Castle

WHITEPAPER

A Guide to Continuous Identity Protection For Your Online Business

Securing Your Customer's Journey from Registration to Recovery

Contents

Introduction

Securing user accounts from abuse and fraud is a fundamental and never ending challenge for any organization that delivers high-value applications. Accounts face continuous assault from bots and attackers attempting account take-overs (ATO), carding attacks, fake account creation, theft, fraud, and more.

Unfortunately, tighter traditional security controls usually create a worse experience for users and more work for security and IT. Aggressive security policies lead to false positives and account lockouts, while lenient security policies lead to compromised accounts and fraud. In either case customers are unhappy.

However, a shift is underway that solves 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 customer identities and the user experience at the center of the security model**.

With customer identities at the center of the security model, organizations can enable the good while stopping the bad.

- **Simultaneously improve customer satisfaction while stopping threats in real time that would be missed by traditional security measures.**
- **Fully automate threat prevention and account recovery with less work for staff and less downtime for users.**
- **Configurable, flexible security that is sticky with existing infrastructure, not bolted on.**

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

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

The Challenges of Securing Customer Accounts

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

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

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

WAF

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 have been notoriously prone to false positives and require constant tuning in order to find a palatable balance between protection and blocking valid users. As a result, WAFs are often deployed in a "detection-only" mode 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). Account take-overs 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.

Anti-Bot Tools

The rise of automated attacks has 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 negatively focused on a 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 and based on anomalies. Without conclusive answers, teams are 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. Anti-automation measures that work well one week can become ineffective the next as attackers adapt. This means that security teams must constantly adapt to new attacker techniques and a never-ending cycle of work in order to achieve unreliable security efficacy.

Anti-Fraud 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.

Access Management

Access management tools provide another approach to securing user accounts. These technologies can range from basic username/password controls to multi-factor authentication tools (MFA) and more. Many MFA tools have also become increasingly "adaptive", although in a relatively limited sense. For example, adaptive authentication may trigger a multi-factor authentication challenge in response to a high-level trait such as a user connecting from a new country.

And while these tools provide a key step in the right direction, they have several limitations. 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 deployed at the initial login phase of an application. Many automated and fraudulent attacks are executed after the initial login. Even if a step-up challenge is configured after login, it can only trigger on a specified event pre-configured in policies without any visibility into context and risk. For example, assuming an attacker has done reconnaissance of an application's security controls, they can do ahead and make changes to the account in an attempt to take control or may make a risky transaction that does not set off any step-up challenges. Not only would adaptive authentication tools lack the intelligence to spot threats that are within the organization's risk tolerance, they also simply don't provide the continuous authentication needed beyond just the login.

Persistent Gaps and Slow Responses

All told, these approaches leave organizations with serious gaps in their approach to customer 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 ability to drive end-to-end account recovery and processes that can run automatically.

An Introduction to Castle's Identity Risk Engine

Castle introduces a fresh approach to customer identity security that drives better outcomes both for security teams and the organization's end users. To deliver on this goal, Castle incorporates customer identities and the user 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. This includes:

From Threat-Focused to Customer-Focused Security	Traditional security models are defined exclusively by what is bad - signatures, traits, and behaviors of threats. This approach is negative, reactive and only takes into account one half of the situation. Castle lets organizations not only see threats, but positively defines security in terms of applications and user identity. Policies can align to specific application functionality, user groups, and overall risk. When something suspicious happens, policies can proactively ask for more information from the user in order to drive the right decision in real time.

