VMware Carbon Black™

Case Study

Carbon Black Overview

Carbon Black is a cybersecurity software that provides endpoint security. The primary product is called the Carbon Black Cloud, which is a cloud-native platform designed to protect endpoints (such as computers and servers) from a variety of security threats. Key features and components of Carbon Black products include:

- **Endpoint Detection and Response (EDR):**
  This functionality helps in detecting, investigating, and responding to advanced threats in real-time. It provides detailed visibility into activities across endpoints and aids in threat hunting and incident response.
- **Next-Generation Antivirus (NGAV):**
  Combines traditional antivirus techniques with advanced machine learning and behavioral analysis to prevent malware and other malicious activities.
- **Threat Intelligence**
  Integrates global threat intelligence feeds to provide insights into the latest threats and helps in proactive threat hunting and defense.
- **Application Control**
  Allows organizations to lock down and control applications on endpoints to prevent unauthorized or malicious applications from running.
- **Threat Hunting and Incident Response Tools**
  Provides tools for security teams to conduct thorough investigations and respond quickly to incidents.
- **Real-Time Monitoring and Alerting**
  Continuously monitors endpoint activities and generates alerts for suspicious behaviors and potential threats.

The Carbon Black Cloud leverages a combination of behavioral analytics, machine learning, and cloud-based threat analysis to deliver comprehensive endpoint security.

.

My Role

In my role as Lead UX Designer for Carbon Black Alerts, I was responsible for leading improvements in the alerting experience. I collaborated closely with the product owner, UI and backend development teams, and the UX group to deliver customer value and functionality that aligned with Carbon Black's organizational initiative goals. Additionally, I was responsible for continuously refining the UX of CB Alerts, ensuring seamless integration and enhancement as new features were implemented in other areas.

Enhancing Carbon Black Alerting Experience

The Carbon Black Alerts Revamp Project set out to transform the user experience for enterprise corporate security analysts by revitalizing the Alerts section, the cornerstone of the CB endpoint protection software. This initiative was pivotal in bolstering early endpoint threat detection and mitigation capabilities.

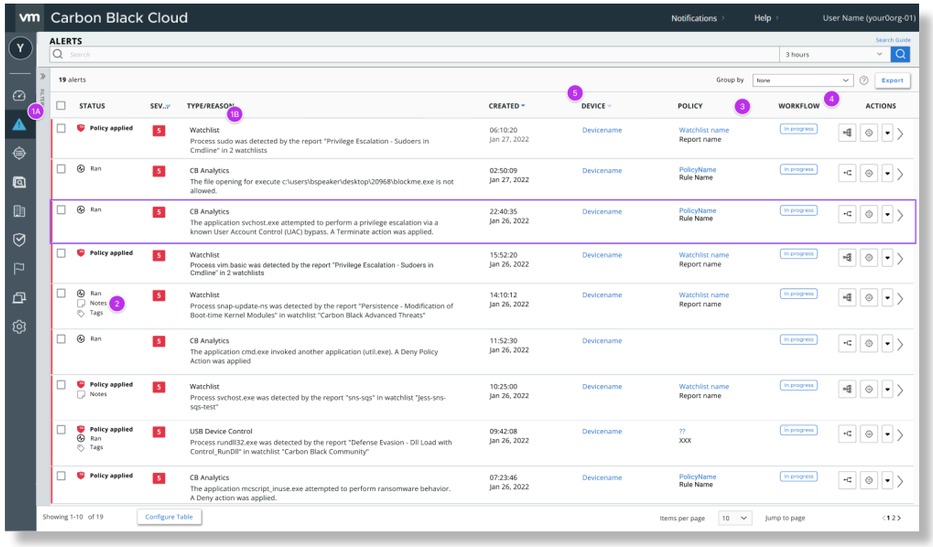
By embracing agile methodology and fostering cross-functional collaboration, the project prioritized crafting a more intuitive, efficient, and user-centric alerts interface. The comprehensive redesign encompassed the development of new alert detail views, alert workflow states, alert grouping by threat ID, and the introduction of an auto-close alerts rule library, all while upholding stringent quality standards and ensuring a uniform UI across systems.

Project Context

Users within the Carbon Black Cloud platform have encountered numerous alerting challenges. Informed by extensive customer feedback, it became evident that users often struggle to effectively triage, scope, investigate, and remediate alerts within the platform. As we geared up for a significant UAE release in August 2023, we explored additional avenues for users to view, organize, and manage their alert queues within their SOC teams.

