Individual Submission
Request for Comments: DRAFT_ID_01.0
Category: Experimental

A. Lewis
Z. Seguin
J. Kemper
T. Pozar
ARDC TAC
August 2023

An Automated WireGuard Virtual Private Network (VPN) to provide enhancements to and solve problems of 44net

Status of this Memo

This document specifies experimental Practices for the Amateur Radio and Internet Communities and requests discussion and suggestions for improvements.

Information about current status of this document, any errata, and how to provide feedback on it may be obtained at:
https://nextcloud.ardc.net/index.php/apps/files/files?dir=/TAC/POP

Distribution of this memo is unlimited.

1. Abstract:

This document specifies an open source architectural map for development of an Amateur Radio community resource that enables interconnectivity of radio nodes and computing device endpoints using IP network technology. The topology defined herein is not limited to amateur radio specific usage, but can be extended to solving similar problems in scientific research including Internet Measurement, self-guided experimentation, education, and similar uses where a simple method of connecting to remotely located devices is needed.

A proposal is presented in this document for a standardized approach for automating the configuration and management of Virtual Private Network(VPN) tunnels using the WireGuard Protocol. The main objectives of this VPN infrastructure are to provide publicly routable IPv4 or IPv6 addresses to solve three primary use-cases. These are:

1. A user wants to allow remote machines to be able to initiate a connection to their local devices (Make publicly accessible through firewalls and Network Address Translation (NAT)).
2. A user wants to access many other isolated devices such as “Ham Networks” simultaneously without polluting each of these networks with Internet traffic.
3. User groups (clubs), want to provide their own Point-of-Presence (PoP) infrastructure but don’t have a machine in a Datacenter to pass data upstream.

As a case study, subnets within the IPv4 address space of Amateur Radio Digital Communications (ARDC) from 44/9, and/or 44.128/10 (known as

44Net)are used to create PoP VPN servers with a shared database for the purpose of developing and refining the technologies proposed within this document.

The aim is to provide a seamless and efficient automated method for setting up and maintaining reliable, secure connections between multiple networks and device endpoints. This document outlines the key components, features, design/implementation considerations, procedures/specifications involved in an automated VPN deployment using WireGuard, highlighting the benefits, and operational considerations for implementing such a system.

2. Introduction:

The rapid growth of interconnected networks has necessitated the development of efficient and secure Virtual Private Network(VPN) solutions. This RFC outlines an automated VPN architecture approach using WireGuard protocol, a modern lightweight VPN protocol known for its simplicity and as a high-performance alternative to traditional VPN solutions. The goal of this system is to simplify the process of setting up and managing VPN connections, allowing communication between networked devices over a publicly routable IPv4 or IPv6; and MAY use any available RFC1174 (NON-RFC1918) IP space such as prefix blocks from ARDC 44net allocations. VPNs are widely used to establish secure communication tunnels over untrusted networks. Automating the configuration and management of VPN connections using WireGuard MUST simplify the deployment process, provide quick access of 44net resources, and ease onboarding of users to allow and enhance exploration of digital communications.

3. Definitions/Terminology:

The following terms are used throughout this document.

- **44Net:** Refers specifically to the ARDC managed address space between 44.0.0.0 and 44.191.255.255.
- **PoP Project:** An open-sourced linux downloadable Point of Presence (PoP) software package used to create web based Dashboard, and tunnel endpoint servers (PoP's) to allow users to seamlessly connect to 44net.
- **VPN:** Virtual Private network created using WireGuard's UDP protocol whereby the VPN server is resident on the PoP and the VPN client is present on the device endpoint.
- **Dashboard:** Web based interface used to update database used in orchestration of controller servers in altering VPN tunnel's and PoP Node Virtual Machine(VM's) interconnection. Dashboard SHOULD have both an end-user facing and admin facing interfaces.
- **Controller:** Backend software residing on CORE & MICRO PoP's responsible for automation and tunnel creation. The controller provides the VPN endpoint for tunnel connectivity.
- **CORE PoP(s)** – ARDC operated VPN controller server. CORE PoP's are a top of fabric (ToF) infrastructure. CORE PoP interactions with MICRO PoP's are similar to the Spine/Leaf Clos configurations used in data-centers.
  1 or more regional upstream (northbound) VPN CORE PoP controller servers are used as either northbound transit providers for downstream (southbound) "MICRO" PoP's, and MAY be used for

direct user VPN connections.

