White Paper

Smart Sense

New Role of Risk Signals

NOTICE TO LICENSEE:

This source code and/or documentation ("Licensed Deliverables") are subject to Nok Nok Labs, Inc. intellectual property rights under U.S. and international Copyright laws.

These Licensed Deliverables contained herein is PROPRIETARY and CONFIDENTIAL to Nok Nok Labs, Inc. and is being provided under the terms and conditions of a form of Nok Nok Labs, Inc. software license agreement by and between Nok Nok Labs, Inc. and Licensee ("License Agreement") or electronically accepted by Licensee. Notwithstanding any terms or conditions to the contrary in the License Agreement, reproduction or disclosure of the Licensed Deliverables to any third party without the express written consent of Nok Nok Labs, Inc. is prohibited.

NOTWITHSTANDING ANY TERMS OR CONDITIONS TO THE CONTRARY IN THE LICENSE AGREEMENT, NOK NOK LABS, INC. MAKES NO REPRESENTATION ABOUT THE SUITABILITY OF THESE LICENSED DELIVERABLES FOR ANY PURPOSE. IT IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED WARRANTY OF ANY KIND. NOK NOK LABS, INC.DISCLAIMS ALL WARRANTIES WITH REGARD TO THESE LICENSED DELIVERABLES, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE. NOTWITHSTANDING ANY TERMS OR CONDITIONS TO THE CONTRARY IN THE LICENSE AGREEMENT, IN NO EVENT SHALL NOK NOK LABS, INC. BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THESE LICENSED DELIVERABLES.

U.S. Government End Users. These Licensed Deliverables are a "commercial item" as that term is defined at 48 C.F.R. 2.101 (OCT 1995), consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (SEPT 1995) and is provided to the U.S. Government only as a commercial end item. Consistent with 48 C.F.R.12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (JUNE 1995), all U.S. Government End Users acquire the Licensed Deliverables with only those rights set forth herein.

Any use of the Licensed Deliverables in individual and commercial software must include, in the user documentation and internal comments to the code, the above Disclaimer and U.S. Government End Users Notice.

Table of Contents

Introduction

Computers operate using binary true/false patterns, making it straightforward for typical programming languages to handle “if x then y, else z” scenarios. But not all scenarios are suitable for binary patterns. Human beings have always grappled with uncertainty and probabilities, and over time have honed the ability to perceive complex information and reduce it to manageable functions in order to excel in the evolutionary “survival of the fittest” game. Take, for example, a farmer deciding whether to plant potatoes today or next month. This decision involves assessing the variables and weighing potential risks, such as the risk of an early or late freeze that would affect young plants. Humans have not always had computers and computer models to assist in decision making, and have developed the skill to interpret observable weather signs to make such predictions, demonstrating the ability to navigate and mitigate certain risks.

This observation can be applied to how online access to computer systems (authentication) has evolved over time. Access to these systems began with a simple (and binary) method: the use of a password. If a password and username match, then access is granted. This approach, based on deterministic comparison and boolean results, is straightforward to implement. However, the industry has learned that it is not the best practice to solely rely on a password. Other information outside of the binary system of the password may be needed in order to address less obvious risks. For example, how does the risk profile change when there is a sign-in attempt from an IP address associated with an internet provider in a location far from where the user is typically located? It is possible, and perhaps even probable that user access should be denied - even when a matching username and password are presented. The additional information about user location is a signal – a signal that is an input into the overall risk profile of the potential transaction. Because this information is available, in addition to the use of the user name and password, we see today many companies that specialize in protecting against fraud by processing hundreds or even thousands of risk signals.

However, there are tradeoffs. Relying too much on risk signals may lead to false declines which causes significant user dissatisfaction. For example, a user may be traveling in another country – and because the IP address comes from a different location, the user may be denied access. The industry has seen a significantly higher false-decline rate in countries that depend on risk based authentication and researchers have observed that “42% of consumers say they will be put off from returning to an app or website following one false decline” [1].

The role of risk signals must move away from being the primary method for protecting against account takeover attacks. Instead, it needs to evolve to become a “helper” signal that detects anomalies and improves user experience at the same time.

