VirnetX

Gabriel Security Platform™

Protect Communications
with Private and Secure Domains

October 2016

Gabriel Integration White Paper

GABRIEL
SECURE COMMUNICATIONS
"Take back your digital life"

Introduction

Each day people wake up to hear or read about the latest data breach. Companies are being compromised at an increasing rate. While the financial impact can be recoverable, the damage to one's brand is almost irreparable. Global cyber-attacks demonstrate the challenge to protect your data.

OUTLINE

The Internet was designed to allow devices to communicate with ease through public domains and Internet addresses. Security was not a priority and add-on solutions came much later. Continuous discovery of new "zero-day" attacks show that these add-on security solutions can only provide *reactive* protection until the next vulnerability is discovered. A completely new approach, one that is **proactive** and built on the foundations of security, is required to counter these growing threats.

VirnetX's Gabriel Connect Technology has been architected and developed from its very inception to address these challenges by integrating VirnetX patented technology with industry standard cryptography, technology and practices. Originally developed for the intelligence community to provide private secure communications and data protection while leveraging the Internet infrastructure, Gabriel Connect Technology can now be used to secure all types of data communications through the use of private secure domains / internet addresses.

Gabriel Connect Technology provides end-to-end encrypted communication between any two devices on the Internet and intranet. It has been designed to enable instant private and secure connections for ANY DEVICE, ANYTIME, and ANYWHERE across the Internet/intranet. This allows users the ability to privately chat, email, talk, video call, and share information regardless of where the users are located or the devices being used. Gabriel Connect Technology can be deployed on devices including, smart phone, tablet, laptop, desktop or servers, IoT, Cameras, etc. on private wired, Wi-Fi, public Wi-Fi or cellular networks.

Following a brief overview of the Gabriel Connect Technology and associated Gabriel Security Platform, this Gabriel *Integration Whitepaper* describes methods by which Application Developers and Systems Integrators can integrate the Gabriel Security Platform into their applications and private networks for securing and protecting communications. For a more in-depth discussion of the Gabriel Connect Technology, Gabriel Security Platform and Gabriel Collaboration Suite, the reader is referred to *Gabriel Secure Communications and Gabriel Collaboration Suite – Whitepaper*.

Gabriel Connect Technology Overview

All Internet communications today, uses an Internet Protocol (IP) address, which is a sequence of numbers, to determine where to send data packets destined for a specific device. The Domain Name System (DNS) was created to simplify reaching public domains by associating a network/IP address to a name for a specific device. The DNS relies on a process, network address resolution, to resolve a domain name into a specific IP address of the device where are all the data packets need to be sent. For example, a user accessing the Acme website would enter the domain name (www.acme.com) for the website. This domain name is sent to a DNS server, which either knows the corresponding IP address or forwards the request to a server, which knows its address. When the address is found, it is returned to the requesting device. This process suffers from vulnerabilities which can make it susceptible to a number of security threats including Distributed Denial of Service attacks (DDoS), DNS cache poisoning, Registrar hijacking, etc.

Unlike traditional DNS, Gabriel Connect Technology, mitigates these vulnerabilities, by using a private secure domain name (e.g. www.acme.scom). Gabriel Connect Technology enhances the step of network address resolution by automatically determining the need for initiating a Virtual Private Network (VPN) and setting it up automatically to the destination device.

Figure 1, illustrates how the Gabriel Connect Technology intercepts the domain name lookup before it is sent to the legacy DNS and determines if the network address resolution request involves a Secure Domain Name. If the domain name in the resolution request is determined to be a Secure Domain Name, a VPN is provisioned and automatically setup for secure communication between the requesting device and the destination (or target) device. The VPN uses a secure, private IP address, which is then returned to the user's requesting application. This IP address is then used by the application to connect to the target device through a secure VPN link. If the domain name lookup request is determined to be for an unsecure, legacy domain name then the request is forwarded to the legacy DNS.

