Challenge Description

The domain in question: krampus.csd.lol

We need you to perform a full DNS reconnaissance sweep. The Syndicate thinks they're clever, hiding their infrastructure in plain sight but they're wrong.

Map their infrastructure. Find what they're hiding. Report back before they realize we're onto them.

Type	Purpose
A	IPv4 address
AAAA	IPv6 address
MX	Mail servers
CNAME	Alias
TXT	Freeform text records
SRV	Service records

NS	**Nameservers**

Solutions:

1st Step:
Run this Command in terminal

```
nslookup -type=TXT krampus.csd.lol
```

This command tells your computer to ask a DNS server for the **TXT records** belonging to the domain **krampus.csd.lol**.

🔹 nslookup

This is the program you are using.
It means:

> "Look up DNS information."

🔹 -type=TXT

This tells nslookup *what type of DNS record you want.*

TXT = **Text record**.

TXT records are used for:

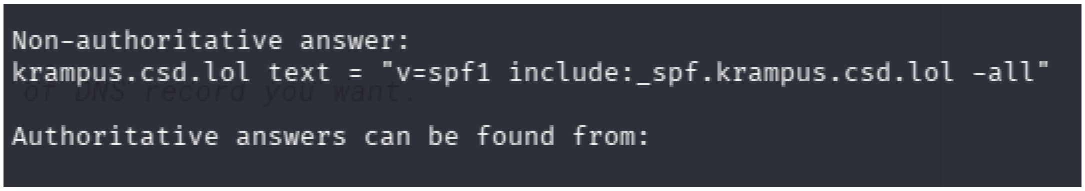
- SPF (email security)
- DKIM keys
- DMARC policies

- Verification strings
- Human-readable notes

🔹 krampus.csd.lol

This is the **domain name** you are querying.

1st command output:



```
Non-authoritative answer:
krampus.csd.lol text = "v=spf1 include:_spf.krampus.csd.lol -all"

Authoritative answers can be found from:
```

When I ran the TXT lookup, my computer sent the DNS query to its local resolver at **XX.XX.XX.XX**, which returned a cached response containing the domain's SPF policy. The TXT record for krampus.csd.lol is v=spf1 include:_spf.krampus.csd.lol -all, which means the domain uses SPF version 1, delegates its allowed mail-sender rules to the _spf.krampus.csd.lol record, and applies a strict -all policy that rejects any email sent from servers not explicitly listed in that included SPF file.

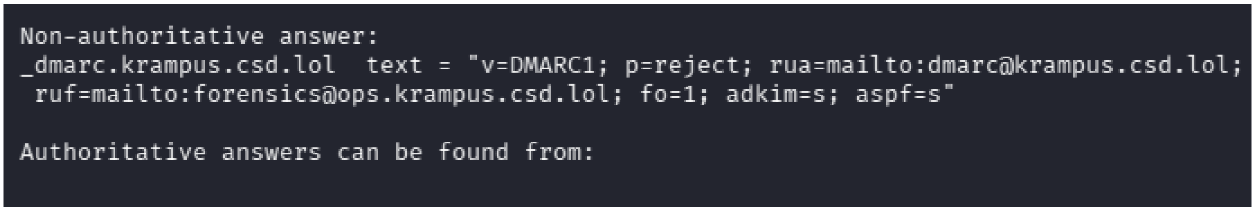
2nd Step:
Run this Command in terminal
nslookup -type=TXT _dmarc.krampus.csd.lol

DMARC (Domain-based Message Authentication, Reporting & Conformance) is always stored as a **TXT record** in DNS.

It contains rules that tell mail servers:

- How to verify emails from the domain
- Where to send reports
- What policy to apply if verification fails

2nd Command Output:



```
Non-authoritative answer:
_dmarc.krampus.csd.lol  text = "v=DMARC1; p=reject; rua=mailto:dmarc@krampus.csd.lol;
 ruf=mailto:forensics@ops.krampus.csd.lol; fo=1; adkim=s; aspf=s"

Authoritative answers can be found from:
```

🧩 1. v=DMARC1

This identifies the version of DMARC.