Project Goals

1. Increase ease-of-use of alert management in the CBC and reduce alert resolution friction
2. Introduce simplified alert workflow enhancements to customers with the goal of reducing users' time-to-value when investigating, triaging, and closing alerts
3. Surface relevant metadata to customers so they have the right level of information at their fingertips throughout the end-to-end alert triage and closure workflow



We refined our project goals into more granual list of objectives:

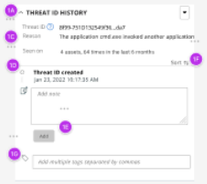
1. Incorporate new alert metadata such as process command line and username, parent and child process information, netconn data, additional device fields, MITRE categorization, and more
2. New customizable alert facets and table columns
3. In-product alert workflow management, allowing the analyst to mark alerts as "In Progress"
4. Classify alerts as True or False Positive
5. Better note management with the ability to add notes to both individual alerts as well as threats
6. Enhanced Alert History visibility which shows a history of all alert state transitions (ie. Open -> In Progress), comments, determination, closure information, etc.

Alert ID History & Threat ID History

Annotation Threat ID History

Default Threat ID History feature presentation

...Threat ID History 24



Tool Tips & Mouse overs

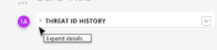
Threat ID History 23

Modifications based on data entry

(tool tips, mouseovers & interactions)

Threat ID History 22

Collapsed presentation



Annotation Alert ID History

Default Alert ID History feature presentation

...Alert ID History | Used

Tool Tips & Mouse overs

Alert ID History | Used

Modifications based on data entry

(tool tips, mouseovers & interactions)

ALERT ID HISTORY | USED

EPIC 1: Alert Workflow

User Stories, Design Requirements

In delineating the enhanced alert experience, I've drawn upon previous customer research and design insights, where customers expressed a desire for alert workflow states. This allows the team of analysts to collaborate on alert resolution progress effectively.

User Epic: As a Customer, I want to be able to transition an alert between Open, In Progress, and Closed states, while maintaining a comprehensive history of alert workflow activity within my environment.

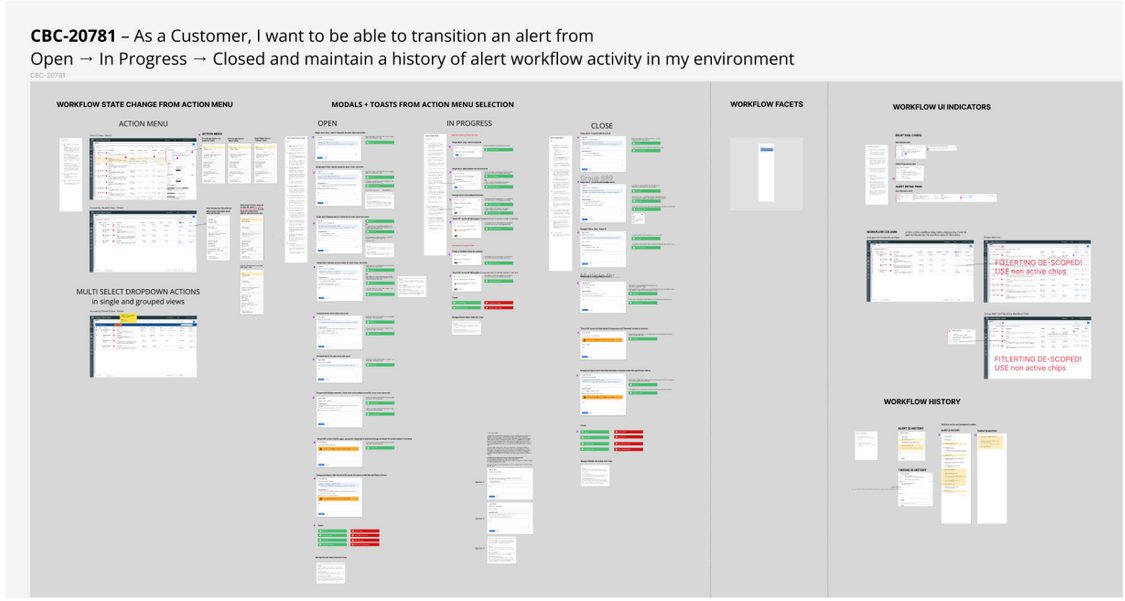
As a cross-functional alert team comprising of product owner, content writer, UX researcher, UI devs and sometimes backend devs, we unearthed various edge cases during our exploration, necessitating further refinement of design requirements. We conducted weekly design reviews to deliberate technical specifications and assess their impact on our design trajectory.