Understanding Risk Signals

Not all risk signals are the same. We have seen the IP address as one very commonly used risk signal. A variety of other signals are in use today such as: (i) the “User Agent” strings provided by web browsers, (ii) list of installed fonts, (iii) accepted download types, and (iv) even user typing behavior. These legacy risk signals were typically introduced to compensate for the weaknesses in password-based authentication. Many of these risk signals were designed to recognize *known devices*, i.e. devices that were previously used by the legitimate user. However, all these signals have one thing in common: they are publicly observable. Meaning, any relying party – including a MITM attacker – would be able to harvest this information by convincing/tricking the user to visit their (not legitimate) web site / phishing site.

Some of those signals are provided by an *unverifiable* client component (i.e., a client component with security characteristics that are unknown to the relying party and that do not provide methods to remotely assess its integrity) – typically the user’s web browser. In this case, it is the browser providing the user agent information, the list of installed fonts, accepted download types and even the typing behavior is reported by JavaScript code running in an unknown client environment. The IP address is different, however, because the IP address is endorsed by network components different from the client itself.

Therefore, signals that are (1) publicly observable and (2) depend on an unknown client component could easily be forged by a malicious entity. More on this point to follow.

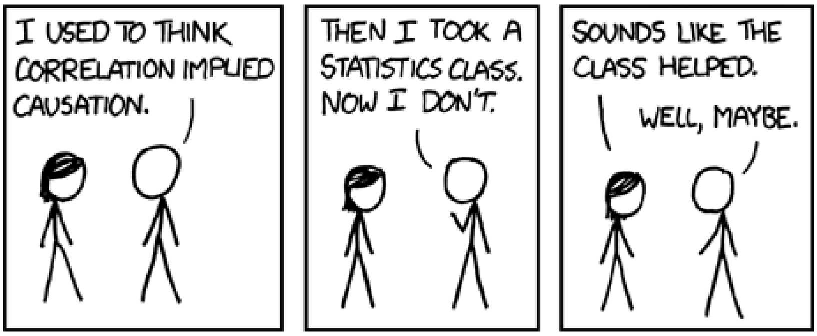
Correlated Risk Signals

Often it makes sense that a single risk signal might not significantly reduce fraud on its own. But, if there are hundreds of combined signals, ideally with AI/ML, it “adds up”. Behind this theory, there is the math of conditional probabilities [2]. In an ideal case, all risk signals would be completely independent and the resulting probability of combining the various signals would be a simple multiplication of the probability of each individual risk signal being incorrect. So if each probability is just slightly below 1, the result would be really small - and the method could be statistically effective.

However, if the risk signals are correlated, this approach doesn’t work. In the worst case two or more risk signals may be interdependent and they may also be jointly influenced.

Noisy Risk Signals

Risk signals do not necessarily indicate the root cause of an attack. Instead, there is merely a correlation between the signal and an increased level of fraud.


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

XKCD Correlation [3]

As a result, there are *false positives* and *false negatives*. Meaning: sometimes the risk signal incorrectly indicates a higher likelihood of fraud in the case of legitimate transactions (false positive), and other times the risk signal may incorrectly indicate a lower likelihood of fraud in the case of fraudulent transactions (false negative). The latter is expected - 100% of the fraudulent cases won't be detected using risk signals. But, the former (i.e., false positives) leads to a poor user experience. In the worst case, the system will block a legitimate transaction from being conducted. This outcome frustrates users and ultimately impacts revenue.

Note, this scenario and result is not just theoretical. Statistical studies show that the false decline rate in the US is approximately twice as high as the false decline rate in the EU. The US primarily uses risk based methods for payment authentication, whereas in the EU the PSD2 directive requires strong customer authentication in many cases – which is a deterministic method [1].

Robustness of Risk Signals

There is one more difference worth noting: Risk signals can also be intentionally manipulated. Sometimes attackers intentionally manipulate the system to trigger a false negative. This weakness against manipulation has an impact on the assumption of risk signals to be uncorrelated. Said differently, it is easy for a man-in-the-middle (MITM) to report any user agent, any list of installed fonts, any accepted download types and even simulate typing behavior. This shortcoming is the Achilles heel of risk signals in general as the use of hundreds of risk signals that can be jointly attacked does not deliver increased security that is expected from (independent) signals.

To make it more concrete, there are attackers that collect the observable signals [4] from real devices, so they can be replayed for attacks. In other words, a relying party cannot distinguish between passwords or risk signals coming from the legitimate user/device or an attacker. And attackers can easily replay all publicly observable risk signals. While this is currently not an industry reported issue for typing behavior or mouse movements, there is no reason why it couldn't be done in the near future.