- **MICRO PoP(s)** – Downstream (southbound) VPN controller servers able to be autonomous or connected to a northbound CORE PoP. MICRO PoP MAY share a common DB with CORE PoP's or create their own. Like CORE PoP's, MICRO PoP MAY also provide end user tunnels and as a northbound connection to southbound connected MICRO PoPs.

4. **Motivation:** Motivating reasons for creating VPN PoP network and a few problems it aims to solve are:

   a. **Easy Onramp and barriers to entry:**
   To democratize technology for those outside of the network engineering profession who want to connect to remote devices that need to be discoverable through firewalls and Carrier Grade Network Address Translation (CGNAT). VPN PoP's address this with both automation and an open source blueprint that reduces barriers to entry which may arise from knowledge gaps or the prohibitive costs associated with hosting VPN servers for experimentation purposes.

   b. **Public IP Provisioning:**
   Automated provisioning of Public IPs that are advertised to autonomous systems using Border Gateway Protocol (BGP) will enable remote devices to initiate access to VPN user's internal resources, such as repeater control, or Raspberry Pi node services, while maintaining a logical tunnel with end-to-end encryption to a VPN PoP (point of presence) residing in data centers. The Core PoP's (ARDC operated VPN tunnel controllers) MUST have a pool of public IP addresses that SHOULD be dynamically assigned to VPN clients as a dedicated (also known as fixed/static IP) endpoint for user devices to be reachable, externally, on the internet.

   c. **Isolated Network Integration:** The VPN solution MAY facilitate the connection of isolated/island amateur radio networks, allowing secure communication between different network segments. Each PoP MUST not pollute an isolated network with Internet traffic nor allow IP address transversal not originating on isolated networks. For example: if on AREDN, only an RFC 1918 address such as 10.2.1.23/32 would be allowed, then from the perspective of the internet (as an example) it SHOULD be NET-mapped to a public IP such as 44.31.0.23/32. This will enable seamless collaboration and resource sharing while maintaining network isolation boundaries.

   d. **Roaming Public IP Addresses:** The automated VPN MUST support the roaming of IP addresses, allowing VPN clients to connect from various locations while maintaining consistent IP address connectivity. System MUST auto re-establish tunnels without having the need for the user to manually update source IP of internet connectivity. This will enhance flexibility and convenience for users without compromising security. Roam-able IPv4 allows you to route your publicly routable IP anywhere in the world; attached to devices such as amateur radio "Ham" Towers, Rigs, phones, Internet-of-Things (IoT's), etc.

If using a cellular connection as the internet connectivity, new possibilities are created such as being able to give Raspberry Pi or RadioSonde balloons a public IP which could enable the ability to read telemetry from anywhere in the world.

e. **IPv4allocation digestion reduction:** Since about 10% of users requesting /24 from ARDC will know how to use a prefix allocation; by providing an automated VPN, users MAY request smaller allocations from the PoP's. Using the current methods (without an automated VPN service), experienced users that request a /24 (256 IP's) will set up their own virtual machine at a datacenter, BGP advertise to an upstream (northbound) transit provider, and setup a few tunnels back to their home, Tower or ham shack. The final quantity of IPv4usage will typically be 5 to 10 IP's out of the 256 (254 usable) allocation. Users not having network know how SHOULD be able to directly request 1 or more IP's from CORE PoP(s). For users wishing to provide their own infrastructure, by giving the tools and knowledge needed to provide Micro PoP's that MAY hang off (virtually) from upstream Core PoP's, ARDC MAY give downstream (southbound) connected Micro PoP's a smaller allocation out of the upstream Core PoP's allocation, such as a /28 (16 IPs). This process will free up large swaths of the 44net allocations controlled by ARDC for use by others.

