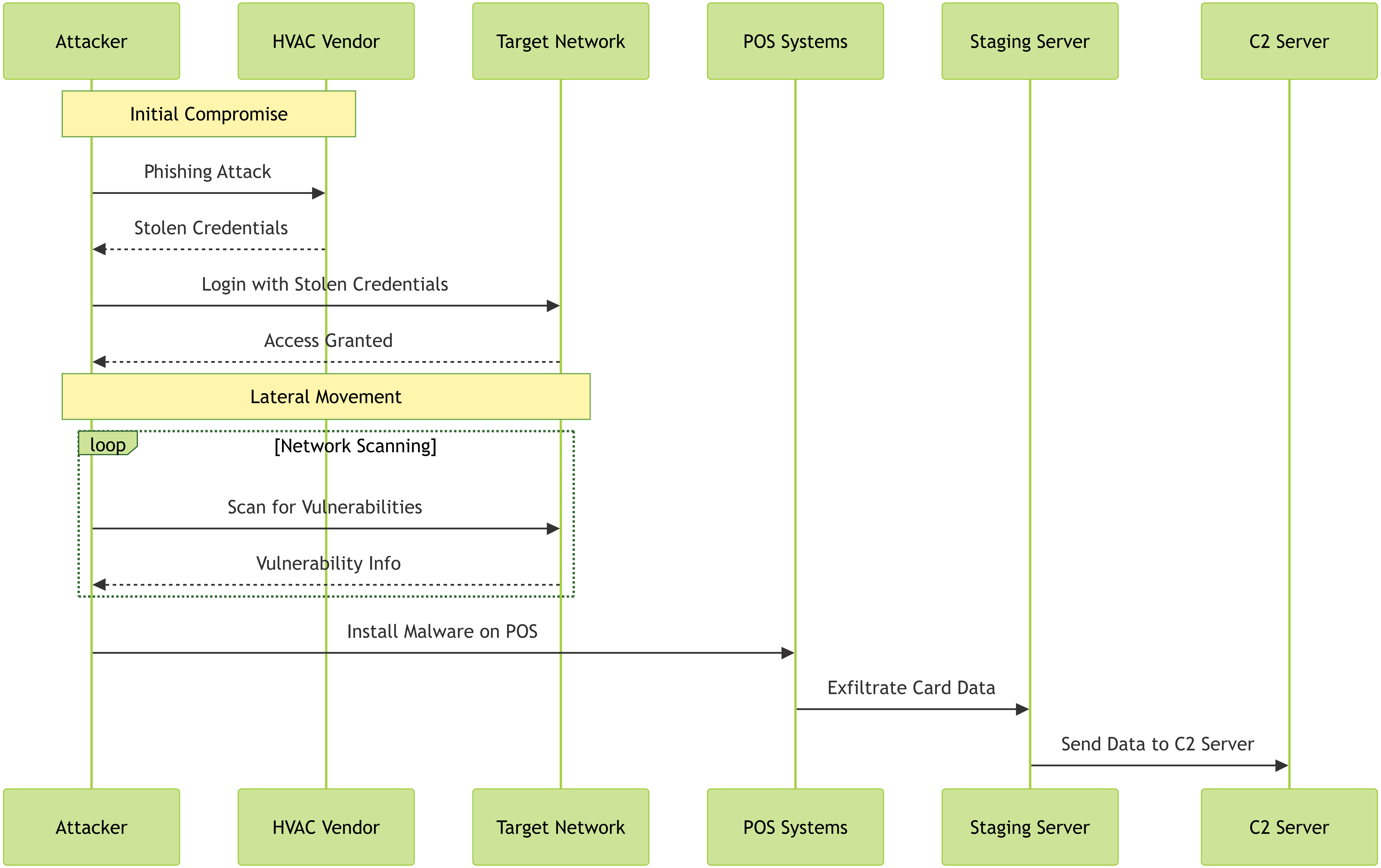
Network Security Security Architecture

For preventing Target & similar retail Breaches

Recap: The Target Breach (2013)

A failure of classic defense-in-depth against a determined APT.



Hands-on: Simulating Basic Network Scan (Nmap)

Attackers scan networks to find vulnerable systems for lateral movement.

```
# On Linux/macOS (requires Nmap installed)
# Scan a target IP for common open ports
# WARNING: Only scan systems you have explicit permission to test
# Example: Scan localhost
sudo nmap -sV -T4 127.0.0.1

# In Docker (using nmap image)
docker run --rm instrumentisto/nmap -sV -T4 <target_ip>
```

Note: Replace <target_ip> with an authorized target IP.

"The Target breach started with compromised HVAC vendor credentials - a classic 'Trusted Relationship' initial access (MITRE T1199). Once inside, weak internal segmentation allowed attackers to move laterally from a low-value network segment to the highly sensitive Point-of-Sale environment. Their RAM scraping malware wasn't detected, and the massive data exfiltration went unnoticed for weeks. This highlights the failure of relying solely on perimeter defenses and signature-based detection."

