Workiva updated on	1/17/2024
MAP Due Date	11/30/2023
IA Validation Date	12/18/2023
Validation Performed By	Yojita Ialpuria
Validation Reviewed By	Yanan Li
Review Date	01/10/2024
MAP Remediation Status	Remediated

I. Audit Observation:

IGD Private Messaging Restrictions: 1:1 Messaging
The Messenger Wellbeing team does not always prevent unconnected adults from initiating a 1:1 thread with teens on IGD. For example, on 6/10/2023, IA observed that out of 2.7M new threads initiated by an adult with a teen globally, 238K (9%) were from unconnected adults.

II. Management Action Plan:

The Messenger Wellbeing team will:
1.Establish thread initiation event-level logging and perform analysis to determine the effectiveness in unconnected adult - teen intervention in 1:1 threads.
2.Implement remediation to prevent the unconnected adults from initiating 1:1 threads with the teens.
3.Establish monitoring to track adult-side 1:1 messaging effectiveness and set up automated alerts to notify oncall in case of anomalies.

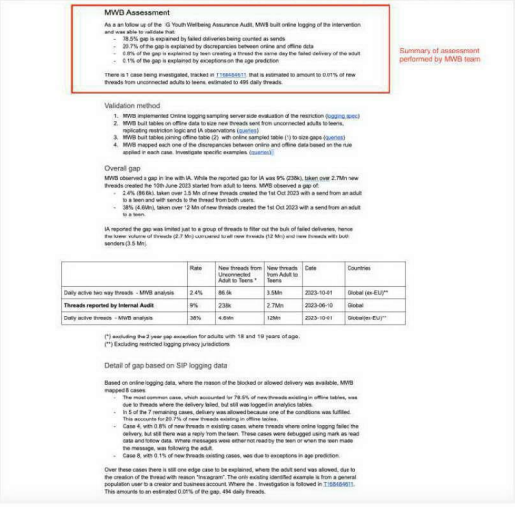
III. Management Remediation Approach:

1. MWB performed detailed analysis to determine the effectiveness of unconnected adult - teen intervention in 1:1 threads.
2. MWB implemented Online logging sampling server side evaluation of the restriction.
3. MWB built tables on offline data to size new threads sent from unconnected adults to teens, replicating restriction logic and IA testing logic.
4. MWB built tables joining offline table with online sampled table to size gaps.
5 Using online logging, MWB could validate a sample of the discrepancies identified by offline data based on the rule applied in each case and was able to validate that:
- 78.5% gap was due to failed deliveries being counted as sends
- 20.6% of the gap was explained by discrepancies between online and offline data
- 0.8% of the gap was explained by teen creating a thread the same day the failed delivery of the adult
- 0.1% of the gap was explained by exceptions on the age prediction
6. MWB established Dashboards for monitoring as follows: Audit dashboard , Server side logging . Information from the following tables was used to extract information for dashboard: agg_mwb_unami_evaluations for dashboards and monitoring, fct_mwb_unami_evaluations for case by case debugging
7. MWB set up automated alerts to notify oncall in case of anomalies as follows: Audit alerts , Composer block alerts

IA Validation Approach:

Step 1: Based on the explanation provided by MWB, IA reviewed the analysis document and noted that 99.99% of the leakage was explained as false positives.

Screenshot from the analysis document showing summary of assessment performed by MWB team



MWB Assessment

As a an follow up of the IG Youth Wellbeing Assurance Audit, MWB built online logging of the intervention and was able to validate that:
- 78.5% gap is explained by failed deliveries being counted as sends
- 20.7% of the gap is explained by discrepancies between online and offline data
- 0.8% of the gap is explained by teen creating a thread the same day the failed delivery of the adult
- 0.1% of the gap is explained by exceptions on the age prediction

There is 1 case being investigated, tracked in T166589671 that is estimated to amount to 0.01% of new threads from unconnected adults to teens, estimated to 495 daily threads.