f. **IPIP Gateway Depreciation:** Reduce dependency on the aging and low bandwidth IPIP gateway hosted at UCSD. Existing methods for using IPIP Mesh are fairly complex to set up. In order to improve ease of connectivity, PoP VPN's MUST auto provision a tunnel and SHOULD send connection credentials to a user's email, only requiring the user to download a WireGuard Client, and import configuration file containing auto-generated PKI credentials to connect. During the transition phase, the PoPs MAY connect into the IPIP network to allow connectivity to/from pre-existing networks.

g. **NAT Traversal:** To overcome the challenges posed by network address translation (NAT), the VPN solution MUST implement techniques to "punch through" single or double NAT'd networks. This will ensure that VPN connections MAY be established even in scenarios where traditional VPN protocols encounter difficulties.

5. **Protocol Overview:** The automated VPN deployment system leverages the power of [WireGuard], a modern and high-performance open source VPN protocol. It provides a simple and secure way to establish encrypted communication channels over the internet or private networks. It utilizes state-of-the-art cryptographic primitives and aims to reduce complexity without compromising security. This protocol operates at the network layer, enabling transparent routing of IP traffic.

6. **Wireguard:** Is an IP-layer protocol (operating at layer 3) designed as an alternative to IPsec for certain use cases. It uses UDP to encapsulate

IP datagrams between peers. Unlike most transport security protocols, which rely on Public Key Infrastructure (PKI) for peer authentication, WireGuard authenticates peers using pre-shared public keys delivered out of band, each of which is bound to one or more IP addresses. As a protocol suited for VPNs, WireGuard offers no extensibility,negotiation, or cryptographic agility. WireGuard isn't limited to L3 unicast connectivity; with the help of IGMP proxies, such as avahi, multicast may also transverse Wireguard tunnels.

7. **Design & Technical Details:** The design section describes the proposed design or solution for meeting the requirements outlined in the document. This section contains the main content of the document, explaining the design choices, technical details, and any algorithms, data structures, or protocols described in the RFC.

   Design Considerations:

   a. User Interface: Develop a user-friendly interface to configure and manage VPN connections. This interface SHOULD provide a clear and intuitive way to add, modify, and delete VPN tunnels. Users SHOULD be sent a preconfigured file for small routers like GL-INET or Mikrotik along with Linux, Windows, and Mac Operating systems. Additionally QRcode techniques MAY provide an easy onramp for mobile phones. This SHOULD be performed by scanning QRcode inside WireGuard app which MUST auto provisions the users device for immediate connectivity.
   b. Code reuse/mobility: In order to have a maintainable codebase, both CORE & MICRO PoP's SHOULD use the same code. This will provide a uniform PoP package that allows users to choose to set up a MICRO pop that performs the same functionality as CORE PoP's.
   c. Key Management: Implement a secure key management system to generate and distribute public and private keys required by the WireGuard protocol.
   d. Network Topology: Support various network topologies, including point-to-point, site-to-site, and peering with isolated Ham Networks using protocols such as PPTP. BGP and OSPF SHOULD be used to provide equal-cost-multi-path ECMP routing.
   e. Scalability: Design the automated VPN solution to handle a large number of VPN tunnels efficiently. VPN solution SHOULD be designed to scale horizontally to accommodate a growing number of VPN clients and connected networks. This MAY be achieved by employing load balancing techniques for dashboard and utilizing additional VPN servers as the user base expands.
   f. Performance: From anecdotal evidence, with moderate data bandwidth used, it is expected performance to be 30-250 tunnels per 2vCPU virtual machine. Capacity of VPN infrastructure SHOULD be evaluated as an ongoing adjustment. With bandwidth being a premium, either a hardware server with unlimited transit bandwidth or additional virtual PoP nodes MAY be added as bandwidth requirements increase. Use Qdisc for bandwidth shaping.
   g. Failover: Each PoP location (especially CORE PoP's) MUST have

a failover Virtual Machine or Hardware Machine using a floating IP and BGP preference to allow a failing machine (or ones that need patching or maintenance) be able to be pulled out of service, with users only experiencing a minimal amount of unreachability while the tunnels handshakes are re-established.

