

# IN PROGRESS

## Prerequisites

The winbind package must be installed and working.

## Changed files

- clients.conf
- proxy.conf
- sites-available/govroam
- sites-available/govroam-inner-tunnel
- mods-available/eap
- mods-available/govroam\_logs

Delete any other links in the sites-enabled directory ('status' can be left/added if you're allowing status checks). Attempting to run 'govroam' and 'default' will likely result in problems stating the RADIUS server.

### clients.conf:

```
# Configure a Network Access Server (e.g. wireless controller) to accept
traffic from.

client NAS {
    secret = something
    ipaddr = 10.10.20.1
}

# Configure the JISC NRPS as a client as it will be sending request from
your people abroad.

client roaming0 {
    secret = something
    ipaddr = 192.168.0.1
    operator = "NRPS"
}

# Configure your local IdP as a client. (Omit for Visited Only ORPS)
client localidp1 {
    secret = something
    ipaddr = 10.10.10.31
    operator = "localidp1"
}
```

proxy.conf:

```
# Blackhole (REJECT) where the realm is missing.

realm NULL {
}

# Realms that don't match any other listed send to the pool of govroam
servers
realm "~^[^@\. ]([a-zA-Z0-9-]+\.)+[a-zA-Z]{2,6}$" {
    auth_pool = govroam
    nostrip
}

# Pool of govroam servers
home_server_pool govroam {
    home_server = roaming0
    type = client-port-balance
}

#Govroam server configuration
home_server roaming0 {
    ipaddr = roaming0.govroam.uk
    port = 1812
    type = auth
    secret = something
    status_check = status-server # Checks status of govroam server
    operator = "NRPS"
}

# Handle requests for the realm 'localnet'. (Omit for Visited Only ORPS)
realm localnet {
    nostrip
    auth_pool = ad_auth
}

server_pool ad_auth {
    type = client-port-balance
    home_server = localidp1
}

home_server localidp1 {
    status_check = status-server
    ipaddr = 10.10.10.31
    secret = something
    port = 1812
    type = auth
}
```

```
operator = "localidp1"

}
```

### sites-available/govroam:

```
server govroam {
    # Listen on the default port on all IP addresses
    listen {
        type = auth
        ipaddr = *
    }

    authorize {
        preprocess
        update request {
            Operator-Name = %your.domain # Adds the Operator
Name attribute to the request, if it doesn't already exist.
        }
        auth_log
        suffix # Identifies the realm
        files
        cui
        mschap # used for plain/non-eap ntlm_auth testing
        eap {
            ok = return
        }
    }

}

authenticate {
    ntlm_auth
    Auth-Type MS-CHAP {
        mschap
    }
    eap
}

preacct {
    preprocess
    suffix
}

accounting {
    detail
}

post-auth {
    # Lots of logging
    reply_log
}
```

```
        # Only send F-TICKS to Jisc when proxying between sites.
        if ( "%{home_server:operator}" != "NRPS" && "%{client:operator}" !=
"NRPS") {
            f_ticks
        }
        govroam_log
        cui
        Post-Auth-Type REJECT {
            attr_filter.access_reject
            reply_log
        }
    }

    pre-proxy {
        pre_proxy_log
        cui
        if("%{Packet-Type}" != "Accounting-Request") {
            attr_filter.pre-proxy
        }
    }

    post-proxy {
        post_proxy_log
        attr_filter.post-proxy
    }
}
```

And then create a symlink from sites-enabled/govroam to sites-available/govroam.

### **sites-available/govroam-inner-tunnel**

```
server inner-tunnel {

    authorize {
        preprocess
        auth_log
        suffix
        update control {
            Proxy-To-Realm := LOCAL
        }
        eap {
            ok = return
        }
        files
        pap
        mschap
    }
    authenticate {
```

```
        ntlm_auth # Just for testing plain/non-EAP auth
        files
        Auth-Type PAP {
            pap
        }
        Auth-Type MS-CHAP {
            mschap
        }
        eap
    }

    post-auth {
        cui
        reply_log
        govroam_log
        Post-Auth-Type REJECT {
            reply_log
            govroam_log
        }
    }
}
```

And then create a symlink from sites-enabled/govroam-inner-tunnel to sites-available/govroam-inner-tunnel.