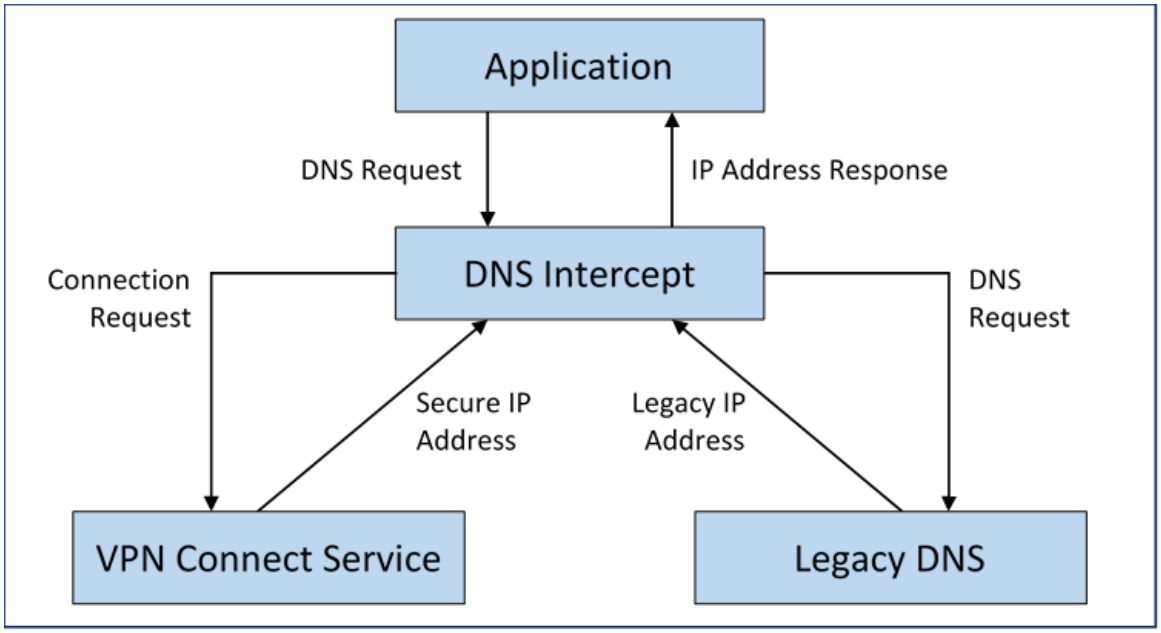

Figure 1 – Gabriel Connect Technology

Gabriel Connect Technology has been implemented as Gabriel Instant Secure Connect software and is now available for Android, iOS, Linux, Mac OSX and Windows. Developers and system integrators have the option to support Gabriel Connect software at either the network or application layer depending on the type of device. Mobile operating systems, traditionally, do not allow the deployment of 3rd party network drivers and applications that modify the system network settings. Due to these operating system restrictions, Gabriel Connect software must be deployed at the application layer on mobile devices. On non-mobile devices the Gabriel Connect software can be deployed at either the network or application layer. For example, Windows is more convenient to implement Gabriel Connect software at the application layer so 3rd party Secure Suites/Firewalls do not need to be configured to accommodate Gabriel Connect Technology.

Figure 2 illustrates Gabriel Connect Technology implemented at the network layer (non-mobile operating systems). It shows how Gabriel Connect Technology can be directly connected into the operating system network stack where it can route IP traffic and capture all system DNS requests.

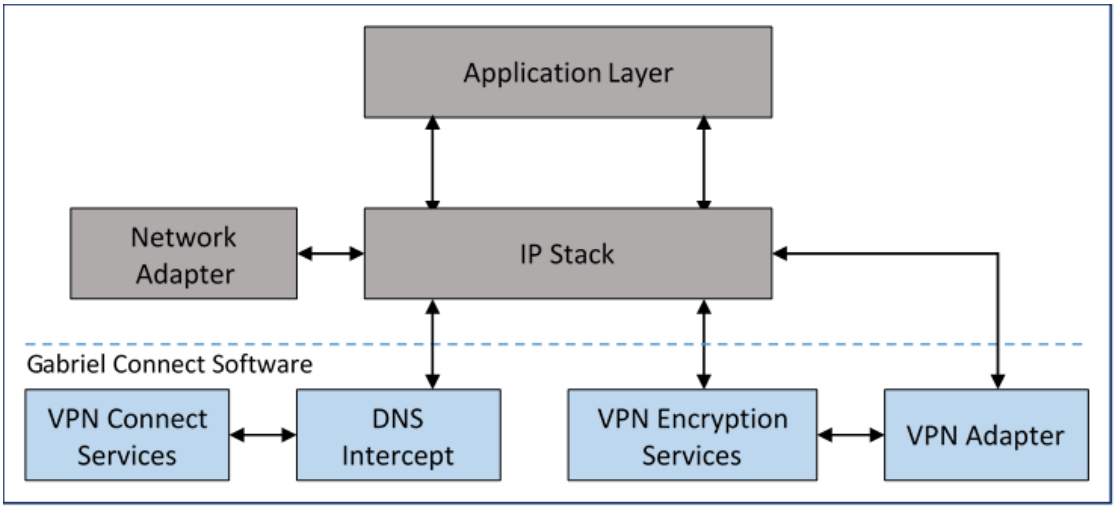

Figure 2 – Gabriel Connect Software Implementation at Network Layer

Figure 3 illustrates an alternative Gabriel Connect Technology implementation at the application layer (mobile devices and optionally other operating systems). Gabriel Connect Technology interfaces with the operating system IP stack through TCP/UDP sockets and legacy DNS resolution. Gabriel Connect Technology performs secure DNS, routing to secure VPNs, encryption/decryption, and packet encapsulation/de-encapsulation is done at the application layer. This is referred to as Gabriel Secure Virtual Tunnel (VTUN).