Semi-structured Interview + Usability Testing

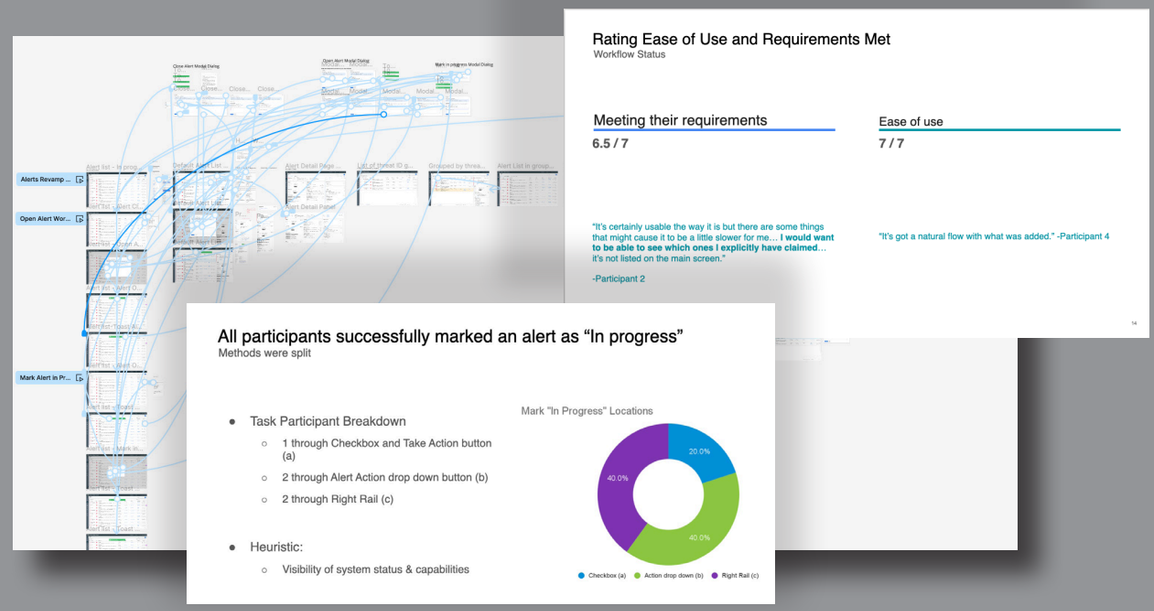
We validated the workflow UX design by user research study that was
Semi-structured interview + usability testing. We had 5 participants (1 internal, 1 Partner, 3 Customers)
The study was based on an interactive prototype that I've created for research task flows
I sat in all research sessions and notated customer feedback.

Key findings:

- Overall, new improvements were very well received by participants
- Ratings were 6+ for meeting their requirements and 7 for ease of use across the board
- New features did not prevent participants from completing their tasks and were mostly a value added
- Some hindrance but no major obstructions to completing the tasks
- Needing to click or go into Alerts ID history to view who is working on an alert because did not know "In progress" was clickable
- Having Close being separate from Determination
- Not being able to use alert_id key in search



Design Validation UXR > Research Prototyping



EPIC 2: Group Alerts by Threat ID

Background
Customers have reported experiencing alert fatigue due to the CBC system generating thousands of alerts based on the root alert trigger. With the introduction of Threat ID, the CBC system can now group similar alerts into a Threat ID alert group, thereby reducing the number of alerts users need to mitigate.

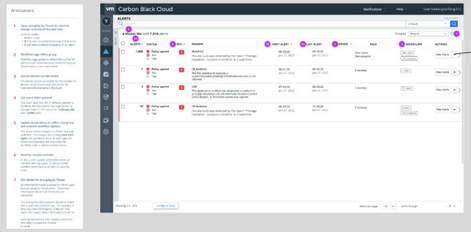
Customer Pain Points
1. *Lack of Clarity at a Glance:*
   - It's not immediately clear that alerts are grouped.
   - The grouping criteria are not transparent.
2. *Confusing Alert Details:*
   - The group right rail takes sample data from one alert and displays an "Alert Detail" card instead of an aggregate of data across the group, which is confusing for users.
3. *Visibility Issues:*
   - There is no clear indication of how many alerts are in a group from the initial grouped table view.
4. *Unhelpful Labeling:*
   - Generic labels such as "results" in group navigation do not help users understand that the list represents groups of alerts.

