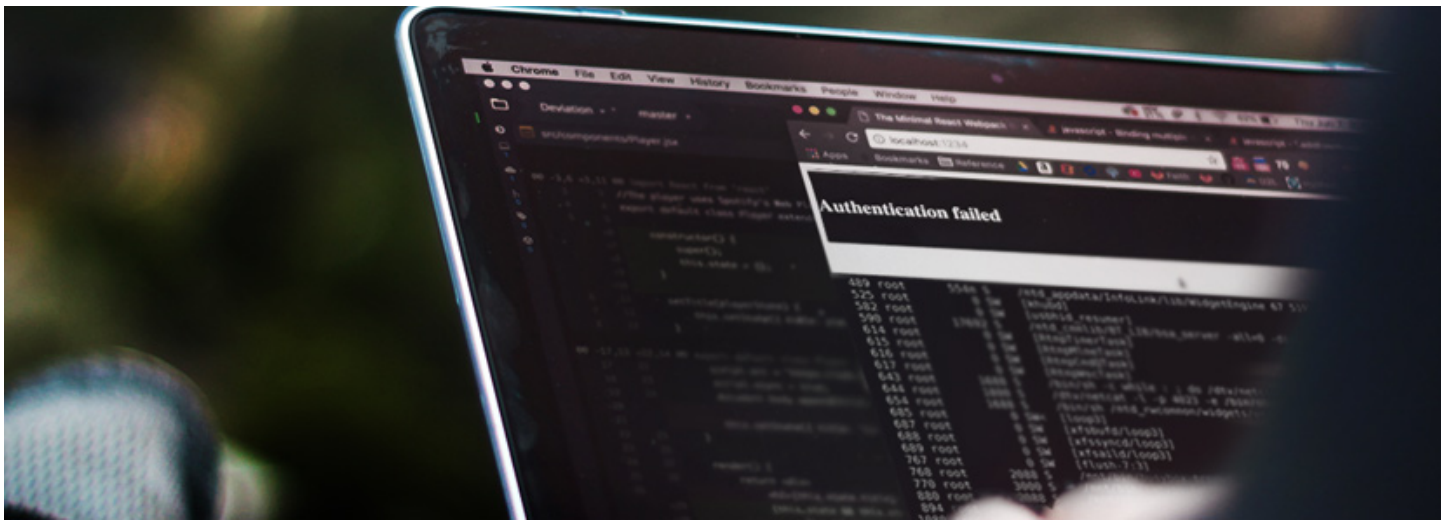
February 04, 2019

Role-based Access Control vs Attribute-based Access Control: How to Choose

Category: Security



Access control is a fundamental element of the security infrastructure of any company. Every security officer wants to apply the principle of less privilege, zero-trust, segregation of duties, and other best practices without harming the company workflow.

There are several approaches to organizing an access management system. In this article, we analyze the two most popular access control models: role-based and attribute-based. We'll talk out the pros and cons of each model, compare them, and see if it's possible to combine them.

What is role-based access control (RBAC)?

Role-based access control (RBAC) is an access control method based on defining employee roles and corresponding privileges within the organization. The idea of this model is that every employee is assigned a role. Every role has a collection of permissions and restrictions. An employee can access objects and execute operations only if their role in the system has the relevant permissions.

For example, a company's accountant should be allowed to work with financial information but shouldn't have access to client contact information or credit card data.

A user might be assigned to one or several roles. When a new employee comes to your company, it's easy to assign a role to them. And when someone leaves the company, you don't need to change the role parameters or a central policy.

