Castle

WHITEPAPER

A User-Centric Approach to Preventing Threats Beyond Account Takeover

Designing Security as a User Experience

Contents

Introduction

Securing user accounts from both bots and human attackers has become one of the most fundamental challenges of delivering modern applications and services. Attackers continuously develop ever more sophisticated techniques for taking over user accounts, while constantly adapting in order to evade security controls. Organizations have likewise tried to keep pace by deploying new classes of threat detection and analytics.

Unfortunately, this ongoing battle between attackers and defenders has left valid users caught in the middle. Aggressive security policies lead to false positives and needlessly lock valid users out of their account, while lenient security policies lead to compromised accounts and fraud. Either case results in unhappy users.

However, a shift is underway that can solve this no-win situation both for application providers as well as their users. The change is as much about a new philosophy of account security as it is about new technology. Instead of focusing exclusively on the threat, this new approach puts the user experience at the center of the security model.

Instead of simply being locked out of their account with no context, users can actively participate in security in low-friction ways that actively keep their accounts safe.

Instead of simply being locked out of their account with no context, users can actively participate in security in low-friction ways that actively keep their accounts safe. Instead of making block-allow decisions based on inconclusive data, security teams can get authoritative answers from the users themselves. And instead of having conflicting goals, developers and security teams can build and design with a common goal of user satisfaction.

At the highest level, a user-centric model allows security to focus on enabling the good in addition to stopping the bad. The benefits of this approach extend throughout the enterprise. Organizations can simultaneously improve customer satisfaction while stopping threats in real time that would otherwise be missed by traditional security measures. Security and user experience can be easily designed in during the development phase instead of being bolted on after the fact. Simply put, by putting the user at the center of the security model, all the various teams of an enterprise can work toward the same goal leading to better security, better applications, and happier customers.

In this paper, we introduce the key concepts of a user-centric approach to security and then dive into how it works in a real-world environment.

The Challenges of Traditional Account Security

Before building a new approach to account security, it is important to understand exactly how and why the existing approaches are failing. And while account takeover (ATO) and fraud can be addressed in many ways, most solutions fall into three high-level categories:

- **Application Security - WAF, anti-automation and anti-bot tools**
- **Fraud Detection - Commercial or custom-built analytics and fraud detection**
- **Traditional Adaptive Authentication - Adaptive MFA and access controls**

While these technologies can be incredibly valuable to an organization, they generally suffer from some common and fundamental challenges.

Application Security - WAF, Anti-Automation and Anti-Bot Tools

Web application firewalls have been a standard component of the application security stack for years, and unfortunately many organizations have also become very familiar with their limitations. WAFs are heavily signature-based and prone to false positives. These two factors mean they often require constant tuning and work from security teams in order to keep the WAF up to date and find an acceptable balance between protection and blocking valid users. In fact, in many cases WAFs are used in a purely detection-based mode of deployment to ensure that users are not inadvertently affected.

Secondly, WAFs are best suited for finding traditional attacks against vulnerabilities such SQL injection (SQLi) or cross-site scripting (XSS). ATO's and other modern types of abuse often work at the application layer and abuse valid application functionality in unintended ways. For example, a credential stuffing or carding attack will simply use the exposed, valid capabilities of the application for a malicious purpose. In these cases, there is often nothing overtly malicious for the WAF to detect or block beyond obvious spikes in IPs or user agents.

The rise of automated attacks and the limitations of WAFs have led many organizations to adopt various anti-bot and anti-automation tools. However, these products have not proven to be a panacea either. Like WAFs, anti-bot tools remain focused on their particular subset of threats, and attempt to detect the presence of bots in a wide variety of ways. However, unlike detecting an exploit or known piece of malware, bot detection is often uncertain. Without conclusive answers, teams are often once again stuck making uncomfortable trade-offs between security and customer happiness.

Additionally, bots and automated attacks have proven to be highly adaptable.

Whether the goal is ATO, carding, or any number of automated threats, most attackers will adapt their tools and techniques as they encounter new detections and security countermeasures. This often leads to an ongoing and ever-changing battle between attackers and security staff. A technique that worked to mitigate a headless browser may be completely ineffective if the attacker decides to employ cheap human attackers as part of a labor farm.

