Réparer ceci

Page de vérification

Voici le document à télécharger télécharger

September 21, 2026

From DanielG's desk. Freelancer member since : February 24, 2012

Subject : Looking for a «virus/security administrator»

Synthesis : We're looking for someone to repair (permanent) the TelePartout.org Tiki email communication system.

We're having difficulty with the DKIM – PFS & system set up of Open Source ecosystem.

Website: https://telepartout.org
CMS: Tiki Wiki CMS Groupware / https://tiki.org
Project: TelePartout.org
Primary objective: Correct and document the email authentication and outbound communication infrastructure once and for all.

1. Background

TelePartout.org is a community-oriented website running on **Tiki Wiki CMS Groupware**.

2. Current working environment

The contractor must be comfortable working in an environment involving:

- Tiki Wiki CMS Groupware;
    - Linux web hosting;
- Gitlab;
    - web-server and application configuration;
- Email deliverability troubleshooting.

There are presently different people responsible for different portions of the infrastructure.

3. Main use case

A registered TelePartout.org user should be able to receive legitimate messages generated by Tiki without admin messages being rejected, incorrectly classified as spam, or failing authentication because of an infrastructure configuration problem.

4. Scope of work

Quickly communicate with the our current email config administrator so that he can put you up to speed about th **audit and fixing he did** before you change anything.

A. Tiki application audit

That info will be provide by Project Manager, i.e. Daniel Gauthier, owner and main administrator of the TelePartout.org web site. You as a subcontractor will work with our email administrator and become the responsible person who communicates with our sysadmin using GitLab (we use GitLab instead of GitHub, yet, the two infrastructures are the same. If you are used to work with GitHub, you will have no issue working on our GitLab admin space. Altough you will able to accomplish specific privileged actions, you will not receive the

permanent root credential. After all, we only profess what and why working in « sh » (shell) is the best process ever to safeguard data integrity

B. Deliverability testing

You will perform real email tests to multiple providers where practical.

5. GitLab workflow

Changes affecting our server configuration or application configuration must follow our GitLab workflow.

As such, if you succeed Phase 1 diagnostic milestone, we will grant you access to our Wiki Groupware so that we all four administrators (You, the email admin, the sys admin and the owner of the website TelePartout) can document what is done and what problems are solved.

You should:

1. identify the configuration requiring modification;
2. prepare the change in Git where appropriate;
3. create a clearly described commit;
4. submit the commit or pull request to our system administrator;
5. explain what the change accomplishes;
6. identify any required manual deployment step;
7. provide a rollback procedure.

The contractor must **not bypass the system administrator by making undocumented production changes**.

6. Security requirements

This project must follow a least-privilege security model.

The following practices are specifically required:

- No request for the permanent root password.
- No sharing of passwords by ordinary email or chat.
- No undocumented modification of production configuration.
- Temporary credentials must be used where privileged access is unavoidable.
- Access must be limited to what is required for the task.
- Temporary credentials and keys must be revoked when the project is completed.
- Existing credentials should not be placed in a Git repository.
- Private DKIM keys must never be committed to a public repository.
- Sensitive information must not appear in Git commit history.
- Changes must be reversible.

If root-level work is required, the sysadmin will provide the command or configuration changes.

7. Acceptance criteria

The contract will be considered successfully completed when:

- Tiki-generated test messages are successfully transmitted;
- SPF passes for legitimate outgoing messages;
- DKIM passes for legitimate outgoing messages;
- DMARC alignment passes;
- DNS records are syntactically valid;
- there are no duplicate SPF records;
- the DKIM public key corresponds to the active signing configuration;
- test messages can be independently verified from their headers;
- all required GitHub commits have been reviewed;
- no root password has been disclosed to the contractor or email administrator;
- the final configuration is documented;
- a rollback procedure has been provided.

9. Contractor qualifications

Please indicate your experience and pricing

Attached is a virus checker's report sample we ran FYI.

http://telepartout.org/

Daniel G

11 / 90

Community Score

URL report for 2026-09-04 14:04:00 UTC.