Summary of assessment performed by MWB team

Validation method

1. MWB implemented Online logging sampling server side evaluation of the restriction (logging spec)
2. MWB built tables on offline data to size new threads sent from unconnected adults to teens, replicating restriction logic and IA observations (queries)
3. MWB built tables joining offline table (2) with online sampled table (1) to size gaps (queries)
4. MWB mapped each one of the discrepancies between online and offline data based on the rule applied in each case. Investigate specific examples (queries)

Overall gap

MWB observed a gap in line with IA. While the reported gap for IA was 9% (238k), taken over 2.7Mn new threads created the 10th June 2023 started from adult to teens. MWB observed a gap of:
- 2.4% (86.6k), taken over 3.5 Mn of new threads created the 1st Oct 2023 with a send from an adult to a teen and with sends to the thread from both users.
- 38% (4.6bln), taken over 12 Mn of new threads created the 1st Oct 2023 with a send from an adult to a teen.

IA reported the gap was limited just to a group of threads to filter out the bulk of failed deliveries, hence the lower volume of threads (2.7 Mn) compared to all new threads (12 Mn) and new threads with both senders (3.5 Mn).

	Rate	New threads from Unconnected Adult to Teens *	New threads from Adult to Teens	Date	Countries
Daily active two way threads - MWB analysis	2.4%	86.6k	3.5Mn	2023-10-01	Global (ex-EU)**
Threads reported by Internal Audit	9%	238k	2.7Mn	2023-06-10	Global
Daily active threads - MWB analysis	38%	4.6bln	12Mn	2023-10-01	Global(ex-EU)**

(*) excluding the 2 year gap exception for adults with 18 and 19 years of age.
(**) Excluding restricted logging privacy jurisdictions

Detail of gap based on SIP logging data

Based on online logging data, where the reason of the blocked or allowed delivery was available, MWB mapped 8 cases
- The most common case, which accounted for 78.6% of new threads existing in offline tables, was due to threads where the delivery failed, but still was logged in analytics tables.
- In 5 of the 7 remaining cases, delivery was allowed because one of the conditions was fulfilled. This accounts for 20.7% of new threads existing in offline tables.
- Case 4, with 0.8% of new threads in existing cases, where threads where online logging failed the delivery, but still there was a reply from the teen. These cases were debugged using mark as read data and follow data. Where messages was either not read by the teen or when the teen made the message, was following the adult.
- Case 8, with 0.1% of new threads existing cases, was due to exceptions in age prediction.

Over these cases there is still one edge case to be explained, where the adult send was allowed, due to the creation of the thread with reason "Instagram". The only existing identified example is from a general population user to a creator and business account. Where the . Investigation is followed in T166589671.
This amounts to an estimated 0.01% of the gap, 494 daily threads.

Step 2: IA evaluated the logging spec for implementation done around online logging sampling server side evaluation of the restriction.

Screenshot reflecting the logging specifications of the online table

Name	Data type	Description
actor_id	int	User_id
actor_struct	**int**	**Optional - Acting profile id (for SOAP)**
other_user_id	int	Subject id (Recipient_id)
thread_id	string	thread_id
~~other_id_type~~	~~string~~	~~Values: User / thread / content~~
conformed_interface	string	indicates it's coming from msgr android / IG iOS / ....
session_id	string	Identifier of execution
use_case	string	Example: IG_CLIENT_INTERVENTIONS
spec	string	Spec under evaluation Example: SafetyInterventionMinorToHighRiskAdultMediaShareEducationSpec
spec_version	int	Optional - Version number of spec
evaluators_seq	vec<string>	"SafetyInterventionHighRiskAdultEvaluator, SafetyInterventionTeenEvaluator, SafetyInterventionThreadAgeNewerThanEvaluator..."
evaluators_result	vec<int>	1 = true, 0 = false, 2 = not evaluated
evaluators_exception	vec<int>	1 = exception, 0 = no exception

