SEON

Guide

Guide to Fraud Detection Rules & How to Choose a Solution

SEON Technologies Ltd.
seon.io

info@seon.io
+44 20 8089 2900

Table of Contents

Fraud detection rules are the cornerstone of a good risk management strategy and, importantly, of scaling it along with your growth as a company.

Let's examine them in more detail below, including concrete examples you can leverage today.

What Is a Fraud Detection Rule?

A fraud detection rule is a condition that helps you decide if an activity is fraudulent or not. The rule will be put in place to let you **review, decline, or approve** a user action. It can be based on correlation, statistics, or logical comparison.

A fraud rule needs data to be activated. For instance, the data could be an IP address.

Let's say that you know that an IP address belongs to a fraudster and it appears on a blacklist. The most basic type of rule based on if/then logic could be:

```
if IP address = 192.168.1.1, then block website access.
```

Of course, having access to more data (particularly via data enrichment) allows you to get more precision when deciding what is considered fraud or not.

Meanwhile, **stacking multiple fraud rules allows you to perform fraud scoring**, which businesses use to mitigate risk how they see fit.

Rule-based fraud detection runs the gamut from basic, static rules, all the way to complex velocity checks, where you look at certain actions within a specific time frame. The latter is designed to analyze user behavior rather than looking at single data points.

Rule-Based Systems for Fraud Detection

Not all fraud rules are created equal. Let's look at five different types of rules with clear examples based on the SEON platform.

Static Rules for Fraud Detection

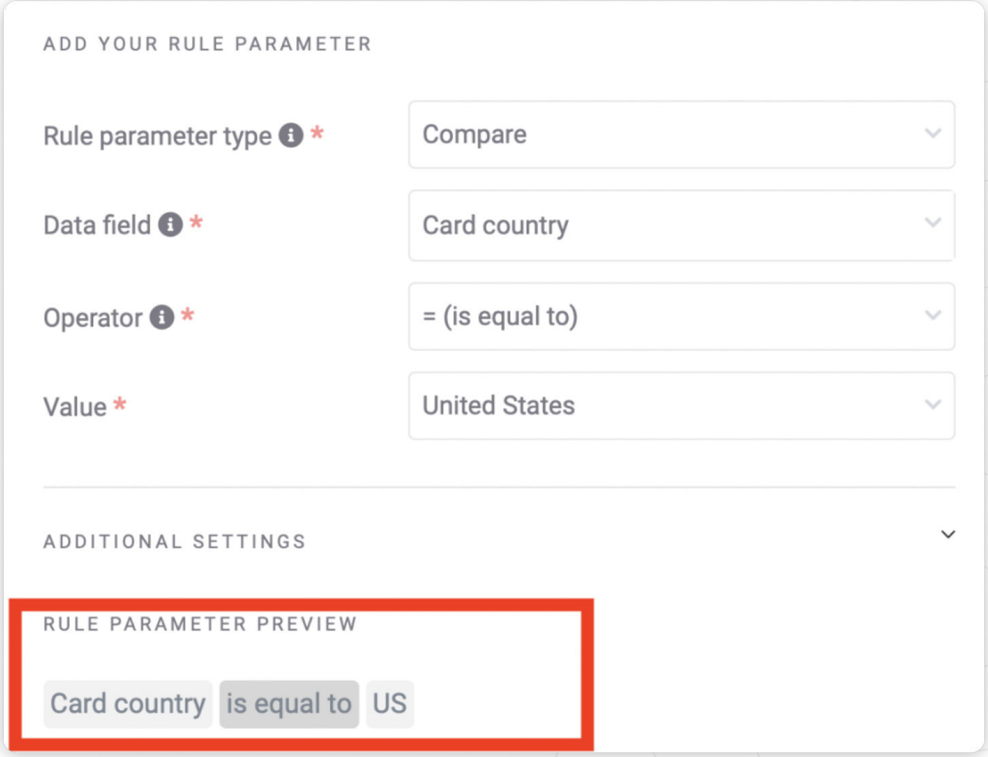
A static rule is the most basic form of fraud rule and tends to follow a simple if/then logic. It is considered static when the outcome of the rule is strict and inflexible – for instance, blocking a user action.

Historically, the earliest static rules had to do with IP addresses. If an IP was found on a blacklist, the user would be blocked.

However, **static rules can be as creative as you want** based on the data you have at hand.

Let's say, for instance, that you have access to IP information and credit card details. Your static rule could be made up of two parameters.

The first parameter will flag credit card details that point to the US.