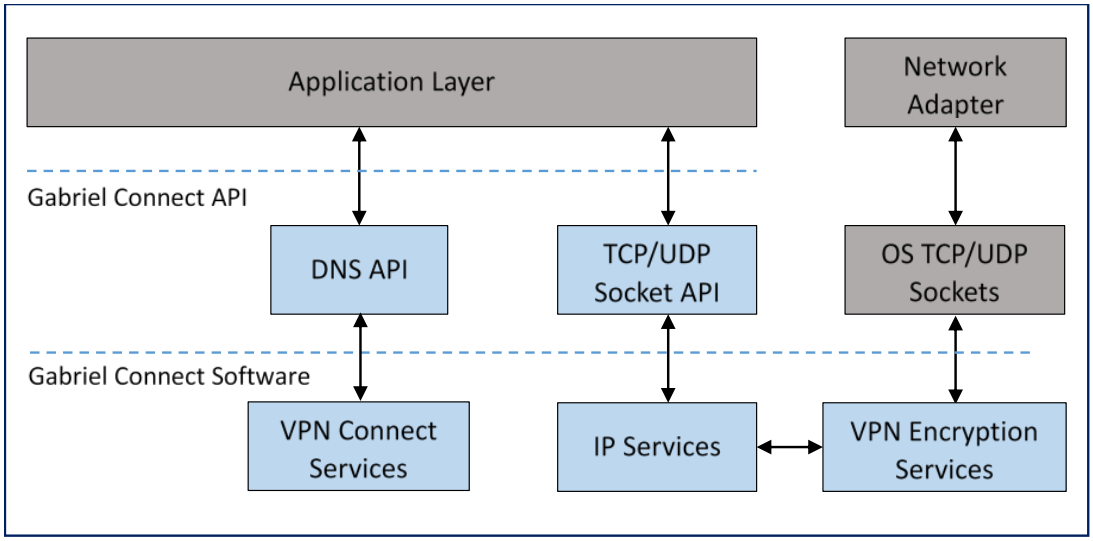

Figure 3 – Gabriel Connect Software Implementation at the Application Layer

Gabriel Security Platform Overview

The Gabriel Security Platform was engineered to provide the highest levels of privacy, security, policy control, and mitigate threat exposures. When combined with other security software (virus scanning, malware detection, audit, strong identity management, etc.), the platform provides the strongest possible defense against current and evolving cyber-attacks.

GABRIEL SECURITY PLATFORM
Provides all infrastructure and services required to create a cryptographically secure private network, manage access and enforce policies.

The Gabriel Security Platform automatically establishes private network enclaves and supports heterogeneous devices. The authenticated access controls and strong encryption peer-to-peer links are enforced within the private enclave.

Figure 4 illustrates the architecture of the Gabriel Security Platform and the collection within the components of Gabriel Connect Technology required for deploying on ANY DEVICE, ANYTIME, and ANYWHERE across the Internet/intranet. The security platform has been designed for maximum flexibility and broadest protection across most IP enabled devices. Developers and system integrators can pick and choose the appropriate components of the platform depending on the device and functionality being supported.

GABRIEL INSTANT SECURE CONNECT
Intercepts the domain name look up before it is sent to legacy DNS to determine if it is a SECURE DOMAIN NAME. Performs cryptographic authentication of peering devices

SECURE DOMAIN NAME SERVICE
Provides address request lookup for VPN initiation, secure address resolution, and reverse address lookup.

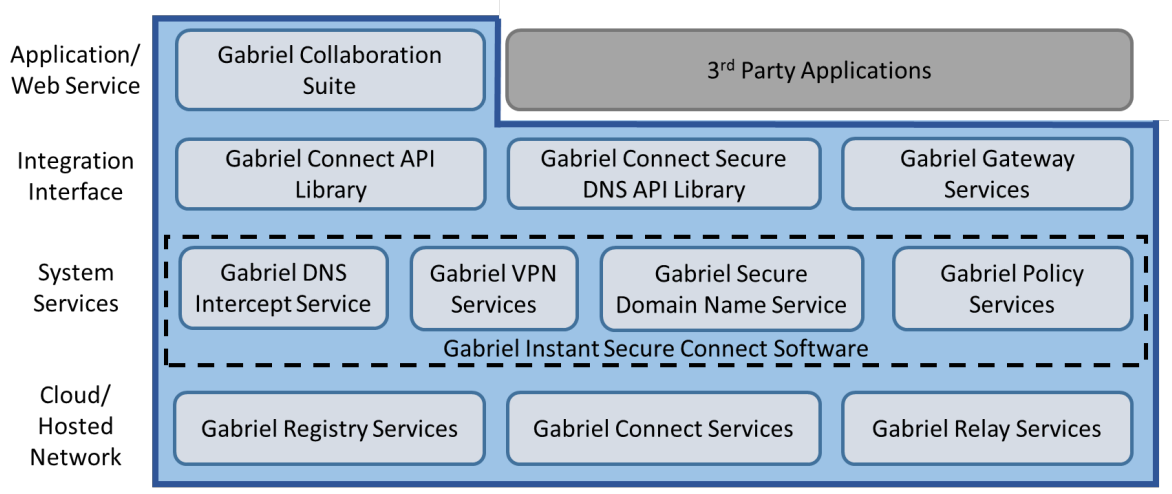

Figure 4 – Gabriel Security Platform Architecture

The complete security platform includes the following components and sub-components:

- Device/ Web Services Components
  - Gabriel Instant Secure Connect software:
    - Gabriel DNS Intercept Service
    - Gabriel VPN Services
    - Gabriel Secure Domain Service
    - Gabriel Policy Services
  - Gabriel Connect API Library
  - Gabriel Connect Secure DNS API Library
  - Gabriel Gateway Services
- Cloud/Hosted Infrastructure Components
  - Gabriel Registry Services
  - Gabriel Connect Services
  - Gabriel Relay Services

Gabriel Security Platform provides the networking and cryptographic infrastructure that enables the following:

- Private Secure Domain Name / Internet Address Registration,
- User defined security policy,
- Secure peer presence discovery,
- Domain Name Lookup interception and Secure Domain Name request determination by *Gabriel Instant Secure Connect,*
- Seamless efficient VPN initiation and management,
- *Secure Domain Name Service* address request lookup feature:
  - Automatic VPN initiation,
  - Remote peer secure address resolution, and
  - Certified peer IP reverse address lookup
- Cryptographic peer authentication,
- Network Address Translation (NAT) firewall discovery and relay services when needed, and

The Gabriel Security Platform implements both Registry and Registrar Services to register users and/or IP devices, provides network presence services, and supports Secure DNS for Gabriel protected devices throughout the Internet/intranet.

GABRIEL SECURE
Any network, application or IP enabled device protected by Gabriel Security Platform and the use of private and secure domains and internet addresses

By using the Gabriel Security Platform, developers can create their own "On-Demand" protected networks with secure access through mutually authenticated users and devices. ALL data packets are encrypted and transported through encrypted tunnels using the existing Internet infrastructure.

