PENTERA | Pentera Labs™ Research Series

Bypassing “air-gapped” networks via DNS

How adversaries can cross “air-gapped” networks via DNS & what can be done to prevent it.

Uriel Gabay

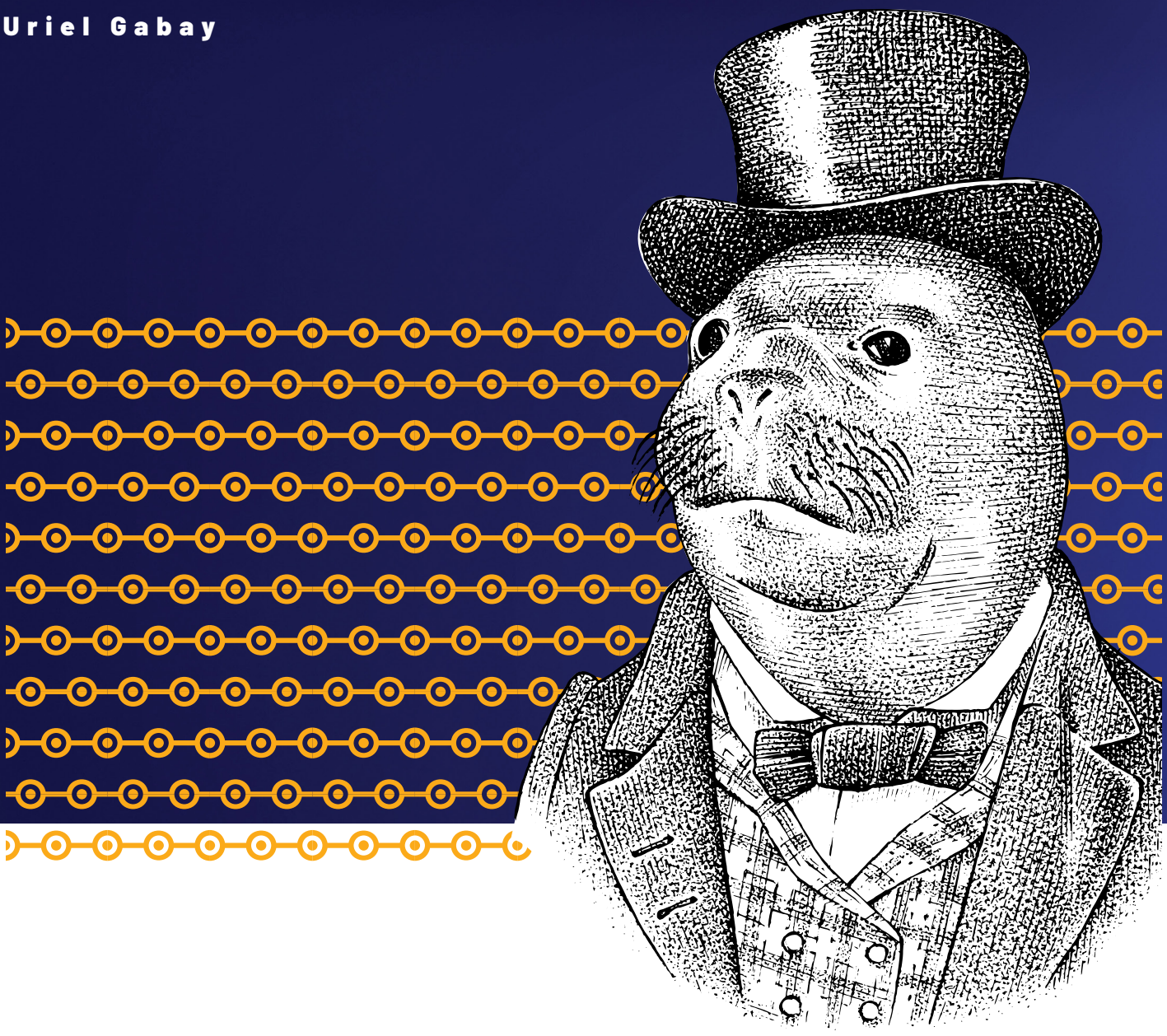


Table of contents

Purpose

Learn how an attacker can communicate over DNS to attack “air-gapped” networks and what steps your security teams can take to mitigate this threat.