ADD YOUR RULE PARAMETER

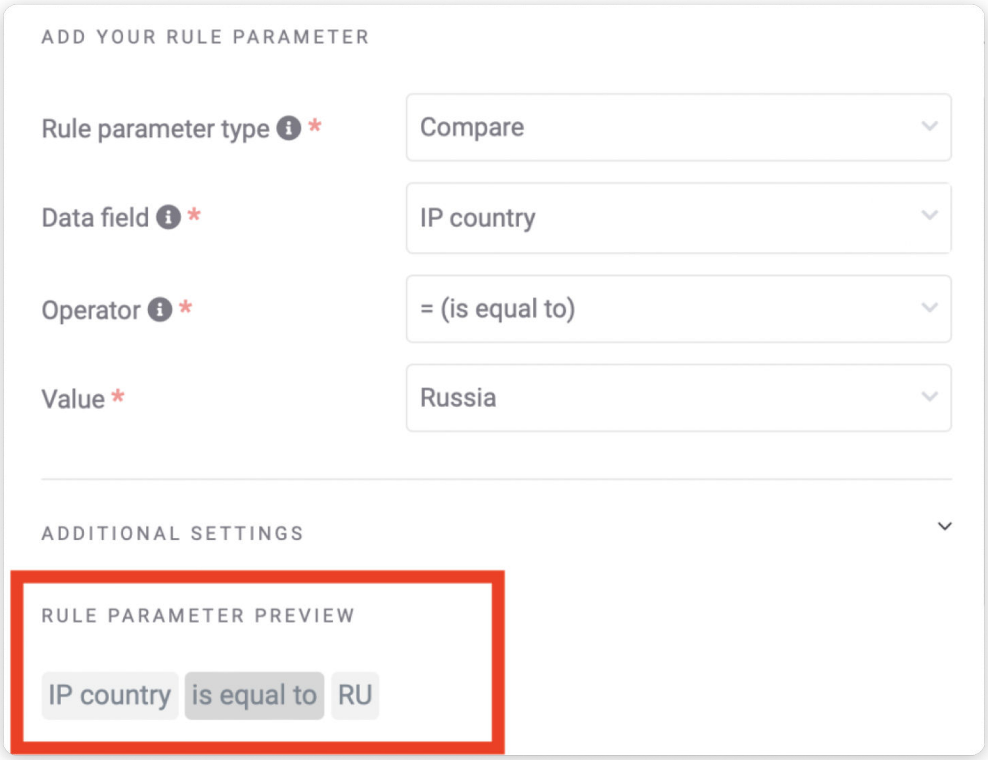
Rule parameter type *	Compare
Data field *	Card country
Operator *	= (is equal to)
Value *	United States

ADDITIONAL SETTINGS

RULE PARAMETER PREVIEW

Card country is equal to US

The second parameter will look at IP addresses that point to Russia.



ADD YOUR RULE PARAMETER

Rule parameter type *: Compare

Data field *: IP country

Operator *: = (is equal to)

Value *: Russia

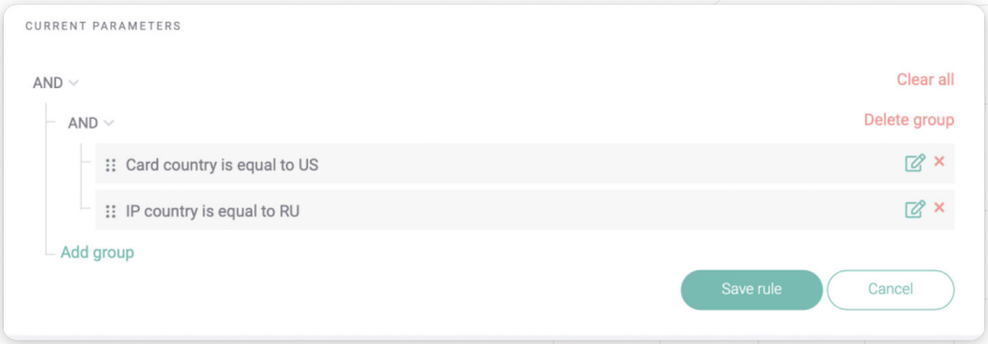
ADDITIONAL SETTINGS

RULE PARAMETER PREVIEW

IP country is equal to RU

Then our rule can be as follows:

```
If a user's credit card country points to the US but their IP points to
Russia, then the transaction should be blocked.
```



CURRENT PARAMETERS

AND

AND

Card country is equal to US

IP country is equal to RU

While static rules can be extremely sophisticated, one clear downside is their lack of flexibility. In the fraud prevention world, this is reflected in high numbers of **false positives**.