✔️ Means: “This is a DMARC record.”

🧩 2. p=reject

This is the **policy**.

- **reject** → If an email fails DMARC checks, **reject the message completely**.

✔️ This is the strictest setting.
⚠️ Indicates someone is deliberately protecting the domain or simulating a hardened security posture.

🧩 3. rua=mailto:dmarc@krampus.csd.lol

Aggregate reports should be sent to:

→ **dmarc@krampus.csd.lol**

Aggregate reports = High-level daily logs about all email authentication activity.

🧩 4. ruf=mailto:forensics@ops.krampus.csd.lol

Forensic reports should be sent to:

→ **forensics@ops.krampus.csd.lol**

Forensic reports contain **full copies of failed emails**.

This is extremely sensitive in real-world setups — attackers rarely expose this unless it's deliberate (e.g., a puzzle).

Important:
ops.krampus.csd.lol is a **new subdomain discovered through DMARC**.

🧩 5. fo=1

Failure options.

- fo=1 → Send a forensic report whenever **any** DMARC test fails.

🧩 6. adkim=s

Alignment mode for DKIM = **strict**

Meaning:

- The DKIM signature domain must exactly match krampus.csd.lol.

🧩 7. aspf=s

Alignment mode for SPF = **strict**

Meaning:

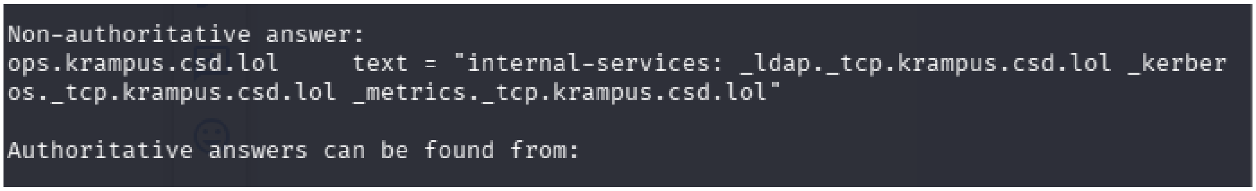
- SPF authenticated domain must exactly match the From: domain.

3rd Step:
Run this Command in terminal

```
nslookup -type=TXT ops.krampus.csd.lol
```

ops = operations
Ask the DNS server to show all TXT records that belong to the domain ops.krampus.csd.lol

3rd Command Output:



```
Non-authoritative answer:
ops.krampus.csd.lol     text = "internal-services: _ldap._tcp.krampus.csd.lol _kerber
os._tcp.krampus.csd.lol _metrics._tcp.krampus.csd.lol"

Authoritative answers can be found from:
```

This TXT record is acting like an **internal service directory**.

It lists **three internal service endpoints** used by the Krampus Syndicate:

1. **_ldap._tcp.krampus.csd.lol**
2. **_kerberos._tcp.krampus.csd.lol**
3. **_metrics._tcp.krampus.csd.lol**

These are not normal public-facing services.
They are core **infrastructure services** that organizations use internally for authentication and data access.

1️⃣ _ldap._tcp.krampus.csd.lol

LDAP = **Lightweight Directory Access Protocol**
Used for:

- User directory
- Credentials
- Group policies
- Authentication backends

🎯 Indicates Krampus Syndicate has an **internal directory server**.

Run this command:

```
nslookup -type=SRV _ldap._tcp.krampus.csd.lol
```

2️⃣ _kerberos._tcp.krampus.csd.lol

Kerberos = **Ticket-based authentication system**

Used for:

- Secure logins
- Service authentication
- Enterprise identity management

🎯 This suggests a **Kerberos realm** exists → very significant.
Kerberos realms often hide hostnames or secret keys.

Run this command:

```
nslookup -type=SRV _ldap._tcp.krampus.csd.lol
```

3️⃣ _metrics._tcp.krampus.csd.lol

Metrics = internal monitoring.