Step 3: IA validated the queries for tables built on offline data to size new threads sent from unconnected adults to teens, replicating restriction logic which is as follows:
-Restriction is applied if ALL the following conditions are TRUE:
1. Thread is 1:1
2. The message recipient is teen (17 years old or less)
3. Message sender is adult (18 years old or more)
4. Teen is NOT following the adult
5. Teen has not sent messages in the past to the adult in the thread.
6. Restriction is not applied if adult stated age is 18 or 19 AND age difference with teen stated age is 1 or 2 years (e.g. users aged 19 years can message users aged 17 years but not users aged 16)

Screenshot reflecting the details of the restrictions implemented

```
SELECT
    ds,
    'Daily active threads' thread_type,
    COUNT_IF(
        starter_is_adult AND other_user_is_teen AND no_previous_sends
    ) new_threads_from_adults_to_minors,
    COUNT_IF(
        starter_is_adult AND other_user_is_teen AND no_previous_sends AND (NOT
is_2y_exception)
        AND NOT other_user_follows
```

Teen has not sent messages in the past

Sender is adult and recipient is teen

Teen is not following adult

```
    ) new_threads_from_acults_to_minors_exc_2y_exception

FROM tmp_mwb_isla_unami_offline
GROUP BY
    1

UNION ALL

SELECT
    ds,
    'Two way thread' thread_type,
    COUNT_IF(
        starter_is_adult AND other_user_is_teen AND no_previous_sends
    ) new_threads_from_acults_to_minors,
    COUNT_IF(
        starter_is_adult AND other_user_is_teen AND no_previous_sends AND (NOT
is_2y_exception)
            AND NOT other_user_follows
    ) new_threads_from_adults_to_minors_exc_2y_exception
FROM tmp_mwb_isla_unami_offline
WHERE
    both_thread_members_senders
GROUP BY
    1
```

```
-- STAGING QUERY CREATE TABLE tmp_mwb_isla_uamr_offline_stg
WITH new_threads_by_initiator AS (
        SELECT
            consistent_thread_fbid,
            resur_info_30d,
            FB_FBID_TO_IGID_V2(
                IF(
                    CAST(resur_info_30d['resurrector'] AS BIGINT) =
latest_participant_ids[1],
                    latest_participant_ids[1],
                    latest_participant_ids[2]
                )
            ) thread_starter_igid,
            FB_FBID_TO_IGID_V2(
                IF(
                    CAST(resur_info_30d['resurrector'] AS BIGINT) =
latest_participant_ids[1],
                    latest_participant_ids[2],
                    latest_participant_ids[1]
                )
            ) other_user_igid,
            num_sends_datelist_user,
            rtc_call_duration_datelist,
            ARRAY_SUM(TRIM_ARRAY(MAP_VALUES(num_sends_datelist_user)[1], 1)) = 0
                AND ARRAY_SUM(TRIM_ARRAY(MAP_VALUES(num_sends_datelist_user)[2], 1)) = 0
no_previous_sends,
            ARRAY_SUM(MAP_VALUES(num_sends_datelist_user)[1]) > 0
                AND ARRAY_SUM(MAP_VALUES(num_sends_datelist_user)[2]) > 0
both_thread_members_senders,
            ds
        FROM dim_threads.messages
        WHERE
            ds = '<DATEID>'
            -- 1G to 1G 1v1 threads
            AND interop_state = 2
            AND thread_type = 1
            -- NEW THREADS
            AND firstactiontime = '<DATEID>'
            -- With thread starter info, participants ids and sends data of both
participants
            AND CARDINALITY(latest_participant_ids) = 2
            AND resur_info_30d['resurrector'] IS NOT NULL
            AND COALESCE(
                TRY(
                    CONTAINS(
                        latest_participant_ids,
                        CAST(resur_info_30d['resurrector'] AS BIGINT)
                    )
                ),
                FALSE
```