h. Security: System MUST implement robust security measures to protect the confidentiality and integrity of the VPN traffic. It is crucial to protect against unauthorized access, data breaches, and other security threats. Use best password practices such as NIST 800-63B. You MUST use the tightest firewall settings. You MUST move the server's wireguard interface, (which provides the endpoint for all tunnels), into its own namespace. MUST use namespaced virtual interfaces to isolate management and access interfaces from PoP's wireguard tunnels. Application should only collect minimal logs to diagnose program issues. Any user facing application interfaces such as the dashboard MUST have DDoS protection.
i. Isolated Network Interconnectivity: To allow interconnectivity to isolated Ham networks such as AREDN & HAMNET, connected users MUST appear to originate on the isolated Ham Network, without polluting the networks with internet traffic. To accomplish this, each isolated network MAY be peered over BGP with a /24 allocated from each isolated network, mapped to each VPN's Point of presence (PoP's), which MAY be Netmapped to the internet routable /24 for each (PoP). One PoP VPN tunnel MAY give access to all the Networks attached at each regional CORE PoP. BGP peering between Isolated Ham Network's and PoP's only have the Ham Network's IP routed through the link. No other routes are shared with the isolated networks, which keeps everything clean.
j. Monitoring and Logging: To ensure seamless operation and troubleshooting, the VPN solution SHOULD include monitoring and logging capabilities. This will allow administrators to track VPN performance, detect anomalies, and analyze logs for security incidents or operational issues.
k. Data: Each CORE PoP MAY be used to provide Data Analytics and Research for the Internet Measurement projects.
l. Redundancy: To aid in maintaining system availability and limiting the impacts of failures:
    i. ARDC Regional PoPs SHOULD be connected to at least 2 ARDC Core PoPs.
    ii. PoP's SHOULD advertise, at least, a default route into the PoP network.
    iii. ARDC Core PoPs MAY advertise the same 44net address space to the Global Routing Table.
    iv. ARDC Core PopS MAY be inter-connected as a full-mesh (ie. a Core PoP interconnects with all other Core PoPs).

Security Considerations:

The automated VPN system SHOULD prioritize security

throughout the setup and management processes. This includes the use of strong encryption algorithms, robust authentication mechanisms, and secure key exchange protocols. Additionally, the system SHOULD undergo regular security audits and updates to address emerging threats.