Reanalyze More

http://telepartout.org/ → https://telepartout.org/

telepartout.org 172.67.209.117

text/html password-input external-resources

Status	Content Type	Last Analysis Date
200	text/html; charset=utf-8	15 days ago

SUMMARY DETECTION DETAILS COMMUNITY

Security vendors' analysis

ADMINUSLabs	Malicious	alphaMountain.ai	Phishing
BitDefender	Phishing	Chong Lua Dao	Malicious
ESET	Phishing	Forcepoint ThreatSeeker	Phishing
G-Data	Phishing	Gridinsoft	Phishing
Lionic	Phishing	Sophos	Phishing
Webroot	Malicious	Abusix	Clean
Acronis	Clean	AILabs (MONITORAPP)	Clean
AlienVault	Clean	Antiy-AVL	Clean
BlockList	Clean	Blueliv	Clean
Certego	Clean	CINS Army	Clean
CRDF	Clean	CTX AI	Clean
Cyble	Clean	CyRadar	Clean
desenmascara.me	Clean	Dr.Web	Clean
EmergingThreats	Clean	Emsisoft	Clean
ESTsecurity	Clean	Fortinet	Clean
Google Safe Browsing	Clean	GreenSnow	Clean
Heimdal Security	Clean	IPsum	Clean
Juniper Networks	Clean	Kaspersky	Clean
LevelBlue	Clean	Malwared	Clean
MalwarePatrol	Clean	OpenPhish	Clean
Phishing Database	Clean	Phishtank	Clean
PREBYTES	Clean	Quick Heal	Clean
Quttera	Clean	Rising	Clean
Sangfor	Clean	Scantitan	Clean
SCUMWARE.org	Clean	Seclookup	Clean
StopForumSpam	Clean	Sucuri SiteCheck	Clean
ThreatHive	Clean	URLhaus	Clean
Viettel Threat Intelligence	Clean	ViriBack	Clean
VX Vault	Clean	Xcitium Verdict Cloud	Clean
Yandex Safebrowsing	Clean	ZeroCERT	Clean
0xSI_f33d	Unrated	AlphaSOC	Unrated
ArcSight Threat Intelligence	Unrated	AutoShun	Unrated
Bfore.Ai PreCrime	Unrated	Bkav	Unrated
ChainPatrol	Unrated	Criminal IP	Unrated
CSIS Security Group	Unrated	Cyan	Unrated
DNS8	Unrated	Ermes	Unrated
Fortra	Unrated	GCP Abuse Intelligence	Unrated
GreyNoise	Unrated	Guardpot	Unrated
Hunt.io Intelligence	Unrated	K7AntiVirus	Unrated
Lumu	Unrated	MalwareURL	Unrated
Mimecast	Unrated	Netcraft	Unrated
PhishFort	Unrated	PrecisionSec	Unrated
SafeToOpen	Unrated	Sansec eComscan	Unrated
Snort IP sample list	Unrated	SOCRadar	Unrated
URLQuery	Unrated	VIPRE	Unrated
ZeroFox	Unrated		

From SF support dept, read this article :

How emails are delivered from Salesforce

Publish Date: Aug 18, 2026
https://help.salesforce.com/s/articleView?id=004633801&type=1

VERP :

More info – UseCase using a visible sender (yogapartout@satoshi.yoga)and recipient (subscriber@gmail.com)

Application originating the message: potentially your Tiki installation at `telepartout.org`

The most important distinction is that an email has **two layers**: the SMTP **envelope**, which the recipient normally doesn't see, and the message **headers**, including the visible `From:` address.