Approach
To address these issues, we undertook the following steps:
1. *Design Workshops:*
   - Conducted workshops to brainstorm solutions and align on design direction.
2. *User Research:*
   - Gathered insights from existing user feedback and additional user interviews to understand pain points in detail.
3. *Prototyping and Testing:*
   - Created prototypes of the proposed solutions and tested them with users to validate effectiveness.

Solution Exploration
1. *Improved Grouping Visualization:*
   - Enhanced the visual cues to clearly indicate that alerts are grouped.
   - Made the grouping criteria more transparent.
2. *Aggregate Alert Details:*
   - Redesigned the group right rail to show an aggregate view of data across the group rather than sample data from one alert.
3. *Clear Group Size Indication:*
   - Added a clear indicator of the number of alerts in each group within the initial grouped table view.
4. *Informative Labeling:*
   - Replaced generic labels with more informative ones that help users understand the list represents groups of alerts.

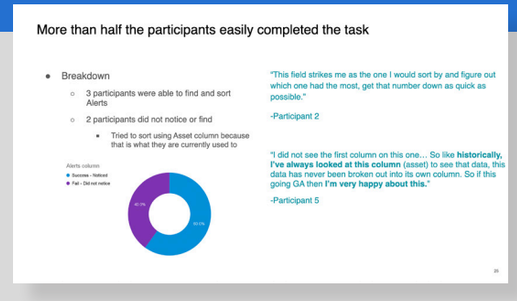
Design Specification Wireframes

Threat ID List View

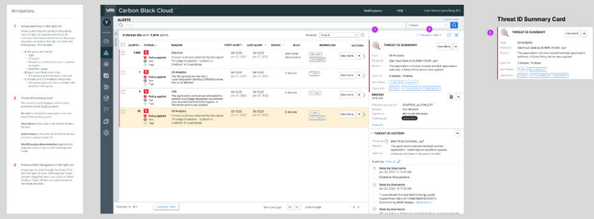


Action Menu Update

Outcome
The new design provides a clearer and more intuitive grouping of alerts by Threat ID, reducing user confusion and enhancing the overall user experience. These improvements have been positively received in user testing, with users reporting reduced alert fatigue and a better understanding of the grouped alerts.

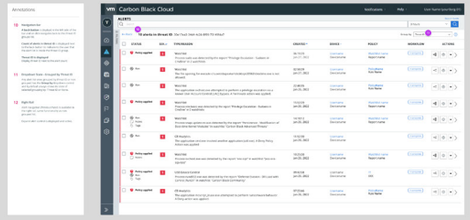


Threat ID Summary

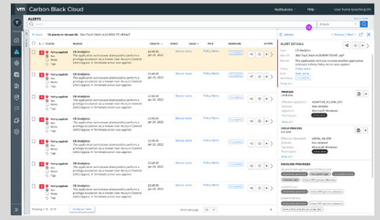


Alert Detail in a Group

Threat ID > Group Alert List



Threat ID > Group Alert List > Right Rail



EPIC 3: Alert Detail Page

User Stories, Design Requirements

User Story
As a SOC or Security Analyst, I want a holistic alert details view that shows me relevant alert information so I can perform the majority of alert investigation and relevant actions for the selected alert without **pivoting** to other page

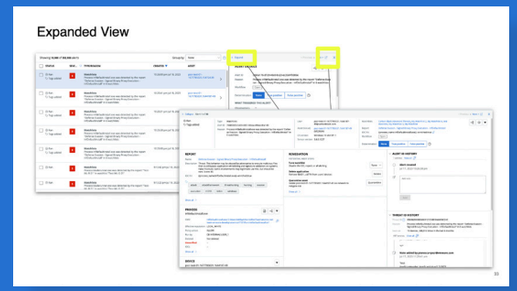
To add richer experience we anabled the user to view right rail alert information in a full page view to ease data digestion and collation. The new navigation through this detail view to the top of the alert list giving a context of alert data.