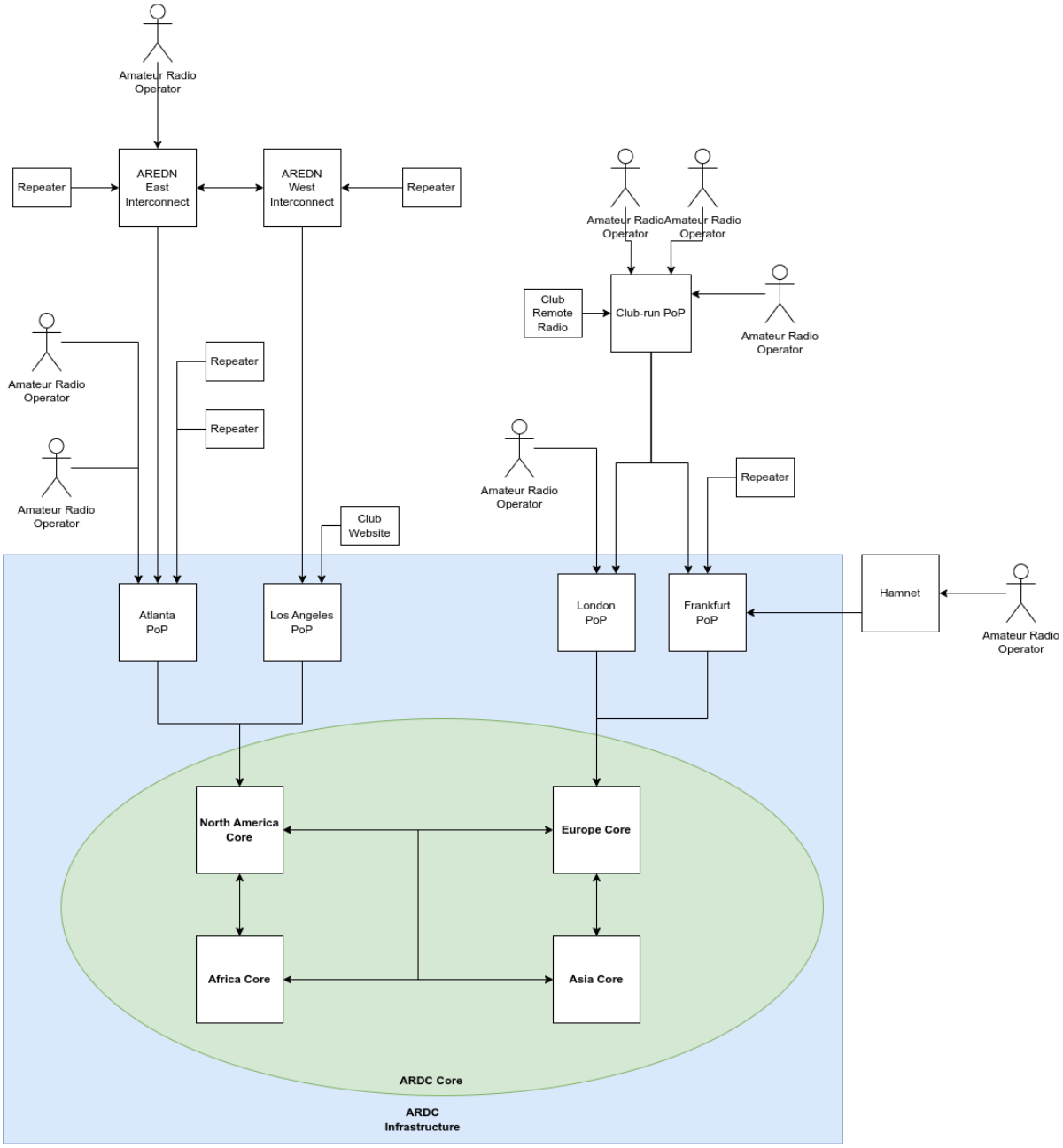
m. Strong Encryption: WireGuard employs state-of-the-art cryptography to ensure secure communication.
n. Key Management: Administrators SHOULD securely manage private keys to prevent unauthorized access.
o. Firewall Configuration: Properly configure firewalls to allow VPN traffic and block unauthorized access.
p. Regular Updates: Keep the VPN server and client software up-to-date to address security vulnerabilities.
q. Namespace Isolation: Using network namespaces, isolation of tunnels and BGP aids in PoP server tunnels being isolated from management interfaces. Removing direct access of local PoP server machines via tunnels.

System Architecture: The system consists of the following components:

r. Nodes: Servers, either virtual or physical, which form the PoP system. Nodes SHOULD be geographically distributed to reduce latency. Nodes are hierarchical:
    i. Core Nodes (ARDC operated): Provide connectivity between regions. Core nodes SHOULD announce the 44net address space to the Global Internet Routing Table. (E.g, North America, Europe, Asia) End users SHOULD NOT be connected to Core nodes.
    ii. Regional Nodes (ARDC operated): Regional nodes connect the users (either Amateur Radio Operators or Micro PoPs) within a region (e.g., Canada East, Canada West, U.S. East, U.S. West, Europe East, Europe West, etc.)
    iii. Local Nodes (ie. Micro PoP) (Community operated): A local node is community-operated for a local group of users (e.g., an amateur radio club)
s. Node Controller: The node controller is responsible for reconciling the PoP routing configuration on each node from the Database. This includes configuring Wireguard interfaces and peers, and Bird BGP configuration.
t. Interconnect: An interconnect is a “special” interface and peer which connects two Nodes together. This interconnect permits network traffic from one node to reach another node. Interconnects MUST be performed using trusted nodes, as full routes are exchanged between nodes.
    i. Interconnects with community-operated PoP MUST be treated as a regular peer. This will ensure that normal protections, such as route filters, are applied.
u. Database: The database contains the authoritative source of PoP routing configuration. ARDC-managed PoPs SHOULD use a singular, globally distributed database.

Community-operated PoPs MAY deploy their own database.

v. Administration Management Dashboard: Allows administrators to configure and manage client devices, and allows complicated alterations to Database that could cause disruption if improperly performed by an end user. Admin dashboard updates a common database used in endpoint controller orchestration. Roles SHOULD be used in the Administration dashboard to limit access to either their own PoP nodes (as a regional admin), or provide access to all PoP nodes as a global admin.
w. User Dashboard: Provides user login. Updates and reads a database used in controller orchestration of tunnel creation, deletion, enabling. Also provides tunnel information and security keys in a convenient interface.
x. VPN Client: Open sourced WireGuard Client application installed on client devices, it establishes secure connections with the VPN CORE & MICRO PoP servers.


Amateur Radio Operator
Repeater
AREDN East Interconnect
AREDN West Interconnect
Repeater
Amateur Radio Operator
Amateur Radio Operator
Amateur Radio Operator
Club Remote Radio
Club-run PoP
Amateur Radio Operator
Repeater
Repeater
Amateur Radio Operator
Amateur Radio Operator
Repeater
Club Website
Atlanta PoP
Los Angeles PoP
London PoP
Frankfurt PoP
Hamnet
Amateur Radio Operator
North America Core
Europe Core
Africa Core
Asia Core
ARDC Core
ARDC Infrastructure

Implementation Details:

i. Decentralized VPN Server: The VPN solution will be based on a decentralized VPN server model responsible for managing its own VPN connection, handling key exchanges, and routing traffic between connected users and other networks. The server SHOULD authenticate VPN clients and enforce security policies to ensure secure communication.

ii. VPN PoP servers such as CORE PoP's MAY use a shared/common Database for orchestration. Each VPN controller SHOULD periodically, at 5-10 second intervals determined empirically, poll a database (such as MongoDB or similar) and compare its own connections with the state of each tunnel from the DB; if records are returned that are different than local configurations, controllers MUST update itself. This system SHOULD provide a user-friendly interface for network administrators to manage VPN connections, assign public IP addresses, and monitor network traffic.

iii. Automated Configuration Management: Develop a configuration management system to store and retrieve VPN tunnel configurations. The VPN system SHOULD employ an automated configuration process, simplifying the deployment and management of VPN connections. This will involve the use of scripts and tools to automatically generate VPN client configurations, handle key distribution, modification, and deletion of VPN tunnels.

Deployment The deployment process involves the following steps:

iv. Provisioning the VPN Server:
   1. Install the necessary software packages and dependencies.
   2. Generate the server's private and public keys.
   3. Configure network interfaces and firewall rules.

v. Setting Up Clients:
   1. Each user installs the wireguard VPN Client software on the user's devices.
   2. VPN server Generate client keys and configure network settings. Send automatically to the user.
   3. The VPN server pins up a tunnel waiting for connection.

vi. Establishing Connections:
   1. Clients authenticate with the VPN server using their private keys. The VPN server verifies client credentials and establishes secure connections.
   2. Traffic between clients is encrypted and routed through the VPN server.
   3. Configuration Options Administrators MAY customize on the VPN deployment using the

following configuration options:

a. VPN Server IP Address: Specify the IP address for the VPN server.
b. Subnet and IP Range: Define the IP range for client devices.
c. DNS Settings: Configure DNS servers for client devices.
d. Firewall Rules: Set up custom firewall rules for the VPN server.

