UNDERSTANDING DOMAIN NAME SYSTEM (DNS) FUNCTIONALITY AND SECURITY THREATS WITH EXAMPLES AND SIMULATIONS

BY IZZMIER IZZUDDIN

Table of Contents

DOMAIN NAME SYSTEM (DNS)

What Is DNS?

DNS is one of the foundational technologies of the internet. It's often referred to as the "phone book" of the internet because it translates human-readable domain names (e.g., www.example.com) into IP addresses (e.g., 192.168.1.1) that computers use to locate resources on the internet.

How DNS Works

1. Humans interact with websites and services via domain names (like example.com) because they are easier to remember than numerical IP addresses. Computers, however, use IP addresses to communicate.

2. When you type a URL into your web browser, your device (e.g., computer, smartphone) needs to find the IP address of the server hosting the website. It starts by querying a DNS resolver, which is typically provided by your Internet Service Provider (ISP) or a third-party DNS service (e.g., Google DNS or Cloudflare DNS).

3. Here's a step-by-step breakdown of what happens during a DNS query

    - **Step 1: DNS Cache Check**
      The DNS resolver first checks its local cache to see if it already knows the IP address for the requested domain name. If the address is cached, the resolver returns it to the client without further steps.

    - **Step 2: Root Server Request**
      If the IP address is not in the cache, the resolver queries one of the DNS root servers. These root servers are the starting point for DNS and are operated by organisations like ICANN. The root server doesn't know the IP address but can direct the resolver to the appropriate Top-Level Domain (TLD) server (e.g., for .com, .net, .org).

    - **Step 3:TLD Server Request**
      The DNS resolver queries the TLD server (e.g., .com server). The TLD server will respond with the authoritative DNS server responsible for the domain name (e.g., for example.com).

    - **Step 4: Authoritative DNS Server Request**
      Finally, the resolver queries the authoritative DNS server for the domain (e.g., ns1.example.com). This server knows the exact IP address for the requested domain and returns it to the resolver.

    - **Step 5: Response to Client**
      The DNS resolver now has the IP address and sends it back to the client (your

device). Your web browser can now use the IP address to send requests to the web server and display the website.

4. **Recursive vs. Iterative DNS Queries**

   - **Recursive Query**
     When a DNS resolver queries another server on behalf of a client until it gets the final answer (IP address).

   - **Iterative Query**
     When a DNS server refers the resolver to another DNS server (as seen in the TLD and authoritative server steps), allowing the resolver to continue the process.

Example How DNS Works

Scenario: You want to visit www.example.com on your web browser.

Step-by-Step Simulation

1. **User Request**

   - You open your browser and type www.example.com in the address bar.
   - Your computer needs to know the IP address for www.example.com to retrieve the website.

2. **DNS Resolver**

   - The browser first checks if it has cached the IP address of www.example.com.
   - If the IP address is not cached, the browser sends a DNS query to the recursive resolver, usually provided by your Internet Service Provider (ISP) or a public DNS resolver like Google DNS (8.8.8.8).

3. **Checking Local Cache**

   - The recursive resolver checks its own cache to see if it has the IP address for www.example.com. Let's assume it doesn't have it cached, so it moves forward.

4. **Query to Root DNS Server**

   - The resolver sends a request to a Root DNS Server, asking for the address of www.example.com. The Root server does not know the IP address of www.example.com, but it knows which Top-Level Domain (TLD) server is responsible for the .com domain.
   - The Root DNS server responds with the address of the .com TLD DNS server.

5. **Query to TLD Server**

- The recursive resolver now queries the TLD DNS server for .com domains and asks for the DNS server responsible for example.com.
- The .com TLD server responds with the address of the Authoritative DNS server for example.com.

6. **Query to Authoritative DNS Server**
   - The resolver sends a query to the Authoritative DNS server for example.com, asking for the IP address of www.example.com.
   - The Authoritative DNS server looks up its records and responds with the IP address: 192.0.2.1.
7. **Returning the IP Address**
   - The recursive resolver now has the IP address for www.example.com (192.0.2.1) and it caches this information for future use.
   - The resolver sends the IP address back to your computer.
8. **Accessing the Website**
   - Your browser receives the IP address and uses it to send a request to the server at 192.0.2.1 to load the webpage.
   - The server responds by delivering the content of www.example.com and the website appears on your screen.

Diagram of DNS Resolution

User ---> DNS Resolver (ISP/Google) ---> Root DNS Server ---> TLD DNS Server (.com) ---> Authoritative DNS Server (example.com) ---> Website IP (192.0.2.1) ---> User's Browser

DNS Record

Here's a breakdown of different DNS records you might encounter in this process

- **A Record:** example.com A 192.0.2.1
  - This is the IP address record that maps example.com to its IPv4 address.
- **CNAME Record:** www.example.com CNAME example.com
  - This record indicates that www.example.com is an alias of example.com.
- **MX Record:** example.com MX 10 mail.example.com
  - This specifies the mail server responsible for handling emails for example.com.

- **NS Record:** example.com NS ns1.example.com
    - This indicates the authoritative name server for example.com.

Example How DNS Works In Real-World

Let's take an example of resolving the domain www.google.com.

Here’s what happens when you visit www.google.com

1. **Query from Browser**
   Your computer queries the DNS resolver to resolve www.google.com.
2. **Root Server Reply**
   The root server refers the DNS resolver to the .com TLD server.
3. **TLD Server Reply**
   The .com TLD server directs the resolver to the authoritative DNS server for google.com.
4. **Authoritative Server Reply**
   The authoritative DNS server for google.com responds with the IP address for www.google.com (for example, 142.250.190.68).
5. **Connection to Google**
   Your browser then uses this IP address (142.250.190.68) to request the website from Google’s servers.

DNS IN CYBERSECURITY

Importance Of DNS in Cybersecurity

1. **Network Gateway**
   DNS serves as the gateway to the internet for almost all online activities. Every time a user or a machine attempts to access an external or internal resource, a DNS lookup is usually involved. This makes it a critical control point for monitoring and securing traffic.

2. **DNS Logging for Threat Intelligence**

   - Monitoring DNS requests can reveal early indicators of compromise (IoCs). For instance, attackers often use domain names for phishing sites or command-and-control (C2) servers.
   - DNS logs can be a treasure trove for security analysts to track anomalous requests, potentially identifying malicious domains that are part of a malware campaign or botnet communication.

3. **DNS for Blocking Malicious Content**
   Many cybersecurity solutions (like firewalls, DNS filtering tools and Secure Web Gateways) use DNS to block access to malicious domains. By preventing the resolution of known bad domains organisations can stop users from accessing phishing sites or downloading malware.