Thread is 1:1

```
        AND du.userid = t.thread_starter_igid
)
SELECT
    COALESCE(
        ia.recommended_adult_minor_flag,
        IF(
            COALESCE(du.stated_age, 0) BETWEEN 13 AND 17,
            '13_17',
            'other'
        )
    ) = '13_17' other_user_is_teen,
    COALESCE(
        t.thread_starter_am_flag,
        IF(COALESCE(t.thread_starter_stated_age, 0) > 17, '18_plus', 'other')
    ) = '18_plus' starter_is_adult,
    -- 2y exception, a 18yo should be able to message a 16yo
    COALESCE(
        (
            du.stated_age BETWEEN 13 AND 17
        ) AND (
            t.thread_starter_stated_age BETWEEN 18 AND 19
        ) AND t.thread_starter_stated_age - du.stated_age <= 2,
        FALSE
    ) is_2y_exception,
    ia.recommended_adult_minor_flag other_user_am_flag,
    ia.predicted_output_adult_classifier_api other_user_predicted_age_api,
    du.stated_age other_user_stated_age,
```

2 year exception rule

Step 4: IA validated the query for tables built by joining offline table with online sampled table to size gaps.

Screenshot reflecting the details of the join established

```
Unset
-- CREATE TABLE tmp_mwb_isla_unami_offline_and_online

SELECT
        t.follow_taps_from_target_same_day_min_ts,
        t.other_user_follows,
        t.other_user_follow_date,
        t.other_user_is_teen,
        t.starter_is_adult,
        t.is_2y_exception,
        t.rtc_call_duration_datelist,
```

```
t.team_sends_direct_at_user,
t.sends_from_thread_starter,
t.other_user_am_flag,
t.other_user_predicted_age_min,
t.other_user_status_age,
t.other_user_account_tiers,
t.thread_starter_am_flag,
t.thread_starter_predicted_age_api,
t.thread_starter_stated_age,
t.starter_account_tiers,
t.consistent_thread_fbid,
t.recur_info_30d,
t.no_previous_sends,
t.both_thread_members_senders,
t.thread_starter_igid,
t.other_user_igid,

s.spec 'IS NOT NULL' in_online_logging,

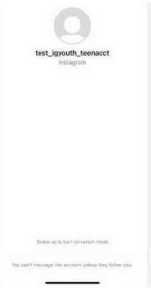
--  s.actor_id <>  s.other_user_id,
s.thread_id nip_thread_id,
s.session_id,
s.use_case,
s.spec,
s.spec_version,
s.evaluators_req,
s.evaluators_result,
s.evaluators_execution,
s.device_id,
s.device_type,
s.user_country,
s.device_os,
s.device_os_version,
s.app_id,
s.mobile_app_version,
s.confirmed_interface,
s.device_model,
--s.actor_was_actor,
s.spec_result,
s.ig_app_event_id,

FROM tmp_mvb_idle_users_offline t
LEFT JOIN safety_integrations.messages s
    ON s.ds = '<DAILY>'
    AND s.use_case = 7
    AND FB_FBID_TO_IGID_V2(s.actor_id) = t.thread_starter_igid
    AND FB_FBID_TO_IGID_V2(s.other_user_id) = t.other_user_igid
```