Executive summary

Segregating internal networks is a known practice for protecting an organization’s ‘crown jewels’. These high-value assets are critical for maintaining business continuity and are therefore, in most cases, separated from the ‘users’ network, which has access to the Internet.

These internal networks are called "air-gapped" networks. "Air-gapped" networks are only supposed to have internal access without any additional external interface.
However, "air-gapped" networks still require DNS services and this is where things can often be derailed.

Does it apply to my organization?

Yes, if you currently use an "air-gapped" network with DNS access and if your DNS services are connected to another DNS service, which has access to the internet. (See figures 1 & 2 below.)

Who should read this?

Defenders (CISOs) to know how to monitor "air-gapped" networks and researchers/ethical hackers who want to improve their skills & capabilities.

Intro

In order to protect an organization's critical assets from Internet access, IT teams often create isolated or "air-gapped" networks. These networks are often considered inherently untouchable.

While air-gapped networks may not have direct access to the Internet, they still often require DNS services to communicate internally. In a true air-gapped network, the network should not be connected (directly or indirectly) to public DNS on the internet.. However, as seen in the diagrams below, a common misconfiguration chain connects the DNS of an "air-gapped" network to a public DNS. This breaks the isolated nature of a true air-gapped network, allowing an attacker to then potentially abuse DNS for command and control.

Below are two common architectures of how DNS is connected to "air-gapped" networks:

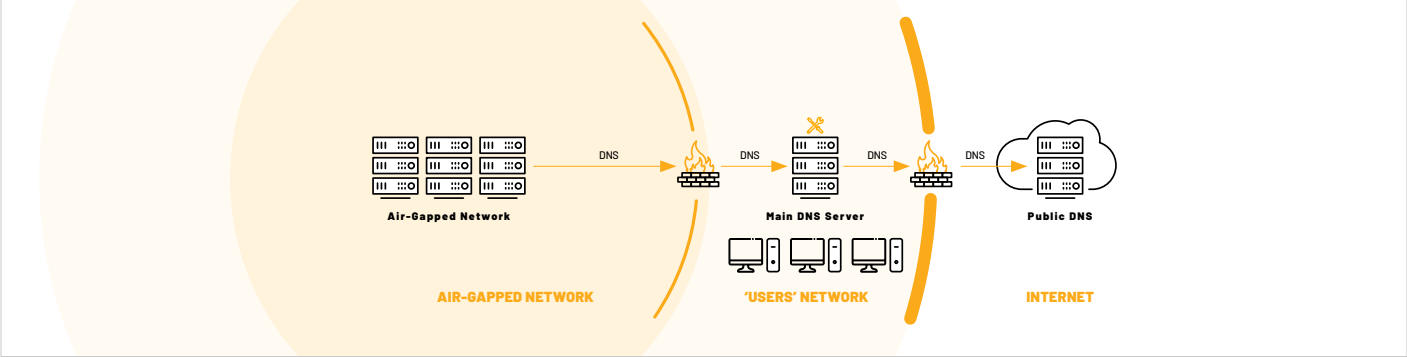


Figure 1 - "Air-gapped" network DNS service connected directly to the main DNS service