Gabriel Secure private networks CANNOT be accessed using legacy DNS; *therefore, you can't hack what you can't see.*

Gabriel Integration Methods

This section describes the various integration methods available to an application developer or the system integrator to create Gabriel Secure networks, applications or protection for IP enabled devices by taking advantage of the functionality provided by the Gabriel Security Platform.

INTEGRATION METHODS:

GABRIEL SECURE GATEWAY SERVICES protect IP enabled devices without Gabriel Secure resident on the device itself.

GABRIEL CONNECT API provides access to Gabriel Connect library functions for securing the application/service for use across all devices.

GABRIEL CONNECT DNS API intercepts DNS requests and recognizes secure domain access request.

1. ***Gabriel Secure Gateway Services*** – Developers / integrators can secure IP enabled applications and devices using the Gabriel Secure Gateway Services without making any modifications other than directing communications through the secure gateway. IP devices installed on local/private networks can communicate securely over the public Internet using the Gabriel Security Platform without modification by using an externally provided Gabriel Secure Gateway Service. If the network service secured by Gabriel Secure Gateway Services is a web service, then Gabriel can act as a client to securely access the web service. On the server side, Gabriel does not need to reside on the same device as the secured service. An example is described later in protecting IP enabled devices (e.g. security cameras) against unauthorized outside access.

2. ***Gabriel Connect Application Program Interface (API)*** – Developers have the option of adding network service calls to Gabriel API libraries directly into the applications. While this requires modifying the application to support Secure Domain Names, it is the most efficient approach as it allows securing the application's communication for use across all devices. A developer can include all the Gabriel Instant Secure Connect software dependencies in the installation package.

3. ***Gabriel Connect Secure DNS API*** – The Gabriel Instant Secure Connect Client software on non-mobile platforms runs at the network device driver layer of the operating system, allowing Gabriel to intercept DNS requests from the application. For Secure DNS requests, the Gabriel Instant Secure Connect software instantiates a VPN channel before returning a secure IP address. Once this secure VPN is established, all data from Secure DNS enabled applications is routed through the encrypted tunnel to a secure IP address. An application can be secured by simply modifying its domain name request to a Gabriel Secure Domain Name request. As an example, an application could request a Gabriel Secure Domain Name e.g. *server.acme.**scom*** instead of a regular domain name, such as *server.acme.com* A Gabriel Instant Secure Connect software provides an integrated firewall for VPN traffic. The specific communication ports required by

the application needs to be enabled within the Gabriel Instant Secure Connect software configuration. The Gabriel Instant Secure Connect software must be installed and running on all devices.

Gabriel Secure Gateway Services

The Gabriel Secure Gateway Services will secure the communications of a 3rd party application or IP enabled device not running the Gabriel Secure software. The Gabriel Secure Gateway Services provides a network proxy so that communications to either a local or remote device on the local/ private network, communicates securely to a device running Gabriel Instant Secure Connect software. The devices are proxied to that final destination for the communications. A web-based network service can be securely accessed from a Gabriel Connect Technology enabled device, without any modification to the service or application. When using the Gabriel Secure Gateway Services, Gabriel Connect software must be installed on both ends of the communication. On the server side (e.g. private/home network), Gabriel Instant Secure Connect software does not need to reside on the IP device. As an example, a user could install Gabriel Secure Gateway Service server on any machine inside his/her home network and protect IP enabled devices (e.g. security cameras, lights, garage door) against unauthorized outside access.

GABRIEL SECURE GATEWAY SERVICES
protect IP enabled devices without Gabriel Secure resident on the device itself.

- No Need for Opening Firewall Ports
- Secure Domain Name Addressing
- Digital Certificate Authentication
- All Connections Encrypted
- Owner Authorizes Access
- Unauthorized Access Blocked