4. **Incident Response**
   DNS logs are invaluable during incident investigations. They can help track the domains and IP addresses that compromised machines have communicated with, enabling faster containment and response.

Most Common Types Of DNS Attacks

1. DNS Spoofing (Cache Poisoning)

Overview: DNS spoofing (also known as DNS cache poisoning) occurs when attackers insert fake DNS records into the DNS resolver's cache. This leads to the resolver returning incorrect IP addresses, sending users to malicious websites instead of the intended legitimate site.

How It Works

- The attacker tricks the DNS resolver into storing a false IP address for a legitimate domain (e.g., bank.comresolves to a malicious IP instead of the real bank's server).
- When users try to visit bank.com, they are redirected to a fraudulent site where attackers may steal login credentials or sensitive information.

Impact

- Users are tricked into visiting malicious websites that look like legitimate ones (e.g., phishing sites).
- Attackers can steal sensitive information (like usernames, passwords or credit card numbers) or distribute malware.

2. DNS Tunnelling

Overview: DNS tunnelling is a method of abusing the DNS protocol to exfiltrate data or establish a communication channel between a victim's machine and an attacker's server. Attackers often use DNS tunnelling to bypass network security controls (e.g., firewalls).

How It Works

- Attackers encode data within DNS queries and responses, allowing them to send or receive information via DNS.
- This traffic often goes unnoticed because DNS traffic is usually not inspected as rigorously as other types of traffic.

Impact

- Sensitive data (e.g., login credentials, intellectual property) can be stolen using DNS tunnelling.
- Attackers may use DNS tunnelling for command-and-control (C2) communication, allowing them to control compromised systems remotely.

3. DNS Hijacking

Overview: In DNS hijacking, attackers redirect DNS queries to malicious servers by compromising either the DNS settings on a victim's device, router or DNS server.

How It Works

- Attackers modify DNS settings on a device or router to point to a rogue DNS server.
- When a user tries to visit a legitimate site, the rogue DNS server redirects the user to a malicious website or an advertisement page.

Types of DNS Hijacking

- **Local DNS Hijacking**
  Attackers alter DNS settings on a victim's device.
- **Router DNS Hijacking**
  Attackers gain control over a router and modify its DNS settings.
- **Man-in-the-Middle DNS Hijacking**
  Attackers intercept DNS queries and send false responses to redirect users to malicious sites.

Impact

- Users are redirected to fraudulent websites, often designed to steal personal information or spread malware.
- Compromised devices or routers may continue to direct all DNS queries to malicious servers, affecting the entire network.

4. Domain Generation Algorithm (DGA) Attacks

Overview: Malware can use Domain Generation Algorithms (DGAs) to generate a large number of pseudo-random domain names. This helps attackers communicate with a Command and Control (C2) server, evading detection by security teams.

How It Works

- Malware generates multiple domain names using an algorithm, such as abc123.com, xyz456.com, etc.
- The malware will attempt to contact one of these generated domains to receive further instructions or send stolen data.
- Security tools find it difficult to block such domains because they are frequently changing.

Impact

- Malware can maintain communication with the attacker even if certain C2 domains are blocked or taken down by defenders.
- Security teams may struggle to detect and block every domain generated by the algorithm, increasing the persistence of the malware.

5. DNS Amplification (DDoS) Attack

Overview: DNS amplification is a type of Distributed Denial of Service (DDoS) attack where attackers send small DNS queries with a spoofed IP address to DNS servers. These servers then respond with large replies to the spoofed address (the victim), overwhelming the target with traffic.

How It Works

- Attackers send a large number of DNS requests with the victim's IP address as the source (IP spoofing).
- The DNS servers respond to these requests with much larger replies (often 10x-100x the size), overwhelming the victim's server or network.
- Attackers can exploit certain DNS record types, such as ANY, which return all available information about a domain, resulting in large response sizes.

Impact

- The victim's network or server is flooded with traffic, causing a denial of service and preventing legitimate users from accessing the network.
- DNS amplification is highly effective because it uses minimal resources from the attacker while creating a massive impact on the victim.

6. Fast Flux DNS Attacks

Overview: Fast flux is a technique used by botnets to rapidly change the IP addresses associated with a domain name. This makes it difficult for security teams to block access to malicious domains.

How It Works

- Attackers register a domain and associate it with a constantly changing set of IP addresses, usually linked to compromised devices in a botnet.
- Each time the domain is queried, the DNS returns a different IP address, distributing the malicious traffic across many different hosts.

Impact

- Fast flux makes it challenging to take down malicious sites or identify the source of attacks because the IP addresses are constantly changing.

- This technique is often used in phishing and malware distribution campaigns, where the malicious domains remain online longer by evading blacklists.

7. NXDOMAIN Attack

Overview: An NXDOMAIN attack is a Denial-of-Service (DoS) attack where an attacker floods a DNS server with queries for non-existent domains, overwhelming the server and causing legitimate DNS requests to be dropped or delayed.

How It Works

- The attacker sends an excessive number of requests for domains that do not exist (e.g., abc123nonexistent.com).
- The DNS resolver repeatedly queries authoritative servers, trying to resolve these non-existent domains.
- This wastes the server's resources and leads to increased response times or complete failure for legitimate queries.

Impact

- Legitimate users experience slow or no response when trying to resolve domains, causing service disruption.
- DNS servers may become overwhelmed and go offline, affecting large sections of the network.

8. Typosquatting and DNS Phishing

Overview: Typosquatting involves registering domain names that are similar to legitimate ones but contain slight misspellings (e.g., goggle.com instead of google.com). Attackers use these domains to trick users into visiting phishing sites or downloading malware.

How It Works

- Attackers register a domain that looks like a legitimate one, often with minor misspellings or swapped letters.
- Users who mistype the legitimate domain name are directed to the attacker’s site, where they might enter sensitive information or unknowingly download malware.

Impact

- Users can be tricked into entering their credentials into fake login pages (phishing), leading to identity theft or compromised accounts.
- Typosquatting domains may also be used to distribute malware or exploit kits.

EXAMPLES AND SIMULATIONS OF DNS THREAT AND INCIDENT RESPONSE

Scenario 1: DNS Spoofing (Cache Poisoning) Attack

Incident Background

Victim
A financial institution's website (e.g., www.bankbruno.com).

Attacker
Cybercriminal looking to steal credentials from bank customers.

Objective
Redirect customers trying to access www.bankbruno.com to a malicious website that looks identical to the real site, tricking them into entering their login credentials.