False positives are bad for your bottom line (because you block legitimate customers), and terrible for your business reputation. Good users whose actions are automatically blocked will be understandably frustrated and potentially take their custom elsewhere.

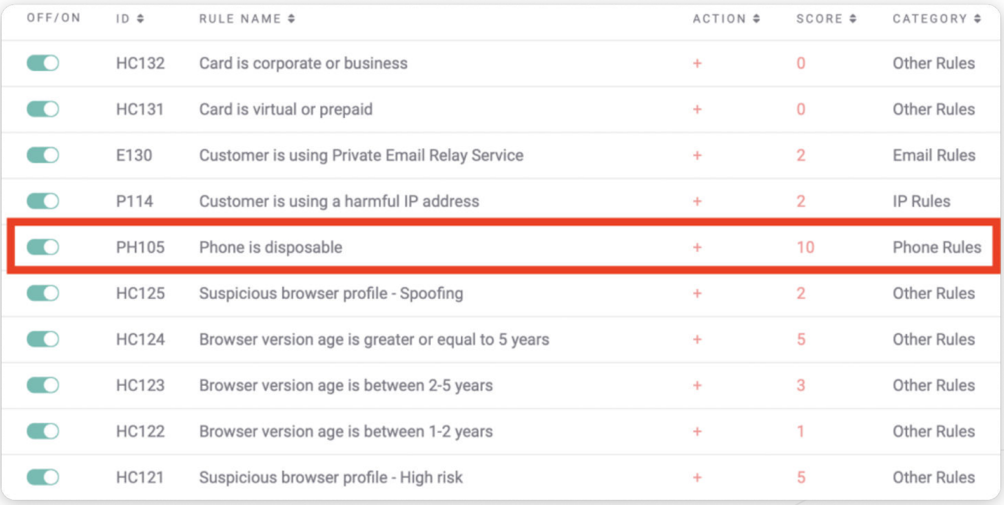
They could even spread the word to their friends and acquaintances.

This is why it's often better to use fraud rules as indicators. Enter scoring fraud rules...

Scoring Rules for Fraud Detection

Fraud rules don't always have to block actions. Sometimes, you want them to inform your risk strategy, and this is done thanks to risk scores.

Below are examples of risk scoring fraud rules. Notice the "score" column: Each of them adds a number of points to ultimately calculate how risky this user action is – and thus what the system should do about it.

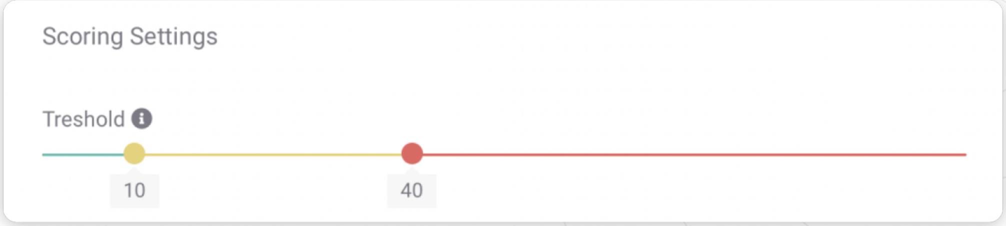


OFF/ON	ID ⬍	RULE NAME ⬍	ACTION ⬍	SCORE ⬍	CATEGORY ⬍
●	HC132	Card is corporate or business	+	0	Other Rules
●	HC131	Card is virtual or prepaid	+	0	Other Rules
●	E130	Customer is using Private Email Relay Service	+	2	Email Rules
●	P114	Customer is using a harmful IP address	+	2	IP Rules
●	PH105	Phone is disposable	+	10	Phone Rules
●	HC125	Suspicious browser profile - Spoofing	+	2	Other Rules
●	HC124	Browser version age is greater or equal to 5 years	+	5	Other Rules
●	HC123	Browser version age is between 2-5 years	+	3	Other Rules
●	HC122	Browser version age is between 1-2 years	+	1	Other Rules
●	HC121	Suspicious browser profile - High risk	+	5	Other Rules

This business has clearly decided that the most damning evidence that someone is a fraudster is a disposable phone number. They have assigned it a score of +10.

Note that fraud scores aren't standardized. Some platforms, for instance, decide that the lower a score, the riskier the customer. And the scoring works along the same lines too.