Conclusion

With this ability to harvest and replay observable signals, almost all risk signals could be strongly correlated when under attack. As a result, signals won't continue to provide value. These signals are not needed when *not* under attack, but when under attack, they don't end up adding much value – depending on the type of attack, of course. The apparent user could be asked if this is an attack, where honest users would say no and honest attackers would say yes!

The IP address signal is slightly different. The client is unable to report arbitrary IP addresses because the IP address is assigned by the network infrastructure. In practice, clients get slightly different IP addresses every day. But if an attacker has access to a global botnet, an infected machine with an IP address similar to the attacked user could be used - as IP addresses are also publicly observable.

Robust Signals

In order to make authentication robust against attacks, we need to use a method that doesn't merely send publicly observable data from the client to the server. In other words, we need to move away from bearer tokens and use cryptographic challenge response protocols. FIDO and passkeys implement such an approach. Using FIDO/Passkeys uplevels the security level to trust-on-first-use (TOFU) [5]. This makes it much harder for attackers. With cryptographic authenticator attestation, even the highest security levels (like NIST SP 800-63 AAL3) can be reached [6].

FIDO/passkeys provide a strong signal on whether to "let the user in". It is in use globally in large organizations in both workforce and customer scenarios. Multiple companies have reported a significant reduction in successful account takeovers when using FIDO [7]. Over time, industry will likely see some successful attacks on passkey providers and potentially other successful attacks to trick users on using malicious passkey providers - attacking the TOFU model. But the fraud levels will be significantly lower than what we see with current combinations of passwords and risk-based authentication.

Sign-in

With FIDO/passkeys, the security of the ~99% use case of sign-in is mostly addressed - including reauthentication on the same device and FIDO cross device authentication.

This leaves the remaining 1% use case, i.e., *sign-up* and *device migration*.

Note: FIDO/passkeys are not effective against fake emergency scams [8]. But traditional risk signals aren't either. A solution is still needed to assess how "free" the user was in the decision to authorize a sign-in or a specific transaction [9].

Sign-up

When new users sign-up, relying parties need to perform some form of identity verification (ID&V). In most practical cases, relying parties only verify that the user has access to an email address or a phone number. Only in the case of high-security needs (such as banking), some kind of document-centric remote ID proofing is used. This means the user presents a selfie or a video stream plus an image or a video stream of an acceptable picture ID. Both methods, i.e., email-/SMS-OTP and document centric ID proofing, do not provide robust security and hence benefit from the use of risk signals. The “deepfakes” created by new Generative-AI tools are an increasing challenge [10] for these legacy ID proofing methods. Using AI to detect deepfakes [11] results in an “arms-race”, a race where the protection approaches are half a step ahead but where the attackers are not far behind. Or vice versa. A different approach is needed to change that “game”.

The industry is working on a more robust solution based on Identity Wallets and verifiable credentials & presentations [12]. Similar to FIDO and passkeys, this approach will lead to robust security for the sign-up use case.

And a new approach is needed! With approximately 100 accounts per user [13], sign-up happens every 1.5 years on average. New account signups constitute 60% of the 1% use case.

Device Migration

With a smartphone upgrade cycle of 3.6 years [14], a tablet upgrade cycle of 6 years [15], and a PC upgrade cycle of 5 years [16], users bootstrap a new device approximately every 2 years. Device migrations constitute the remaining 40% of the 1% use case.

Here we have the “happy-path”, in which the user still has access to the old device and the “unhappy-path”, in which the old device was lost, stolen or is otherwise unavailable. The unhappy-path will have to be addressed similar to the sign-up use case, meaning risk signals and similar methods are relied on until Identity Wallets and verifiable credentials are practical to use.

Today, there is no easy solution with robust security for the happy-path. However, platform vendors/passkey providers could implement a method using the old device for endorsing new keys on the new device. Physical proximity could be leveraged to achieve robust security, but for all the user’s keys in a single step, as opposed to “key by key” as relying party apps could implement it.

Until this new method is practical to use, users must use relying party specific methods for migrating keys in the KeyStore and passkey provider specific methods for migrating synced passkeys to new devices. Today, these methods leverage approaches that do not provide robust security (such as passwords and OTPs). As a result, these methods also benefit from the use of risk signals.