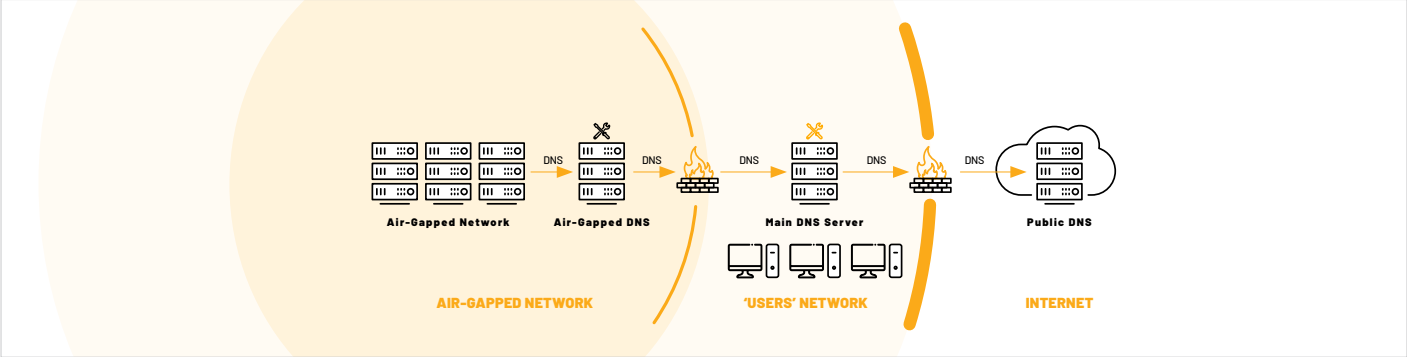


Figure 2 - "Air-gapped" network DNS service connected to an air-gapped DNS

In the second example, many organizations often make the mistake of thinking that by routing communication over an internal DNS server they are preventing a potential breach. However, they are still susceptible as the internal DNS server can still connect with a public DNS server.

This research explores how an attacker can communicate over DNS to access an 'air-gapped' network and what can be done to prevent it.

The DNS threat landscape

DNS attacks, in general, are more common than ever with 88% of organizations reporting some type of DNS attack in 2022 according to the latest IDC Global DNS Threat Report. More specifically, attackers often abuse DNS to establish command and control (C2) to gain unauthorized access to the network. One type of these attacks, DNS Tunneling, which is discussed in this paper, accounted for 28% of DNS attacks in 2022, an increase of just over 16% year over year.

There are many examples of how DNS has been abused as a C2 channel.

In this article, we'll show an example of how an attacker could leverage DNS communication to an "air-gapped" network, but before we do that, let's talk about what exactly is DNS.

DNS in a nutshell

DNS is a protocol used to get a record of a specific name. In this example, we'll focus on TXT and NS DNS record types.

While there are many types of records, two are of particular interest:

1. **TXT** - The text record of a provided name - this can be any text written by the DNS administrator.
2. **NS** - The Name Server record of a provided name - this record basically returns the name servers that hold our provided name.

For example, we can see the AWS Name Servers hold the 'pentera.io' name.