### mods-available/eap

```
eap {
    default_eap_type = mschapv2
    timer_expire     = 60
    ignore_unknown_eap_types = no
    cisco_accounting_username_bug = no
    max_sessions = ${max_requests}

    md5 {
    }

    tls-config tls-common {
        # Generate and install a server cert and a CA ROOT.
        private_key_password = whatever
        private_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
        certificate_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
        ca_file = /etc/ssl/certs/ca-certificates.crt
        dh_file = ${certdir}/dh
        ca_path = ${cadir}
        cipher_list = "DEFAULT"
        cipher_server_preference = no
        ecdh_curve = "prime256v1"

        cache {
```

```
        enable = no
        lifetime = 24 # hours
    }

    verify {
    }

    ocsdp {
        enable = no
        override_cert_url = yes
        url = "http://127.0.0.1/ocsdp/"
    }
}

tls {
    tls = tls-common
}

# This is the config for PEAP/MSCHAPv2 i.e. username/password.
peap {
    tls = tls-common
    default_eap_type = mschapv2
    copy_request_to_tunnel = no
    use_tunneled_reply = no
    virtual_server = "inner-tunnel" # Make sure that this points to the
govroam inner tunnel
}

mschapv2 {
}

}
```

And then create a symlink from mods-enabled/eap to mods-available/eap, if one doesn't already exist.

### mods-available/govroam\_logs:

```
# F-TICKS
linelog f_ticks {
    filename = syslog
    format = ""
    reference = "f_ticks.%%{%reply:Packet-Type}:-format}"
    f_ticks {
        Access-Accept ="F-
TICKS/govroam/1.0#REALM=%{Realm}#VISICOUNTRY=GB#VISINST=%{Operator-
Name}#CSI=%{Calling-Station-Id}#RESULT=OK#FEDID=XX#" # Replace XX with your
supplied ID
```

```

    }

    linelog govroam_log {
        filename = syslog
        format = ""
        reference = "govroam_log.%%{reply:Packet-Type}: -format}"
        govroam_log {
            Access-Accept = "govroam-auth#ORG=%{request:Realm}#USER=%{User-Name}#CSI=%%{Calling-Station-Id}:-Unknown Caller Id}#NAS=%%{Called-Station-Id}:-Unknown Access Point}#CUI=%%{reply:Chargeable-User-Identity}:-Unknown}#MSG=%%{EAP-Message}:-No EAP Message}#RESULT=OK#"
            Access-Reject = "govroam-auth#ORG=%{request:Realm}#USER=%{User-Name}#CSI=%%{Calling-Station-Id}:-Unknown Caller Id}#NAS=%%{Called-Station-Id}:-Unknown Access Point}#CUI=%%{reply:Chargeable-User-Identity}:-Unknown}#MSG=%%{reply:Reply-Message}:-No Failure Reason}#RESULT=FAIL#"
        }
    }
}

```

And then create a symlink from mods-enabled/govroam\_logs to mods-available/govroam\_logs.

Use the **details.log** file in mods-available to configure how the local logs are formatted and stored. The format below stores the logs by date and time making it easier to use logrotate or similar to archive off older logs.

```

detail auth_log {
    detailfile = ${radacctdir}/%Y%m%d/request-detail-%H:00
    header = "%{Packet-Src-IP-Address} - %t"
    detailperm = 0600
    suppress {
        User-Password
    }
}

detail reply_log {
    detailfile = ${radacctdir}/%Y%m%d/reply-detail-%H:00
    header = "%{Packet-Src-IP-Address} - %t"
    detailperm = 0600
}

detail pre_proxy_log {
    detailfile = ${radacctdir}/%Y%m%d/pre-proxy-detail-%H:00
    header = "%{Packet-Src-IP-Address} - %t"
    log_packet_header = yes
    detailperm = 0600
    suppress {
        User-Password
    }
}

detail post_proxy_log {

```

```
detailfile = ${radacctdir}/%Y%m%d/post-proxy-detail-%H:00
header = "%{Packet-Src-IP-Address} - %t"
detailperm = 0600
}
```

Once configured you can test using [eapol\\_test](#)

From:  
<https://wiki.govroam.uk/dokuwiki/> - Govroam

Permanent link:  
[https://wiki.govroam.uk/dokuwiki/doku.php?id=siteadmin:basic\\_freeradius\\_orps\\_and\\_idp\\_configuration](https://wiki.govroam.uk/dokuwiki/doku.php?id=siteadmin:basic_freeradius_orps_and_idp_configuration)

Last update: **2023/04/05 11:58**