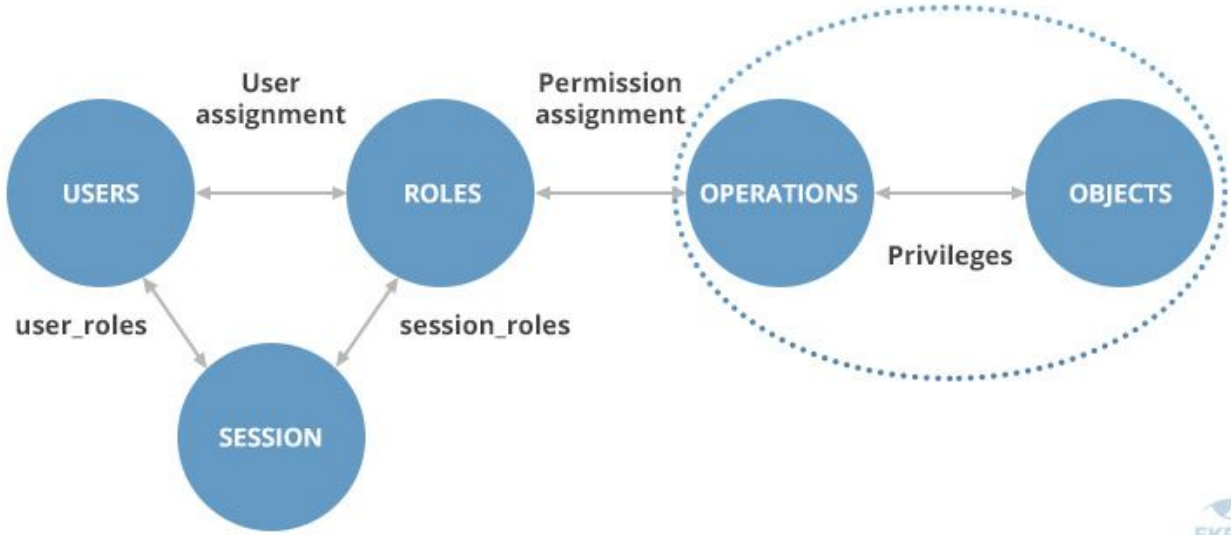
Let's consider the main components of the role-based approach to access control:

User – an individual (with UID) with access to a system

Session – a mapping between a user and a set of roles to which the user is assigned in the context of a working time
Object – a system resource that requires permission to access
Operation – any action in the protected network

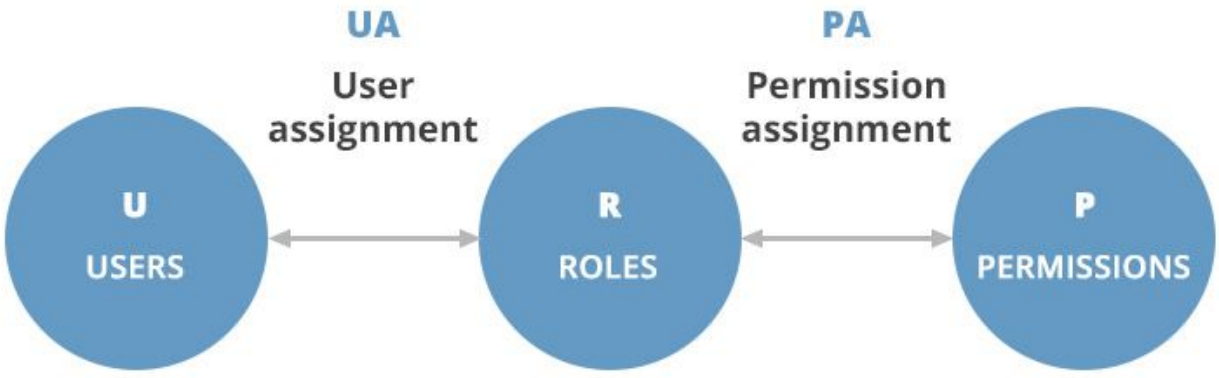
The basic rules of RBAC are:

A user can execute an operation only if there is a role assigned to the subject.
Identification and authentication are not considered operations.
All user activities are carried out through operations.



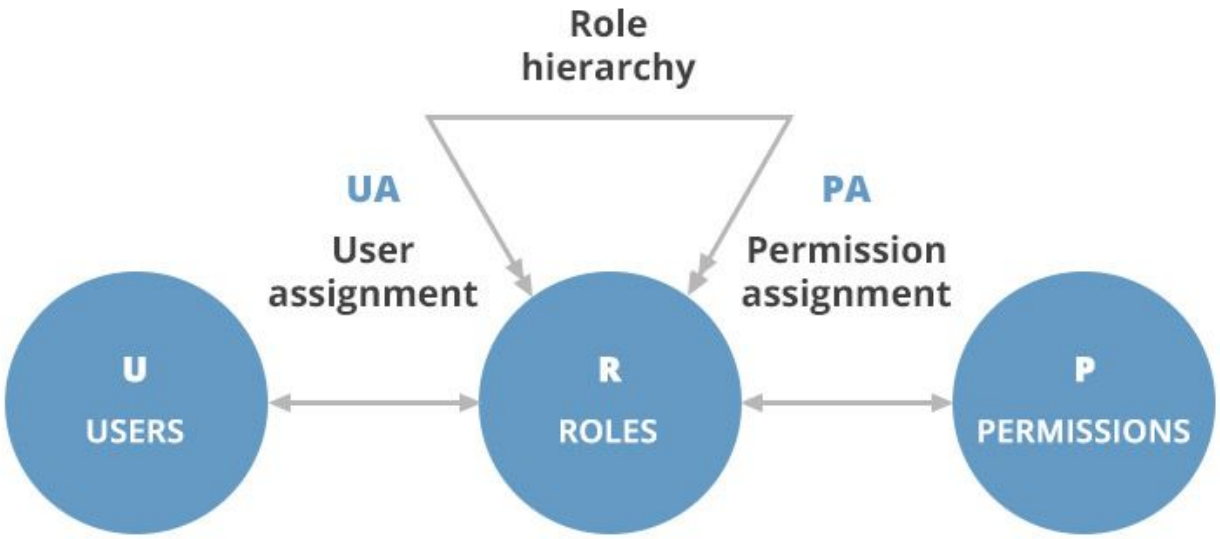
RBAC can be implemented on four levels, according to the **NIST RBAC model**. Each subsequent level includes the properties of the previous. Let's take a look at them:

1. Flat RBAC is an implementation of the basic functionality of the RBAC model. All users and permissions are assigned roles. Users obtain the permissions they need by acquiring these roles. There may be as many roles and permissions as the company needs. A single user can be assigned to multiple roles, and one role can be assigned to multiple users.

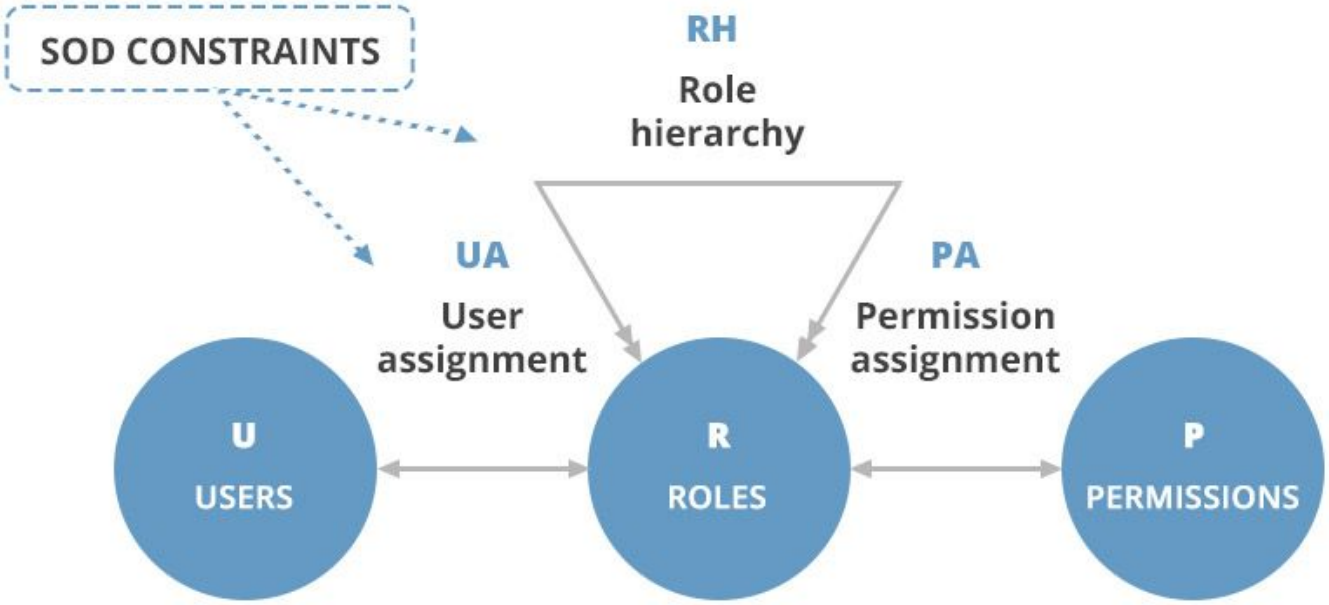


2. Hierarchical RBAC, as the name suggests, implements a hierarchy within the role structure. This hierarchy establishes the relationships between roles. Users with senior roles acquire permissions of all junior roles, which are assigned to their subordinates. The complexity of the hierarchy is defined by the needs of the company.