Automated VPN Setup: The automated VPN setup involves the following steps:

vii. Discovery and Authentication: The participating devices or networks MUST discover and authenticate each other before establishing a VPN tunnel. This MAY be achieved through various methods such as pre-shared keys, certificates, or public keys.

viii. Configuration Exchange: Once the devices have authenticated each other, they MUST exchange configuration information required for establishing the VPN tunnel. This includes IP addresses, encryption keys, and other parameters specific to the WireGuard protocol.

ix. Tunnel Establishment: Based on the exchanged configuration, the devices MUST establish a VPN tunnel using the WireGuard protocol. The tunnel allows encrypted communication between the participating networks or devices.

x. Automated Management: Automated management of the VPN tunnels includes the following aspects:

1. Dynamic Routing: The automated VPN system SHOULD support dynamic routing protocols, enabling efficient communication between networks. This ensures that changes in network topology or device availability are automatically reflected in the VPN configuration.
2. BGP Configuration: On PoP interconnections, each PoP node MUST auto create both ends of the BIRD BGP configurations.
3. Key Generation & Rotation: To maintain a high level of security, the automated VPN system SHOULD support key rotation of public, private and pre-shared keys for each VPN tunnel. This involves periodically updating the pre-shared encryption keys used for securing the VPN tunnel. The rotation process SHOULD be seamless and not disrupt the ongoing communication.
4. Monitoring and Logging: Effective monitoring and logging mechanisms SHOULD be in place to track the performance and security of the automated VPN system. This includes monitoring capabilities and

the status of VPN tunnels, bandwidth usage, and detecting any potential security breaches.

5. Updating DNS records for each tunnel on a remote platform.

Maintenance:

xi. Conduct thorough testing and evaluation of the solution in a controlled environment.

xii. Document the installation and configuration process for easy deployment

xiii. Provide comprehensive documentation and support materials for users.

xiv. Regularly update and maintain the VPN solution to address security vulnerabilities and compatibility issues.

Testing & Validation:

xv. An outline for testing and validation for VPN automation

5. **Conclusion:** Automating the configuration and management of VPN tunnels using WireGuard offers numerous benefits in terms of simplicity, scalability, security and providing added utility for end users with use of publicly routed IP addresses to devices. This document provides a foundation for the development and implementation of an automated VPN system based on the WireGuard protocol. By following the guidelines outlined in this document, an organization, club or individual can easily help others establish secure tunnel endpoints. By simplifying the setup process and offering customization options, the PoP Project enables organizations to deploy VPNs with ease, opening up the possibilities of connecting island networks and having publicly routable, roamable IP's.

6. **Acknowledgments:**

The author(s) would like to thank Jason A. Donenfeld, creator of WireGuard, and the WireGuard community, along with the TAC members of ARDC for their valuable contributions and insights in developing this process.

Author(s):
Adam William Lewis (KC7GDY)
6015 Saskan Ranch Circle, Flagstaff, AZ 86001
Phone: 928-637-8132
Email: adam@airgapped.io

Zachary Tyler Seguin (VA3ZTS)
Email: zachary@va3zts.ca

Jon Kemper (KA6NVY)
Email: jon@ardc.net

Appendices / References:

Document(s):
44net_Architecture_V2.pdf

Problem Solved_UseCase 1 - 3

https://nextcloud.ardc.net/index.php/apps/files/files?dir=/TAC/POP

[WireGuard] Donenfeld, J., "WireGuard: Next Generation Kernel Network Tunnel", <https://www.wireguard.com/papers/wireguard.pdf>.

RFC 2119: Key words for use in RFCs to Indicate Requirement Levels

FINAL NOTES:
This document is a proposal and does not represent a finalized standard. Feedback and contributions are welcome to further refine the proposed approach.