Attack Execution

1. **Reconnaissance**
    - The attacker identifies a vulnerable DNS resolver (let's call it dns-resolver.com) that provides services for the financial institution.
        - The resolver has weak security configurations, allowing cache poisoning.
2. **Poisoning the Cache**
    - The attacker sends a large number of DNS queries to dns-resolver.com, requesting the IP address of www.bankbruno.com.
    - Simultaneously, the attacker forges DNS responses, tricking the resolver into accepting a fake IP address for www.bankbruno.com (e.g., 192.0.2.10, an attacker-controlled server).
    - The resolver caches this fake record for www.bankbruno.com, meaning any future user queries to dns-resolver.com for www.bankbruno.com will return the malicious IP address.
3. **Redirection**
    - Legitimate users of the bank try to access www.bankbruno.com by entering the URL into their browser.
    - Their DNS query is sent to dns-resolver.com, which now holds the poisoned cache. It returns the malicious IP address (192.0.2.10), directing users to the attacker's fake site.
    - The fake website looks identical to the legitimate bank's website and users unknowingly enter their login credentials, which are captured by the attacker.

Consequences

- User credentials are stolen.
- Customer trust is compromised.
- The financial institution's reputation is at risk.
- The bank is potentially liable for damages caused to its customers.

Incident Response

1. **Detection**
    - **Indicators of Compromise (IoCs)**
        - Incident response teams notice unusual traffic patterns, including high volumes of DNS requests and responses.
        - Customers may report suspicious activities on their accounts or issues accessing www.bankbruno.com.
        - Security monitoring tools detect DNS responses pointing to an unknown or unusual IP address for www.bankbruno.com.
        - Analysis of DNS logs shows that the IP address 192.0.2.10 is being returned instead of the legitimate IP for www.bankbruno.com.
    - **Security Operations Centre (SOC) Action**
        - A Security Information and Event Management (SIEM) system generates an alert about possible DNS poisoning based on traffic patterns and DNS response anomalies.
        - The SOC team immediately starts investigating by correlating logs from DNS servers and firewalls.
2. **Containment**
    - **Isolate the DNS resolver**
        - The team temporarily takes the affected DNS resolver (dns-resolver.com) offline to stop any further DNS queries from returning the malicious IP.
    - **Redirect users**
        - Affected users are informed to manually clear their DNS cache on their local machines and use an alternative DNS resolver (e.g., Google Public DNS) until the issue is resolved.

- A banner or email notification is sent out, alerting users of the phishing attempt.

- **Block malicious IPs**
  - Firewall and intrusion prevention systems (IPS) are configured to block outbound traffic to the attacker-controlled IP (192.0.2.10) to prevent users from reaching the malicious site.

3. **Eradication**

- **Flush the cache**
  - The DNS resolver's cache is cleared to remove the poisoned entry for www.bankbruno.com and restore the legitimate DNS resolution.

- **Patch and harden the DNS resolver**
  - Apply necessary security patches to fix the vulnerability that allowed cache poisoning.
  - Implement security measures such as
    - Configuring the DNS resolver to only accept responses from trusted authoritative DNS servers.
    - Enforcing randomisation of query IDs and source ports to prevent forged responses.

- **Implement DNSSEC**
  - DNS Security Extensions (DNSSEC) are deployed to ensure that all DNS responses are digitally signed and verified, preventing future spoofing attacks.

4. **Recovery**

- **Restore services**
  - Once the DNS resolver is patched and cache is flushed, the resolver is brought back online and DNS resolution resumes as normal.

- **Monitoring and validation**
  - Continuous monitoring of DNS logs and network traffic is conducted to ensure that no further cache poisoning or spoofing activities are happening.
  - Users are asked to validate any suspicious activity on their accounts and those affected are directed to change their passwords immediately.

5. **Lessons Learned**

- **Incident Review Meeting**
    - The incident response team conducts a post-incident review to analyse how the attack occurred, what went well in the response and what could have been done better.
    - A detailed timeline of events is created to document the detection, containment and eradication steps.
- **Enhancing Security Posture**
    - The organisation enforces security best practices for DNS infrastructure, including
        - Using encrypted DNS protocols (DNS-over-HTTPS or DNS-over-TLS) to secure DNS queries.
        - Regular vulnerability scanning and patching of DNS infrastructure.
        - Increased monitoring for suspicious DNS activity, such as unusually high request rates or unknown IP addresses being returned.
- **User Awareness Campaign**
    - Customers are educated on DNS attacks and phishing, encouraging them to verify the authenticity of websites, especially before entering sensitive information.

Key Insights From The Incident

1. **Vulnerabilities Exploited**

- The DNS resolver was vulnerable to cache poisoning due to inadequate security measures.
- Lack of DNSSEC allowed the attacker to forge DNS responses without any validation mechanism.

2. **Impact Analysis**

- The attack primarily impacted customers who were redirected to a phishing website and potentially had their credentials stolen.
- The financial institution's reputation and trust were damaged, possibly leading to financial losses and customer attrition.

3. **Potential Attack Vector Detection**

- If proper DNS monitoring was in place, the cache poisoning attack could have been detected earlier. Security teams should have been alerted when the DNS resolver started returning an unusual IP address for a frequently accessed domain.

Mitigation Strategy Going Forward

- Deploy DNSSEC to prevent spoofing attacks by ensuring DNS records are authenticated and tamper-proof.
- Regular cache flushing policies and updates to DNS resolver configurations to avoid caching of outdated or false records.
- Network segmentation to ensure that DNS servers are isolated from other critical parts of the infrastructure.
- Implementation of rate-limiting to prevent DNS amplification and cache poisoning attempts by limiting the number of queries from suspicious sources.
- User education to recognise phishing sites and avoid entering sensitive information on unknown websites.

Scenario 2: DNS Amplification DDoS Attack

Incident Background

Victim
A large e-commerce website (www.shoprashford.com).

Attacker
A group of hackers aiming to disrupt the website's operations during a major sale event to cause financial losses and reputational damage.

Objective
Overwhelm the target's network with massive amounts of traffic by exploiting DNS servers, causing a service outage.

Attack Execution

1. **Reconnaissance**
    - The attackers identify multiple open DNS resolvers on the internet that are misconfigured to respond to any request from any source.
    - They also discover that these DNS servers respond to ANY queries, which return large amounts of DNS data in response (amplifying the response size significantly).
2. **Attack Setup**
    - The attackers create a botnet (a network of compromised devices) capable of sending DNS requests to the identified open DNS resolvers.
    - Instead of using their real IP addresses, the attackers spoof the IP address of the victim (www.shoprashford.com), making it appear as though the DNS requests are coming from the victim.
3. **Amplification Process**
    - The botnet sends a large number of small DNS queries (such as ANY queries) to the open DNS resolvers, using the victim's IP address as the source.
    - Each DNS server receives the small request and responds with a significantly larger DNS response, sending it back to the victim's network (www.shoprashford.com).
4. **Flooding the Network**
    - The victim's network is overwhelmed by a flood of massive DNS responses from multiple DNS servers across the globe.