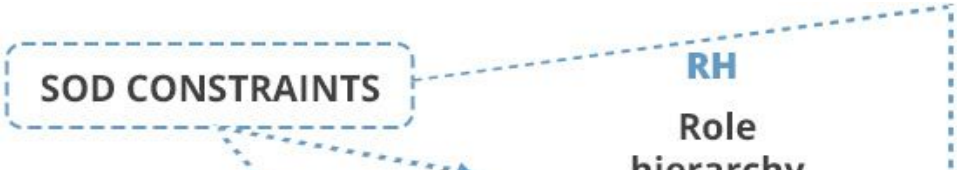
Contact us

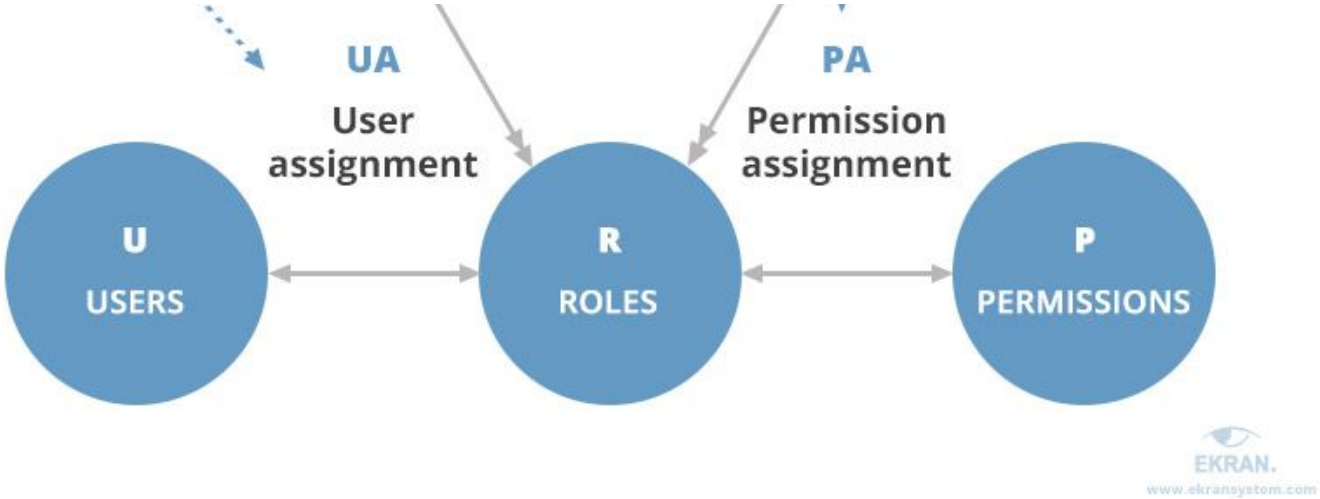


3. Constrained RBAC adds a **separation of duties** (SOD) to a security system. SOD is a well-known security practice when a single duty is spread among several employees. It's quite important for medium-sized businesses and large enterprises. Separation of duties guarantees that no work can introduce fraudulent changes to your system that no one else can audit and/or fix.



4. Symmetric RBAC supports permission-role review as well as user-role review. It allows identification of the permissions assigned to existing roles (and vice versa). For example, by identifying permissions of a terminated employee, the administrator can revoke the employee's permissions and then reassign the role to another user with the same or a different set of permissions.





Defining a role can be quite a challenge. You have to consider all the permissions a user needs to perform their duties and the position of this role in your hierarchy. If you assign too many permissions to a role, it will break the least privilege principle and may lead to privilege misuse.

Role-based access control is most commonly implemented **in small and medium-sized enterprises**. Such organizations typically have simple workflows, a limited number of roles, and a pretty simple hierarchy, making it possible to effectively determine and describe user roles.

Once all the necessary roles are set up, this model doesn't require a lot of maintenance and support from the IT department. Implementing RBAC can help you meet IT security requirements without much pain. On the other hand, creating a complex role system for a large enterprise may be challenging. The organization with thousands of employees can end up with a few thousand roles. This is known as role explosion, and it's unavoidable for a big company.

What is attribute-based access control (ABAC)?