> **Q:** Why was network segmentation so critical in the Target breach?
>
> **A:** Proper segmentation would have prevented the attackers, who initially gained access to a less sensitive network zone (likely for vendor management), from being able to directly reach or scan the POS systems in the PCI-controlled zone. It acts as a firewall *inside* the network."

Introduction to Modern Security Architecture

Key components to prevent breaches like Target's: Zero Trust Architecture, Micro-segmentation, Extended Detection and Response (XDR), Security Orchestration, Automation, and Response (SOAR). Using AWS services to implement these components.

Zero Trust Architecture on AWS

Always verify access based on identity, device, and behavior. Use AWS IAM for least privilege and MFA.

Hands-on: Setting up IAM Roles

```
# Create a policy
aws iam create-policy --policy-name EC2ReadOnly --policy-document

# Create a role
aws iam create-role --role-name EC2Role --assume-role-policy-docu

# Attach policy to role
aws iam attach-role-policy --role-name EC2Role --policy-arn arn:a
```

Ensure policy.json and trust.json are properly configured.

"In Zero Trust, we don't trust any user or device by default, even if they are inside the network. AWS IAM helps enforce least privilege by allowing us to define precise permissions for each role. Here, we're creating a role for EC2 instances that can only describe EC2 resources, nothing more."

> **Q:** What is the benefit of using IAM roles over user accounts for EC2 instances?
>
> **A:** IAM roles provide temporary credentials that are automatically rotated, reducing the risk of credential theft. They also allow for easier management of permissions at scale."

Network Segmentation in AWS

Isolate systems using Amazon VPCs, security groups, and network ACLs to limit lateral movement.

Hands-on: Configuring Security Groups

```
# Create security group
aws ec2 create-security-group --group-name POS-SG --description "

# Allow SSH from management IP
aws ec2 authorize-security-group-ingress --group-id sg-12345678 -

# Allow HTTP from API Gateway security group
aws ec2 authorize-security-group-ingress --group-id sg-12345678 -
```

"In this setup, the POS systems are in their own VPC, and access is strictly controlled. Management can SSH via a jump host, and vendors interact through APIs, not direct network access. Security groups act as virtual firewalls for instances, allowing only specified traffic."

> **Q:** What’s the difference between security groups and network ACLs?
>
> **A:** Security groups are stateful and associated with instances, while network ACLs are stateless and associated with subnets. Security groups are generally used for finer-grained control at the instance level."

Threat Detection with AWS Services

Use Amazon GuardDuty for intelligent threat detection, AWS CloudTrail for API logging, AWS Security Hub for centralized monitoring.

Hands-on: Enabling GuardDuty

```
# Create GuardDuty detector
aws guardduty create-detector --enable

# List findings (after some time)
aws guardduty list-findings --detector-id <detector-id>
```

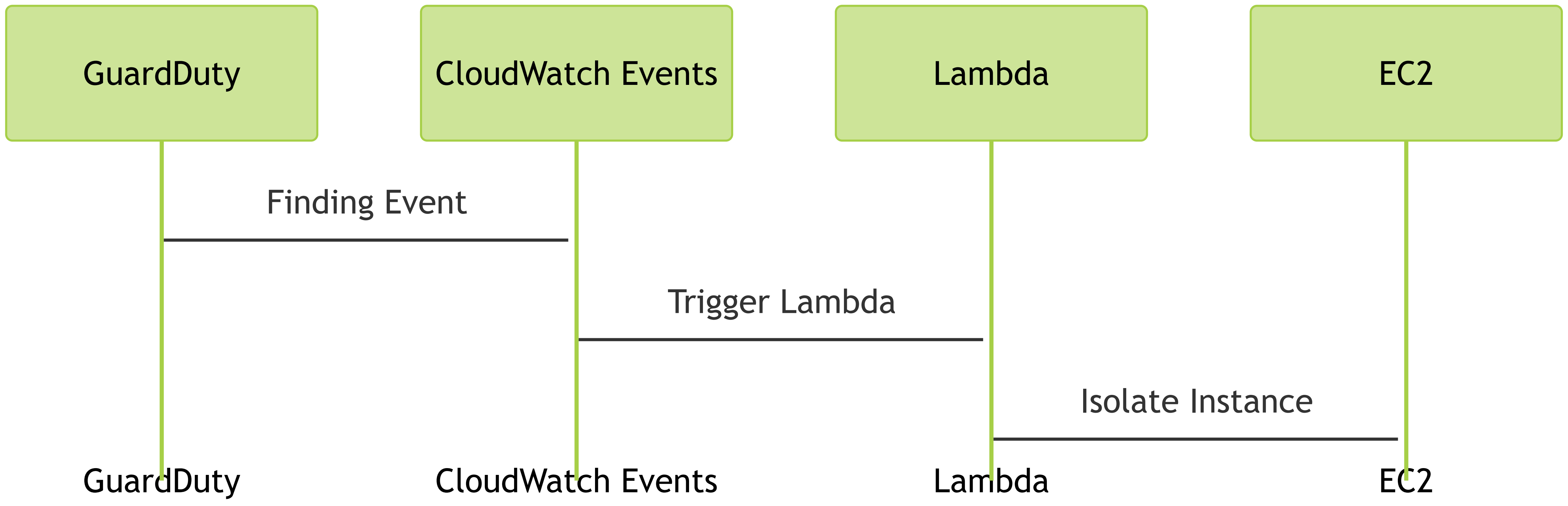
"GuardDuty analyzes continuous streams of data from your AWS accounts and generates findings when it identifies potential security issues. It's a managed service that requires minimal configuration, making it easy to enhance your security posture."