- The incoming traffic, amplified by the DNS servers, far exceeds the capacity of the victim's network infrastructure, leading to service degradation and eventual outage.

Consequences

- The e-commerce website (www.shoprashford.com) becomes inaccessible to customers.
- During the critical sale event, the website experiences a major downtime, causing financial losses and damaging the company's reputation.
- Customers are frustrated, leading to complaints on social media and potentially lost future business.

Incident Response

1. **Detection**
    - **Monitoring Tools and Alerts**
        - The Security Operations Centre (SOC) monitors the e-commerce website's network and notices unusually high traffic hitting the website's servers.
        - Traffic analysis tools and intrusion detection systems (IDS) generate alerts about an ongoing DDoS attack targeting DNS infrastructure.
        - Network logs reveal that the traffic is originating from multiple DNS servers and is directed to the victim's IP address.
    - **Indicators of Compromise (IoCs)**
        - Massive amounts of DNS responses, far larger than the typical query size, are observed in the network logs.
        - The source IP addresses in the DNS response packets belong to legitimate DNS servers worldwide, making it clear that the attack is exploiting DNS amplification techniques.
2. **Containment**
    - **Traffic Filtering**
        - The SOC team implements rate limiting on incoming DNS traffic to prevent the flood of DNS responses from overwhelming the network.
        - Firewalls and DDoS protection tools are configured to drop packets coming from DNS servers that are sending unusually large DNS responses (especially for the ANY query type).

- **IP Blacklisting**
    - The team identifies a pattern in the traffic and temporarily blacklists the source IP addresses of the open DNS resolvers being used in the attack.
- **Third-Party Mitigation**
    - The e-commerce website's SOC team contacts its internet service provider (ISP) and cloud-based DDoS protection service (if applicable) to assist in mitigating the attack.
    - DDoS scrubbing services are employed to filter and block malicious traffic before it reaches the victim's network.

3. **Eradication**
    - **DNS Traffic Analysis**
        - The security team works with DNS experts to trace the sources of the amplification. They identify the misconfigured open DNS resolvers used in the attack.
    - **Notify DNS Resolver Owners**
        - The team collaborates with third-party organisations and DNS resolver administrators, notifying them about their DNS servers being exploited as part of the amplification attack.
        - Administrators of open DNS resolvers are encouraged to implement best practices such as rate limiting, disabling recursive queries for external clients and using DNS Response Rate Limiting (RRL) to prevent further abuse.
4. **Recovery**
    - **Restore Service**
        - Once the incoming DNS flood is under control, traffic levels return to normal and the website's services are restored.
        - The team closely monitors the network for any signs of a repeat attack or residual effects.
    - **Reputation Management**
        - The incident response team collaborates with the public relations department to release a statement explaining the service disruption and providing assurance that the issue has been resolved.

- Measures to improve future resiliency are highlighted to maintain customer trust.

5. **Lessons Learned**

- **Incident Review Meeting**
  - A post-incident review is conducted to analyse how the attack was executed and evaluate the response efficiency.
  - The team creates a detailed report with a timeline of events and identifies areas for improvement in the detection, containment and recovery phases.
- **Enhancing DDoS Protection**
  - The organisation strengthens its DDoS protection by implementing additional network filtering and traffic analysis tools.
  - Future DNS-based attacks are mitigated by integrating anycast routing and cloud-based DDoS protection, which helps distribute and absorb large volumes of traffic.
- **Security Awareness**
  - The team conducts internal training to improve awareness of DNS-related threats.
  - The importance of securing DNS infrastructure, even for third-party resolvers, is emphasized to prevent further amplification attacks.

Key Insights From The Incident

1. **Vulnerabilities Exploited**

- The attackers exploited poorly configured open DNS resolvers to perform a reflection and amplification attack. These resolvers were not secured with rate limiting or access controls.
- The attackers used spoofed IP addresses (the victim's IP address) to make it appear that the victim was requesting DNS data, leading to the massive amplification of responses.

2. **Impact Analysis**

- The attack caused significant downtime for the e-commerce website, resulting in loss of sales and potential long-term customer dissatisfaction.
- The organisation faced reputational damage, particularly as the downtime occurred during a high-traffic sales event.

3. **Detection and Response Time**

- The SOC team detected the attack relatively quickly through the use of traffic monitoring tools.
- However, the response was slightly delayed due to the need to identify and mitigate the amplified DNS traffic.

Mitigation Strategy Going Forward

- **Implement DNS Rate Limiting**
    - DNS servers should be configured with DNS Response Rate Limiting (RRL) to prevent amplification attacks. This reduces the number of responses to repeated queries from the same source.
- **Disable Open Resolvers**
    - DNS servers should be configured to reject requests from external sources (unless explicitly required). Only internal or trusted clients should be allowed to use the DNS service.
- **Deploy DDoS Protection**
    - Use a cloud-based DDoS protection service or anycast network routing to distribute DNS traffic and prevent a single point of failure.
- **Monitoring and Alerting**
    - Enhanced monitoring tools should be deployed to detect large volumes of DNS traffic, particularly in the case of amplification attacks. Alerts should be generated when abnormal DNS traffic patterns are detected.
- **Collaborate with DNS Providers**
    - The organisation should establish partnerships with DNS resolver administrators to ensure they follow best practices for DNS configuration and security.

Scenario 3: DNS Tunnelling For Data Exfiltration

Incident Background

Victim
A multinational corporation (GarnachoX) specialising in technology products.

Attacker
A state-sponsored advanced persistent threat (APT) group targeting intellectual property.

Objective
Steal sensitive information (e.g., product designs, financial data) from GarnachoX using DNS tunnelling to avoid detection by traditional security controls (e.g., firewalls and intrusion detection systems).

Attack Execution

1. **Initial Compromise**
    - The attacker uses a phishing email to compromise an employee's device at GarnachoX. The email contains a malicious attachment, which when opened, installs malware (a backdoor) on the victim's system.
2. **Command-and-Control (C2) Setup**
    - The malware is designed to communicate with the attacker's command-and-control (C2) server, but instead of using HTTP or other common protocols, it uses DNS requests.
    - The malware encodes data from the victim's system (such as file contents) into DNS queries. The queries are sent to a domain under the attacker's control, such as attacker-controlled.com.
3. **DNS Tunnelling**
    - The compromised device sends out a DNS query for a subdomain, such as data.exfil.attacker-controlled.com. However, the data is not a legitimate DNS query but rather an encoded snippet of sensitive information (e.g., part of a design document).
    - These DNS queries are routed through GarnachoX's internal DNS server, which forwards them to public DNS resolvers (e.g., Google Public DNS). The public DNS servers query the attacker's domain (attacker-controlled.com), which points to the attacker's C2 server.
    - The attacker's DNS server receives the queries, decodes the information hidden in the subdomain and reconstructs the stolen data.