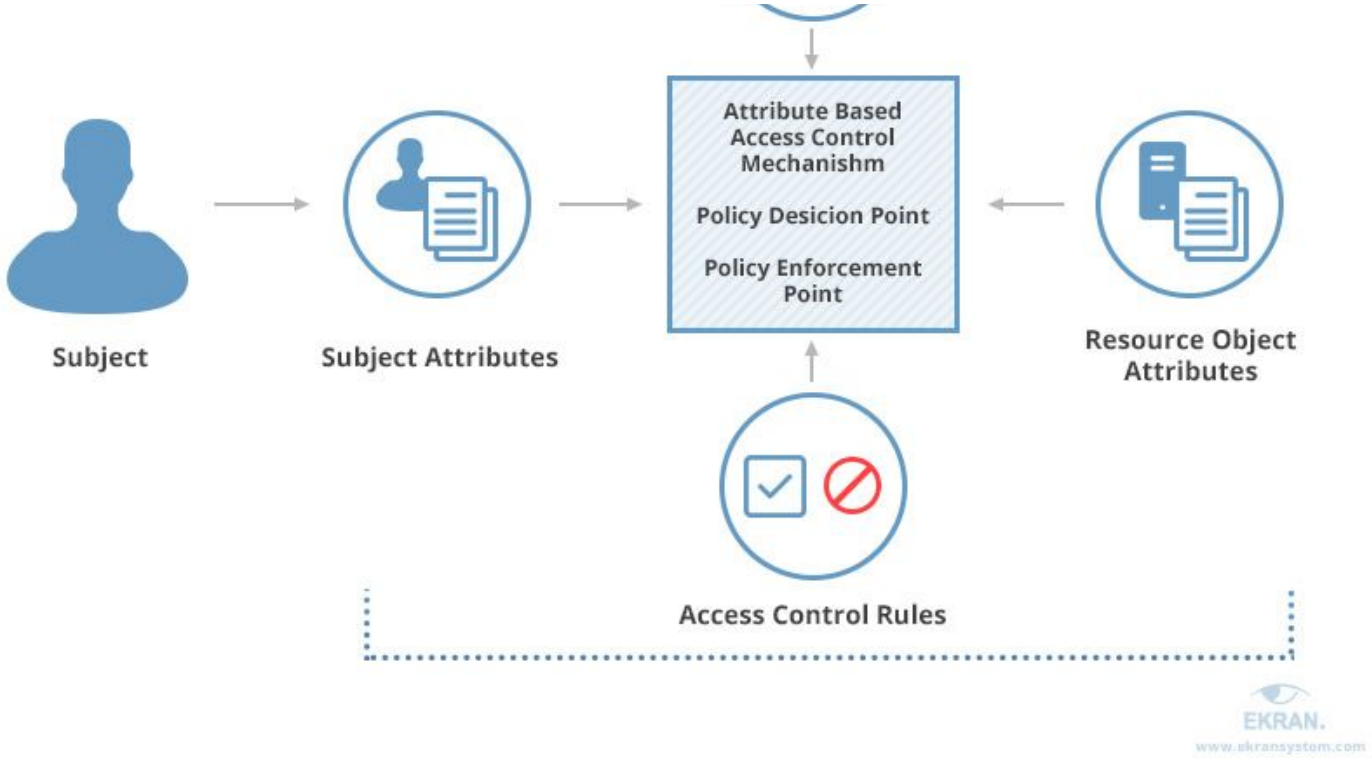
Attribute-based access control is a model that evolved from RBAC. This model is based on establishing a set of attributes for any element of your system. A central policy defines which combinations of user and object attributes are required to perform any action.

Let's consider the **main components** of the ABAC model according to NIST:

Attribute – a characteristic of any element in the network. An attribute can define:
- **User** characteristics – employee position, department, IP address, clearance level, etc.
- **Object** characteristics – type, creator, sensitivity, required clearance level, etc.
- Type of **action** – read, write, edit, copy, paste, etc.
- **Environment** characteristics – time, day of the week, location, etc.

Subject – any user or resource that can perform actions in the network; a subject is assigned attributes in order to define its clearance level
Object – any data stored in the network; objects are assigned attributes in order to describe and identify them
Operation – any action taken by any subject in the network
Policy – a set of rules allowing or restricting any action in your information retrieval system; rules are "IF/THEN" statements based on attributes of any element (user, resource, environment)

Unlike in RBAC, in ABAC you can even use attributes that aren't yet registered in the system but will appear during the work process.



This approach is **suitable for a company of any size** but is mostly used for large organizations. ABAC requires more time and effort than RBAC at the deployment and configuration stage, as security administrators need to define all attributes of the system. At first, you need to assign attributes to each system component manually.

But once you've created policies for most common job positions and resources in your company, you can simply copy them for every new user and resource. This is similar to how a role works in the RBAC model, but in the ABAC model, attributes can be modified for the needs of a particular user without creating a new role. Attributes make ABAC a more fine-grained access control model than RBAC.

