Policies

Chris Watson

November 2018

1 Policy 1

Attempting to download JOOSE2 coursework

A student wants to login to the service and browse/search for resources of their JOOSE2 course. They may even be searching for the JOOSE2 coursework file directly. This file should show to the student if (and only if) they have certain attributes such as enrollment in the JOOSE2 course. The metadata of the resource is compared to the student's account for this show/hide service. If they find the resource then the student would like to download the file and decrypt the contents using their Key.
Below are the defined attributes for the policy, where ***Subject*** **s**, ***Environment*** **e** and ***Resource*** **r** are defined as:

s ⇒ role, jobField, level, enrolledCourses, startDate, endDate, demonstratorClasses, accountStatus
r ⇒ releaseDate, owner
e ⇒ currentDate, network

With the policy defined as:

$$
\begin{aligned}
\text{Policy}(\mathbf{s},\mathbf{r},\mathbf{e}) \leftarrow\ & \text{owner}(\mathbf{r}) \equiv \mathbf{s} \\
& \vee\ (\text{role}(\mathbf{s}) \equiv \text{Staff} \\
& \quad \wedge\ \text{jobField}(\mathbf{s}) \equiv \text{Research \& Teaching}) \\
& \vee\ (\text{role}(\mathbf{s}) \equiv \text{Student} \\
& \quad \wedge\ \text{studentLevel}(\mathbf{s}) \equiv 2 \\
& \quad \wedge\ \text{enrolledCourses}(\mathbf{s}) \equiv 2001,\ 2008 \\
& \quad \wedge\ \text{currentDate}(\mathbf{e}) \geq \text{releaseDate}(\mathbf{r}) \\
& \quad \wedge\ \text{network}(\mathbf{e}) \equiv \text{Internal}) \\
& \vee\ (\text{studentRole}(\mathbf{s}) \equiv \text{Demonstrator UG} \\
& \quad \wedge\ \text{studentLevel}(\mathbf{s}) \equiv 4,\ \text{M},\ \text{PG} \\
& \quad \wedge\ \text{startDate}(\mathbf{s}) \leq \text{currentDate}(\mathbf{e}) \\
& \quad \wedge\ \text{endDate}(\mathbf{s}) \geq \text{currentDate}(\mathbf{e}) \\
& \quad \wedge\ \text{demonstratorClasses}(\mathbf{s}) \equiv \text{2JP},\ \text{2OOSE})
\end{aligned}
$$

Below we define an instance scenario with Subject $\mathbf{s}_0$, Resource $\mathbf{r}_0$ and Environment $\mathbf{e}_0$ that is intentionally designed to demonstrate an instance of the Policy which resolves to `True` (*and thus grants access to the resource*). This instance also acknowledges the existence of Subject $\mathbf{s}_1$, however we do not define their attributes, as the only important fact is that $\mathbf{s}_0 \neq \mathbf{s}_1$.

$\mathbf{s}_{0attrs}$ = role: Student
X studentLevel: 2
X enrolledCourses: 2001,2003,2007,2008,2021,2028

$\mathbf{r}_{0attrs}$ = owner: **s1**
X releaseDate: <DateTime>:2018-09-17 10:00:00.000Z

$\mathbf{e}_{0attrs}$ = network: Internal
X internalNetwork: DCS
X currentDate: <DateTime>:2018-09-19 16:14:36.000Z

Hence, we can fill out the above policy with the above attributes, defined for this instance of Policy($\mathbf{s}_0$,$\mathbf{r}_0$,$\mathbf{e}_0$) with the calculated values shown in **bold**:

Policy($\mathbf{s}_0$,$\mathbf{r}_0$,$\mathbf{e}_0$) $\leftarrow$ $\mathbf{s}_1 \equiv \mathbf{s}_0$
$\vee$ (**Student** $\equiv$ Staff
$\wedge$ **NULL** $\equiv$ Research & Teaching)
$\vee$ (**Student** $\equiv$ Student
$\wedge$ **2** $\equiv$ 2
$\wedge$ **2001,2003,2007,2008,2021,2028** $\equiv$ 2001, 2008
$\wedge$ **2018-09-19 16:14:36.000Z** $\geq$ **2018-09-17 10:00:00.000Z**
$\wedge$ **Internal** $\equiv$ Internal)
$\vee$ (**NULL** $\equiv$ Demonstrator UG
$\wedge$ **2** $\equiv$ 4, M, PG
$\wedge$ **NULL** $\leq$ **2018-09-19 16:14:36.000Z**
$\wedge$ **NULL** $\geq$ **2018-09-19 16:14:36.000Z**
$\wedge$ **NULL** $\equiv$ 2JP, 2OOSE)

As shown by the third set of conditions (highlighted in dark red) above, the Policy($\mathbf{s}_0$,$\mathbf{r}_0$,$\mathbf{e}_0$) instance resolves to `True`, since all 5 conditions within the third set evaluate to `True`. This resolves correctly, since the policy consists of 4 sets of conditions linked by `or` operators, thus it can be calculated to `False` $\vee$ `Unknown` $\vee$ `True` $\vee$ `Unknown` which clearly resolves to `True`.

2 Policy 2

Downloading minutes from class rep meeting

Student attended (or missed - we don't care) the class rep meeting but would like to find and download the official minutes from the meeting. There may be many minutes for class rep meetings available, and the student isn't necessarily sure if the latest meeting's minutes are uploaded yet. As such they need to browse all available minutes to find the latest uploaded. Only current class reps should have access to the minutes and other students should not even be able to see the minutes on the server. After finding the latest minutes available, the student's Key should be used with a CLI tool to decrypt the minutes for the student to view them.
Below are the defined attributes for the policy, where ***Subject*** **s**, ***Environment*** **e** and ***Resource*** **r** are defined as:

$\mathbf{s} \Rightarrow$ role, accountStatus, studentRole, startDate, endDate, demonstratorClasses, accountStatus
$\mathbf{r} \Rightarrow$ releaseDate, owner
$\mathbf{e} \Rightarrow$ currentDate

With the policy defined as:

$$
\begin{aligned}
\text{Policy}(\mathbf{s},\mathbf{r},\mathbf{e}) \leftarrow\ & \text{accountStatus}(\mathbf{s}) \equiv \text{Active} \\
& \wedge\ (\ \text{owner}(\mathbf{r}) \equiv \mathbf{s} \\
& \quad \vee\ (\ \text{role}(\mathbf{s}) \equiv \text{Student} \\
& \quad\quad \wedge\ \text{studentRole}(\mathbf{s}) \equiv \text{Class Rep} \\
& \quad\quad \wedge\ \text{startDate}(\mathbf{s}) \leq \text{releaseDate}(\mathbf{r}) \\
& \quad\quad \wedge\ \text{endDate}(\mathbf{s}) \geq \text{releaseDate}(\mathbf{r}) \\
& \quad\quad \wedge\ \text{startDate}(\mathbf{s}) \leq \text{currentDate}(\mathbf{e}) \\
& \quad\quad \wedge\ \text{endDate}(\mathbf{s}) \geq \text{currentDate}(\mathbf{e})\ ) \\
& \quad \vee\ (\ \text{role}(\mathbf{s}) \equiv \text{Staff} \\
& \quad\quad \wedge\ \text{startDate}(\mathbf{s}) \leq \text{currentDate}(\mathbf{e}) \\
& \quad\quad \wedge\ \text{endDate}(\mathbf{s}) \geq \text{currentDate}(\mathbf{e})\ )
\end{aligned}
$$