CSIRT Scenario: Takeout

Service background: Takeout is a service that allows customers to export their Customer Data from *YourExampleProduct* and download it. There are two methods of requesting a takeout-bundle (aka “dog-bag”):

1. Log in through the GUI to the service a dog-bag is requested for: Settings > GDPR Data Portability > Export. Select the desired data for export, and the front end infrastructure makes an API call to the Takeout Servers for a dog-bag to be generated.
2. Customers can use their credentials to authenticate to the Takeout service directly and leverage documented APIs to generate dog-bags in an automated fashion.

As soon as the customer downloads the data and confirms receipt, the data is destroyed from the staging area. There’s also a 7 day expiry from data generation to deletion – if the data hasn’t been downloaded by then, it is destroyed and a new takeout process needs to be started.

- Dog-bags can contain EVERYTHING in the customer product instance (except password hashes), or just a subset of data, for example, just documents that have been uploaded, or just the user database.
- Runs in Amazon AWS with Takeout API servers running on EC2 compute and storing data in ephemeral S3 buckets. All of the buckets are owned by the Takeout AWS account.
- S3 buckets have IP-based permissions so only *YourExampleProduct* application server IPs can access the data when generated via Application front end. When generated by direct call to takeout server API, part of the requirement is that the request must contain an IP list to be configured in an allowlist for downloading the resulting dog-bag.
- Containment is easy, the takeout team can just remove this VPC’s internet access, preventing anybody from talking to the API. This won’t take down access to existing S3 buckets, that’ll be “harder, and will take a while”. This will break the functionality for existing customers though. It’s used on average 23 times per week.

For the scenario, when answering questions, assume this is a generally well architected and secure service. Previous pentests have identified several API endpoints that didn’t require authentication, that were ***probably*** *fixed up, per the SME.*

Compromise Scenario: An alert has fired against this account for hitting a spending threshold. SMEs started investigating and determined that there were a number of buckets storing dog-bags that hadn't expired and weren't auto-destroyed. Retaining this excess data has caused the overage in costs and triggered the alert. SME's investigation determined that these dog-bags hadn't been provisioned via the approved flow. The oldest dog-bag observed is 10 months and 3 days old. Legitimate dog-bags are named in the format Service-Customer-Identifier-YYYY-MM-DD-HH-MM-SS. Illegitimate dog-bags are named in the format ######-{GUID}, incrementing. There are dog-bags that are 7.1 days old, indicating the issue is ongoing.

IF requested, a list of download URLs can be provided to the IM. These contain <string> which can be searched for in SIEM for finding evidence of download. If this is investigated, searches will show that the dog-bag is downloaded consistently ~10 hours (jitter of a few seconds) after it's generated. If requested, a search will show the most recently created dog-bag has not been downloaded. All downloads go to a VPN IP address belonging to a third-party VPN provider (always the same provider). The last unauthorized dog-bag was created 2 hours before the incident notification (inject) was sent.

Attack Timeline:

Note that a negative time value is pre-inject, a positive time value is post-inject, and subject to absence of corrective incident response procedures.

Time	Event
-10 months 6 days	Attacker starts probing the API endpoint for the takeout application for vulnerabilities
- 10 months 3 days	Attacker exploits takeout service for the first time, and uses it to create a complete takeout package for a single customer
…	~1100 unauthorized dog-bags are created, at a rate of around 23 per week
-2 hours	Last unauthorized dog-bag is created
+8 hours	Last unauthorized dog-bag created prior to inject is downloaded (in the absence of containment actions)

Inject:

> Hey Security,
>
> The takeout team has found an issue where dog-bags have been created and aren't following the normal rules of the service (aren't auto-expiring). This is impacting us because our hosting costs are becoming pretty astronomical. Looks like this has been going on for a while. Looks like their creation might be unauthorized, so my manager wanted me to let you know.
>
> I'll be banging my head against this for the next few hours, holler if you have questions!
>
> Tom Bombadil

Suggested questions for leadership/handoff to ask:

1. What is a dog-bag, or the takeout service?
2. What data is contained in the dog-bags?
3. Of those we've identified, what is the geo of the customers impacted?
4. When were we made aware of the issue?
5. How sure are we that these dog-bag creations and downloads are malicious?
6. How long has this been going on for?
7. How were the malicious dog-bags generated?
8. Have we taken down the existing malicious dog-bags?
9. Could a new dog-bag be created maliciously right now? (i.e. are we contained?)
10. Is <key customer> impacted?
11. Do we have EDR in that environment?
12. What should be investigated next as a priority?