As a result, security measures may work well for one week, but then become ineffective as attackers adapt. It then falls to security teams to deconstruct the attacker's new techniques and build new counter-measures. This leads to a never-ending cycle of work in order to achieve unreliable security efficacy.

Fraud Detection Tools

While WAFs and anti-bot tools must render decisions quickly and often with limited information, anti-fraud tools tend to take a more data-rich approach. These systems are typically highly customized to the unique application or organization, and can include data collected from local applications as well as data feeds from external sources. In many cases, a dedicated data science team focuses on developing models and tools to drive fraud detections for the organization.

These tools can help organizations adapt to broad changes in their environment, and to detect compromised accounts and fraud that is in progress. And while this insight is invaluable to the organization, conclusions are often rendered after an account has been compromised.

Traditional Adaptive Authentication

Traditional adaptive authentication provides another approach to securing user accounts. These tools will often offer step up authentication challenges based on basic anomalies. For example, adaptive authentication may trigger a multi-factor authentication challenge in response to a user connecting from a new country.

And while these tools provide a key step in the right direction and can be an essential component of a user-centric security strategy, they have several limitations that keep them from providing protection from ATO, credential stuffing, and other automated attacks. First, the detection logic is fairly rudimentary, typically being based on IP address and lacking threat-based detection of actual malicious automated behaviors. This means that while they may see the simple example of a user logging in from a new country, they would miss actual credential stuffing behaviors indicative of a true attack.

Secondly, adaptive authentication tools are typically limited to the initial login phase of an application. Many automated attacks are executed after the initial login. For example, an attacker may make changes to the account in an attempt to take control or may make a transaction within the application. Not only would adaptive authentication tools lack the intelligence to see these threats, they also simply don't provide the continuous authentication needed to stop them.

Persistent Gaps and Slow Responses

All told, these approaches leave organizations with serious gaps in their approach to account security. Teams are forced to either make early decisions without enough information via WAFs, anti-bot tools, or adaptive authentication, or alternatively make more reliable decisions later in the attack after damage has been done. Neither option is ideal, and in both cases customers end up unhappy.

And unfortunately the challenges are not limited to the user experience. GDPR and an array of regulations are consistently pushing for faster responses and user notification. The same lack of certainty that limits enforcement efforts, likewise limits response and recovery efforts to slow, manual processes that are expensive and ultimately ineffective. In order to bridge this gap, organizations need the functional ability to drive end-to-end account recovery, they need to have high confidence detections and conclusions that can allow the process to run automatically.

An Introduction to User-Centric Account Security

Castle introduces a fresh approach to account security that drives better outcomes both for security teams and their end users. To deliver on this goal, Castle incorporates end users and their experience directly into the security model.

This simple shift allows us to rethink some of the fundamental assumptions and traits that have consistently hamstrung security efforts for years.

Rethinking Security Fundamentals. It's Time for a Shift.

- From Reactive to Proactive Security
- From Negative to Blended Detection
- From Frustrated to Empowered Users

Shifting Security Fundamentals

From Reactive to Proactive Security

If security technologies focus exclusively on detecting the threat, then the attacker is given the first-mover advantage. The attacker can change tactics and is the defender's job to react and try to implement a near-perfect detection model. By incorporating user feedback, security teams gain proactive options. When teams see something suspicious, they can proactively ask for more information that drives the right decision in real time. Instead of chasing, we can challenge.

From Negative to Blended Detection

Traditionally detection models are only trained by what is bad. Models learn the traits of threats, and can learn baselines for what is normal for a particular user. By involving users, detection models get active feedback from the user in terms of what is good for that particular user. By actively training to recognize the good as well as bad, the detection model becomes highly tuned to the realities of the individual user. So instead of constantly challenging the user with repetitive auth or MFA challenges, the model learns, becomes more customized to the user and reduces user friction over time.