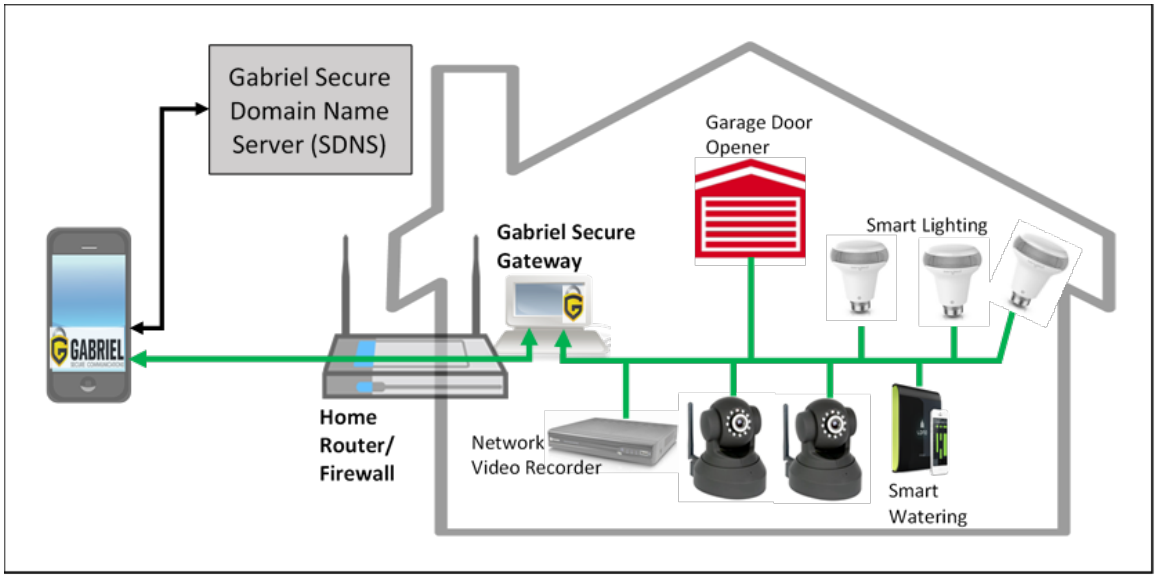

Figure 5 – Securing Smart Home Using Gabriel Secure Gateway Services

With additional configuration on the Gabriel Connect software, it can automatically be configured at the same time as the Gabriel Secure Gateway Service. Figure 5 shows a 3rd party application on the client can access their IP device through the VPN tunnel. The tunnel is first established to the Gabriel app on the client device and then from the Gabriel Secure Gateway Services to the final IP device or 3rd party application. Figure 6 represents an example of a Gabriel Secure Gateway Services configuration for an IP camera.

```
MappedUdpPort=8080
TargetUdpPort=80
MappedTcpPort=8080
TargetTcpPort=80
TargetIp=192.168.1.3
AllowedPolicies=camera
```

Figure 6 – Secure Gateway Configuration for IP Camera

Figure 7 below identifies some of the security vulnerabilities created by the widely used approach of opening ports in the home router/firewall to access IP devices installed in the home network. Some of the key issues include:

- Specific ports need to be opened in order to allow incoming connections. Hackers can randomly scan for vulnerable devices.
- Public Domain Name Addressing using legacy DNS can be discovered.
- Routers/ Firewalls are configured by default to accept all connection requests.
- Weak password authentication.
- Data encryption is not always available.

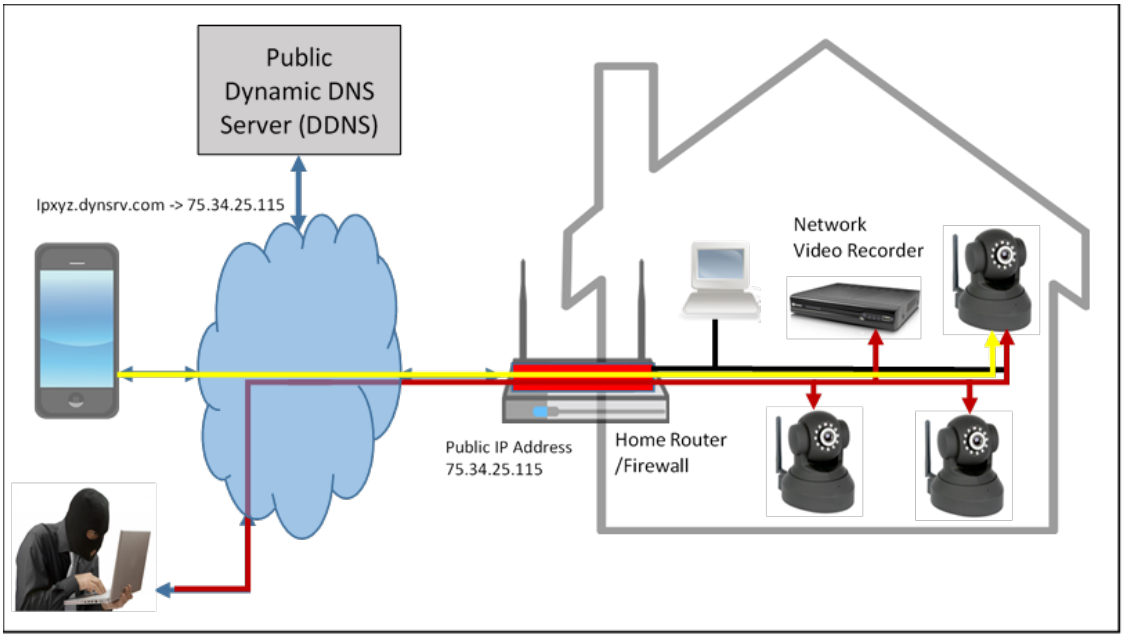

Figure 7 – Unsecured Camera Access in a Smart Home using legacy DNS

In Figures 5 and 6, Gabriel Instant Secure Connect software listens on UDP and TCP port 8080, and forwards the traffic to UDP and TCP port 80 on the target IP address 192.168.1.3. The local / private network communicates between the Gabriel Secure Gateway Services and the IP camera at 192.168.1.3 is protected by Gabriel Instant Secure Connect software from unauthorized access. Many of these IP devices, such as cameras are secured by being hard wired or using wireless security protocols (e.g. WEP, WPA or WPA2). The communication over the Internet to the Gabriel Secure Gateway Services on the local/private network is secured. It is not necessary to modify the network's router configurations to enable UPnP or port forwarding for achieving the secure access of your local IP device applications.

An IP camera available to those on the same local/private network at IP address `192.168.1.3,` is now securely available via Gabriel Secure Gateway Services to other users in the "CameraUsers" policy. This provides secure authenticated access to the IP camera without needing to install the Gabriel Connect software directly on the IP camera.

Note: The "TargetIp" in a Gabriel Secure Gateway Services configuration could also be set to Localhost IP 127.0.0.1. This would expose another network service running on the same device as the Gabriel Instant Secure Connect software. An example of this may be a Linux server running both Gabriel Instant Secure Connect software and an Apache Web server.

