We have an increasing number of customers that have either migrated to cloud productivity solutions like Office 365 and G-Suite or plan on doing it soon. The migration is usually paired with a decent dose of anxiety over the degree of security visibility that is relinquished in the process.

As stewards of the tough to tackle security challenges “estate,” we feel obligated to share the various ways that our MDR+ security team addresses this problem. While we can’t disclose all of the ingenuity we provide to our customers, there are certainly a few things we can share that every organization can use to bolster their G-Suite security posture.

This is the part in the story where I tell you to turn on multi-factor authentication and all your worries will be addressed. Unfortunately, there’s plenty of recent evidence reconfirming that MFA is not a knight in shining armor. It turns out, that sand-filled “DEFENSE-IN-DEPTH” sock that you’ve been pummeling your executive leadership with still carries the best message. So with that groundwork laid, here’s the approach we recommend for our customers:

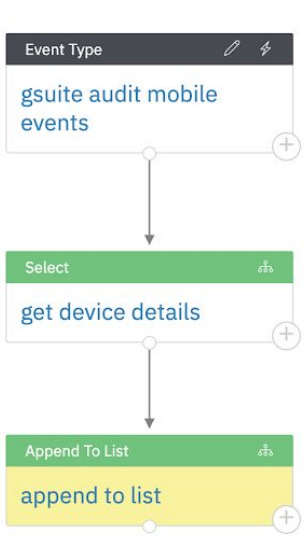
1) Yes, we do still think you should enable and enforce 2-step verification
2) Enable and enforce Mobile Device Management
3) Enable and enforce Endpoint Verification or Managed Browsers
4) Require administrative approval of managed mobile devices and endpoints/browsers
5) Set up a device loss reporting process

The mobile device management endpoint/browser features I’ve referenced above are built into G-Suite at no additional cost, so you don’t necessarily need to make a purchase to achieve these capabilities. But there is a caveat. G-Suite’s built-in endpoint verification and browser management only work with the Google Chrome browser. That said, it’s not the worst requirement. In fact, the managed browser capability provides some policy enforcement features that might be worth looking into all on their own. It’s also worth mentioning that steps four and five above involve a bit of manual labor so our final post in this series will discuss the methods we use for automating those processes. But for now, let’s focus on the threat detection benefits that come from the features described above.

The benefits of 2-step verification are well established so I’ll focus on suggestions 2 through 5. If every user in your organization is accessing email through a managed mobile device or browser, any login from a non-managed device is instantly suspicious and should thus require a few additional hurdles to obtain access. Forcing access to go through an administrative approval process can be one of those hurdles, which gives your team (and ours if you’re an MDR+ customer!) an opportunity to take a closer look at the situation. For example, maybe the login is associated with a new user that doesn’t have a managed device yet.

We also perform a series of machine learning supported user behavior modeling techniques to further determine if such a login is a significant deviation from the norm - such as a login from a country that hasn’t formerly been observed for a particular user. Such detections layered on top of this already strong enforcement serve to increase our confidence as to whether a login is a threat or not. In the world of incident response, a high degree of confidence improves our ability to take immediate (and in some cases even automated) action. I’ll cover a few of the machine learning supported threat detection techniques we use in the next post of this series. Let’s take a look at some simple ways we can leverage managed device details in an account takeover scenario.

When a user registers a managed device, a mobile audit event is generated that we can use to build a device tracking list.




