CloudGen Access Application

https://campus.barracuda.com/doc/93201509/

CloudGen Access Agent

The CloudGen Access agent intercepts requests at the network layer. When a DNS request matches a protected resource, it injects a reply, pointing the domain to an internally accessible marker IP address that represents the resource. The agent then, in parallel, initiates a request to the Access Policy Engine to grant permission to the resource and establishes an mTLS connection to the Access Proxy that is associated with the resource.

The CloudGen Access Agent also includes a DNS security engine that blocks requests that match blacklists configured at the Enterprise Console level.

See also:

- How to Deploy the CloudGen Access Application.
- Troubleshooting the CloudGen Access Application

Supported Operating Systems

- Android 6.0 (API level 23) or higher
- macOS Catalina (10.15) or higher
- iOS 12 or above
- Linux (Arch, Ubuntu 20.04 (or higher) and Redhat)
- Windows 10 or higher

> Note: The CloudGen Access App requires the user to re-authenticate their devices once a month, refreshing the certificate. The user will receive a reminder 22 days after the last refresh of the certificate to authenticate in order to continue having access.

CloudGen Access App UI

Besides configuration and enrollment, the CloudGen Access app UI has two main responsibilities:

1. It guides the user through remediations to be able to access a resource.
2. It provides a friendly UX for blocked DNS resources.

Both flows start with a notification that is triggered when the user either tries to access a resource without fully complying with the corresponding policy or when the user tries to access a blocked domain.

Windows

DNS Leak Prevention

CloudGen Access operates as a split tunnel (see https://en.wikipedia.org/wiki/Split_tunneling) and intercepts all DNS traffic. This DNS interception works in the following way:

1) If the domain name on the DNS query matches a configured resource, a "virtual" IP address will be returned so that it can be used for TCP/UDP communication over the tunnel to the proxy
2) If the domain name is considered malicious or harmful, an NXDOMAIN is returned (see https://www.dnsknowledge.com/whatis/nxdomain-non-existent-domain-2).
3) All other domain name queries are forwarded to the system's DNS servers or the DNS servers configured on the enterprise console.

In its default configuration, the Windows 10 DNS resolver queries all DNS servers of all network interfaces concurrently and uses the first positive reply. This leads to DNS leaks (see https://en.wikipedia.org/wiki/DNS_leak) where a given query will go through a local network interface, thus bypassing CloudGen Access DNS filtering.

As an example, if you have a resource with domain name `example.org` and try to resolve that domain name, the IP address will be given by the first DNS server to reply, and if it is the *system* DNS server, then communication with that resource will not go through CloudGen Access. However, this is highly improbable because resource domain names are kept in a local database, and no network traffic is involved to resolve them. Also, note that filtering malicious or harmful domain names is problematic, because CloudGen Access might reply with an NXDOMAIN, while a DNS server of a local interface might reply with a valid IP address.

To prevent this issue, CloudGen Access has a mechanism to prevent DNS leaking by installing Windows Filtering Platform rules to prevent DNS traffic to all programs except the CloudGen Access process.

This mechanism might be incompatible with other DNS filtering solutions. If you already have a solution for DNS filtering, you can disable DNS leak prevention by setting the following registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Barracuda\CloudGen Access]
"DisableDnsRestrict"=dword:00000001
```