Q: What types of threats can GuardDuty detect?

A: GuardDuty can detect threats like reconnaissance, instance compromise, account compromise, and bucket compromise, among others."

Automated Response using AWS Lambda

Set up automated actions in response to security events using Lambda and CloudWatch Events.



Hands-on: Creating a Lambda Function

```
# Lambda function code (Python)
import boto3

def lambda_handler(event, context):
    ec2 = boto3.client('ec2')
    instance_id = event['detail']['resource']['instanceDetails'][
    ec2.stop_instances(InstanceIds=[instance_id])
    return {
        'statusCode': 200,
        'body': f'Stopped instance {instance_id}'
    }

# Create CloudWatch Events rule
aws events put-rule --name GuardDutyFindingRule --event-pattern '
```

"This setup allows for immediate action when a threat is detected. The Lambda function stops the compromised instance, preventing further damage. You can customize the response based on the type of finding or other criteria."

> **Q:** What are some other actions you can automate with Lambda in response to security events?
>
> **A:** You can snapshot the instance for forensics, revoke IAM credentials, send notifications, or even trigger a full incident response workflow."

Data Protection and Encryption

Use AWS KMS for encrypting data at rest and in transit, Amazon Macie for data discovery.

Hands-on: Encrypting an S3 Bucket

```
aws s3api put-bucket-encryption --bucket my-bucket --server-side-
```

"Encrypting data ensures that even if attackers access storage, they can't read the data without the key. AWS KMS makes this easy to manage and integrate with other services."

Q: Why is encryption important for retail data?

A: Retail data, especially payment information, is highly sensitive. Encryption protects it from unauthorized access, ensuring compliance with regulations like PCI DSS."

Application Security with AWS WAF and API Gateway

Protect web applications with AWS WAF, secure APIs with API Gateway.

Hands-on: Setting up WAF Rules

```
aws wafv2 create-web-acl --name MyWAF --scope REGIONAL --default-
```

"WAF protects against common web exploits like SQL injection, while API Gateway ensures secure and authenticated API access, critical for customer-facing retail applications."

> **Q:** How does WAF help in preventing application-layer attacks?
>
> **A:** WAF filters and monitors HTTP/S traffic, blocking malicious requests like SQL injection or cross-site scripting, protecting the application layer."

Unified Visibility with XDR

Correlates data across domains for better detection using tools like Amazon Detective or third-party XDR solutions integrated with AWS.

Hands-on: Analyzing GuardDuty Findings

```
# List GuardDuty findings
aws guardduty list-findings --detector-id <detector-id>

# Get details of a finding
aws guardduty get-findings --detector-id <detector-id> --finding-
```

"XDR provides a unified view by correlating data from various sources. In AWS, services like GuardDuty and Detective help in identifying and investigating threats across your environment."

> **Q:** How does XDR differ from traditional SIEM?
>
> **A:** XDR integrates deeply with endpoint, network, and cloud data, using AI/ML for advanced threat detection and response, whereas SIEM primarily focuses on log collection and correlation."

Automated Response with SOAR

Orchestrate security tools and automate response actions using AWS Step Functions or third-party SOAR platforms.

Hands-on: Simulating Automated Response

```
# Similar to the Lambda example, but can be extended with Step Fu
# For example, creating a Step Function that triggers multiple ac
# This requires defining a state machine, which is more involved.
# For simplicity, refer to the Lambda example.
```

"SOAR platforms automate incident response workflows. In AWS, you can use Step Functions to orchestrate multiple services and actions in response to security events, ensuring rapid containment and remediation."

> **Q:** What are the benefits of using SOAR in security operations?
>
> **A:** SOAR reduces response times, minimizes human error, and allows security teams to focus on more strategic tasks by automating routine incident response activities."

Preventing Breaches Like Target’s

Modern security architecture addresses key failures: Vendor Access Control with strict IAM policies and MFA, Network Segmentation with isolated VPCs, Malware Detection with GuardDuty and XDR, Exfiltration Monitoring with CloudTrail and flow logs.

"By implementing these measures, organizations can significantly reduce the risk of breaches similar to Target's. Each component works together to create a defense-in-depth strategy that is resilient against modern threats."

Q: What’s the most critical measure to prevent vendor-related breaches?

A: Strict vendor access control with IAM policies, MFA, and limited scopes is critical to prevent initial access, as seen in the Target breach."

Conclusion and Q&A

Modern network security architecture leverages Zero Trust, micro-segmentation, advanced detection, and automation to protect against sophisticated threats. Using AWS services, organizations can implement these principles effectively.

"We've covered how to build a secure architecture using AWS to prevent breaches like the Target incident. Now, let's open the floor for any questions you might have."