The key, however, is to **adapt the risk scores to your own business needs**. You can also automate fraud prevention by setting thresholds based on the fraud scores.



Scoring Settings

Treshold

In the example above, green (0–10 points) means **APPROVE**, yellow (10–40 points) means **REVIEW**, and red (40+ points) means **DECLINE**. On the SEON platform, these can be easily adjusted at will.

Velocity Rules for Fraud Detection

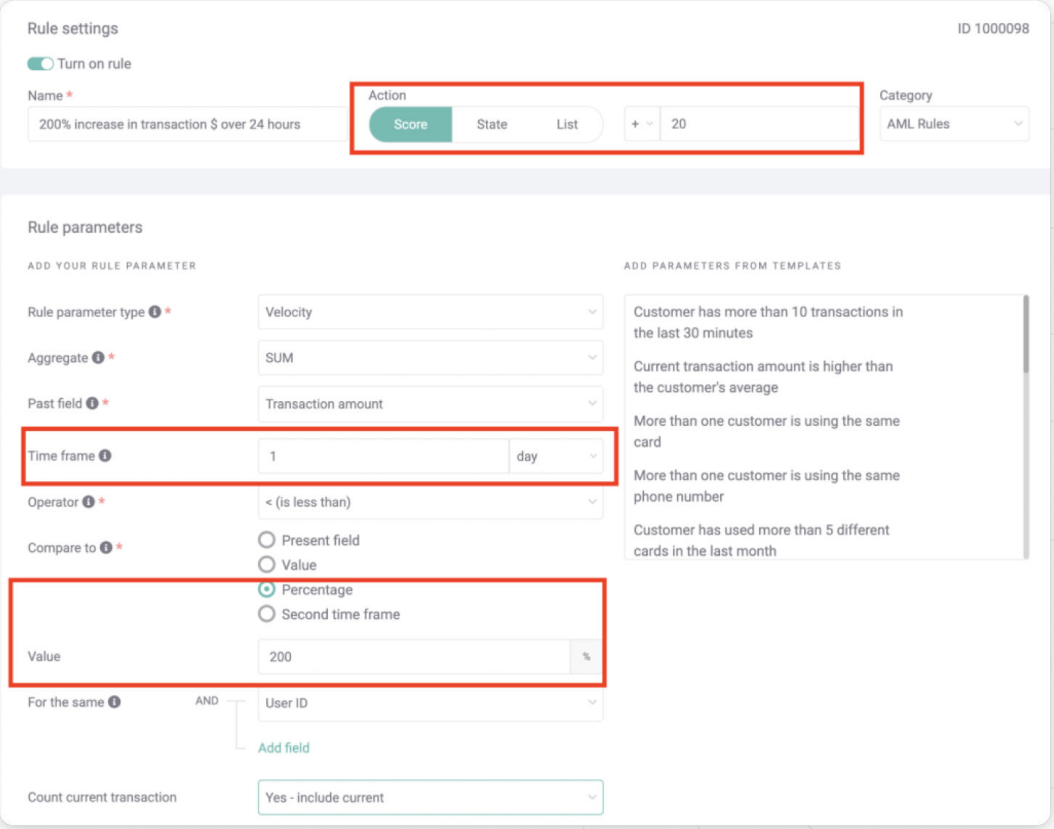
Going up a notch in sophistication, we have velocity checks, or velocity rules.

These rules attempt to **understand user behavior by looking at set actions over a time period**.

A good example of a velocity rule would be at the login stage. One way to prevent an account takeover attack (ATO) from succeeding is to make sure fraudsters can't attempt too many logins. If you have no barriers there, they can use credential stuffing or brute force to crack the login and password details.

This is where a velocity rule would come in handy. Your rule could simply look at the number of login attempts within, say, 5 minutes.

Another example would be to monitor suspicious movements of money. This can be useful in the context of AML (anti money laundering).



Rule settings

ID 1000098

Turn on rule

Name *: 200% increase in transaction $ over 24 hours

Action: Score | State | List + 20

Category: AML Rules

Rule parameters

ADD YOUR RULE PARAMETER

Rule parameter type *: Velocity

Aggregate *: SUM

Past field *: Transaction amount

Time frame: 1 day

Operator *: < (is less than)

Compare to *:
○ Present field
○ Value
● Percentage
○ Second time frame

Value: 200 %

For the same AND User ID

Add field

Count current transaction: Yes - include current

ADD PARAMETERS FROM TEMPLATES

Customer has more than 10 transactions in the last 30 minutes

Current transaction amount is higher than the customer's average

More than one customer is using the same card

More than one customer is using the same phone number

Customer has used more than 5 different cards in the last month

It may look intimidating, but in essence, it's quite simple. Here, you are looking at an increase in spending (more than 200%) over a 24-hour period. If that happens, it will trigger this rule.