4. **Ongoing Data Exfiltration**
    - The malware continues to send out DNS queries in small packets over time, each containing different fragments of the sensitive data.
    - The attackers slowly exfiltrate data without raising any alarms because the network security devices treat the DNS queries as normal traffic.

Consequences

- Sensitive product designs, intellectual property and financial information are stolen.
- GarnachoX risks losing its competitive advantage if the stolen information is leaked or sold to competitors.
- The company faces potential legal and regulatory consequences for failing to protect customer and proprietary data.

Incident Response

1. **Detection**
    - **Indicators of Compromise (IoCs)**
        - The network security monitoring system detects an unusually high volume of DNS queries from the compromised host.
        - Security analysts identify that some of the DNS queries are being sent to a domain not typically used by GarnachoX employees (attacker-controlled.com).
        - Network traffic analysis tools detect DNS query packets containing patterns that suggest data encoding rather than legitimate DNS queries (e.g., unusually long subdomains).
    - **Alert Generation**
        - A Security Information and Event Management (SIEM) system generates alerts based on abnormal DNS traffic patterns. The system correlates this with data exfiltration attempts, indicating a possible DNS tunnelling attack.
2. **Containment**
    - **Immediate Actions**
        - The SOC team isolates the compromised host to prevent further communication with the attacker's C2 server and to halt ongoing data exfiltration.

- **Block the Attacker’s Domain**
    - Firewalls are updated to block DNS queries to the attacker's domain (attacker-controlled.com) and any known associated IP addresses.
    - Outbound DNS requests from internal systems are restricted to a limited set of trusted DNS servers.
- **Monitor DNS Activity**
    - DNS logs are reviewed to identify all devices making suspicious DNS queries. This helps locate any additional compromised systems within the network.

3. **Eradication**
    - **Remove the Malware**
        - The SOC team deploys endpoint detection and response (EDR) tools to scan and remove the malware from the infected device.
        - Full malware analysis is conducted to understand the malware’s capabilities and identify all the systems affected by it.
    - **DNS Resolver Hardening**
        - Internal DNS resolvers are hardened by enabling logging, ensuring they only forward queries to trusted DNS servers and blocking unnecessary outbound DNS traffic.
    - **Restrict DNS Queries**
        - Implement DNS filtering tools that block or alert on queries to known malicious or suspicious domains.
        - Deploy DNS tunnelling detection tools that can spot traffic patterns commonly associated with data exfiltration.
4. **Recovery**
    - **Verify Network Integrity**
        - After malware removal, the SOC team performs network scans to ensure no additional backdoors or malware remain.
        - The compromised host is reimaged and restored with updated security patches before being reintroduced into the network.
    - **Change Credentials and Rotate Keys**

- As a precaution, the team resets any credentials and encryption keys that might have been compromised by the data exfiltration.

- **Notify Stakeholders**
  - Internal stakeholders (e.g., legal, compliance, IT leadership) are informed of the incident and potential data loss.
  - If required by regulations (e.g., GDPR), the appropriate authorities and affected customers are notified.

5. **Lessons Learned**

- **Post-Incident Analysis**
  - The SOC team conducts a post-incident review to analyse how the malware bypassed existing defences and how DNS tunnelling was used for data exfiltration.
  - A detailed timeline is created to document key events, including detection, containment and eradication.
- **Implement DNS Tunnelling Detection Tools**
  - Network security is improved by deploying specialised DNS tunnelling detection tools, which look for DNS queries with abnormal patterns (e.g., excessively long domain names or large volumes of queries).
- **User Awareness Training**
  - Employees are trained to recognise phishing emails and other common social engineering tactics to prevent future malware infections.
- **Strengthen Endpoint Security**
  - Endpoint security tools are enhanced to detect and block malware before it can establish a foothold in the network. This includes stricter application whitelisting and regular patching schedules.

Key Insights From The Incident

1. **Vulnerabilities Exploited**

- The attackers leveraged a compromised host to initiate DNS tunnelling, exploiting the fact that DNS traffic is often less scrutinised than other types of network traffic.
- The organisation's DNS resolvers allowed DNS queries to external servers without properly filtering or monitoring outbound traffic.

2. **Impact Analysis**

- The attackers exfiltrated sensitive data over a period of time using small, encoded DNS queries. The stolen data could significantly impact GarnachoX's competitiveness and market standing.
- The use of DNS tunnelling made it difficult to detect the attack using traditional security tools, which focus more on web and email traffic.

3. **Response Effectiveness**

- The SOC team's ability to detect abnormal DNS traffic patterns through advanced monitoring tools allowed for timely detection, though some data was already exfiltrated.
- Containment efforts were successful in cutting off the attacker's communication channel and preventing further data loss.

Mitigation Strategy Going Forward

- **Deploy DNS Security Solutions**
    - Use DNS security solutions to monitor and filter DNS queries, blocking requests to suspicious domains and alerting on abnormal query patterns.
- **Limit DNS Traffic**
    - Restrict outbound DNS traffic to only trusted DNS servers, preventing compromised devices from using external DNS resolvers controlled by attackers.
- **Encrypt DNS Traffic**
    - Use DNS-over-HTTPS (DoH) or DNS-over-TLS (DoT) to encrypt DNS queries, making it harder for attackers to manipulate DNS traffic.
- **Monitor DNS Queries for Anomalies**
    - Implement continuous monitoring of DNS logs to detect abnormal patterns, such as high volumes of DNS queries or unusually long subdomain names, which can indicate DNS tunnelling.

Scenario 4: DNS Cache Poisoning Attack

Incident Background

Victim
A large online retail company (RetailUgarte) that relies heavily on e-commerce for revenue.

Attacker
A cybercriminal group aiming to steal customers' credit card information.

Objective
Poison the DNS cache of the company's DNS servers to redirect customers to a fake website resembling the legitimate RetailUgarte site, thus capturing sensitive data.