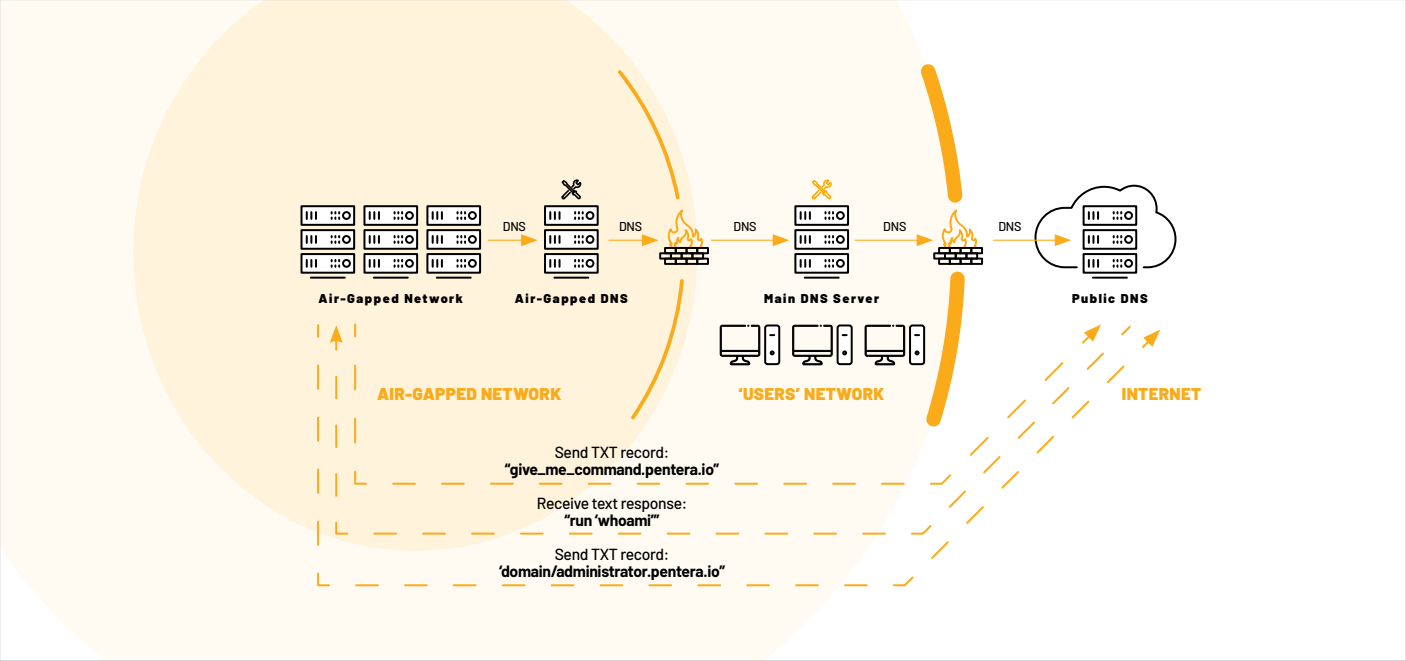
```
c:\>nslookup
Default Server:  UnKnown
Address:  2a00:7c40:0:1:279::3

> set type=NS
> pentera.io
Server:  UnKnown
Address:  2a00:7c40:0:1:279::3

Non-authoritative answer:
pentera.io      nameserver = ns-1300.awsdns-34.org
pentera.io      nameserver = ns-1913.awsdns-47.co.uk
pentera.io      nameserver = ns-268.awsdns-33.com
pentera.io      nameserver = ns-898.awsdns-48.net
```

It's also important to remember that while DNS protocol can run on TCP, it is mostly based on UDP. Each protocol has a different use, but UDP does not have a built-in security mechanism, which will play a role later in this article.

How it works: Send/receive info over DNS

Sending information over DNS can be done by requesting a record and putting the information into the first part of the record's name. For example, to send back to C2 server a 'whoami' command output: **domain\administrator**.pentera.io.

Receiving information over DNS can be done by requesting a TXT record and receiving a text response for this record.