In the example above, we've set up the system so that the rule adds a score of +10, but that's up to you. A realistic option could also be automatically pausing the transaction while it's sent for manual review.

Machine Learning Rules for Fraud Detection

Last but not least, we can look at what happens when you have a system designed to learn programmatically from your previous fraud cases.

Such a system is built on artificial intelligence and the idea is simply that it can analyze copious amounts of data to **spot suspicious patterns** as they emerge. The system will then make suggestions that are relevant to you only.

Of course, AI (or machine learning in our case) won't know if the rules are good or not. You'll need to manually review and test them yourself – either on historic data or on live data in a sandbox environment.

On the SEON platform, the machine learning engine runs in real-time and gathers data from transactions and scoring as it happens.

As a result, the more you use it, the more accurate the suggestions are going to be – while their remain easily **explainable** thanks to the **whitebox** nature of the module.

Benefits of a Rules-Based Fraud Prevention Solution

As you can see from the examples above, rule-based fraud prevention varies greatly depending on the kind of rules you deploy. However, there are undeniable benefits, including:

The ability to scale and automate

While it’s possible for a small business to manually review each user action, you can only really scale your fraud prevention strategy by automating checks based on fraud rules.

Dynamic friction

A great benefit of rule-based detection is that you can still manually review cases that fall within a grey area. That means obvious fraudsters are blocked and legitimate users are let through with no friction. To ensure you don't lose any good customers, you can introduce extra checks for those who require a little extra vetting instead of blocking them outright.

Reduce fraud rates over time

Rule-based fraud prevention is data-driven. This means you can monitor results, set KPIs, and tweak rules to align with your strategy. Thanks to machine learning, you can even get suggestions that are relevant based on your unique business use case.

How to Choose a Rule-Based Fraud Detection Solution

There are as many fraud prevention strategies as there are businesses. Still, some key fundamentals should apply to your rule-based fraud detection solution:

- **Is it effective?** Are you getting the results you were hoping for? Some fraud providers favor an approach based on shared blacklists, while others prefer relying on the power of real-time data.
- **Is it flexible?** While rule-based systems sound inflexible, it doesn't have to be the case. Finding a solution that lets you stack rules, use fraud scoring, custom fields, or even machine learning suggestions will reap better rewards in the long run. And more importantly, consider whether you can deploy the right rules based on your vertical. An iGaming company has very different needs from, say, a neobank.
- **Is it scalable?** How many rules does your system support? Can you run **thousands of API calls per day**? And, if it's hosted online, what kind of uptime rate are you looking at?
- **Is it expensive?** Will you pay per API call or as a lump sum for a multi-year contract? Micro-fee vs chargeback guarantee?
- **Is it easy to integrate?** Could you get started today? How many sprints would it take to fully integrate with your platform?

How SEON Does Rule-Based Fraud Detection

At SEON, we put flexibility and customization at the core of all our anti-fraud features That includes the ability to:

- **Deploy the solution rapidly, without any disruption.**
- **Get started with risk rules based on similar fraud cases in your industry.**
- **Completely customize them to your unique business needs.**

- **Receive suggestions from a powerful whitebox machine learning engine.**
- **Harness the benefits of blackbox AIs to spot new patterns.**
- **Easily add and edit rules, as well as activate or deactivate them.**

It's all designed to score, understand, and monitor risk at your company however you see fit – and with the same degree of flexibility when it comes to integration and payment.

FAQ

Why choose a rules-based fraud prevention solution?

All good fraud prevention solutions should offer rule-based settings. These are designed to let you automatically approve, review, or block user actions. The advantage over manual review is that you can automate fraud prevention and scale as your business processes more applications, transactions, or logins.

What is the most common detection method for fraud?

The most common fraud detection method is static rules that follow an if/then logic. For instance, if the IP address appears on a blacklist, then you should block the user from accessing your website.

What type of algorithm is used for fraud detection?

Fraud prevention algorithms allow companies to recommend whether you should block or allow a user action based on a scoring system. The algorithms tend to be proprietary. However, you can manually adjust the thresholds of your risk rules to take control over the algorithms manually.

SEON

SEON Technologies Ltd.
seon.io

info@seon.io
+44 20 8089 2900