Attack Execution

1. **Compromise of a DNS Resolver**
    - The attacker identifies a vulnerable open DNS resolver within RetailUgarte's network, typically one that accepts DNS queries from anyone without validating the responses properly.
    - The attacker starts sending numerous DNS queries for a domain controlled by RetailUgarte (e.g., RetailUgarte.com). While waiting for the legitimate DNS resolver to respond, the attacker floods the DNS server with falsified DNS responses that contain malicious IP addresses.
2. **DNS Cache Poisoning**
    - One of the attacker's falsified DNS responses reaches the server first, corrupting the DNS resolver’s cache. This causes the DNS resolver to believe that RetailUgarte.com resolves to the attacker's malicious IP address instead of the legitimate one.
3. **Redirection of Users**
    - When users attempt to visit RetailUgarte.com, the DNS resolver provides the malicious IP address from its cache, unknowingly redirecting users to a fake website set up by the attackers. The fake website is a near-perfect replica of the legitimate one, complete with login forms and payment portals.
4. **Data Theft**
    - Customers enter sensitive information such as usernames, passwords and credit card numbers on the fake website. All this data is transmitted directly to the attackers.

- Since the attack is happening at the DNS level, users do not notice the redirection, as the URL (RetailUgarte.com) appears to be correct in their browser.

Incident Response

1. **Detection**
    - **Initial Signs**
        - Customers start reporting issues with logging in and completing purchases on RetailUgarte's website.
        - A surge of customer complaints related to suspicious transactions triggers a deeper investigation by RetailUgarte's SOC team.
    - **Indicators of Compromise (IoCs)**
        - Security monitoring tools detect a significant increase in DNS queries related to RetailUgarte.com being resolved to an unfamiliar IP address.
        - Network traffic analysis shows unexpected outbound connections from customers' browsers to an IP address not associated with RetailUgarte.
    - **Alert Generation**
        - A SIEM system generates alerts when anomalous DNS resolutions are detected, particularly when DNS responses point to an IP address outside of RetailUgarte's infrastructure.
2. **Containment**
    - **Immediate Action**
        - The SOC team identifies the corrupted DNS cache entries on the resolver and immediately flushes the DNS cache to remove the malicious records.
    - **Take Down Malicious Site**
        - The incident response team collaborates with law enforcement and domain registrars to take down the malicious website hosted at the attacker's IP address. This reduces the risk of further customers falling victim to the scam.
    - **Isolate Affected DNS Servers**
        - The compromised DNS resolvers are temporarily taken offline to prevent further cache poisoning while the team investigates how the attack occurred.
    - **Update DNS Configuration**

- Security teams disable vulnerable open DNS resolvers and implement security measures such as rate limiting and query validation to prevent future attacks.

3. **Eradication**
   - **DNS Security Hardening**
     - RetailUgarte configures its DNS servers to use DNSSEC (Domain Name System Security Extensions), which ensures that DNS responses are digitally signed and verified, preventing unauthorised changes to DNS records.
   - **Patch DNS Software**
     - The team identifies that outdated DNS resolver software contributed to the vulnerability. All DNS resolvers are patched to the latest version, including security fixes against cache poisoning attacks.
   - **Monitor DNS Traffic**
     - The SOC sets up continuous monitoring of DNS queries and responses to identify and block any suspicious or anomalous DNS traffic in real-time.
4. **Recovery**
   - **Notify Affected Customers**
     - RetailUgarte's incident response team works with the legal and public relations teams to notify affected customers about the data breach. Customers are advised to change their passwords and monitor their bank accounts for fraudulent activity.
   - **Rebuild Trust**
     - A new, secure DNS configuration is deployed and RetailUgarte provides public updates about the incident, ensuring transparency and detailing the steps taken to secure their systems.
   - **Verify DNS Integrity**
     - After making DNS security changes, the SOC team conducts internal and external tests to verify that RetailUgarte's DNS records are correctly resolving to legitimate IP addresses and that no further cache poisoning is taking place.
5. **Lessons Learned**
   - **Post-Incident Review**

    - A detailed analysis of how the DNS cache poisoning occurred is conducted. The team finds that the open resolver allowed the attack and that proper DNS security measures were not in place.

- **Implement DNSSEC**
    - As part of the post-incident improvement, RetailUgarte rolls out DNSSEC across all its DNS servers. This ensures that DNS responses are cryptographically signed, preventing unauthorised modification.
- **Harden Open DNS Resolvers**
    - The team disables open DNS resolvers on RetailUgarte's network and ensures that only authenticated and internal DNS queries are allowed.
- **Customer Awareness Campaign**
    - RetailUgarte launches a customer awareness campaign to inform users about phishing and DNS-related attacks, encouraging vigilance and advising on secure online behaviour.

Key Insights From The Incident

1. **Vulnerabilities Exploited**
    - The attackers took advantage of an open DNS resolver that was susceptible to cache poisoning.
    - The lack of DNSSEC on the DNS servers allowed attackers to inject falsified DNS records without detection.
2. **Impact Analysis**
    - Customers' sensitive data, including credit card information, was stolen and used in fraudulent transactions. This resulted in significant financial losses for both customers and RetailUgarte.
    - RetailUgarte suffered reputational damage due to customers losing trust in the company's ability to secure their online transactions.
3. **Response Effectiveness**
    - The SOC team's swift response, including flushing the DNS cache and taking down the malicious site, mitigated further damage. However, the attack highlighted critical gaps in DNS security that needed addressing.

Mitigation Strategy Going Forward

- **Implement DNSSEC**

    - RetailUgarte enables DNSSEC to ensure that DNS responses are signed and verified, making it difficult for attackers to inject falsified DNS records.

- **Harden DNS Resolvers**
    - All DNS resolvers are reconfigured to reject queries from unauthenticated or external sources.
    - Rate limiting and query validation are implemented to prevent DNS flooding or brute-force cache poisoning attacks.
- **Regular DNS Monitoring**
    - The SOC team deploys real-time DNS traffic monitoring tools that analyse DNS query patterns for signs of attack, including abnormal DNS resolutions and suspicious domain lookups.
- **Customer Notifications**
    - Customers are notified more rapidly in future incidents and RetailUgarte adopts a proactive approach in communicating with customers about potential phishing and fraud risks.
- **Third-Party Audits**
    - Regular third-party audits of RetailUgarte's DNS infrastructure are scheduled to ensure ongoing security and compliance with best practices.

Scenario 5: DNS Amplification Attack

Incident Background

Victim
A medium-sized web hosting company, HojlundHostCo, which provides hosting services for various e-commerce and business websites.

Attacker
A cybercriminal group aiming to disrupt HojlundHostCo's services and make their hosted websites unavailable by overloading their DNS infrastructure.

Objective
Conduct a DNS amplification attack to flood the company's DNS servers with a large amount of traffic, leading to a denial of service (DoS) and taking down the company's hosting services.