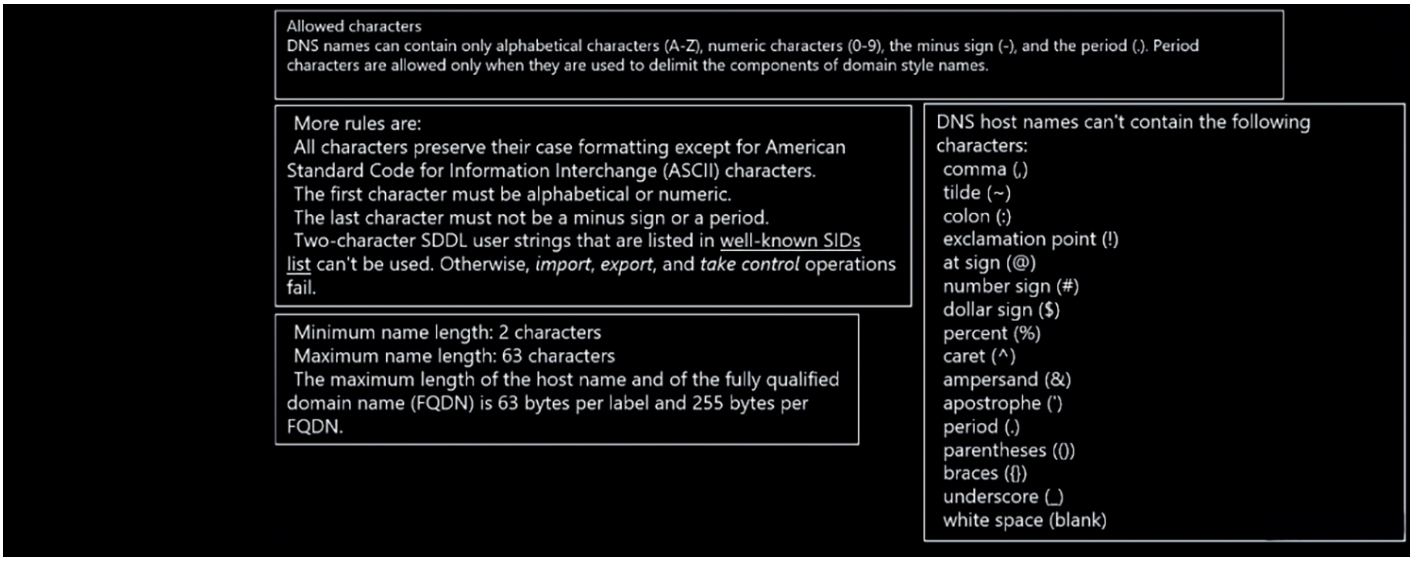
Sending information over DNS can be done by requesting a record and putting the information into the first part of the record's name. For example, to send back to C2 server a 'whoami' command output: domain\administrator.pentera.io.

Receiving information over DNS can be done by requesting a TXT record and receiving a text response for this record.

Challenges attackers face when communicating over DNS

When communicating over DNS, attackers need to take a few things into account:

1. When communicating over DNS, attackers need to take a few things into account:
    A. There isn't a built-in error detection mechanism (as is found in TCP, for comparison).
    B. There is no control over the flow or sequence of data transmission.
2. Compatibility with the DNS protocol
    A. DNS has restrictions on the types of characters it accepts, so not all characters can be sent. Characters that can't be sent are so-called, "bad characters".
    B. There is a limit on the length of characters that can be sent.



How attackers can overcome these challenges

Below are methods that can be used to overcome the challenges outlined in the previous section in communicating over DNS.

How an attacker could overcome UDP challenges:

1. **Error detection** - Prior to sending, compress the payload before sending and immediately decompress after receiving. This can be done with any other encoding (e.g. base64).
2. **No control over the flow**
    A. Notify the server which packet should be buffered (while still sending) and what is expected as the last package (connecting to all buffered packages)
    B. A package should not be sent unless we know that the previous one successfully arrived.

How an attacker could overcome DNS protocol challenges:

1. **Avoiding bad characters** - apply base64 on data sent right before sending
2. **Avoiding length limit** - Data is sliced into pieces and sent one by one

Here's one example of how an attacker could communicate over DNS to an "air-gapped" network:

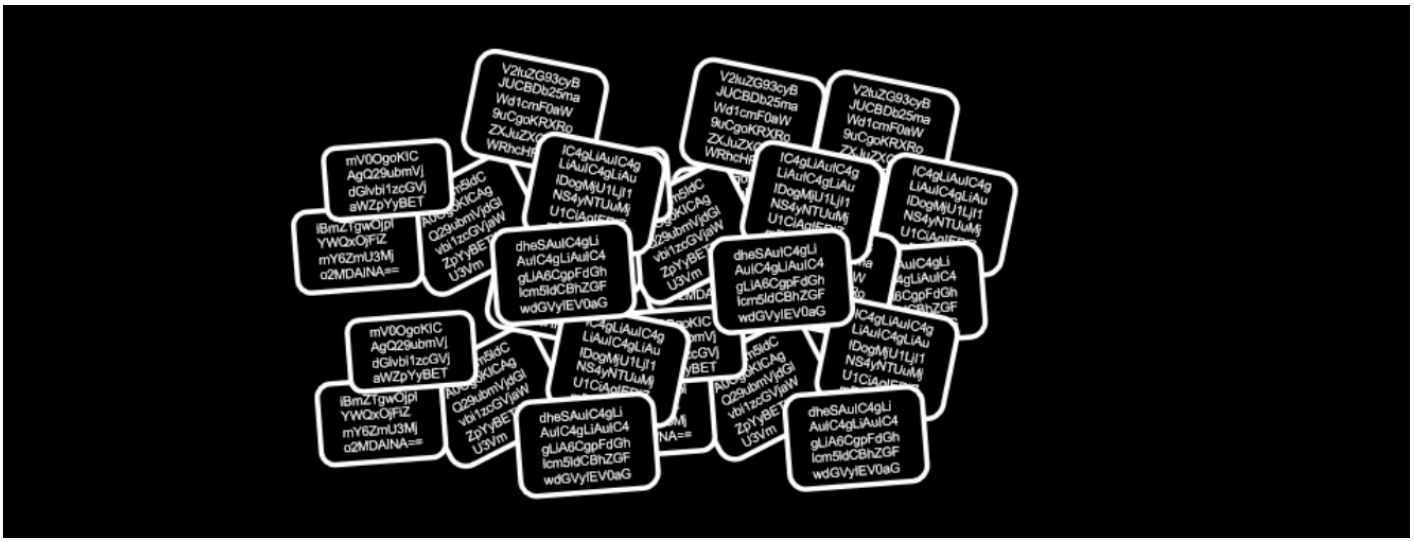
1. Required data sent (output of "ipconfig")

```
Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::241e:7b37:faed:7233%13
   IPv4 Address. . . . . . . . . . . : 192.168.146.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:
```

2. Avoid syntax errors by compressing, obfuscating or encrypting the message sent.

```
V2luZG93cyBJUCBDb25maWd1cmF0aW9uCgoKRXRoZXJuZXQgYW
RhcHRlciBFdGhlcm5ldCA0OgoKICAgQ29ubmVjdGlvbi1zcGVjaWZpYy
BETlMgU3VmZml4ICAuIDoKICAgTGluay1sb2NhbCBJUHY2IEFkZHJlc
3MgLiAuIC4gLiAuIDogZmU4MDo6NzExYjoxMjJhOmUwNTk6OTgyYiU
xNwogICBJUHY0IEFkZHJlc3MuIC4gLiAuIC4gLiAuIC4gOiAxM
C4yMTMuMjEzLjEKICAgU3VibmV0IE1hc2sgLiAuIC4gLiAuIC4
gLiAuIDogMjU1LjI1NS4yNTUuMjU1CiAgIERlZmF1bHQgR2F0ZXdheS
AuIC4gLiAuIC4gLiA6CgpFdGhlcm5ldCBhZGFwdGVyIEV0aG
VybmV0OgoKICAgQ29ubmVjdGlvbi1zcGVjaWZpYyBETlMgU3VmZml4
ICAuIDoKICAgSVB2NiBBZGRyZXNzLiAuIC4gLiAuIC4gLiAuI
DogMmEwMDo3YzQwOmMxNTA6YTM6NmMwMzo2MDZmOjM5OTc6
NTBiYgogICBUZW1wb3JhcnkgSVB2NiBBZGRyZXNzLiAuIC4gLiAuIC4
gOiAyYTAwOjdjNDA6YzE1MDphMzo3MTY2OjJjZTc6NzlmZjpjNjAxCiA
gIExpbmstbG9jYWwgSVB2NiBBZGRyZXNzIC4gLiAuIC4gLiA6IGZlOD
A6OjZjMDM6NjA2ZjozOTk3OjUwYmllINAogICBJUHY0IEFkZHJlc3MuI
C4gLiAuIC4gLiAuIC4gLiAuIC4gOiAxOTIuMTY4LjEuMjYKICAgU3Vibm
V0IE1hc2sgLiAuIC4gLiAuIC4gLiAuIDogMjU1LjI1NS4yNTUuM
AogICBEZWZhdWx0IEdhdGV3YXkgLiAuIC4gLiAuIC4gOiBmZ
TgwOjplYWQxOjFiZmY6ZmU3Mjo2MDAlNA==
```