Used for:

- Prometheus-style monitoring
- Health checks
- Logging endpoints
- Internal dashboards

🎯 This is likely a **monitoring server**.

Run this command:
nslookup -type=SRV _metrices._tcp.krampus.csd.lol

4th Step:
We found something in this
nslookup -type=SRV _metrics._tcp.krampus.csd.lol

🧩 1. nslookup

A tool used to query DNS servers.

It asks DNS for information about a domain.

🧩 2. -type=SRV

This tells DNS:

> “Give me the SRV record for this service.”

SRV = Service Record, used to locate servers for specific services such as:

- LDAP
- Kerberos
- SIP
- Metrics
- Minecraft
- Game services
- Internal monitoring services (like Prometheus)

An SRV record contains:

Field	Meaning
Priority	Lower value = preferred
Weight	Load balancing
Port	The port the service runs on

Target	The hostname providing the service

🧩 3. _metrics._tcp.krampus.csd.lol

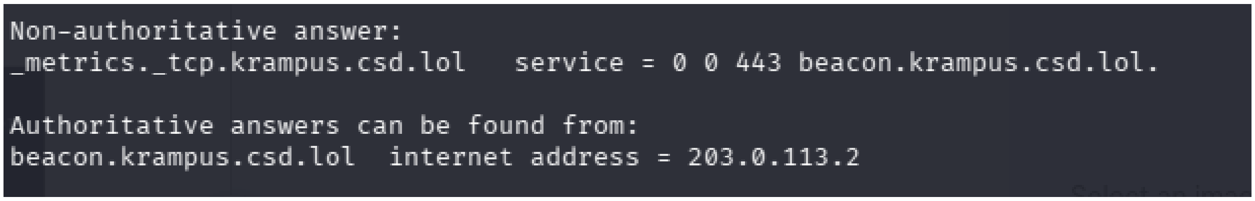
This is the DNS service name you are querying.

It means:

- Service name: metrics
- Transport: TCP
- Domain: krampus.csd.lol

This type of entry usually points to internal monitoring endpoints.

4th command output:



```
Non-authoritative answer:
_metrics._tcp.krampus.csd.lol   service = 0 0 443 beacon.krampus.csd.lol.

Authoritative answers can be found from:
beacon.krampus.csd.lol  internet address = 203.0.113.2
```

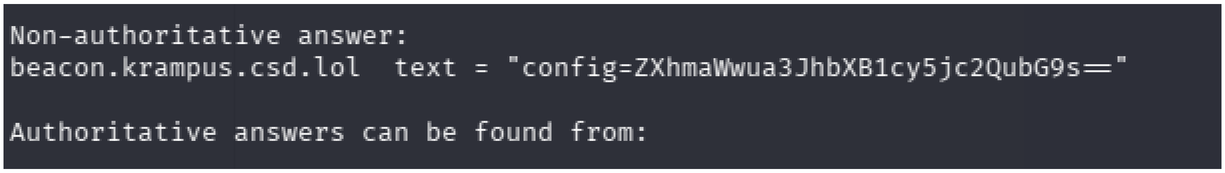
This breaks down as:

Field	Value	Meaning
Priority	0	Highest priority (no other entries compete)
Weight	0	No load balancing
Port	443	HTTPS port → metrics exposed over secure web
Target	beacon.krampus.csd.lol	The server providing the metrics

5th Step:
Run this command
nslookup -type=TXT beacon.krampus.csd.lol

Give me all TXT records associated with the domain beacon.krampus.csd.lol

5th command output:



```
Non-authoritative answer:
beacon.krampus.csd.lol  text = "config=ZXhmaWwua3JhbXB1cy5jc2QubG9s=="

Authoritative answers can be found from:
```