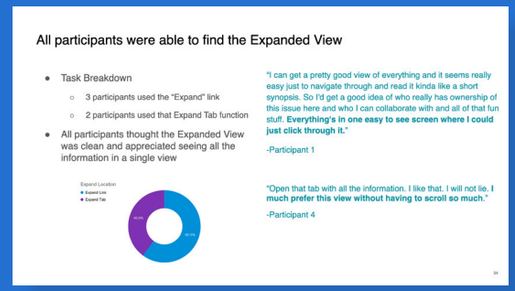
User Goal:
Reviewing a specific alert, determining True Positive/False Positive, beginning investigation

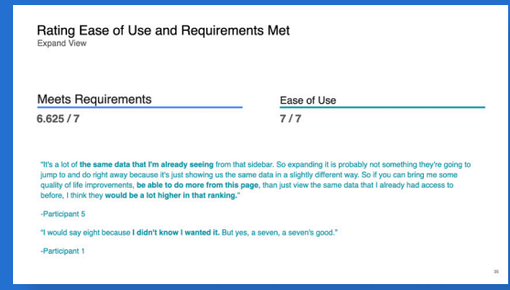
Expanded view of Alert Details with summary header An expanded view of the Alert details gives analysts a full view of the alert in one screen.

The alert summary header (maintained from the expanded details view) gives users a sense of “place” while triaging an alert. Users can navigate here from the alert list page to quickly reference individual alert information. Affordances like open in a new tab allow users to easily leverage multiple monitors.