3. Avoid length limitations by slicing the messages



4. Include required information (e.g. purpose symbol) within the package for the server to know its purpose.
   b_ - Should be buffered
   f_ - Finish sending
   h_ - Heartbeat

5. Send the DNS requestt

Advance C2 over DNS by leveraging DGA (Domain Generation Algorithm)

Since defenders can easily block these requests by blocking access to "*.pentera.io," the next step in this research was to ask how an attacker can get around this.

One thing an attacker could do is generate domain names based on variables that both sides know and expect. While the executable is not necessarily difficult, an attacker or group would need the infrastructure to continue to buy root records as we will see in the example below.

In the below example, we will use a date - December 29, 2021 - as an example.

The attacker may configure their malware to generate a domain based on the date - at the 29/12/2021 to communicate with "**29122021pentera.io**" or with "**29pentera122021.io**" or with "**2912pentera2021.io**".

Below is a simple code example of how to generate DNS based on date:

```
from datetime import date

today = date.today()
domain1 = today.strftime("%d%m%Y") + "pentera.io"
domain2 = today.strftime("%dpentera%m%Y") + "pentera.io"
domain3 = today.strftime("%d%mpentera%Y") + "pentera.io"
print(domain1, domain2, domain3)
```

The result is:

- 29122021pentera.io
- 29pentera122021pentera.io
- 2912pentera2021pentera.io

Because the attacker can constantly send new requests over DNS using a new, known root domain, DGA over DNS will prove challenging to organizations using static methods or even with basic anomaly detection to detect and prevent.

How to mitigate attacks over DNS using C2 on "air-gapped" networks

Today, there are two recommended ways that organizations monitor and protect against attacks using DNS to C2.

- Use a dedicated, offline DNS server for air-gapped networks and monitor any outside access attempts
- DNS filtering - Use a secure DNS service with advanced anomaly DNS analysis such as:
  A. DNS requests with big length
  B. Amount of DNS requests per minutes/hour/day

Conclusion: Even air-gapped segments are at risk

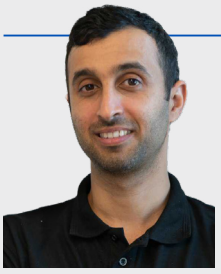
As discussed, an air gap is a commonly used security countermeasure based on the idea of creating an impenetrable barrier between a digital asset and a malicious actor. Organizations worldwide, in every industry and of every size, use this technique to isolate sensitive networks.

Many defenders may be convinced that by air-gapping their most sensitive information, or even relying on physical isolation and offline backups, their critical assets are protected. However, the reality is far from it.

As seen in the example attack provided, as long as the supposedly "air-gapped" segment is connected to the same DNS server as the rest of the network, the risk of being breached over DNS is real.

We hope this research will help defenders & IT teams reconsider their current infrastructure and security measures surrounding air-gapped networks.

About the author



Uriel Gabay is a Senior Security Researcher and exploit developer at Pentera.
Prior to joining Pentera, Uriel served in a classified unit in the IDF, specializing in application security and Red Teaming.

For any questions, feel free to reach out at uriel.gabay@pentera.io

About Pentera

PENTERA

Pentera is the category leader for Automated Security Validation, allowing every organization to test with ease the integrity of all cybersecurity layers, unfolding true, current security exposures at any moment, at any scale. Thousands of security professionals and service providers around the world use Pentera to guide remediation and close security gaps before they are exploited.

For more info, visit: pentera.io

PENTERA | Pentera Labs™ Research Series

Bypassing "air-gapped" networks via DNS