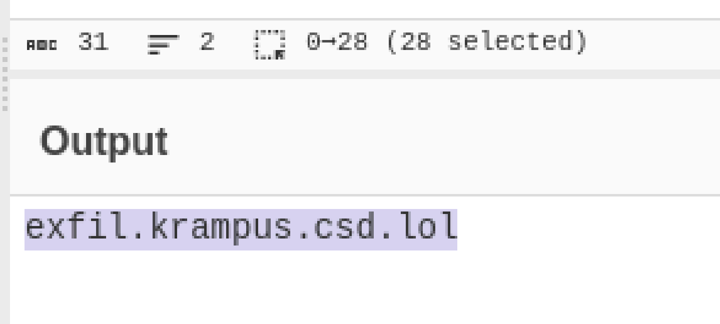
This TXT record contains a parameter:

config = ZXhmaWwua3JhbXB1cy5jc2QubG9s==

This looks **very clearly like Base64**.

TXT records often hide configuration strings this way.

6th Step:
Decode the Base64 string



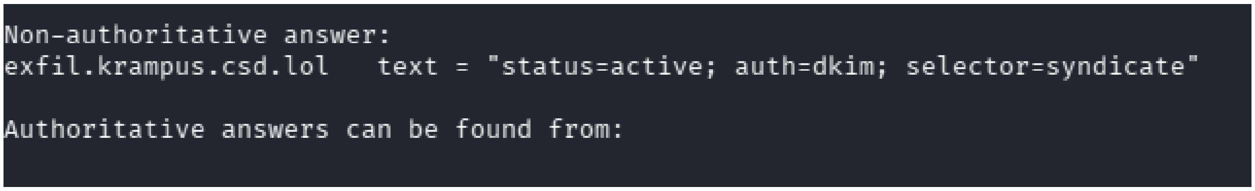
```
exfil.krampus.csd.lol
```

Decoded string : -**exfil.krampus.csd.lol**

7th Step:
Run this command
nslookup -type=TXT exfil.krampus.csd.lol

Show me the TXT configuration attached to the exfil server

7th command output:



```
Non-authoritative answer:
exfil.krampus.csd.lol	text = "status=active; auth=dkim; selector=syndicate"

Authoritative answers can be found from:
```

🧩 1. status=active

This indicates the *exfiltration server* is marked as active in their simulated configuration.

In puzzle or simulation contexts, this usually means:

- This endpoint is "live"
- Further DNS records or clues may exist
- This node is part of the core system

🧩 2. auth=dkim

This means the server uses DKIM authentication.

In normal email systems, DKIM protects outbound mail.
In this puzzle, it suggests:

- This server expects something signed
- DKIM selectors may hide more clues

This points us toward DKIM selector records.

🧩 3. selector=syndicate

This is the DKIM selector name.

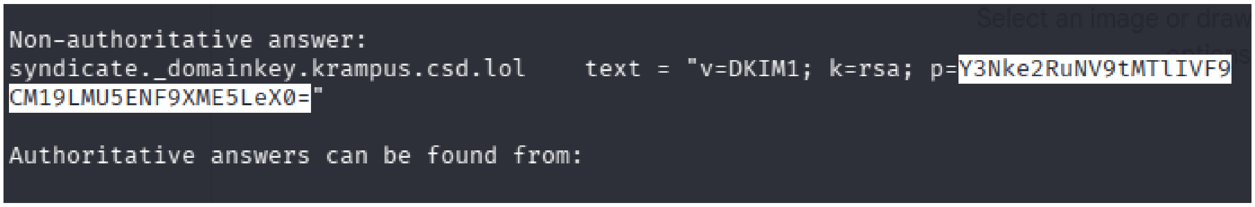
DKIM is stored in DNS under:

<selector>._domainkey.<domain>

So with selector syndicate,

DKIM KEY = **syndicate._domainkey.krampus.csd.lol**

8th Step:
Run this command
nslookup -type=TXT syndicate._domainkey.krampus.csd.lol