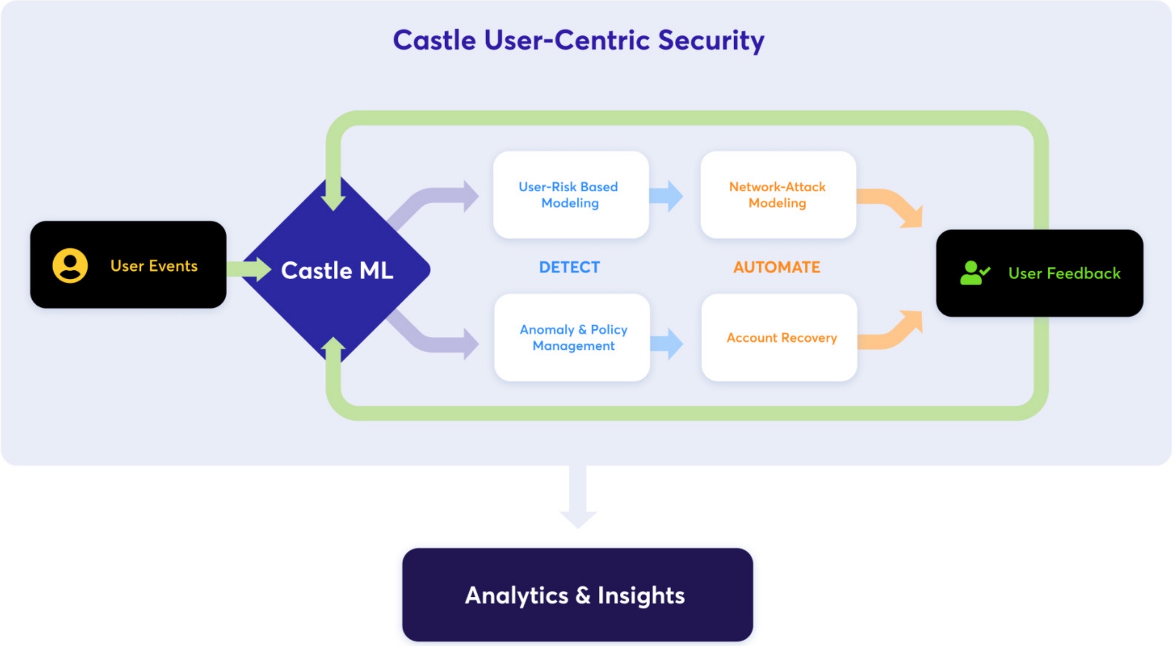
From Frustrated to Empowered Users

For a typical end user, account security is a completely opaque process that happens to them. An account lockout or blocked access often comes with inconvenience and almost no context. By using low-friction methods of incorporating users into their account security, they not only gain insight into the valuable security happening in the background, but they gain agency in decisions without being needlessly locked out of their accounts. And by using automated responses, organizations can streamline the process of account recovery without the need for intervention from support staff.

How Castle Works

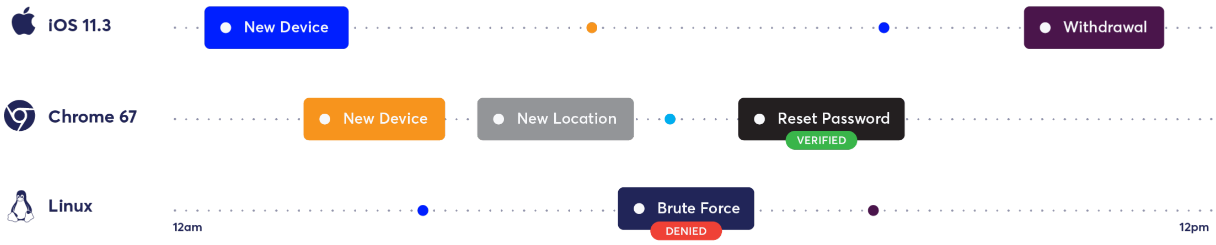
Castle provides an end-to-end approach to stopping account takeovers, credential stuffing, and virtually any attack that relies on humans or bots impersonating your valid users. The solution uses machine learning to understand both users and threats, but it approaches learning in a unique way. With Castle, the machine learning is driven by behavior modeling, threat modeling, as well as feedback from the user. This feedback from the user provides not only a definitive answer for a given event such as a login, but it also provides the real-world feedback for that specific account that can drive ongoing learning. This lets the machine learning models become highly tuned to each individual user over time, meaning security gets stronger with less friction.

