**Data Breach Scenario – Assessment Steps, Recovery, and Response.**

**Overview:**

Security breach via Web application, exploiting an unpatched vulnerability. This allowed unauthorized access to the internal network.

1. **Possible attacks used:** 
   1. SQL Injection attack could be used to hijack the application to return internal data via injected SQL queries, masquerading as an HTTP request.
   2. Container image, code base, or 3rd party dependency vulnerabilities.
   3. If not all management ports secured externally: a Brute-force terminal or RDP/SSH session with:
      1. Simple admin account names, and/or passwords
      2. Privileged system or service account credentials or tokens exposed in code.
      3. Token generation tools not properly obfuscated in code, which can be retrieved via HTTP, REST, or other web calls.
      4. Compromised administrator credentials or hijacked admin account session token.
      5. Zero-day vulnerability
   4. For Azure web app services (PaaS):
      1. SCM / Kudu management web controls and terminal is publicly exposed and accessed via a compromised admin credential, hijacked admin session token.
2. **Incident Response:**
   1. **Gather information for initial short term response steps.**
      1. Consult Microsoft Defender on the compromised local machine(s) orweb app services affected for any anomalous alerts.
      2. Consult Microsoft Defender for cloud:
         1. Check for related anomalies in the security alerts section.
         2. Run an attack path analysis from the web application.
      3. If admin account or other exposed credentials observed to be used by suspected attacker, act to:
         1. Revoke all active sessions for that user.
         2. Reset passwords.
         3. Contact user over the phone, or other a live call to assess the situation.
         4. Disable account if not actively used.
      4. Determine attacker’s IP/Ips and block them at the firewall/NSG/ASG level.
         1. If attacks are coming from a particular region/country this page does not serve, (i.e. Russia, North Korea, etc.) WAF policies will allow for country/region blocking.
      5. Ensure management ports (22,3389,5589, and other ports) used by attackers based on web traffic logs are enabled only for internal access using ASG/NSG/ and or WAF.
         1. Similarly with web-based management tools (SCM/Kudu).
      6. Perform emergency patch procedure on vulnerable web app servers, moving patched web services into place, and removing vulnerable ones, with as little service interruption as possible.
      7. If using Azure Front Door Standard, upgrade to AFD Premium which will enable a suite of Microsoft managed firewall rules via WAFv2 policy, to prevent many common attack vectors, including code and SQL Injection attacks, for web endpoints.
      8. On WAF settings switch to prevention mode if on detection mode.
   2. **Once initial response is complete, assess the data and Web server integrity:**
      1. Review network diagrams, mapping the web app to all attached networks and assets potentially exposed.
      2. Run defender on all public facing or potentially compromised assets and Microsoft for Defender cloud-based scans on the subscription and or resource groups affected.
      3. Review any Log analytics records and Azure Activity Logs against exposed assets.
      4. For SQL or Storage accounts determine what tables, containers, or file systems were accessed.
         1. Analyze for what data was accessed, deleted, or modified for the time period.
         2. Use Point-in-time restore, or backups for these services to restore damaged/deleted/altered data.
      5. Replace compromised VMs with newly created fully patched OS, and latest code base, behind the current WAFv2 enabled endpoint.
         1. Power down and destroy compromised VMs, retaining the OS and data Disks for later analysis, and as needed by data breach policy.
      6. Retain all relevant log analytics data, Firewall logs, VM event logs and disk in an internal only accessible storage account or other secure long-term storage retention solution.
3. **Network Security Measures:**
   1. **Limit VM/Web App internal vnet access by the web app.**
      1. Decouple Vnets from the Web app VNet which are not required for daily access, using an NSG or internal firewall to block outbound local VNet and internet traffic. This should both prevent network discovery and prevent extrication of data.
      2. Add private endpoints to include minimum required SQL servers, storage accounts, and internal API services to an approved subnet.
      3. Use SAS tokens limited to read access and only to the necessary containers required by the web app.
      4. Limit outbound access to only the app used subnets and to just the ports/services needed for minimum communication.
      5. Allow management ports inbound from trusted private networks only.
      6. If downstream service with a with global load balancing public endpoint may be at fault, consider using Cross-Origin Resource Sharing to allow for trusted incoming URLs.
   2. **Replace all weak admin accounts, roll service principals credentials, or replace service principals with User Assigned Managed Identities where possible.** 
      1. If storage account key compromised, roll compromised key, or turn off Key-based access, instead using Entra ID-based credentials and managed identities where possible.
   3. **SQL Servers:**
      1. Ensure that SQL is also patched to current standards
      2. SQL accounts used but the web app going forward are appropriately limited to read only the minimum DBs and tables required to perform its job.
      3. Disable the use of SQL Auth accounts if possible.
      4. Ensure public end points are disabled.
      5. Use dynamic data masking for columns which contain customer PII and other sensitive data.
      6. Assess user and service account access to these resources and reduce access to follow least-privileged principals.
      7. Reduce access to the DB server and DBs to only approved networks, resources, and personnel.
   4. **Create Azure Monitor alert events to trigger based on events discovered during the analysis.** 
      1. Ensure that log events from the affected APIs, firewalls, networks, SQL servers, Storage accounts and other services are integrated into Azure Sentinel or other applicable SIEM for global visibility.
   5. **On the Azure Front Door or Application Gateway WAFv2 policy:**
      1. Consider rate limiting rules on your public endpoints. This can narrow demanding traffic from individual sessions, reducing load and helping to prevent denial of service attacks.
      2. Consider regional blocking to allow for only regions/countries where you expect incoming traffic.
      3. Enable Azure DDoS protection plans your WAF enabled endpoints.