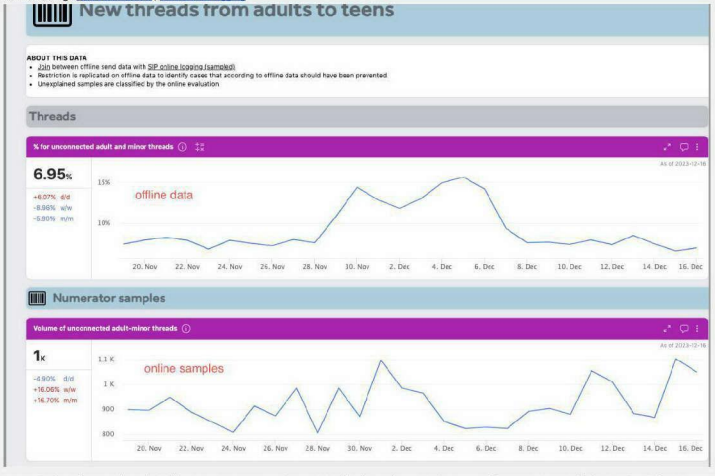
Step 5: IA performed transaction testing to validate if the Unconnected adult <> Minor restriction was enforced. Results of the sample testing are below.

	Sender 1: adult_testacct_IGY	Sender 2: adult_user_auditacct	Recipient 1: test_IGyouth_teenacct	Recipient 2: yoji_testaccount_new
TEST USER DOB				

S.N.	Sender | Recipient	Sender Side Screenshot	Recipient Side Screenshot	IA Comment
1	IGID:adult_testacct_IGY (unconnected adult DOB: Nov 2000) was unable to send a 1:1 message to test_IGyouth_teenacct(13 yr old).		NA	[NO EXCEPTION NOTED] The teen was not following the adult and so the adult was "unconnected". Unconnected adult <> Minor restriction was enforced through composer block or server-side block. Sender was restricted from sending message to receiver
2	IGID: adult_testacct_IGY (unconnected adult DOB: Nov 2000) was unable to send a 1:1 message to yoji_testaccount_new (34 yr old).		NA	[NO EXCEPTION NOTED] The teen was not following the adult and so the adult was "unconnected". Unconnected adult <> Minor restriction was enforced through composer block or server-side block. Sender was restricted from sending message to receiver

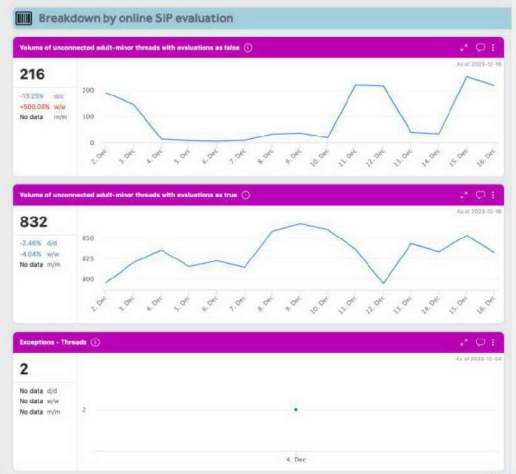
3	IGID:adult_user_auditact (unconnected adult DOB: Nov 1988) was unable to send a 1:1 message to test_IGyouth_teenacct (13 yr old).				NA	[NO EXCEPTION NOTED] The teen was not following the adult and so the adult was "unconnected". Unconnected adult <> Minor restriction was enforced through composer block or server-side block. Sender was restricted from sending message to receiver
4	IGID:adult_user_auditact (unconnected adult DOB: Nov 1988) was unable to send a 1:1 message to yoji_testaccount_new (14 yr old).				NA	[NO EXCEPTION NOTED] The teen was not following the adult and so the adult was "unconnected". Unconnected adult <> Minor restriction was enforced through composer block or server-side block. Sender was restricted from sending message to receiver