Castle's models learn normal user traits and behavior, along with detecting signs of risks or threats. When problems arise, configurable policies let organizations decide how and when to respond and step up authentication. If the user is verified, the results are fed back into the machine learning models to learn from the event. If a threat is detected, Castle policies can document and automate the end-to-end recovery account process to ensure users remain enabled without needing to contact support.



Castle's machine learning continuously analyzes your individual users across a broad spectrum of traits including details about their devices, locations, access patterns, cookies, and more. The system fingerprints each device based on the device type, operating system, browser, user agent, and much more.

Castle also learns behaviors of the user. This includes traits like access times, behaviors on the applications, regions and geographies of access and more. This also extends to behaviors within the application such as making changes to the account, initiating transactions, or virtually any other behavior in the application.



Risk-Based Analysis

Castle also analyzes visitors for signs of potential problems. This could include risk factors such as users connecting from Tor exit nodes, hiding behind proxies, or connecting from a datacenter.

Castle intelligence also looks for other signs of suspicious activity. This could include traits indicating the presence of a bot or malicious automation, such as non-human mouse movement or application usage. Other risk factors could include signs that an account has been compromised or is being abused, such as a user logging in from geographically distant locations within a short period of time. These are just a few examples, but by combining user-facing and risk-facing detection models, Castle can quickly identify and score the overall risk for each user.

Pre-Login and Post-Login Analysis

All of Castle's analysis extends throughout the user experience both before and after login. This is a major shift that treats authentication as a continuous process instead of a one-time gate to be cleared by an attacker. For example, Castle retains insight into user behavior within the application and continues to identify risk, anomalies, and signs of threats.

This is particularly important to account takeover and a variety of other automated attacks. For example, an attacker may try to change the user's password or add an account to receive stolen funds. Additionally, many automated attacks rely on malicious behavior after a login such as performing small transactions to test payment cards, locking up inventory in shopping carts without completing the purchase, or generating fake reviews. Detecting and mitigating these behaviors requires an ongoing and continuous approach to analysis that traditional adaptive authentication solutions lack.

