Duo Security Outage - Responses and Planning for Future

The following page offers advice for planning and handling outages to the Duo Security service.

Monitoring Duo

1. Duo offers a specific status page, https://status.duo.com/ with outage information, and is a good place to start.
    a. what's this – ?] https://urldefense.proofpoint.com/v2/url?u=https-3A__status.duo.com_&d=CwIBaQ&c=y2w-uYmhgFWijp_IQN0DhA&r=lNKHpzGJV6eDhS9ywGFIM09rFusUQguE4Cr_8enAJAA&m=5w6F0y2jFRsDyUkRcpardKxYhYdFAvpe77QS5Jo9M5E&s=gYvqUnFWUuCjNqyK1zEqiwQW3S9XTmKm15su3l2EFDw&e=
2. Duo API host monitoring is prudent

    We learned today that simple ping monitoring of the Duo API host is
    insufficient. Based on today's incident, it appears that you should at
    least perform an HTTP GET on some API host resource and alert on slow
    responses and/or error status codes.

    a. I didn't see any data in the thread on how to actually do the monitoring, so I asked Duo in a support ticket and they recommended a particular link on the API host to pull:
    https://$APIHOST/auth/v2/ping

    where APIHOST is your specific hostname which would look something like:
    api-12345678.duosecurity.com

    This pulls a little JSON packet like so:
    {"response": {"time": 1454437771}, "stat": "OK"}

    The specific recommendation was "The best course of action for monitoring your API hostname to discover when an issue may occur would be to set up a heartbeat ping to your specific API hostname. This can be utilized with a script to automatically email if packets are consistently dropped or if a connection is unable to be made."

    I double-checked with them if this would be superior to the simple ping of the host, and they said yes it does exercise the application.
3. Different DUO services – Duo iFrame (Web SDK integration) VS use of the API; which endpoint to monitor depending on how a site is using DUO

Bypassing Duo

Choosing an approach – fail-open VS fail-closed – implications of choosing each approach

> It's a classic risk/cost balance. The right answer depends on tolerance for risk (in terms of less-secure authentication, as well as loss of the authentication service itself) and what price you're willing to pay.
>
> those applications that must remain protected (HIPPA, FISMA Moderate, in their opinion) remain protected during an incident. However, the much larger (generally speaking) user base of self service and less secure application can continue to operate in an event. This provides a middle of the road solution to protect that which "must" be protected and allow those with lower risk profiles to continue to operate.
>
> Fail-open becomes more defensible as a steady state if the IdP accurately reports the authentication mechanism used. Applications would be able to drop specific assertions or access requests above and beyond the protests of the IAM system, but users that had opted in to two factor wouldn't be locked out of everything.

A few solutions were offered to support a fail-open integration, to allow AuthN to continue in a weakened state:

1. (need cookbooks for both CAS and Shib)
2. It seems like the 'toggle' is something that Warren Curry, Brett Bieber and Rhian Resnick have a really good way of doing on a per-user basis, based on a live/replicated data source, that preserves authentication but can change authN context when needed, based on a service outage.
3. Configure IdP to check group membership before prompting for Duo, and remove users from the group to bypass.
    a. Nebraska uses a CAS Duo Extension configured to check for a specific attribute value memberOf: cn=psp:orgs:idm:DuoEnabled, ou=grouper,ou=group,dc=unl,dc=edu
4. ...

Communicating AuthN results to SPs

I.e., When the IdP is authenticating in bypass/fail-open mode, what should be sent to SPs indicating that the Authentication process that took place didn't include MFA?

There are two basic scenarios in which this question might be asked:.

SP requests/validates and IdP communicates MFA success

In this case, the SP explicitly requests MFA or (or the equivalent) through the requested authentication context(s). The IdP then communicates the fact of Duo (or any other MF) authentication success through a separate AuthN context.

The IdP must not (per SAML standards) assert Duo/MFA success if authentication is done via "fail-open", so if MFA fails, the SP authentication process will fail ("cannot authenticate using MFA") or be modified ("authenticated with password only"). Note that using approved alternatives to a primary MFA mode is not necessarily the same as "fail-open". E.g., if Duo push fails, but Duo authentication can be performed with other mechanisms (texting a code, use of a one-time code, etc.), it would still be acceptable for the IdP to indicate MFA success.

SPs explicitly requesting/consuming MFA contexts therefore need to either accept downtime when MFA systems fail or must be configured to allow for password-only authentication under appropriate circumstances. (E.g., the SP could be easily reconfigurable to allow for "Password Protected Transport" logins in the event of an MFA failure event).

IdP locally (and silently) enforces MFA

In this case the IdP uses local criteria (not based on specific requests from the SP) to decide whether to authenticate the user using MFA (e.g., flags on the user object in the IDM system). Typically in this case the IdP does not communicate the fact of MFA to the SP, instead indicating simply success of a "Password Protected Transport" login.

Here the SP is not able to (or presumably interested in trying to) determine whether MFA actually occurred as part of the authentication event, and is relying on the IdP to make the appropriate "authentication strength" decision. In this case the IdP can indicate authentication success if "fail-open" occurs, presuming that this is consistent with the IdP's normal operating practices. Pragmatically, if SPs are relying on IdP-managed and -enforced MFA support for increased security it is advisable to document MFA failure behavior (of defaulting to "fail-open" or "fail-closed") to ensure that the SP operators are aware of the impacts. That is, if the SP operator knows the IdP operator is enforcing MFA, but the fact of MFA is not communicated explicitly in the SAML assertion to the SP, then whether or not "fail-open" success is "acceptable" for user authentication is a business/SLA form of decision that cannot be inspected as part of the SAML/authentication conversation.

If the IdP supports "fail-open" operation, then SPs that do not wish to have authentication success in a "fail-open" mode would need to be addressed on a case-by-case basis by the IdP operator or, preferably, be reconfigured to explicitly request MFA support (as described above) so that the SP can make the determination of whether MFA was successful and take the appropriate action in response.