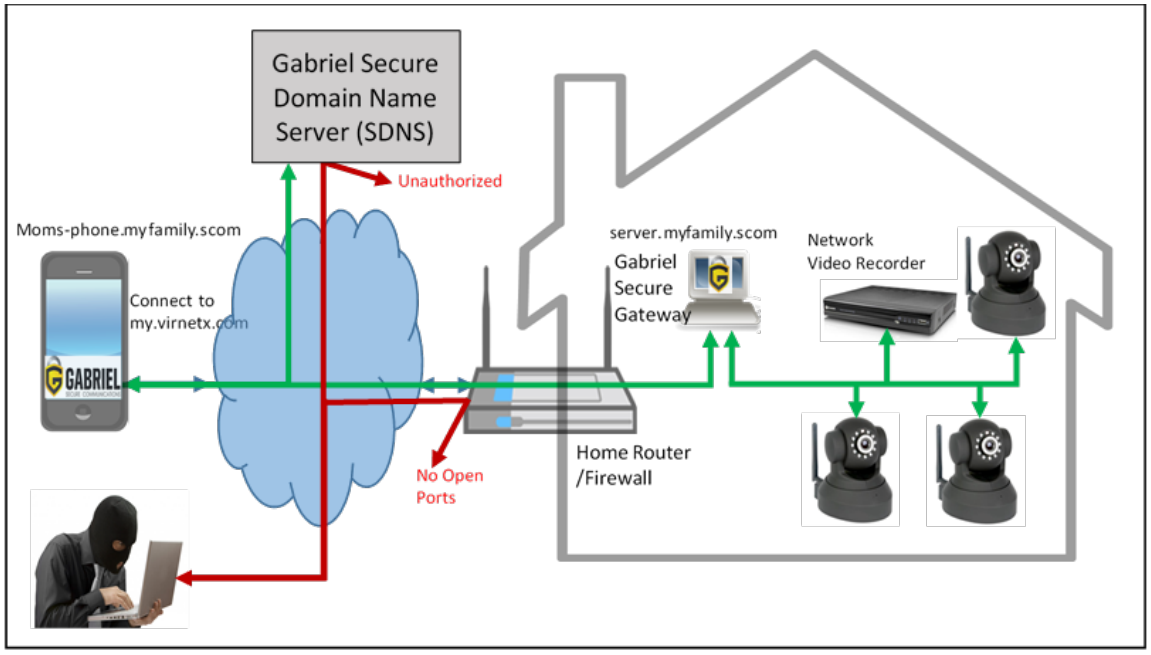

Figure 8 – Secured Camera Access using Gabriel Secure Gateway Services

Figure 8 illustrates significant advantages over the approach shown in Figure 7, including:

- No Need for Opening Firewall Ports
- Secure Domain Name Addressing
- Digital Certificate Authentication
- All Connections Encrypted
- Owner Authorizes Access
- Unauthorized Access Blocked

Gabriel Secure Gateway Services can be installed on any desktop, laptop or headless device for securing a 3rd party application. Gabriel Secure Gateway Service does not need to be resident on the IP device itself. The Gabriel Secure Gateway Services can be utilized in both consumer and commercial private networks.

Gabriel Connect Application Program Interface (API)

Using the Gabriel Connect API, developers can now 'program-in' the Gabriel DNS intercept and VPN transport functions for TCP (transmission control protocol) and UDP (user data gram protocol) packets directly in their applications across all major operating systems including Android, iOS, Mac OSX, Windows, and Linux operating systems.

The Gabriel Connect API goes a step beyond the Gabriel Secure Gateway Services. Instead of configuring your application or device to connect securely by tunneling through Gabriel Instant Secure Connect Software, you can rebuild your application or device software to link directly to the Gabriel Connect software so that the secure connections and security policy are directly embedded into your product.

Figure 9 and 10 illustrates sample code for a local / remote peer demonstrating network calls using the Gabriel Connect API (__GABRIEL_NETWORKING__) and network calls that the code would utilize when not secured by Gabriel. This provides an example of the minor changes required for existing network applications that are modified to communicate securely using Gabriel Secure DNS triggered VPNs.

```
#include <AppApi/GabrielClientApi.h>

// Tcp Connect
// variables set by code:
int port;
...
// Common to both
int acceptFD;
struct sockaddr_in me;
```

```
me.sin_family = AF_INET;
me.sin_addr.s_addr = htonl(INADDR_ANY);
me.sin_port = htons(port);
bool opt = true;

#if defined(__GABRIEL_NETWORKING__)
 acceptFD = gabriel_tcp_listen((struct sockaddr *) &me, sizeof(me));
 setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, (char *) &opt, sizeof(bool));
#else // Traditional
 acceptFD = socket(AF_INET, SOCK_STREAM, 0);
 if (bind(acceptFD, (struct sockaddr*)&me, sizeof(me)) == SOCKET_ERROR)
 {
  printf("could not bind\n");
  exit(1);
 }
 int backlog = 5;
 if (listen(acceptFD, backlog) == SOCKET_ERROR)
 {
  printf("could not listen\n");
  exit(1);
 }
#endif

// run select or a blocking call to accept on the listening socket
// The select call is the same for both
...
int newsock;
struct sockaddr_in clntaddr;

#if defined (__GABRIEL_NETWORKING__)
newsock = gabriel_accept(sock, &clntaddr, sizeof(clntaddr));
#else
newsock = accept(sock, &clntaddr, sizeof(clntaddr));
#endif
...
// close and cleanup the socket
#if defined (__GABRIEL_NETWORKING__)
 gabriel_tcp_close(sockFD);
#else
 close(sockFD);
#endif
```