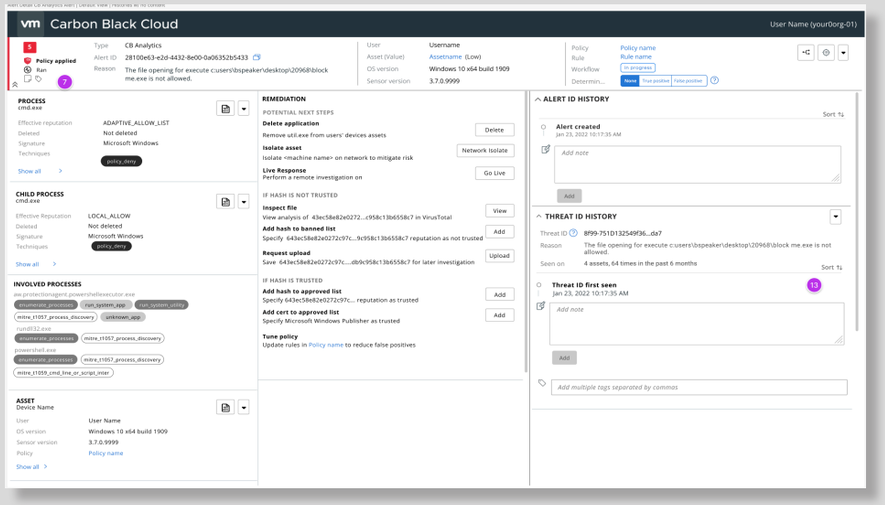
Design and Usability Validation UXR



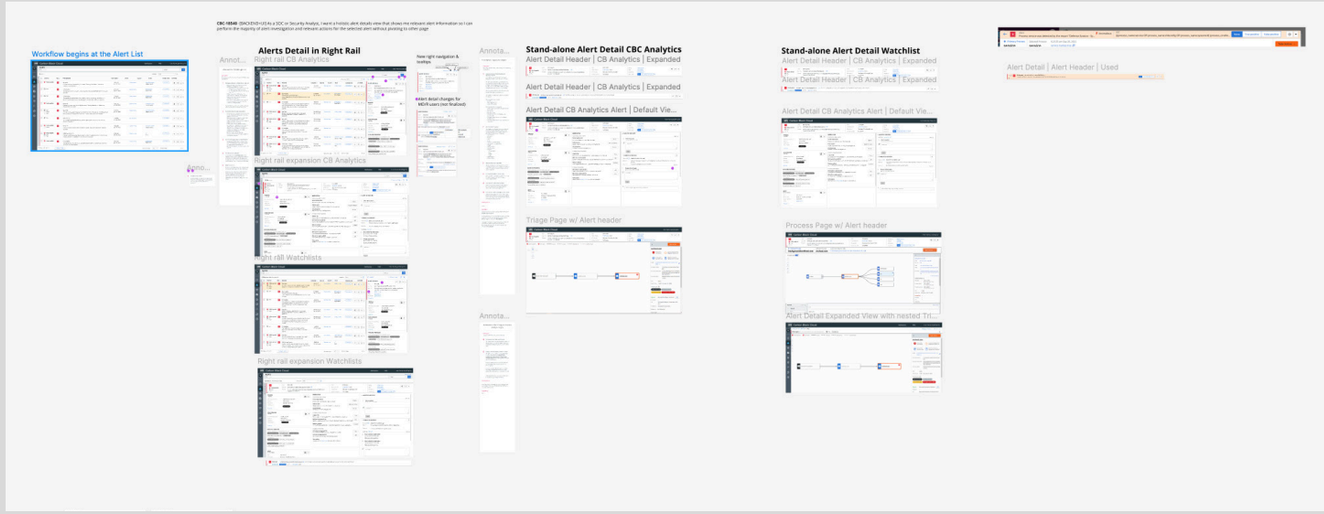




Alert Detail New Full Page Layout



Alert List to Alert Detail Page Navigation and Alert Browse



EPIC 4: Auto Close Enhancements

Addressing Auto-Closed Alert Rule Management for CB Customers

Background
CB customers have consistently complained about the disappearance of auto-closed alert rules into an inaccessible black box. These rules, created during alert closure, cannot be accessed or managed, leading to significant user frustration. Despite being a known issue, the product team faced challenges implementing the necessary enhancements.

Problem
The UX team questioned whether auto-closing alerts was the optimal solution. It was suggested that this feature might be better addressed at the organizational alert policy level, rather than through the current implementation.

Approach
To address this tension, I organized two design workshops aimed at resolving the conflict between these priorities.

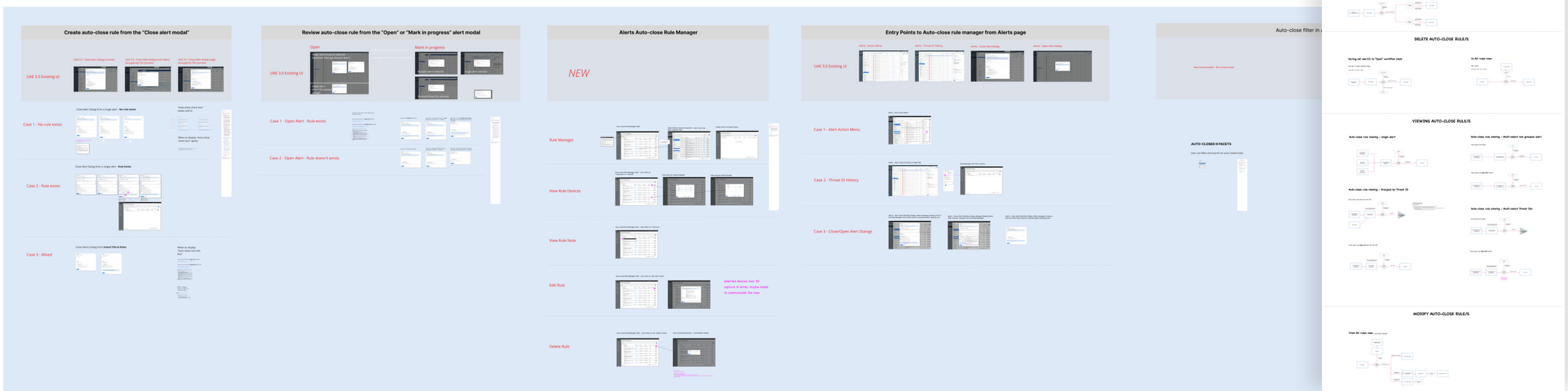
Research
I meticulously reviewed existing research highlighting customer pain points related to alert fatigue and multi-alert management. Key data was presented in design workshops to UX leadership to emphasize the urgency of resolving these issues, especially considering potential dependencies on platform architecture that was not yet ready for implementation.

Solution Exploration
We debated whether to enhance the existing auto-dismissal functionality or design an independent solution to be implemented at the policy level later. Through story mapping and deep dives into potential workflows, we discovered the need for a rule manager for accessing and managing auto-closure rules.

Design Process
During the UX team and alert group meetings, we discussed how customers would access this rule manager and other critical questions. We agreed on a solution that allows users to create auto-close rules during alert closure and access a rule manager to audit and edit these rules. This solution also enables users to filter auto-closed alerts and audit their auto-close triggering rules, providing a safety net and building a trust.

Testing and Validation
This limited experience was tested and validated, demonstrating relief from customer pain. Participants delighted in the new visibility and control over alert auto-close rule.