1. What the complete transaction could look like

Imagine Tiki sends this newsletter:

```
Tiki / TelePartout
        |
        | generates message
        v

SMTP conversation
------------------------------------------------

EHLO mail.telepartout.org

MAIL FROM:
<bounce+ABC123@satoshi.yoga>       <-- VERP + SPF

RCPT TO:
<subscriber@gmail.com>

DATA

From: YogaPartout <yogapartout@satoshi.yoga>   <-- DMARC starts here
To: subscriber@gmail.com
Subject: YogaPartout Newsletter

DKIM-Signature:
 v=1;
 d=satoshi.yoga;                   <-- DKIM domain
 s=mail2026;                        <-- DKIM selector
 ...

[message body]

------------------------------------------------
                  |
                  v
              Gmail
```

Gmail then performs several authentication checks.

---

2. VERP — identifying the subscriber

Instead of sending the email with:

```
MAIL FROM:<yogapartout@satoshi.yoga>
```

your mailing system might use:

```
MAIL FROM:<bounce+ABC123@satoshi.yoga>
```

or:

```
MAIL FROM:<bounce+subscriber=gmail.com@satoshi.yoga>
```

That is essentially where **VERP lives**.

The important part is that VERP normally changes the **local portion** of the envelope address:

```
bounce+ABC123
```

while retaining:

```
@satoshi.yoga
```

If Gmail cannot deliver the message, the failure notice goes back to that unique envelope address, allowing Tiki or your mail system to determine which subscriber bounced.

At final delivery, Gmail may expose that SMTP envelope sender as:

```
Return-Path: <bounce+ABC123@satoshi.yoga>
```

The SMTP standard explicitly permits the Return-Path to differ from the human sender and notes that this is useful for mailing lists and error handling. Technically, the originating application supplies the SMTP **MAIL FROM/reverse-path**, while the final delivery server constructs the `Return-Path:` header from it. (RFC Éditeur)

---

3. SPF checks the server sending that VERP address

Now Gmail sees:

```
MAIL FROM:<bounce+ABC123@satoshi.yoga>
```

Gmail asks DNS, in effect:

> Is this SMTP server/IP authorized to send mail for `satoshi.yoga`?

That is **SPF**.

The SPF record could conceptually resemble:

```
satoshi.yoga TXT
"v=spf1 ip4:203.0.113.25 include:some-mail-provider.example -all"
```

If the TelePartout SMTP server is authorized by that record:

```
SPF = PASS
```

If it isn't:

```
SPF = FAIL
```

SPF is based primarily on the SMTP `MAIL FROM` identity—not simply the address your subscriber sees in the `From:` header. (RFC Éditeur)

This is particularly important with VERP

This:

```
bounce+ABC123@satoshi.yoga
```

still uses the domain:

```
satoshi.yoga
```

So the different `ABC123` values **do not require separate SPF records**.

SPF cares about the domain after the `@`.

---

4. DKIM digitally signs the actual message

Before the message leaves your outgoing mail server, it could receive:

```
DKIM-Signature:
 v=1;
 a=rsa-sha256;
 d=satoshi.yoga;
 s=mail2026;
 h=from:to:subject:date;
 bh=...;
 b=...
```

Two pieces are especially important:

```
d=satoshi.yoga
```

means:

> This message is being cryptographically signed on behalf of `satoshi.yoga`.

And:

```
s=mail2026
```

is the **selector**.

Gmail then queries something resembling:

```
mail2026._domainkey.satoshi.yoga
```

in DNS to obtain the public key.

If Gmail can verify the signature:

```
DKIM = PASS
```

DKIM verifies that the signed parts of the message have not been altered and associates that signature with its `d=` signing domain. (RFC Éditeur)

---

5. DMARC brings SPF, DKIM and the visible From together

Now we arrive at the most important part for your configuration.