New Role of Risk Signals

In addition to the declining need of risk signals as primary fraud reduction method, there is a new and emerging set of use cases that will benefit from signals that correlate with other aspects but are not necessarily causal to it: as a “helper for anomaly detection and usability improvements.”

Step-up Authentication

Assume a user has created a synced passkey in a smartphone some time ago. Now, the synced passkey is used for the first time on a tablet. The question is should the relying party trigger step-up authentication on that new device or not?

There are two obvious cases: (1) the relying party never needed to care about specific devices, not even when using passwords and (2) the relying party is regulated to manage specific devices. These two cases are clear: the relying party in case (1) would not trigger any step-up authentication and the relying party (2) always would do so. But there is also an alternative scenario: the relying party needs to detect the case in which the user might have intentionally shared the key with a friend. This is typically not a fraudulent case, but the security policy or regulation still requires the relying party to ensure that only the original user accesses the account.

The relying party could use *correlation signals* (i.e., signals that indicate correlation rather than root cause relationships) to determine whether the new device was set up by the original user or is used by a friend (for example in an anomalous location).

Time to Create a Passkey

When migrating existing users over to passkeys, legacy authentication methods are used before a passkey can be created. With legacy authentication, there is the same need for risk signals as we have today – but significantly less often. And more specifically, the passkey should only be created, if no anomalies are detected.

There is a second aspect: usability. When suggesting the passkey creation, will the user be inclined to do so, or will it be a distraction? For this aspect, we need to predict the user's sentiment. This is another good example for the need of *correlation signals*, as we want to find the time the user will create the passkey and avoid the explicit ask if the user will see it as a distraction.