Filter Table

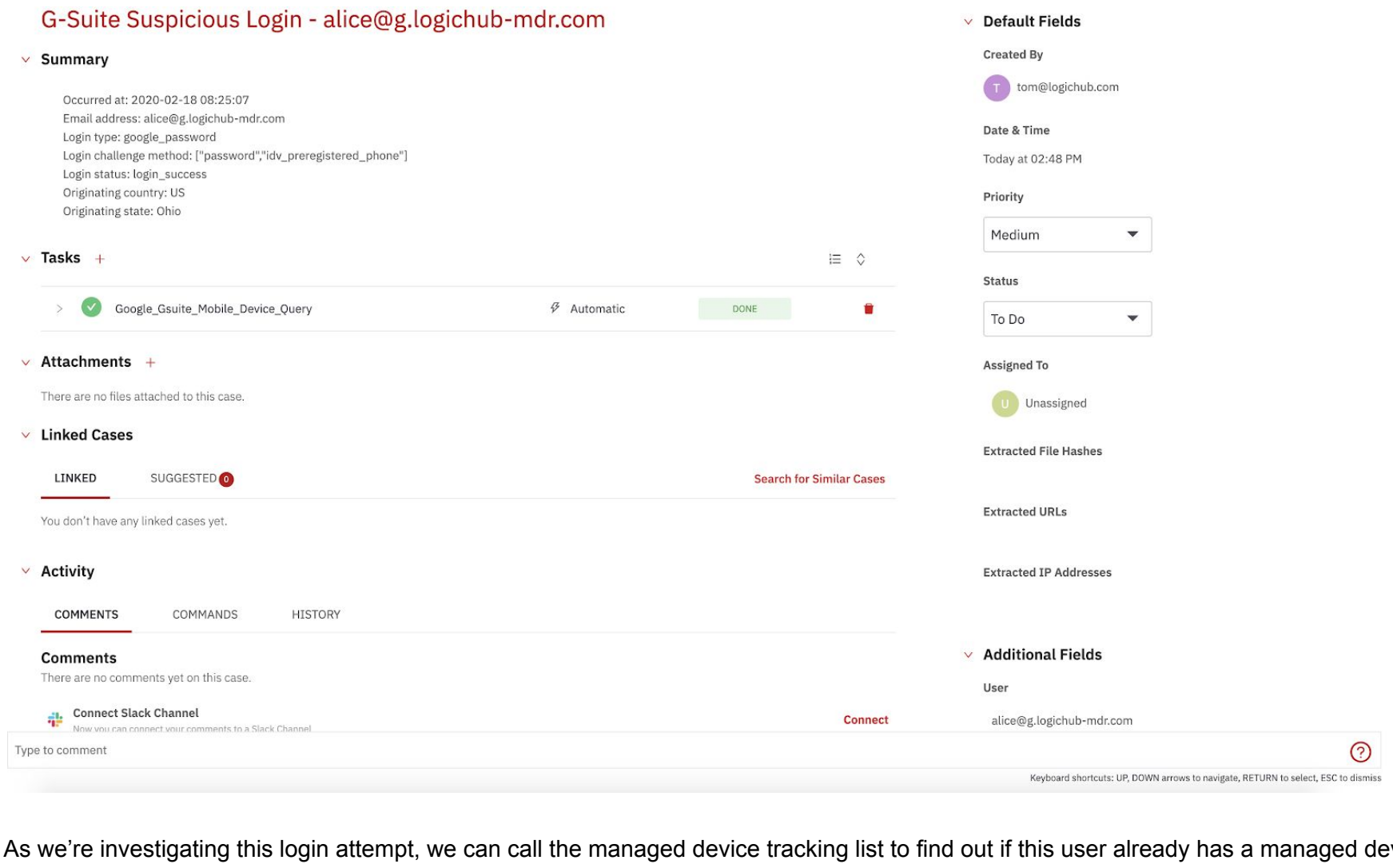
| email | device_id | device_model | serial_number | device_type |
|---|---|---|---|---|
| bob@g.logichub-mdr.com | 34e39441bc7ab80c | SM-G960U | 48434e5952573398 | ANDROID |
| alice@g.logichub-mdr.com | 2HOZ0-WIYODFzWIQQ-ll2O9gI3iDb13hhwHSsoyLkhA | HVM domU | ec23fa66-7e04-925a-3972-8a2d4583ffdc | WINDOWS |

When a user’s managed device is deleted by an administrator, an admin audit event is generated, which we can use to prune the list as needed.