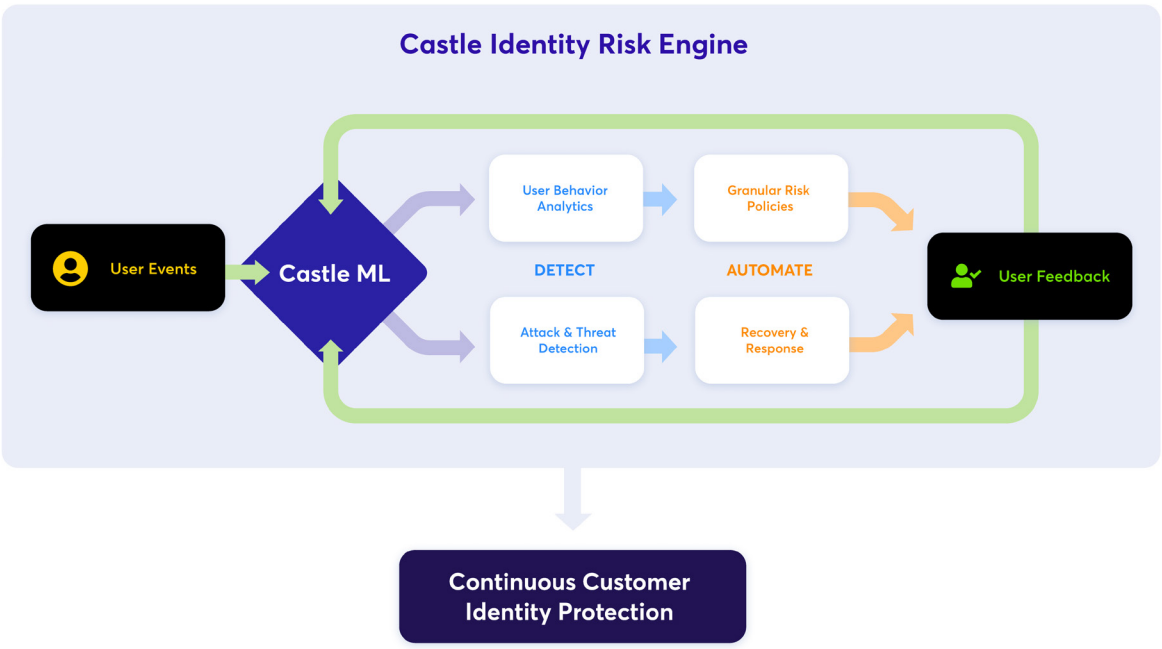
From Login to Lifecycle Security	Security and MFA controls have traditionally been heavily focused on login events when users are authenticated. Castle makes security a continuous process instead of a one-time event. This includes analyzing risk before the user's account is created, during login, and continually monitoring behavior after login. This means that security is continuous and can apply to any application functionality.
From Manual to Automated Response and Recovery	Castle reduces IT work and user downtime by allowing customers to self-heal through automated password resets, account recovery, and other remediation events. By engaging the user in part of the feedback loop, Castle is able to get smarter and faster to meet the demands of a dynamic threat landscape.
From Frustrated to Empowered Users	For a typical end user, traditional account security is often 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.
From Contentious to Coordinated DevSecOps Relationships	Castle's dynamic APIs and webhooks make it easy to integrate with the development process and easily scales with any organization. Developers continue using the same tools they like, with security built in to the fabric of the application. Most importantly, dev and security teams can collaborate on common goals that focus on optimal user experiences while ensuring the overall safety of user information and the integrity of their accounts.

How Castle Works

The Castle Identity Risk Engine provides a continuous approach to stopping account takeovers, fake account creations, and virtually any attack that relies on humans or bots impersonating your valid users. The engine leverages a unique approach to machine learning that automatically distinguishes valid users from threats. Our automated detection models learn from user-centric behavioral analysis, threat-centric behavioral analysis, as well as real-time feedback from the user.

This feedback from the user provides not only a definitive answer for a given event such as a login, it also drives ongoing learning specific to that individual account. This ensures that machine learning models in the engine become highly tuned to each individual user identity over time, so that security gets stronger with less friction.

Castle's configurable policies give organizations fine-grained control over exactly how, when, and for whom authentication should be stepped up. When a user is challenged and verified, the results are fed back into the detection 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.

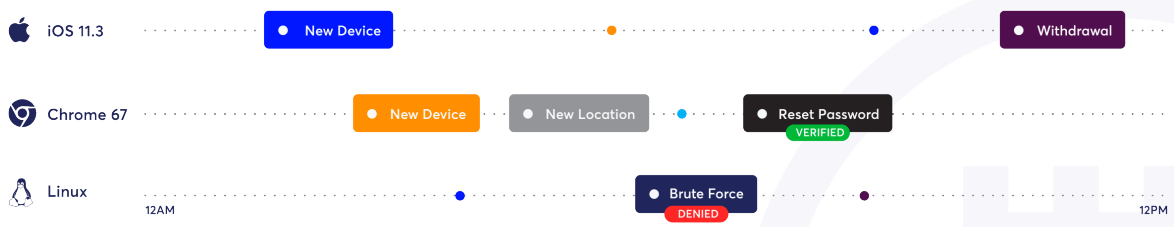


User Behavior Analytics

The Castle Identity Risk Engine continuously analyzes each individual user identity across a broad spectrum of traits, throughout the entire session. This includes details about their devices, locations, access patterns, cookies, and more. The system fingerprints and assigns individual risk scores for 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 after login, 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.

Suspicious or risky behavior can be challenged based on the organization's unique policy. This allows the engine to be specifically trained on approved or "good" behavior from the user. Over time this allows the detection models to be tightly customized to the user and reduce the need for future friction or verifications.



Risk-Based Authentication

Castle's Identity Risk Engine also analyzes visitors for signs of potential problems and then appropriate to the level of risk: such as an allow, deny, or step-up challenge. This could include pre-login risk factors such as users connecting from Tor exit nodes, hiding behind proxies, or connecting from a datacenter. On the other hand, Castle also tracks risk after the user logs in such as initiating money transfers, profile changes, or password resets.