6. IA validated the following Dashboards established for monitoring: Audit dashboard , Server side logging.

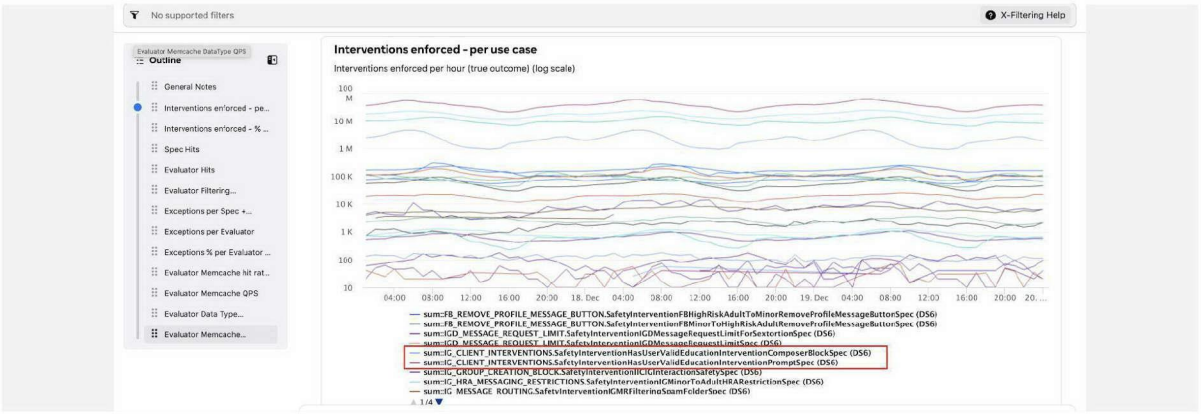


As can be seen from the dashboards below, MWB is monitoring from the samples selected, how many were passed or restricted by the evaluator and corresponding exceptions and false positives. When a message exchange is initiated an evaluator helps determine whether the exchange should happen/should not happen. An evaluator returns a evaluation of "True" if the criteria to implement UNAMI restriction is fulfilled which means message exchange should not happen. If the evaluator returns a evaluation of "False" it means the criteria to implement UNAMI restriction is not fulfilled which means message exchange can happen.The key dashboards being monitored are as follows:

1. "Volume of Unconnected adult-minor threads with evaluations as False" (Where evaluations resulted in false which means messages were allowed to be delivered by the evaluator and the message was actually sent due to reasons such as changed stated age or age prediction at the time of delivery).
2. "Volume of Unconnected adult-minor threads with evaluations as True" (Where evaluations resulted in true which means messages were not allowed to be delivered by the evaluator and the message was actually not delivered. These are failed deliveries and not actual sends.
3. "Exceptions" (actual exceptions found by joining online and offline tables which are not explained by above scenarios)



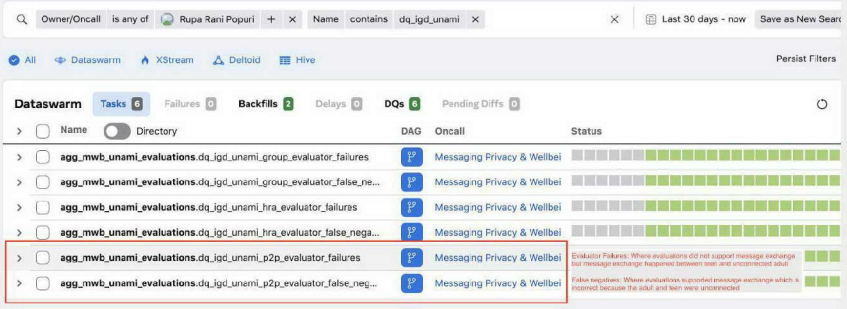




Step 7. IA validated the following automated alerts set up to notify oncall in case of anomalies: Audit alerts , Composer block alerts

Screenshot reflecting the alerts setup to measure Failed interventions and False negatives for 1:1 messaging

The Key alerts are as follows:
1. Failed interventions (where evaluations resulted in true but message exchange happened between teen and unconnected adult)
2. False negatives(where evaluations resulted in false which shouldn't happen since the adult and teen were unconnected)



Screenshot reflecting the alerts setup on composer blocks side for 1:1 messaging. This alert is about a WWW thrift service that detects unconnected adult minor restrictions on IGD. The thrift client uses service router to call from django, and the alert tracks the number of Service Router (SR) errors, which are mostly transient. Ideally, there should be zero errors