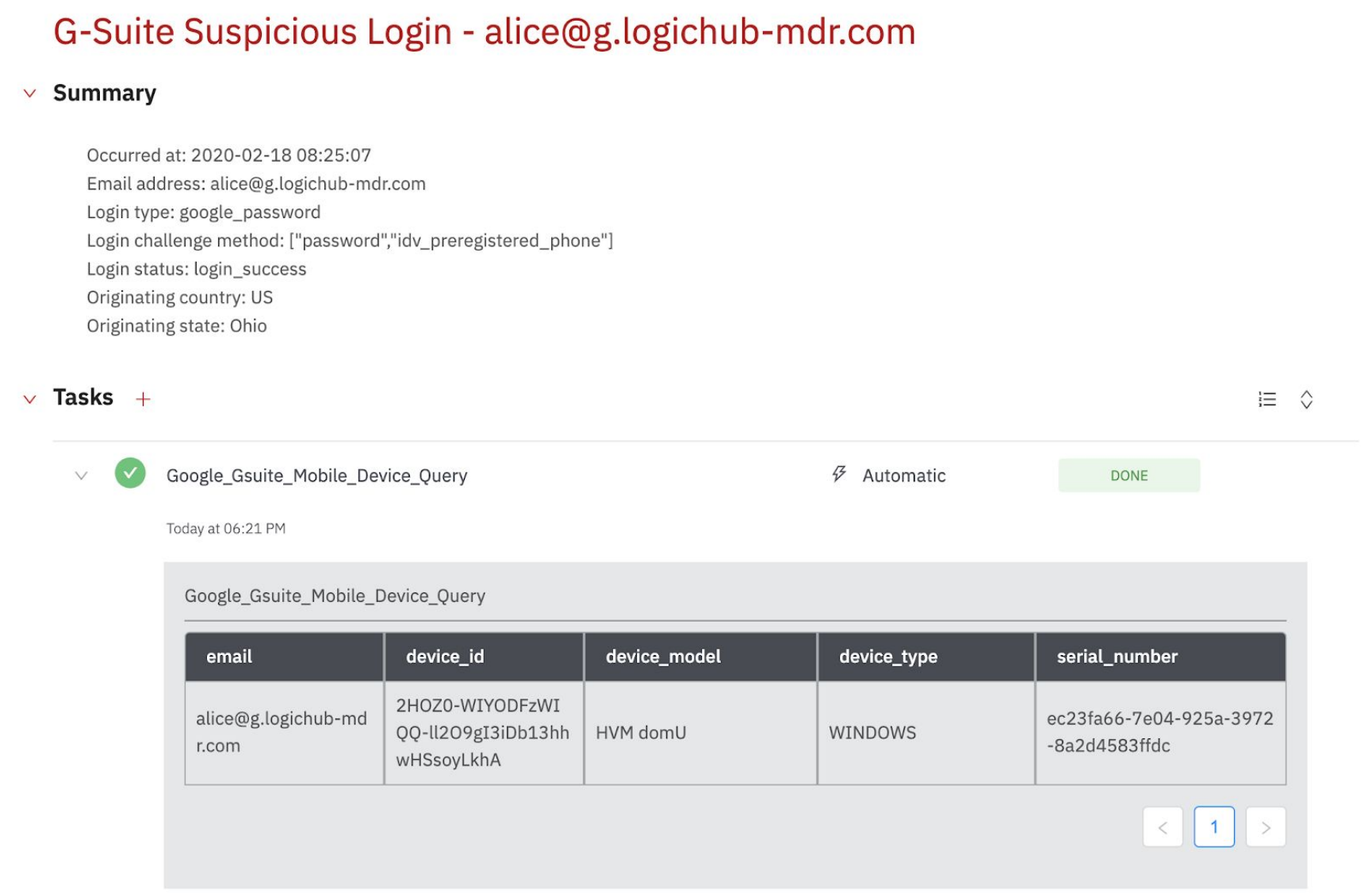

Filter Table

| device_id | device_type |
|---|---|
| XQ4hNbYmaYXY2MNS6M6IkkCjpnibjK71ksg2aleT2pY | WINDOWS |

We can also accomplish this type of device tracking by calling the G-Suite Admin API directly as needed to get a list of a user’s managed devices but that can get resource intensive if the organization isn’t fully migrated to managed devices, so we opt for the list-building from audit events route. With devices under management, it’s fairly easy to track logins that are coming from non-managed devices and generate cases.




As we're investigating this login attempt, we can call the managed device tracking list to find out if this user already has a managed device.




As you can see from the screen capture above, this user already has a managed Windows device so a new login from an unmanaged device could indeed be suspicious.

With the basic concepts covered, it's worth noting that a typical organization is not likely to make a full transition to managed devices overnight. It's also possible that a user might need to occasionally login from a non-managed device, which is why we've built a much more comprehensive threat detection and response playbook that I will cover in the next post of this series. The final post will address how we can automate the manual tasks of mobile device management.

I'll wrap up with a brief note about suggestion #5. As pervasive as BYOD has become, I'm amazed by how little attention is given to device loss reporting. It's a simple process that can provide an opportunity to mitigate a decidedly low-tech (in other words, easy to execute) attack vector. Do you have a stated policy? Better yet, do employees know what to do and who to contact in the event they lose a device?

Thanks for reading! Stay tuned for the next post in the series.