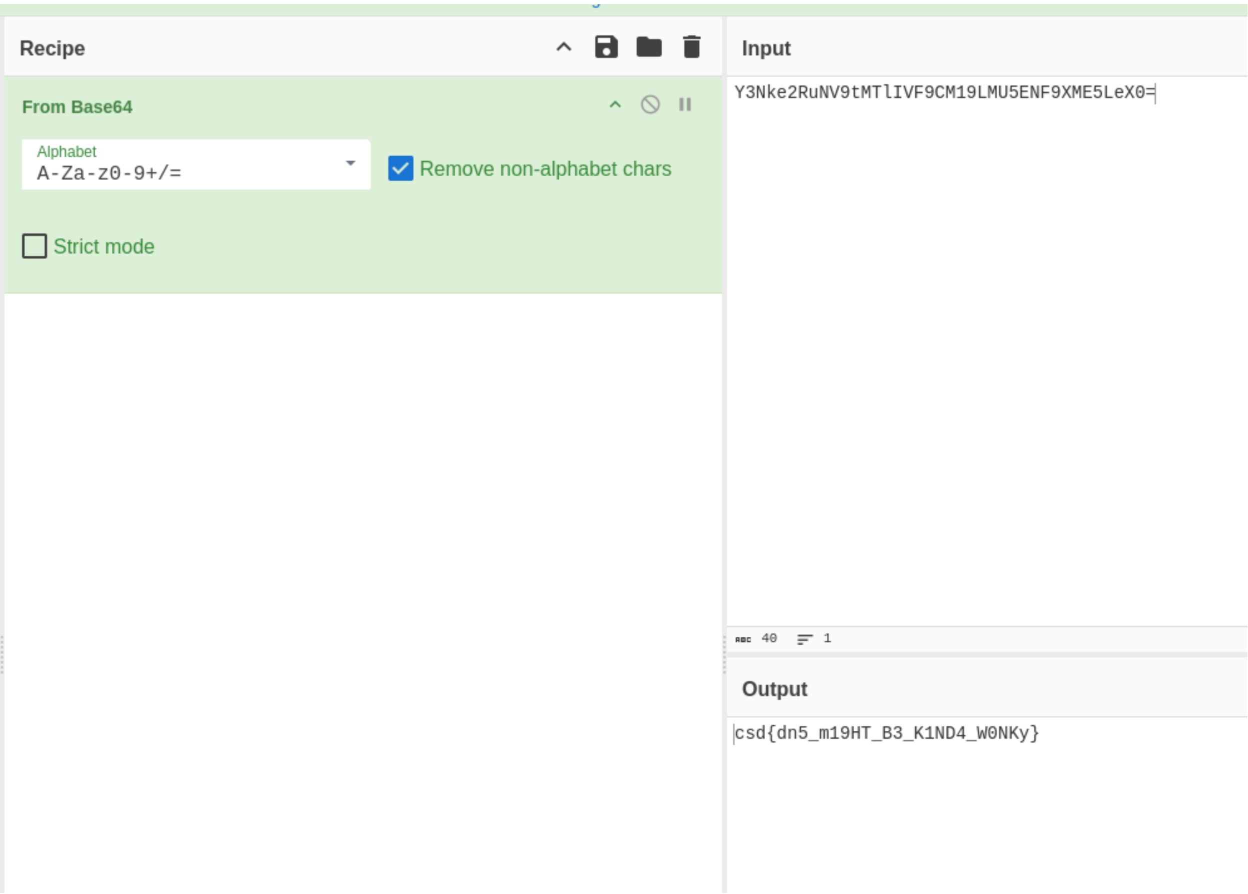


```
Non-authoritative answer:
syndicate._domainkey.krampus.csd.lol    text = "v=DKIM1; k=rsa; p=Y3Nke2RuNV9tMTlIVF9
CM19LMU5ENF9XME5LeX0="

Authoritative answers can be found from:
```

Another BASE64 encoding : **Y3Nke2RuNV9tMTlIVF9CM19LMU5ENF9XME5LeX0=**

Final Step:
Decode this as soon as possible



Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

□ Strict mode

Input

```
Y3Nke2RuNV9tMTlIVF9CM19LMU5ENF9XME5LeX0=
```

Output

```
csd{dn5_m19HT_B3_K1ND4_W0NKy}
```