Outcome
The design process revealed the necessity of a rule manager, where all auto-close rules can be accessed and managed (edit rule endpoints, or delete rule) effectively. Users expected to find rule manager under Enforce section where policy is created.



Design Validation, User stories, Design requirements

UXR Research Goals

1. Validate Auto Closure Workflow:
   - Do users know how to auto-close alerts?
   - Do users understand how auto-closed alerts are defined?
2. Validate Understandability of Rules Manager
   - Where do users go to access the Rules Manager?
   - What information are they looking for in the Rules Manager?

User Story Requirements

Auto Closure Rule Viewing and Management:

- Users can easily understand and see what auto closure rules are applied to any alert.
- Users can view all auto closure rules applied across their entire environment.
- Users can see details about each auto closure rule, including:
  - Rule name
  - Threat/ThreatID
  - Scope (all devices or select devices)
  - Rule duration/expiration
- Users can delete auto closure rules currently in place for the alert or other auto closure rules in their environment.

Validate understandability to Rules Manager

Location

- Overall, participants were able to find the Rules Manager
- Most expected the Rules Manager to be located
  - Sub section Enforce
  - Sub section under Alerts
  - Group By option

Information

- The information presented to participants mostly fulfilled their needs.
  - Ability to see the reason for the rule
  - Who created it
  - What devices are being affected
- Would like to see how many alerts rule has closed
  - How active the rule is

32

Validate Auto-closure Workflow

How to auto-close alerts

- Overall, all of our participants were relatively familiar with how to auto-close future alerts. They have had experience with it before
- Some mentioned they did not like the extra click to expand and access the area because it was extra work

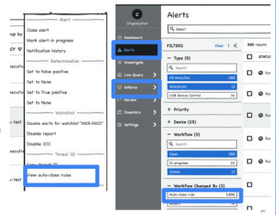
Auto-closed Rules Defined

- Participants knew that auto-close rules were grouped by Threat ID, however Threat ID is not clearly defined to them.
- The lack of confidence in what exactly makes up each Threat ID and how the alerts are exactly grouped causes lack of trust to auto-close future alerts

31

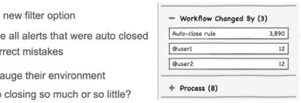
Rules Manager expected to be located under Enforce

- Most participants were able to find the Rules Manager but expected the Rules Manager to be located under Enforce
  - This is where they go for other rules such as Policy, so they believed it should exist in the same place
- Some participants believed it would be located under Alerts
- Some believed they could access it through the "Workflow Changed By - Auto-close Rules" and group the alerts by rule



"Workflow Changed by - Auto closed rules" facet understandable

- Participants noticed the new filter option
- Helpful because can see all alerts that were auto closed and can compare or correct mistakes
- Allows them to easier gauge their environment
  - Why are they auto closing so much or so little?
- When auto closing, participants believed all alerts closed initially should be listed as closed by user and all future can be closed by "Auto-close rule"
- This gives the participant more control and understanding which were closed by a person initially and how many were closed by the rule

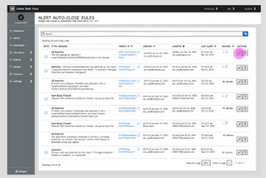


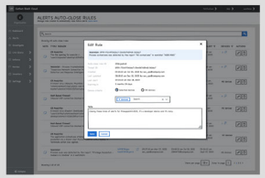
17

https://jira.carbonblack.local/browse/CBCUI-4901

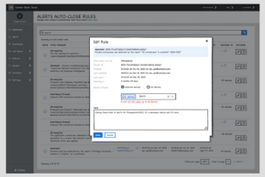
User can edit rule in the Rule Manager

Auto-close Rule Manager View - user clicks on rule "edit" action





Edit Rule Modal - Maximum number of devices

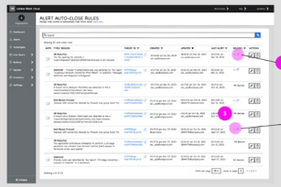


Alert Close Modal

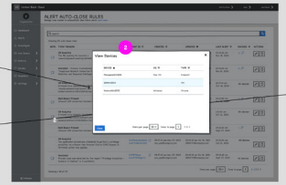
https://jira.carbonblack.local/browse/CBCUI-4899

User can preview rule devices in the Rule Manager

Auto-close Rule Manager View - user clicks on "3 devices" or "1 device"



View devices modal (multiple)



View devices modal (single)

Testing > Research Prototyping