Your subscriber sees:

```
From: YogaPartout <yogapartout@satoshi.yoga>
```

So the **DMARC domain is:**

```
satoshi.yoga
```

Gmail looks for:

```
_dmarc.satoshi.yoga
```

Suppose it contains something conceptually like:

```
v=DMARC1; p=quarantine;
rua=mailto:dmarc@satoshi.yoga
```

DMARC now asks:

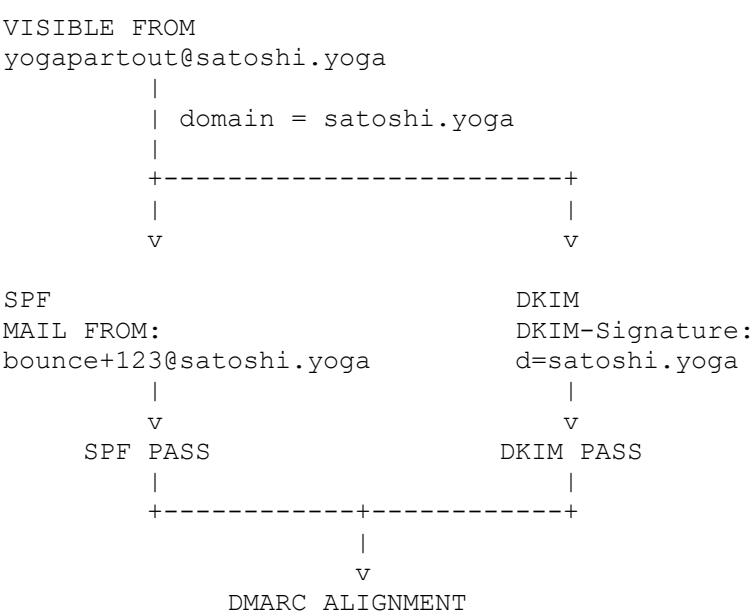
> Did SPF or DKIM authenticate a domain that is aligned with `satoshi.yoga`?

That alignment requirement is the heart of DMARC. (RFC Éditeur)

---

6. Your ideal configuration

With your example, you ideally want something resembling:



```
        |
        v
   DMARC PASS
        |
        v
subscriber@gmail.com
```

This is a very clean arrangement.

---

7. Where TelePartout can accidentally cause trouble

Suppose your Tiki installation produces this instead:

```
From:
yogapartout@satoshi.yoga

MAIL FROM / Return-Path:
bounce+ABC123@telepartout.org

DKIM-Signature:
d=telepartout.org
```

Now look carefully.

Visible From:

```
satoshi.yoga
```

SPF authenticated:

```
telepartout.org
```

DKIM authenticated:

```
telepartout.org
```

Even if both are individually valid:

```
SPF  = PASS
DKIM = PASS
```

you can still get:

```
DMARC = FAIL
```

because neither authenticated identity is aligned with:

```
satoshi.yoga
```

DMARC explicitly distinguishes **authentication** from **alignment**. A DKIM signature can be cryptographically valid while still not satisfying DMARC for the visible From domain. (RFC Éditeur)

That distinction is often responsible for the frustrating situation:

> "SPF says PASS and DKIM says PASS, so why is DMARC failing?"

The answer can be **domain alignment**.

---

8. There is an acceptable mixed-domain configuration

You could conceivably have:

```
From:
yogapartout@satoshi.yoga

Return-Path:
bounce+123@telepartout.org

DKIM:
d=satoshi.yoga
```

Then you might obtain:

```
SPF telepartout.org       = PASS
SPF alignment             = FAIL

DKIM satoshi.yoga         = PASS
DKIM alignment            = PASS

DMARC                     = PASS
```

DMARC needs an appropriately aligned authenticated identifier; it does not necessarily require both SPF and DKIM to align. Google's current Gmail guidance likewise says direct bulk mail must have the `From:` domain aligned with **either the SPF or DKIM organizational domain**, although Google recommends robust authentication generally. (Aide Google)

So this arrangement can work.

However, for a system you administer yourself, I would consider this cleaner:

```
From:        yogapartout@satoshi.yoga
Return-Path: bounce+uniqueID@satoshi.yoga
DKIM d=:     satoshi.yoga
```

That gives you alignment on both authentication paths.