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. By combining user-facing and risk-facing detection models, Castle can quickly identify and score the overall risk for each customer identity and trigger a step up in authentication via SMS, Voice Call, Email Verification, Pin Code, or Push Notification. Risk scores can also be used to drive automated security responses to both protect and recover an affected account without the need for manual effort.

Identity-Aware Bot Detection

Castle's Identity-Aware Bot Detection looks for other signs of bots or malicious automation, such as non-human mouse movement or the use of automated tools such as headless browsers. By combining the behavioral and risk-based analysis described above, Castle's Identity Risk Engine provides an incredibly accurate and low-friction defense for bots and malicious automation. With the ability to analyze both before and after login, Castle applies to a wide variety of automated attacks and business logic abuse. This includes:

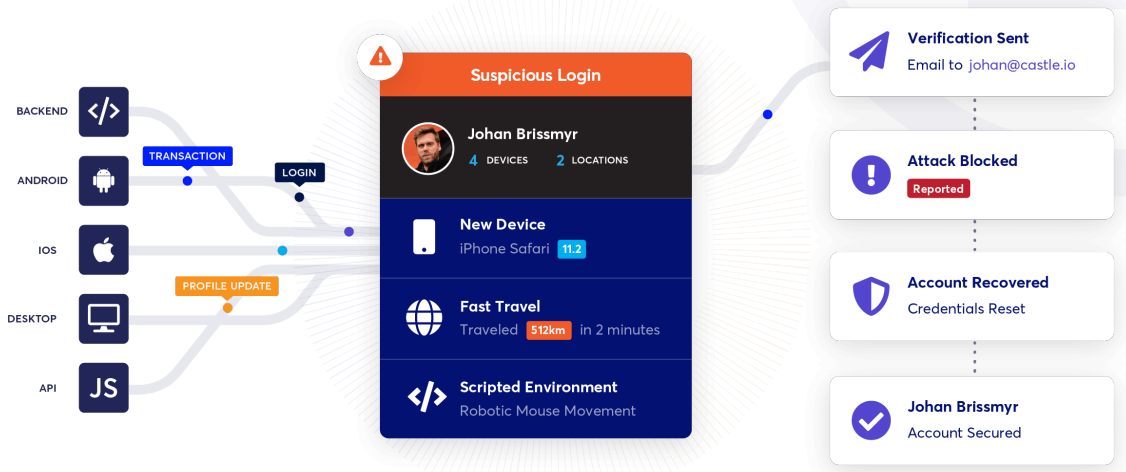
- Fake Account Creation
- Carding Attacks
- Review Spam and Astroturfing
- API Abuse
- More...
- Account Takeovers
- Content Scraping
- Denial of Inventory
- Analytics Abuse

How Castle Offers Unique Value

By putting the focus on customers, Castle allows organizations to define security based on the unique needs of the business. This includes building granular policies tied to specific identities, user groups, applications, or specific application functionality. The organization also retains full visibility into exactly what is going on with each account at any level of granularity in order to drive better overall management and decision making.

Seamless Administration and Controls

Castle's Identity Risk Engine allow you to step up security specifically when it is needed the most - for high value events, transactions, or users. Some users may be more important than others, and some features or transactions may carry more risk than others. Castle recognizes this reality and puts you in control of how to define and challenge risk in your environment. Easily build policies based on identity and device traits, application events, and fine-tune the risk profile and responses for each. Instead of cookie-cutter security rules, Castle's granular policies lets you define security based on the unique requirements of your business and users and tune your risk tolerance and response accordingly.



Actionable Insights and Analytics

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.

Security for the Entire Customer Lifecycle

All of Castle's security begins before a user account is ever created and 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 for stopping fraud, account take-over (ATO) attacks 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.

How Castle Fits Into Your Stack

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-premises and no single point of failure.

Castle believes that simply deployment truly means just a few clicks and were the first to implement a codeless integration with content delivery networks like Cloudflare. Additionally, Castle's platform ensures that you grow at your pace. Staff can start getting visibility in monitor-mode then step up to receiving passive notifications to get value from day 1, and ultimately to active blocking and automation when development resources are available.

Developer-Centric

Castle works just like any API in the development process. This allows dev teams to build in security as part 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 address security early, but also helps reduce an age-old point of friction between security and dev teams.

Developers can also deconstruct Castle into basic blocks with in-app widgets, device list APIs, and security event webhooks to ensure that Castle can align with any UX your team dreams up.

Integrates With Your Existing Tools

Castle works with your existing tools and processes. Use Castle 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 Identity Risk Engine.

The highly flexible Castle API lets organizations truly automate customer 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.

Summary

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 customers 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 user identities either to take over accounts or to abuse the application itself.

Castle's identity-centric approach to customer security 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 how Castle secures the customer journey. If you have additional questions, please contact us at **info@castle.io** or visit **castle.io**