Attack Execution

1. **Reconnaissance**
    - The attacker scans the internet for DNS servers that allow open recursive queries (DNS servers that respond to anyone without restrictions). They find that some of HojlundHostCo's DNS servers are configured as open resolvers.
2. **DNS Amplification Begins**
    - The attacker sends multiple small, spoofed DNS queries to HojlundHostCo's DNS servers. These spoofed queries use the IP address of HojlundHostCo's DNS servers as the target and the attacker makes the queries appear as though they come from the victim's IP range.
    - These queries ask for the largest possible DNS responses (such as a list of all DNS records for a domain), which are much larger than the original query in terms of data size.
3. **Amplified Response Flood**
    - The open DNS resolvers at HojlundHostCo receive these small requests and respond with large DNS responses. Since the source IP address is spoofed, these responses are sent to HojlundHostCo's infrastructure instead of the attacker.
    - Due to the amplification factor, HojlundHostCo's servers are now flooded with traffic that is many times larger than the original queries, leading to resource exhaustion on the DNS servers.
4. **Service Disruption**

- The attack overwhelms the DNS servers, making them unable to respond to legitimate DNS requests from HojlundHostCo customers. This results in downtime for all websites hosted by HojlundHostCo, causing financial losses and reputational damage.

Incident Response

1. **Detection**
    - **Initial Signs**
        - HojlundHostCo's network monitoring tools detect a sudden surge in DNS traffic originating from multiple IP addresses, but all the traffic is directed toward the company's DNS infrastructure.
        - Customers begin reporting that they are unable to access their hosted websites. There are also reports of slow website performance and frequent timeouts.
    - **Indicators of Compromise (IoCs)**
        - Logs show a high volume of DNS requests coming from multiple spoofed IP addresses.
        - Analysis of DNS query logs reveals that the queries are for large DNS records (e.g., ANY records), a typical sign of a DNS amplification attack.
    - **Alert Generation**
        - The SIEM system generates alerts based on the unusual spike in DNS traffic, specifically the large size of the DNS responses being generated by HojlundHostCo's open resolvers.
2. **Containment**
    - **Rate Limiting**
        - The incident response team implements rate limiting on the DNS servers to restrict the number of DNS responses sent per second. This helps reduce the load on the servers while they work on mitigating the attack.
    - **Blocking Spoofed IPs**
        - The team configures firewalls to block inbound traffic from the spoofed IP addresses associated with the attack. However, since the IP addresses are frequently changing, this is only a temporary solution.
    - **Take Open Resolvers Offline**

- The affected DNS resolvers that are configured as open resolvers are taken offline temporarily to stop the attack from continuing. This helps in mitigating the amplification effect.

3. **Eradication**

- **Disable Open Recursive Queries**
  - The team reconfigures the DNS resolvers to disable open recursive queries, ensuring that only authorised users and internal clients can send DNS requests. This reduces the risk of future amplification attacks.
- **Deploy DNS Rate Limiting**
  - Rate limiting is permanently enabled on the DNS servers to limit the number of DNS responses sent to a single client in a given time frame.
- **Use DNS Response Rate Limiting (RRL)**
  - DNS Response Rate Limiting (RRL) is implemented to prevent the servers from responding to repeated requests for the same domain, which is a common tactic used in DNS amplification attacks.

4. **Recovery**

- **DNS Server Restart and Reconfiguration**
  - After reconfiguring the DNS servers to prevent open recursion and enabling RRL, the DNS servers are restarted and put back online. The SOC team monitors the servers closely for any signs of ongoing attack traffic.
- **Restore Website Availability**
  - Once the DNS servers are back online, legitimate customers begin to regain access to their websites. The SOC team continues to monitor traffic and performance metrics to ensure stability.
- **Communication with Clients**
  - HojlundHostCo sends out an official communication to its customers explaining the cause of the outage and detailing the steps taken to secure the infrastructure and prevent future attacks.

5. **Lessons Learned**

- **Post-Incident Analysis**
  - The incident response team conducts a detailed analysis of the attack and identifies that the open DNS resolvers were the primary vulnerability that allowed the DNS amplification to succeed.

- **Implement DDoS Mitigation Solutions**
    - A Distributed Denial of Service (DDoS) mitigation service is deployed to help absorb and filter large volumes of traffic in case of future DDoS or DNS amplification attacks.
- **Regular DNS Security Audits**
    - The company schedules regular security audits of its DNS infrastructure to ensure that no other open resolvers are present and that security best practices are followed.

6. **Continuous Monitoring and Improvement**

- **DNS Monitoring Tools**
    - The SOC team deploys advanced DNS traffic monitoring tools to detect anomalous behaviour, such as unusual query patterns, spikes in DNS traffic or large responses being generated frequently.
- **Staff Training**
    - The team conducts internal training on how to identify and respond to DNS amplification attacks, ensuring that all personnel are aware of the signs of such attacks and can act quickly to contain them.

Key Insights From The Incident

1. **Vulnerabilities Exploited**

- The attackers exploited the fact that HojlundHostCo's DNS servers were open resolvers, allowing them to generate large DNS responses to spoofed queries.
- The absence of rate limiting on DNS responses made it easier for the attackers to amplify their attack and overwhelm the DNS infrastructure.

2. **Impact Analysis**

- The downtime caused by the DNS amplification attack led to a loss of revenue for HojlundHostCoand its customers, as their websites were unreachable for several hours.
- HojlundHostCo faced backlash from its clients, who expected better uptime and service reliability from their hosting provider.

3. **Response Effectiveness**

- The incident response team was able to mitigate the attack by quickly taking down the open resolvers and reconfiguring the DNS servers. However, the attack

highlighted the need for proactive DNS security measures, which were implemented after the fact.

Mitigation Strategy Going Forward

- **Disable Open Recursive Queries**
    - HojlundHostCo permanently disables open recursive queries on all its DNS servers, ensuring that only authorised users can make DNS requests.
- **Deploy DNS Response Rate Limiting (RRL)**
    - DNS Response Rate Limiting is deployed across all DNS servers to prevent attackers from using the servers for amplification attacks.
- **DDoS Mitigation Service**
    - A third-party DDoS mitigation service is contracted to help absorb and filter large volumes of malicious traffic in the event of future DDoS attacks.
- **DNS Security Audits**
    - Regular audits of the DNS infrastructure are scheduled to ensure that no new vulnerabilities, such as open resolvers, are introduced.
- **Client Communication**
    - HojlundHostCo implements a more robust communication strategy to inform clients of any ongoing attacks or service disruptions. This helps maintain transparency and customer trust.

Scenario 6: DNS Tunnelling Attack

Incident Background

Victim
A financial institution, ZirkzeeCorp, that processes sensitive financial transactions and customer data.