Figure 9 – Example Network SERVER Program Using Gabriel Connect API

```
#include <AppApi/GabrielClientApi.h>

// Tcp Connect
// variables set by code:
int port;
char* hostname;
...
// Common to both
struct sockaddr_in dest;
struct hostent* ent;
dest.sin_family = AF_INET;
dest.sin_port = htons(port);

#if defined(__GABRIEL_NETWORKING__)
  ent = gabriel_gethostbyname(argv[1]);
#else // Traditional
  ent = gethostbyname(hostname);
#endif

if (ent == NULL)
{
  print("gethostbyname was null\n");
  exit(-1);
}
memcpy(&dest.sin_addr, ent->h_addr, ent->h_length);

int sockFD;

#if defined (__GABRIEL_NETWORKING__)
  if ((sockFD = gabriel_tcp_connect((struct sockaddr *) &dest,
                          sizeof(struct sockaddr_in))) == BAD_SOCKET)
#else
  sockFD = socket(AF_INET, SOCK_STREAM, 0);

  if (connect(sockFD,
          (struct sockaddr*)&dest,
          sizeof (struct sockaddr_in)) != 0)
#endif
  {
    printf("connect failed on port %d\n", ntohs(dest.sin_port));
    exit(1);
  }
```

```
// The socket is now defined and connected at this point
// You want to use the normal socket read/write commands on the  sockFD
...
// close and cleanup the socket
#if defined (__GABRIEL_NETWORKING__)
  gabriel_tcp_close(sockFD);
#else
  close(sockFD);
#endif
```

Figure 10 - Example Network CLIENT Program Using Gabriel Connect API

Gabriel Connect Secure DNS API

The integration approaches we have already discussed (Gabriel Secure Gateway Services, and Gabriel Connect Secure API) will both function when Gabriel is implemented at either the network or application layer. This next integration approach using the Gabriel Connect Secure DNS API will only function when Gabriel Instant Secure Connect software is implemented at the network layer on the device that is initiating the secure connection.

Gabriel Instant Secure Connect software must be on both sides of the connection for the communication to be secured. Figure 11 illustrates the DNS capture on the initiator side of the communications. The application that will be secured by Gabriel Connect Technology is modified to perform its communication using a private and secure domain name instead of a legacy domain name (acme.*scom* instead of acme.com). Because the Gabriel Connect Technology is implemented at the network layer, it intercepts all DNS requests made by applications on the device. When Gabriel intercepts the DNS request, it examines to see if it is a secure request. If it is a secure request (.SCOM), then Gabriel Connect Technology processes the request through the Secure DNS. The Secure DNS communication is done over secured channels where it examines the request against policy to determine if the communication is allowed. If the communication is allowed, it puts a secure channel in place to the target and returns a secure IP address. If the communication is not allowed, the Secure DNS returns name not found (0.0.0.0). If the DNS request is a legacy request, it is processed through the legacy DNS and the response is returned to the client.

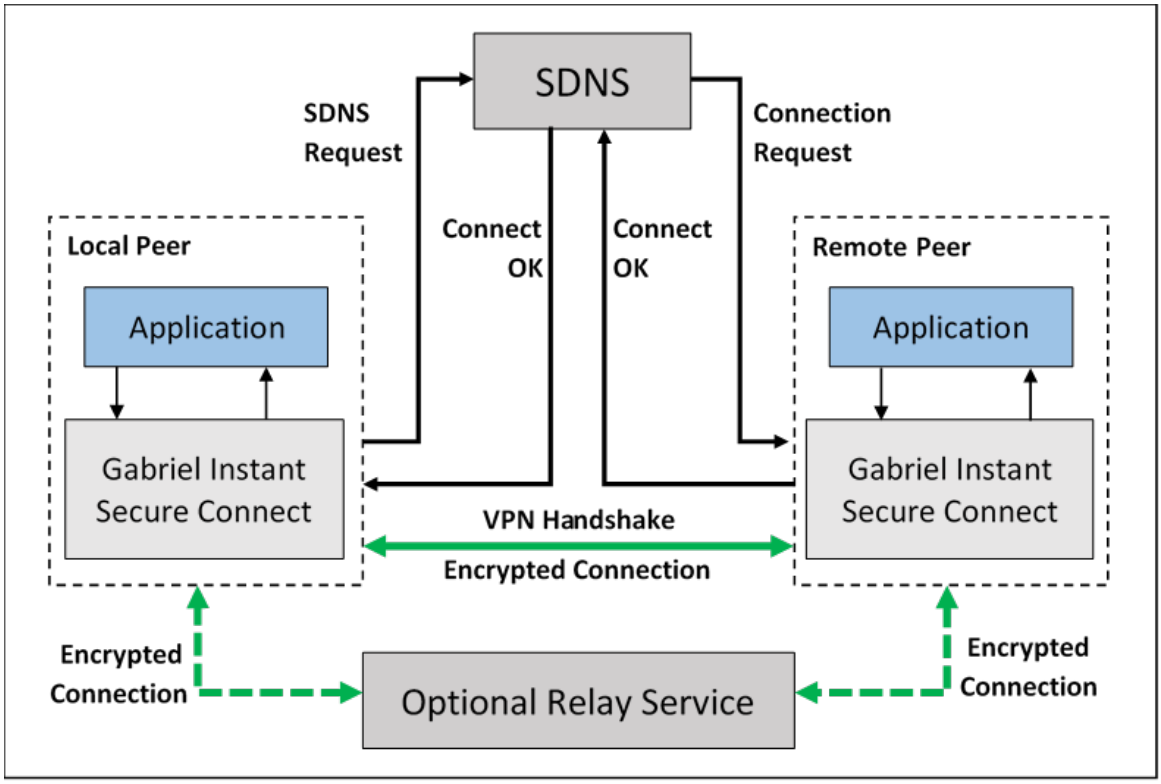

Figure 11 – Gabriel Connect Secure DNS API

Figure 11 illustrates both sides of the Gabriel Secure DNS function for two devices talking to each other via Gabriel Connect Technology. It represents establishing the VPN with a remote peer before responding to the Secure DNS request with a secure IP address.