---

9. What Gmail eventually sees

If everything is correctly configured, examining **Show original** in Gmail might produce authentication information conceptually resembling:

```
Return-Path: <bounce+ABC123@satoshi.yoga>

From: YogaPartout <yogapartout@satoshi.yoga>

Authentication-Results:
    spf=pass
        smtp.mailfrom=satoshi.yoga;

    dkim=pass
        header.d=satoshi.yoga;

    dmarc=pass
        header.from=satoshi.yoga;
```

That is the result I would want to see for your example:

SPF: PASS
DKIM: PASS
DMARC: PASS

and, separately:

VERP: correctly routes a bounce back to the individual subscriber record.

Google currently requires SPF or DKIM authentication for all senders to Gmail and imposes stronger SPF + DKIM + DMARC requirements on bulk senders; Google also explicitly checks alignment between the visible `From:` identity and SPF or DKIM for bulk direct mail. (Aide Google)

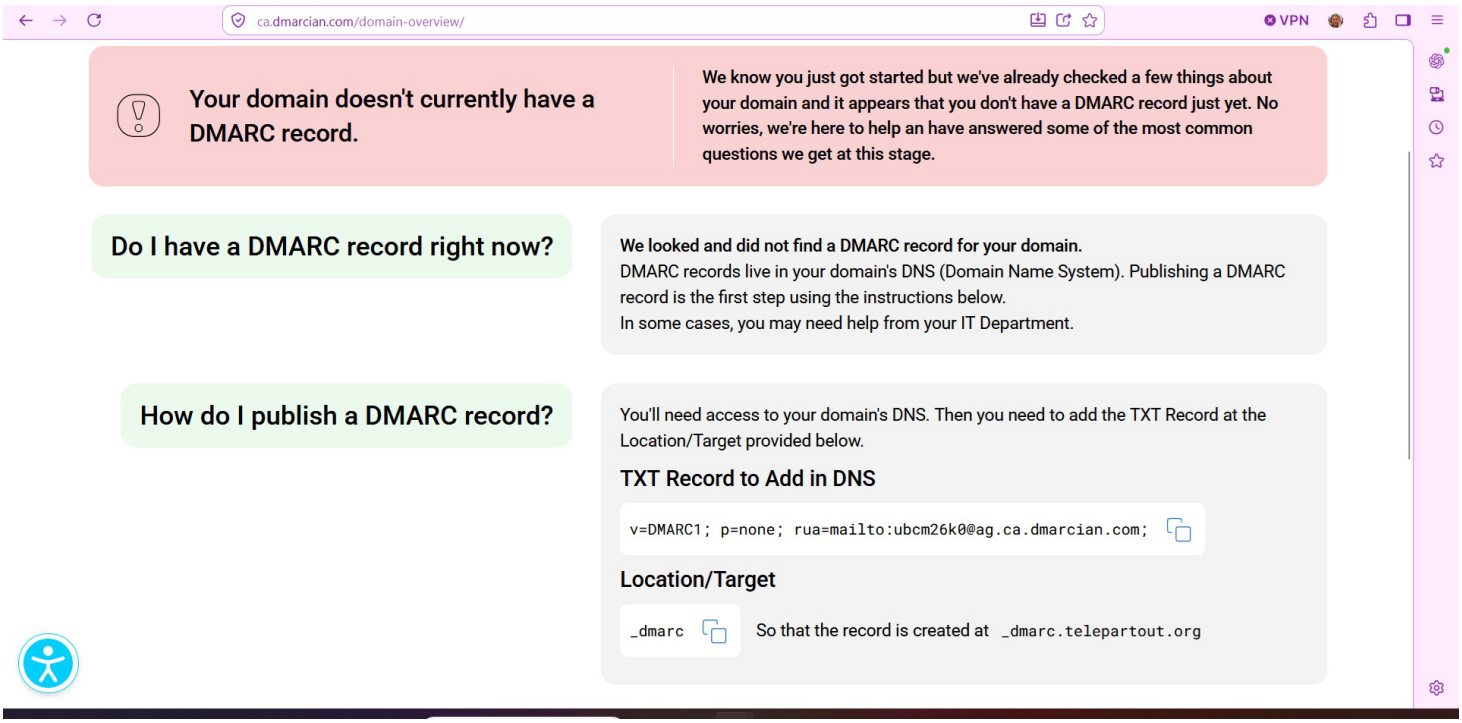
The one sentence to give your email administrator

