The 5 W's of the data breach

What happened?
The security team at Cymbal Retail detected unusual activity within the cloud environment. Upon further investigation, it was confirmed that a significant compromise had occurred. The breach resulted in the exposure of credit card information—including the card numbers, user names, and associated locations—for a substantial number of users. The security team quickly got to work to investigate the security incident, gather information, and analyze all available log data to understand the scope of the breach and its impact.

Who was involved?
The security team at Cymbal Retail was unable to attribute the perpetrator of the data breach.

When did it happen?
The security team is still undergoing forensic analysis to determine the exact times related to the breach.

Where did it happen?
This security incident occurred on Cymbal Retail's cloud environment, where multiple cloud resources were affected.

Why did it happen?
The data breach happened because the malicious actor was able to exploit vulnerabilities in Cymbal Retail's cloud resources, including an insecurely configured firewall, bucket, and virtual machine (VM).

The cc-app-01 VM became infected with malware. The VM was configured with a public IP address and open SSH and RDP ports. This allowed the malicious actor to establish a connection through the open ports. The VM contained a default service account which had full access to APIs, and the malicious actor was able to successfully compromise the default service account, granting them unauthorized access into the VM.

The malicious actor then escalated their privilege and gained access to the service account's managed user key. This allowed them to further escalate the attack and target other services. Leveraging the compromised credentials, the malicious actor targeted BigQuery, where they successfully gained access to sensitive credit card information. To facilitate data exfiltration, the attacker identified and utilized a publicly accessible storage bucket with public internet access enabled. They then exfiltrated the credit card information.

Timeline

The following timeline is based on the available information and forensic analysis conducted during the investigation and details the malicious actor's attack path into the cloud environment.

1. Initial compromise

The initial compromise happened when the malicious actor identified the open and public SSH and RDP ports for the vulnerable application VM cc-app-01.

2. Unauthorized access

The malicious actor then established a connection through the open SSH and RDP ports to gain unauthorized access to the VM cc-app-01. The malicious actor used a brute-force attack to compromise weak credentials and successfully infiltrate the system.

3. Malware

The malicious actor infected the VM with malware to maintain their persistence in the VM.

4. Privilege escalation

The malicious actor gained access into the VM cc-app-01, where they explored the system and identified a user-managed service account key. The presence of this key enabled the malicious actor to compromise both user and service account credentials, granting them unauthorized access. By compromising the user and service account credentials, the attacker escalated their privilege and gained broader access within the Cymbal Retail cloud network.

5. Targeting BigQuery

The malicious actor then identified the service BigQuery, which contained sensitive customer data. Leveraging the compromised credentials, the malicious actor exploited the compromised credentials to gain unauthorized access to BigQuery.

6. Data exfiltration

The malicious actor identified non-encrypted credit card information stored within the compromised system. To facilitate data exfiltration, the attacker utilized a storage bucket that had public internet access enabled. They exfiltrated the credit card information, exporting it for later retrieval.

Vulnerabilities and remediation

The following vulnerabilities were identified and remediated for the affected storage bucket, virtual machine, and firewall.

Storage bucket

The findings in the following table indicate that the storage bucket was insecurely configured to be publicly accessible, lacked appropriate access controls, and did not have proper logging enabled. All of these vulnerabilities exposed the data stored in the bucket to unauthorized access.

Storage bucket			
Severity	**Vulnerability**	**Corresponding compliance rule**	**Remediation actions**
High	Public bucket ACL	Cloud Storage buckets should not be anonymously or publicly accessible.	To address the storage bucket vulnerability, public access to the storage bucket was removed. Fine-grained access was replaced with uniform bucket-level access control to ensure uniform access to all objects in the bucket by using only bucket-level permissions.
Medium	Bucket policy only disabled	N/A	
Low	Bucket logging disabled	N/A	

Virtual machine

The findings in the following table indicate that the virtual machine was insecurely configured with a public IP address, which allowed anyone on the internet to connect to it. The VM was also configured with no secure boot, which allowed the VM to potentially run unauthorized software during the boot process. The VM also used a default service account, which increased the attack surface of the VM. Lastly, the VM was granted full access to all APIs and connected to a known, malicious domain. All of these vulnerabilities and misconfigurations contributed to the data breach and enabled unauthorized access to a malicious actor.

Virtual machine (VM)			
Severity	**Vulnerability**	**Corresponding compliance rule**	**Remediation actions**
High	Public IP address	VMs should not be assigned public IP addresses.	To address the VM vulnerabilities, the infected VM cc-app-01 was shut down and deleted. A new VM, named cc-app-02, was created from a known and trusted snapshot, configured to use a private IP address, and enabled with secure boot.
Medium	Compute secure boot disabled	N/A	
Medium	Default service account used	N/A	
Medium	Full API access	Instances should not be configured to use the default service account with full access to all Cloud APIs.	
Low	Malware: bad domain	N/A	

Firewall

The findings in the following table indicate that the firewall was insecurely configured with open SSH and RDP ports, which allowed unrestricted access to SSH and RDP. The firewall also did not have firewall logging enabled, preventing the recording of information about network connections.

Firewall			
Severity	**Vulnerability**	**Corresponding compliance rule**	**Remediation action**
High	Open SSH port	Firewall rules should not allow connections from all IP addresses on TCP or SCTP port 22.	To address the firewall vulnerabilities, the firewall rules were adjusted to restrict SSH access to only the internal IP range 35.235.240.0/20, and firewall logging was enabled.
High	Open RDP port	Firewall rules should not allow connections from all IP addresses on TCP or UDP port 3389.	
Medium	Firewall rule logging	Firewall rule logging should be enabled so you can audit network access.	