There are several important attributes for setting up the secure connection:

(1) The Gabriel Instant Secure Connect Software needs to be installed on both the local and remote peer. Gabriel needs to be on the local and remote peer for all integration approaches (Gabriel Secure Gateway Services, Gabriel Connect API, and Gabriel Connect Secure DNS API).

(2) The functionality in the Gabriel Instant Secure Connect Software is not limited to connecting devices using *public* IP addresses. Gabriel Connect Technology protects communications via a peer-to-peer VPN even when both devices are behind Network Address Translation (NAT) and not directly connected to the Internet. The encryption channel is always end-to-end, meaning that the encryption starts on the local peer, and the data is not decrypted until it reaches the remote peer. Data can be sent directly to the remote peer (in some cases using UDP NAT traversal depending on the network topology).

If a direct connection is not possible between the two peers, Gabriel Instant Secure Connect software provides a Relay Service to facilitate communications. Data is routed through the relay service but stays encrypted until it reaches the remote peer. Note: *The Relay Service merely routes the encrypted data packets and therefore never stores or is accessible by any 3rd party.*

(3) Gabriel Instant Secure Connect Software encapsulates the application data in either UDP or TCP packets. Encapsulation in UDP is the preferred option. TCP packets will be used when no other option is available.

The following three Tables summarize the application integration approaches for Gabriel Connect Technology that have been previously discussed. Table 1 shows the implementation level of Gabriel Instant Secure Connect Software available by operating system. Gabriel Instant Secure Connect Software can be run at either the network or application layer on all operating systems except for locked down Android and iOS mobile devices. OEMs manufacturing devices with Android Operating System can deploy Gabriel Connect Technology at the network layer. In addition to supporting Linux on Intel platforms, Gabriel Instant Secure Connect Software also supports embedded Linux on devices with ARM processor.

Table 1 – The Implementation Layer of Gabriel Connect Technology Available for Each Operating System

	TUN (Network Level)	VTUN (Application Level)
iOS		X
Android		X
Windows	X	X
Linux Intel	X	X
Mac OSX	X	X
Android Custom (rooted)	X	X
Linux ARM (embedded)	X	X

Table 2 illustrates the Gabriel Connect Technology implementation layer required to support each type of integration into your application. All integration approaches work on all operating systems, except that the Gabriel Connect Secure DNS API approach does not work on locked down mobile devices. In this instance, Gabriel Connect Technology must run at the VTUN (application) layer.

Table 2 – Implementation Layers Supported for Each Integration Method

	TUN	VTUN
GSGS	X	X
App API	X	X
Secure DNS API	X	

Table 3 illustrates the full interoperability of the Gabriel Instant Secure Connect software integration approaches. The client and server side of your application can use a different integration approach with Gabriel Instant Secure Connect software and interoperate with each other. An example is an application that securely accesses a surveillance camera could be secured via Gabriel Connect API, to avoid needing to load custom software on the camera, the responder (server) side, could be integrated using GSGS. Another example would be a camera application on a laptop using the Gabriel Connect Secure DNS API to access a camera secured by GSGS. Or if Gabriel Connect Technology is integrated directly onto the camera, the responder side could be secured via the Gabriel Connect Secure DNS API, and the client application could be secured via GSGS, Gabriel Connect API, or the Gabriel Secure Connect DNS API. This inoperability provides you complete control to use the most appropriate method to integrate Gabriel Instant Secure Connect software for each device participating in the Gabriel Security Platform.

Table 3 – Gabriel Connect Technology on Integration Approaches All Interoperate

Initiator (Client)	Responder (Server)
GSGS	GSGS
GSGS	App API
GSGS	Secure DNS API
App API	GSGS
App API	App API
App API	Secure DNS API
Secure DNS API	GSGS
Secure DNS API	App API
Secure DNS API	Secure DNS API

Summary

As data breaches continue to increase in frequency, size and impact damaging customers globally, there is a new security approach. The Gabriel Security Platform provides new and robust technology for protecting communications of 3rd party applications. Building on the Gabriel Connect Technology, Gabriel Secure Communications and Gabriel Collaboration Suite; these technologies enable customers private secure communications and data protection through private secure domains and Internet addresses. The Gabriel Security Platform offers 3rd party application developers three methods to utilize Gabriel's protection. They include, Gabriel Secure Gateway Services, Gabriel Connect API and Gabriel Connect Secure DNS API.

What was originally developed exclusively for the intelligence community to address the vulnerabilities of the Internet, Gabriel Instant Secure Connect software now provides private secure domains and Internet addresses. The Gabriel Security Platform provides access to secure services such as Real-Time Communications, File, Network and Gateway. ALL data is encrypted and is transported through encrypted tunnels using the existing internet infrastructure. The Gabriel Secure network CANNOT be accessed through normal legacy DNS lookup. The power and flexibility of Gabriel Security Platform is available now.

"Hacker's can't hack what they can't see!"

To get started, contact us at info@virnetx.com.