEM_SYSLOG_FACILITY', LOG_LOCAL2);
```

  If no value is provided, by default it will be set to LOG_LOCAL5. Below is a full list of supported values.

LOG_AUTH	Security/authorization messages (use **LOG_AUTHPRIV** instead in systems where that constant is defined)
LOG_AUTHPRIV	Security/authorization messages (private)
LOG_CRON	Clock daemon (cron and at)
LOG_DAEMON	Other system daemons
LOG_KERN	Kernel messages
LOG_LOCAL0 ... **LOG_LOCAL7**	Reserved for local use. These are not available in Windows
LOG_LPR	Line printer subsystem
LOG_MAIL	Mail subsystem
LOG_NEWS	USENET news subsystem

LOG_SYSLOG	Messages generated internally by syslogd
LOG_USER	Generic user-level messages
LOG_UUCP	UUCP subsystem

LOG Values can also be seen in the official PHP documentation.

Please note that there are no quotation marks used for LOG values, as these have to be set to one of the PHP constants.

Managing SIEM Mappings

The biggest challenge when working with the external SIEM servers is to map messages existing in the system to the correct CEF/LEEF format. In order to allow administrators to have a full control how to represent FileCloud's System Alerts and Audit records in the external SIEM system a special, flexible mapping syntax is supported.

NOTE:

For this step you will need to access **WWWROOT.** It is typically located at:

Windows	Linux (later than Ubuntu 14.04)	Linux (earlier than Ubuntu 14.04)
c:\xampp\htdocs	**/var/www/html**	**/var/www**

Create and access SIEM mappings files:

Navigate to the following directory:

```
WWWROOT/app/siem/maps
```

It contains the following files:

```
auditmap-sample.php
systemalertsmap-sample.php
```

which store mapping samples for audit and system alerts respectively.

Modify the mappings to correspond to your system, and save them as **auditmap.php** and **systemalertsmap.php**.

- **auditmap.php** enables FileCloud to convert audit entries to the valid SIEM messages.
- **systemalertsmap.php** enables FileCloud to convert FileCloud's system alerts to the valid SIEM messages.

NOTE: Mappings are stored in the .php file, so they have to follow all PHP syntax rules as well as the internal mappings rules and syntax. To validate all mappings please navigate to Settings Third Party Integrations SIEM and click the Validate mappings button.

When you upgrade FileCloud, if you previously integrated with SIEM and already have auditmap.php and systemalertsmap.php files, you do not have to recreate or edit them unless you want to change existing mappings.

SIEM mapping format:

A sample SIEM mapping is a PHP array entry, which itself is an array. It contains following fields:

id (Required) - identifies the SystemAlert/Audit entry this map refers to. **NOTE: It can be a string literal which matches the audit operation name or one of the SiemArea values available in FileCloud, an array of values or a wildcard '*' that specifies that the mapping is applied to ALL audit entries/system alerts.**

prefilter (Optional) - A collection of preconditions that event has to meet in order to be processed and sent to the SIEM system. It is an array of filters, where each filter has the following format: property => value, where:

- property is a valid property available for the Audit / System Alert record (TBD - add lists of properties)
- value is a value that has to be matched in order to process the Audit / System Alert record, i.e.

Sample System Alert Mappings

```
'prefilter' => [
    'level' => SysAlert::SYSALERT_LEVEL_MELTDOWN
],
```

specifies that only System Alerts with the Meltdown criticality level would be sent to the SIEM server.

map (Required) - specifies the actual mapping between the FileCloud object being processed and the SIEM-formatted message that will be sent to the SIEM server. SIEM object as to contain the following four fields:

- eventClass - class of the event in the SIEM system.
- eventName - name of the event.
- severity - this is a SIEM side severity, which is a number from the 1-10 range.
- extension - a collection (array) of additional key value pairs that will be stored in the SIEM system (i.e. user that performed the action, ip address of the request, etc.). The key can be any arbitrary string.

To allow a very flexible way to resolve those mappings value a special 'language' was created. Values can be provided in any of the following ways:

- As a literal value (i.e. string or number), i.e.

  Sample System Alert Mappings

  ```
  'eventClass' => 'authentication',
  'eventName' => 'invalid login',
  'severity' => 3
  ```

- As a property biding that will resolve the value, based on the actual value provided by the FileCloud audit, system alert being processed:

  Sample System Alert Mappings

  ```
  'eventClass' => '$siemArea',
  'eventName' => '$description',
  'user' => '$username',
  'ip' => '$ip'
  ```

  Please check a full list of supported properties for more details. (TBD)

- As a method call:

  Sample System Alert Mappings

  ```
  'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'], ['$level']],
  ```

  NOTE: Users can create their own methods that can be utilized here. The first parameter is the PHP callback (class, method name) and the second parameter is the array of values (Optional) that will be processed by that callback. Parameters can be set to literal values or runtime-resolvable properties as described earlier. In FileCloud 19.2 *getSysAlertSeverity* is the only method available out of the box. It converts internal System Alerts severity into the 1-10 range required by SIEM integration in the following way:

  - Meltdown: 10
  - Critical: 7
  - Warning: 4
  - Information: 1

Sample mappings:

System Alerts:

Sample System Alert Mappings

```
//Report all meltdowns
$mappings[] = [
    'id' => '*', //Wildcard denotes all Alerts
    'prefilter' => [
        'level' => SysAlert::SYSALERT_LEVEL_MELTDOWN
    ],
    'map' => [
        'eventClass' => '$siemArea',
        'eventName' => '$description',
        'severity' => 10,
        'extension' => [
            'user' => '$username',
            'ip' => '$ip'
        ]
    ]
];

//AV system alert - infected file found
$mappings[] = [
    'id' => SiemArea::INFECTED_FILE,
    'map' => [
        'eventClass' => 'System Error',
        'eventName' => '$description',
        'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'], ['$level']],
        'extension' => [
            'user' => '$username',
            'ip' => '$ip',
            'path' => '$alertContext.filePath',
            'file' => '$alertContext.fileName'
        ]
    ]
];

//Type mismatch report
$mappings[] = [
    'id' => SiemArea::INVALID_FILE_TYPE,
    'map' => [
        'eventClass' => 'System Error',
        'eventName' => '$description',
        'severity' => [[SiemConversionHelper::class, 'getSysAlertSeverity'], ['$level']],
        'extension' => [
            'user' => '$username',
            'ip' => '$ip',
            'path' => '$alertContext.file'
        ]
    ]
];
```

Audit:

```
//Report all audit events
$mappings[] = [
    'id' => '*',
    'prefilter' => [],
    'map' => [
        'eventClass' => '$operation',
        'eventName' => '$operation',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
            'userAgent' => '$userAgent',
            'ip' => '$ip',
            'notes' => '$notes'
        ]
    ]
];


//Failed login attempt
$mappings[] = [
    'id' => 'loginguest',
    'prefilter' => [
        //List of conditions that audit entry has to met in order to be processed (or filtered out if
excluded option is there)
        'resultCode' => '0', //incidents only
        'exclude' => false// - optional 'include' is used by default
    ],
    'map' => [
        'eventClass' => 'login',
        'eventName' => 'Invalid login attempt',
        'severity' => 2,
        'extension' => [
            'user' => '$userName',
            'ip' => '$ip'
        ]
    ]
];
```