> **When Tiki sends `yogapartout@satoshi.yoga`, make sure the VERP/MAIL-FROM domain, SPF authorization, DKIM `d=` signing domain, and DMARC alignment are deliberately coordinated; a valid SPF or DKIM result alone does not guarantee DMARC alignment.**

For your environment, that is considerably more important than whether VERP itself is enabled. VERP is not inherently dangerous—the potential nuisance comes from **how its Return-Path domain is integrated with SPF, DKIM, DMARC and Tiki's bounce processor**.

DMARC analysis

According to ca.dmarcian.com telepartout.org does not have a DMARC record. See this screen capture :



C:\Users\yogap\Documents\da\Amis\a_Fournisseurs\Freelancer\Bhawan

u: yogapartout@satoshi.yoga p: *8&7?6%5qQv*8&7?6%5qQ

Instruction to the GitLab mngr :

To add the DMARC script, here are the instruction :

1. Double check if the domain does not, indeed, have a DMARC script.

   The dmarcian.com web tool says :

   > We know you just got started but we've already checked a few things about your domain and it appears that you don't have a DMARC record just yet. No worries, we're here to help an have answered some of the most common questions we get at this stage.

2. Dmarian.com says:

   Do I have a DMARC record right now?

   We looked and did not find a DMARC record for your domain.

   DMARC records live in your domain's DNS (Domain Name System). Publishing a DMARC record is the first step using the instructions below.

   In some cases, you may need help from your IT Department.

   How do I publish a DMARC record? You'll need access to your domain's DNS. Then you need to add the TXT Record at the Location/Target provided below. TXT Record to Add in DNS

   TXT Record to Add in DNS

   v=DMARC1; p=none; rua=mailto:ubcm26k0@ag.ca.dmarcian.com;

   Location/Target

   _dmarc.telepartout.org

   Where is my domain's DNS?

   Your DNS could be hosted by a few services. Popular services are: GoDaddy, Cloudflare, DYN, UltraDNS and Route 53. Your DNS Provider seems to be Network Solutions.

   Where can I find help with this step?

   Network Solutions step by step guide to publishing a record

   https://dmarcian.com/registrar-instructions/2/?_gl=1*1roat9c*_gcl_au*MTA0MzgwOTk2OS4xNzg5NjA1NDg4*_ga*MTMwNzUyNzQ1Ny4xNzg5NjA1NDg5*_ga_FH0W8XVQ0W*czE3ODk5MzA2MTYkbzQkZzEkdDE3ODk5MzEwNTgkajYwJGwwJGgw

If you are not using Network Solutions see the instructions just below Step by step guide to publishing a record

https://dmarcian.com/how-to-publish-a-dmarc-record/?_gl=1*1roat9c*_gcl_au*MTA0MzgwOTk2OS4xNzg5NjA1NDg4*_ga*MTMwNzUyNzQ1Ny4xNzg5NjA1NDg5*_ga_FH0W8XVQ0W*czE3ODk5MzA2MTYkbzQkZzEkdDE3ODk5MzEwNTgkajYwJGwwJGgw

3. dmarcian.com says at this url : https://ca.dmarcian.com/domain-overview/

No data received yet.

Data will appear 24-48 hours after publishing a DMARC record.
In the meantime, you can get an idea of what your account will look like.

Tip: You should allow for at least 2 weeks of DMARC data before making changes. Read our guide on how to safely advance your domain's DMARC policy.

4. The following is what your account will look like when data is received.