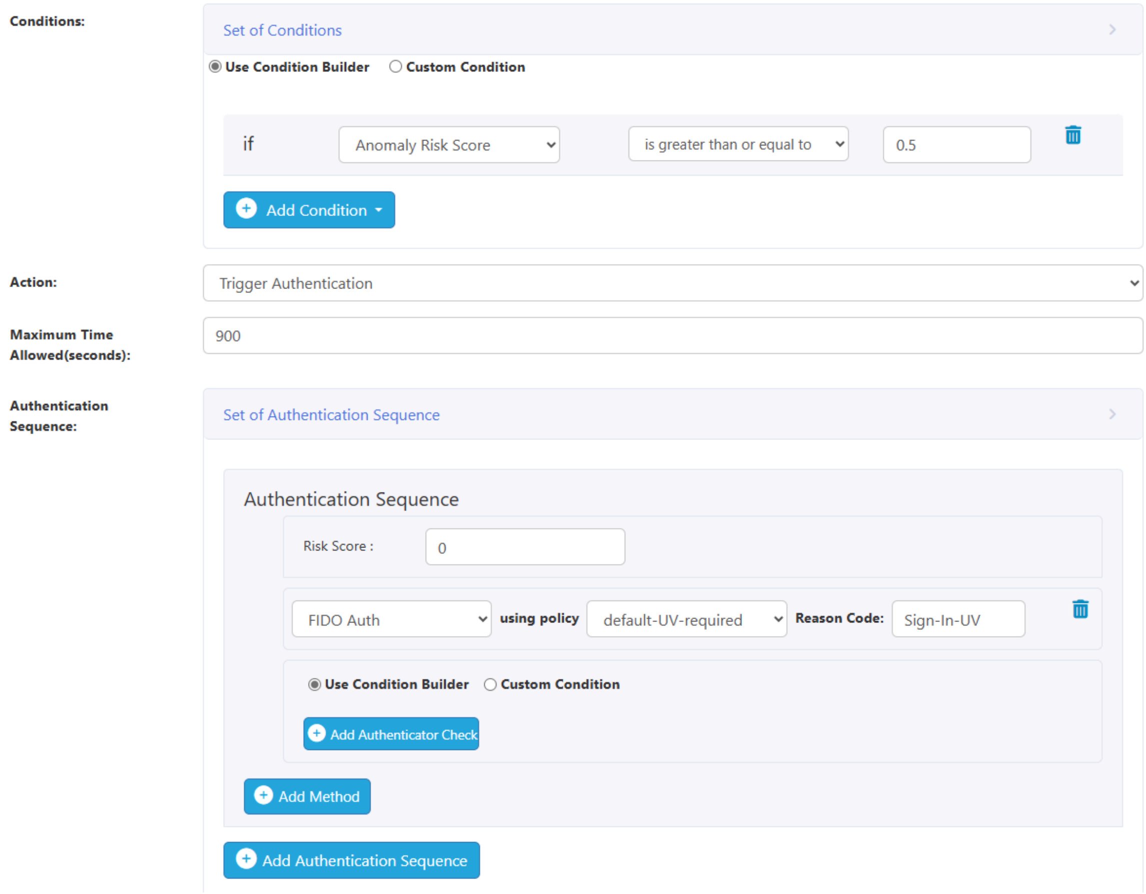
Nok Nok Smart Sense

Nok Nok's mission is to reduce the complexity of managing digital identity with robust security. We co-founded the FIDO Alliance and implemented one of the first solutions supporting passwordless FIDO authentication (aka passkeys) at scale.

Nok Nok Smart Sense uses AI/ML to learn typical user behavior on a per user basis and it provides an "anomaly score" that can be used in the Nok Nok rules engine including location information, network information and more. This capability will lead to a detected anomaly, for example when a user that typically works from home or in the office attempts to authenticate from a *new device* in a *different place* the *first time* – is it legitimate? A passkey shared with a friend? A fraudster? This integrated risk and authentication capability allows organizations to easily adapt the authentication flow to the anomaly score for a particular event through the Nok Nok rules engine – eliminating the need to develop and maintain your own code to respond to risk signals from an external provider.

With this approach, an organization can suggest the creation of a passkey through Nok Nok Registration Rules if the anomaly score is low, which is a sign that it is the legitimate user that was authenticated using a legacy method.

You could also use the new anomaly score to *require* user verification in the case of detected anomalies and leave it at the default value of *preferred* if no anomalies were detected. This makes it more convenient for users using their laptop in “clamshell mode”, when the fingerprint sensor is not easily accessible.



Conditions: Set of Conditions

● Use Condition Builder ○ Custom Condition

if Anomaly Risk Score | is greater than or equal to | 0.5

Action: Trigger Authentication

Maximum Time Allowed(seconds): 900

Authentication Sequence: Set of Authentication Sequence

Authentication Sequence

Risk Score : 0

FIDO Auth **using policy** default-UV-required **Reason Code:** Sign-In-UV

● Use Condition Builder ○ Custom Condition

Based on the anomaly score, an organization can also control whether step-up authentication is required when a synced passkey is used on a new device the first time or whether step-up authentication is required when a passkey provider that doesn’t provide an “authentication intent” signal (see NIST SP 800-63) is used.

With the Nok Nok Smart Sense offering, support is added for AI/ML based anomaly detection to the Nok Nok S3 Suite and Nok Nok Authentication Cloud offerings. This capability makes it easy for companies to combine the use of passkeys with risk signals in their new role to improve user experience and reduce fraud and reduce the reliance on expensive legacy risk signals.

References

1. https://www.checkout.com/guides-and-reports/the-hidden-billion-dollar-opportunity
2. https://en.wikipedia.org/wiki/Conditional_probability
3. https://xkcd.com/552/
4. https://www.bankinfosecurity.com/for-sale-on-cybercrime-markets-real-digital-fingerprints-a-12943
5. https://en.wikipedia.org/wiki/Trust_on_first_use
6. https://www.youtube.com/watch?v=3caHZBiexpQ
7. https://krebsonsecurity.com/2018/07/google-security-keys-neutralized-employee-phishing/
8. https://consumer.ftc.gov/articles/scammers-use-fake-emergencies-steal-your-money#example
9. https://www.imda.gov.sg/how-we-can-help/anti-scam-measures
10. https://www.kuppingercole.com/watch/danger-of-deepfakes-identity-proofing
11. https://www.npr.org/2024/04/05/1241446778/deepfake-audio-detection
12. https://www.kuppingercole.com/watch/convergence-of-passkeys-and-identity-wallets-eic24
13. https://tech.co/password-managers/how-many-passwords-average-person
14. https://www.sellcell.com/blog/how-often-do-people-upgrade-their-phone-2023-statistics
15. https://www.statista.com/statistics/267473/average-tablet-life/
16. https://www.statista.com/statistics/267465/average-desktop-pc-lifespan/