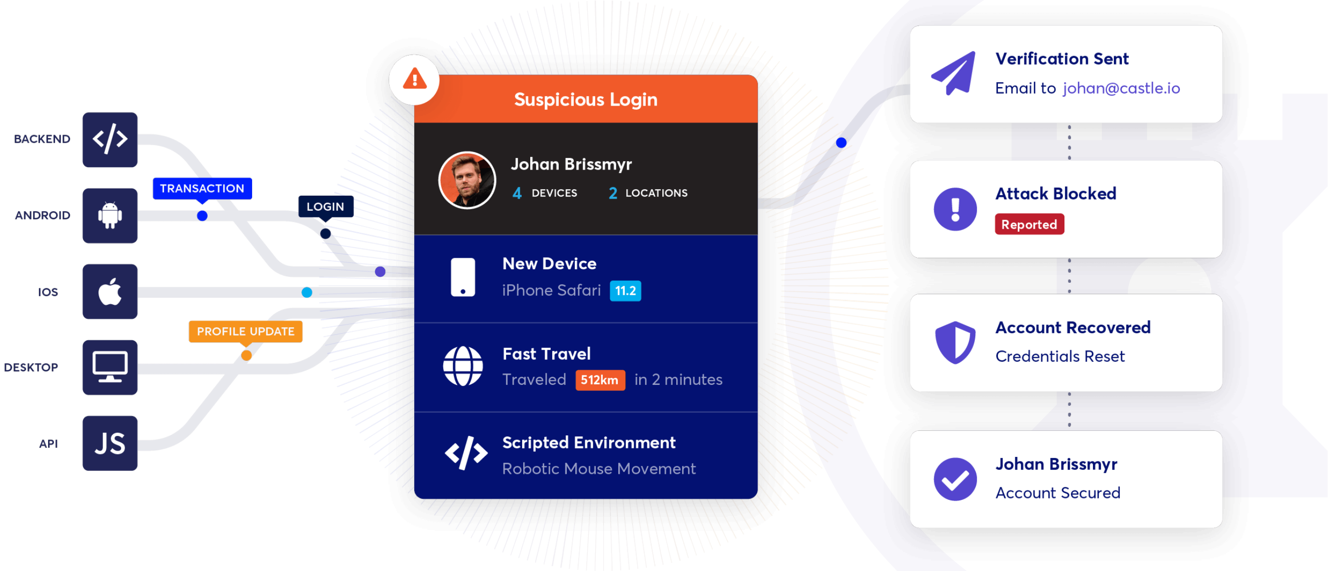
User-Trained Machine Learning

Castle continuously maintains a score for each user that aggregates the risk based on the analysis techniques described previously. Security teams can customize policies to decide when and how to they want to get feedback from the user. Integration with the Castle API allows organizations to trigger any response or challenge of their choosing, thus letting organizations tune responses to their tolerance for both risk and user friction. This could be a simple verification email to the user, a multi-factor authentication challenge, or any other mechanism the organization supports.

The results of the challenge are then fed back into the Castle machine learning engine for additional learning. This allows the engine to be specifically trained on approved or "good" behavior from the user. Over time this allows the machine learning to be tightly customized to the user and reduce the need for future friction or verifications.

Automated Account Recovery

When a threat is detected, Castle documents and can help orchestrate a fully automated recovery of the account. Once again, this process will naturally be customized to the policies and procedures unique to each organization. However this can include steps such as locking the account, coordinating the reset of credentials for the valid user, and unblocking the account.



Continuous Visibility

In addition to providing a highly automated approach to account security, Castle ensures visibility throughout the entire process. Staff can always get insight into any account, track overall risk, or delve into specific types of behavior on an application. While the system ensures that account operations can be run completely hands-free, it also ensures staff have visibility into the key metrics of user behaviors when they need it. Whether via APIs, dashboards, or the Castle user interface, the solution provides transparent insights into every threat signal, risk score, and event tracked per device within a user's account.

And unlike many machine learning-based systems, Castle provides access to any signal collected by the system. Analysts and anti-fraud teams can see exactly why a user has an elevated score, or can dive into particular traits of interest.

How Castle Fits

Castle is incredibly simple to work with and is designed to align with your existing applications, security processes, and development pipelines.

Simple Deployment

Castle makes it easy to get up and running whether you need to protect web, mobile, or API-based applications. Modular APIs lets Castle align to your app's unique UX, not the other way around. Castle is simply accessed as an API meaning that there is nothing to install on-premise and no single point of failure.

Additionally, the solution lets you grow at your own pace. Staff can start getting visibility in monitor-mode then step up to receiving passive notifications, and ultimately to active blocking and automation.

Integrates With Your Existing Tools

Castle works with your existing tools and processes. Use Castle's intelligence to turn static MFA or authentication services into a truly adaptive authentication that responds to changing context. Security event webhooks let teams drive a variety of automated responses based on insight from the Castle engine.

The highly flexible Castle API lets organizations truly automate account security from end to end. Organizations have full control over how users are contacted, authentication mechanisms that should be used, and any automated actions in the application or the account.

Developer-Centric

Castle works just like any API in the development process. This allows dev teams engineer security in as part of the design of the user experience. Developers work in whatever language they choose such as Ruby, Java, PHP, Python, Node, or .NET and the Castle API handles the rest. This not only makes it easy to design security in early, but it also helps to reduce an age-old point of friction between security and dev teams.

Conclusion and Next Steps

For decades, security has been framed as attackers vs defenders - black hats vs the white hats. And while this view is valid, it has consistently left end users caught in the middle of a never-ending fight. The negative impact to users has only gotten worse as attackers have shifted their techniques to impersonating valid users either to take over user accounts or to abuse the application itself.

Castle's user-centric approach to account security provides a modern approach that not only gives security teams a reliable upper hand against threats, it puts user satisfaction at the forefront of the security model. From development to operations and security, the user experience remains central, and the benefits of such alignment can extend across the organization.

This paper provides an introduction to the Castle approach and its capabilities. If you have additional questions or would like to see Castle for yourself, please contact us at castle.io/demo.

Castle

Castle provides an end-to-end user-centric approach to stopping account takeovers, credential stuffing, and virtually any attack that relies on humans or bots impersonating your valid users. Castle's modern approach not only gives security teams a reliable upper hand against threats, but it puts user satisfaction at the forefront of the security model.