Attacker
A nation-state-sponsored Advanced Persistent Threat (APT) group looking to exfiltrate sensitive financial data without being detected.

Objective
To perform DNS tunnelling to covertly extract data from the compromised internal network of ZirkzeeCorp using the DNS protocol, which is less likely to be monitored closely.

Attack Execution

1. **Initial Compromise**
    - The attacker gains initial access to ZirkzeeCorp's internal network by exploiting a vulnerable web application.
    - Once inside, the attacker deploys malware on a key server that stores sensitive financial records.
2. **Command and Control Setup**
    - The malware establishes communication with an external server controlled by the attacker using DNS tunnelling.
    - DNS is typically allowed through firewalls, making it an ideal protocol for covert data exfiltration. The attacker uses DNS queries to communicate with the external Command and Control (C2) server.
3. **Data Exfiltration via DNS**
    - The malware on the compromised server encodes sensitive data (e.g., customer account information) into the subdomain of DNS queries.
    - These DNS queries, appearing legitimate, are sent to an external DNS server controlled by the attacker.
    - The attacker's DNS server receives the queries, extracts the embedded data from the subdomains and replies with legitimate DNS responses to avoid suspicion.
4. **Stealthy Operations**

- Because DNS traffic is rarely scrutinised as thoroughly as other protocols, the attacker can exfiltrate data in small chunks over a period of time without triggering any alerts.
- The attacker encodes the exfiltrated data into a format that looks like normal DNS queries, blending into regular network traffic.

Incident Response

1. **Detection**
    - **Initial Signs**
        - The Security Operations Centre (SOC) team notices abnormal spikes in DNS traffic from a specific server. However, the DNS requests seem valid at first glance because they resolve external domains.
        - Anomaly-based intrusion detection systems (IDS) also flag that certain DNS queries contain unusually long subdomains, which is a red flag for DNS tunnelling.
    - **Indicators of Compromise (IoCs)**
        - Long and complex DNS queries being sent to suspicious external domains.
        - DNS traffic with high entropy in subdomains, indicating possible data encoding.
        - Repeated queries to domains that aren't commonly accessed by ZirkzeeCorp's servers.
    - **Alert Generation**
        - The SOC team generates alerts based on anomalous DNS traffic patterns, such as long subdomains and repetitive access to external DNS servers.
2. **Containment**
    - **Block Communication with Malicious DNS Server**
        - The SOC team identifies the external DNS server used by the attacker's malware. The team quickly blocks outgoing DNS traffic to this server by updating the firewall rules.
    - **Isolate the Compromised Server**
        - The team isolates the infected server from the rest of the network to prevent further data exfiltration and to stop the malware from communicating with external servers.

- **Quarantine the Malware**
    - Using Endpoint Detection and Response (EDR) tools, the SOC team quarantines the malware on the compromised server to stop it from executing any further malicious actions.

3. **Eradication**
    - **DNS Tunnelling Detection Tools**
        - The team deploys a DNS tunnelling detection tool across the network to scan for similar anomalous DNS traffic and identify any other compromised machines.
    - **Update DNS Filtering Policies**
        - The DNS filtering rules are updated to block suspicious or abnormal DNS queries, especially those with unusually long subdomains or encoded data in the queries.
    - **Remove Malware and Patches**
        - The malware is removed from the compromised server and patches are applied to the vulnerable web application that allowed the initial compromise.
4. **Recovery**
    - **Reconfigure DNS Logging**
        - DNS logging is enhanced to provide more detailed information about DNS queries and responses. This helps in identifying abnormal DNS activity early in future incidents.
    - **Restore Server to Normal Operation**
        - After ensuring the malware is fully eradicated, the server is carefully restored to normal operations and reintegrated into the network with additional monitoring.
    - **Conduct a Full Network Sweep**
        - A full network sweep is conducted to ensure that no other servers have been compromised and that no other DNS tunnelling activity is taking place.
5. **Lessons Learned**
    - **Post-Incident Analysis**

    - The incident response team conducts a full investigation of how the DNS tunnelling attack occurred and identifies weaknesses in DNS traffic monitoring and logging.

- **Implement More Robust DNS Monitoring**
    - DNS traffic is often overlooked as a potential attack vector and ZirkzeeCorp implements more robust monitoring tools that can detect DNS tunnelling, anomalous DNS requests and DNS exfiltration attempts.
- **Employee Awareness Training**
    - To prevent similar attacks from happening again, ZirkzeeCorp provides employees with security training on the risks of phishing, web application vulnerabilities and DNS-related attacks.

6. **Continuous Monitoring and Improvement**

- **Threat Intelligence Integration**
    - Threat intelligence feeds are integrated into the SOC to detect known DNS tunnelling domains or malicious DNS servers. This provides real-time data to block such threats proactively.
- **Proactive Hunting for DNS Tunnelling**
    - The SOC team begins proactively hunting for signs of DNS tunnelling across the network by analysing historical DNS logs and using advanced detection techniques, such as entropy analysis and machine learning models.

Key Insights From The Incident

1. **Vulnerabilities Exploited**

- The attackers took advantage of the fact that DNS traffic was not closely monitored, allowing them to covertly exfiltrate data via DNS queries.
- A known vulnerability in a web application was exploited to gain initial access to the network.

2. **Impact Analysis**

- Sensitive financial data, including customer records and transaction histories, was exfiltrated over DNS without detection for several weeks.

- While the exfiltration was contained before critical damage occurred, ZirkzeeCorp faced scrutiny for failing to detect such a covert attack in a timely manner.

3. **Response Effectiveness**

- The incident response was effective once the DNS tunnelling activity was detected, but the attack highlighted significant gaps in ZirkzeeCorp's DNS monitoring and detection capabilities, which were addressed post-incident.

Mitigation Strategy Going Forward

- **Implement DNS Security Solutions**
    - Deploy DNS security solutions that can monitor, analyse and filter DNS traffic in real time to detect anomalies such as DNS tunnelling or data exfiltration attempts.
- **Regular Security Audits**
    - Conduct regular security audits of web applications and network infrastructure to ensure that vulnerabilities are patched and that DNS traffic is adequately monitored.
- **DNS Query Logging and Analysis**
    - Ensure that all DNS traffic is logged and analysed for patterns that may indicate malicious activity. Implement tools that can detect high-entropy subdomains, which are often a sign of DNS tunnelling.
- **Integrate AI-Based Detection**
    - Integrate machine learning models to detect DNS tunnelling by analysing traffic for abnormal patterns that traditional detection methods might miss.
- **Restrict External DNS Traffic**
    - Restrict DNS queries to known and trusted external DNS servers and block outgoing DNS queries to potentially malicious or unknown DNS servers.