RBAC vs ABAC

Let's compare these two popular approaches — role-based access control vs attribute-based access control — to determine the pros and cons of each.

RBAC pros and cons

RBAC is the most popular approach to restricting access. The main advantage of this model is that companies no longer need to authorize or revoke access on an individual basis, bringing users together based on their roles instead. Establishing a set of roles in a small or medium-sized company isn't challenging. On the other hand, setting up such a system at a large enterprise is no easy task.

There are several **limitations to the RBAC** model. You can't set up a rule using parameters that are unknown to the system before a user starts working. Permissions can be assigned only to user roles, not to objects and operations. Also, using RBAC, you can restrict access to certain actions in your system but not to certain data.

ABAC pros and cons

The key **benefit of ABAC** is that it grants access based not on the user role but on the attributes of each system component. This way, you can describe a business rule of any complexity. Even if you need to make certain data only accessible during work hours, it can be easily done with one simple policy. On top of that, ABAC rules can evaluate attributes of subjects and resources

As for **ABAC limitations**, this type of system is hard to configure due to the way policies must be specified and maintained. It's difficult to perform a before the fact audit and determine the permissions available to a specific user. It could be impossible to determine risk exposure for any given employee position.

Gartner **predicts** that **70% of all organizations will use ABAC by 2020**

To sum up, let's compare the key characteristics of RBAC vs ABAC:

Characteristic	RBAC	ABAC
Flexibility	✓ (For small and medium-sized organizations)	✓
Scalability	–	✓
Simplicity	Easy to establish roles and permissions for a small company, hard to maintain the system for a big company	Hard to establish all the policies at the start, easy to maintain and support
Support for simple rules	✓	✓
Support for complex rules	✓	✓
Support for rules with dynamic parameters	–	✓
Customizing user permissions	– (Every customization requires creating a new role)	✓
Granularity	Low	High

Combining RBAC and ABAC

Companies often start with implementing a flat RBAC. This model is easier to set up and maintain. As organizations grow and manage more sensitive data, they realize the need for a more complex access control system. RBAC and ABAC can be used together, with RBAC doing the rough work and ABAC complementing it with finer filtering.

This access model is also known as **RBAC-A**. There are **three RBAC-A approaches** that handle relationships between roles and attributes:

> **Attribute-centric**. A role becomes the name of one of the user attributes. It resembles a job title. The "role" attribute in such a model is used to mark a set of attributes required for a certain position.
> **Role-centric**. Attributes are added to constrain roles. In such a model, attributes can reduce permissions available to a user. This approach strengthens the security of your data.
> **Dynamic roles**. Attributes such as time of day are used to determine the subject's role. In some cases, a user's role can be fully determined by dynamic attributes.

In addition, there's a new method called **next generation access control (NGAC)** that's currently being developed by **NIST**. Its based on ABAC but implements a more refined approach to policies. For example, NGAC supports several types of policies simultaneously, including ones that are applied both in the local environment and in the network.

Conclusion

Access management is an essential component of any reliable security system. Whether you choose role-based or attribute-based access control, you'll need a robust instrument to authenticate and identify your users.

Erkan System offers **identity management** (two-factor authentication, secondary authentication, etc.) and **access management** (PASM, one-time credentials, etc.) functionality that works on a wide range of platforms and supports virtually any network architecture. Thanks to its flexible licensing scheme, Ekran System is suitable for both small businesses and large

YOU MAY ALSO LIKE

Ekran System Gets Two Prestigious Awards From FinancesOnline

February 01, 2019

Ekran System is appreciated by our customers and recognized by industry experts as one of the best insider threat prevention platforms. And we're proud to...

Zero Trust Model: Can Trusting No One Be the Answer to Your Cybersecurity Problems?

January 24, 2019

When trying to create a safe network, organizations usually use a classic perimeter strategy. This strategy presumes that all users, devices, and endpoints...

How Can EkranSystem Protect You Against Infected USB Devices?

August 31, 2018

Most stationary workstations have at least two USB devices plugged in all the time: a keyboard and a mouse. Apart from that, we occasionally connect mass storage...

Portrait of Malicious Insiders: T Characteristics, and Indicators

June 12, 2018

While organizations are spending a g deal of money protecting against unauthorized access from the outsic various industry analyses have revea

TRY IT NOW

Get started today by deploying a trial version in your company or try a free demo online

DOWNLOAD TRIAL VERSION

TRY ONLINE

SUBSCRIBE TO UPDATES

Your email

SUBSCRIBE