White Paper

Cisco pxGrid: Automate Multi-Platform Communications through a Unified Architecture

What You Will Learn

IT environments are drowning in a deluge of network and security information, produced by an array of security systems that do not allow multi-platform communication. This complicates security operations because it increases the amount of time and manual effort an operator needs to stitch together information from disparate systems. Traditional APIs – which facilitate integrations between one platform and another – are too limited, insecure, and platform-specific to provide a practical solution. Cisco Platform Exchange Grid (pxGrid) provides a way for diverse multivendor platforms to exchange information securely, in a tightly controlled, bi-directional manner. By automating intersystem communication in real time, Cisco pxGrid eliminates reliance on platform-specific APIs.

This paper discusses:

- The operational challenges to maintaining a secure IT environment when dealing with multivendor security systems that don't communicate or interoperate
- The ways in which Cisco pxGrid enables immediate, automated intersystem communications
- The Cisco pxGrid architecture and operations
- How developers can integrate their platforms with pxGrid to enable context exchange between their platforms and security products, as well as other pxGrid-enabled development partners
- How violations in an organization's security policy can trigger an Automated Threat Response, such as reducing the risks of spreading malware throughout the rest of the network or being non-compliant in meeting an organization's compliance standards.
- How ecosystem partners can publish IOT asset information into ISE for device classification. This enables an organization to define secure network access for IOT devices, based on their organization's security policy.
- The technical differences between pxGrid 1.0 and pxGrid 2.0

A Growing Security and Operational Challenge

To keep the IT environment secure and running smoothly, businesses leverage a wide range of security platforms from different vendors. These can include identity and access management (IAM) platforms, security and event management (SIEM) systems, policy platforms, and threat defense systems. All of these tools are critical to protecting businesses and safeguarding their operations, but the resulting inability for multi-platform communication poses a huge operational challenge by creating fragmented silos of security information.

Swiveling from one tool to another adds a lot of complexity – and cost – to security operations. It also reduces the overall efficacy of IT security because it requires a greater amount of time and manual effort to obtain the necessary information from each of these tools in order to take the appropriate security actions. That's time businesses can't afford when an advanced attack is seeking to burrow deeper into the environment or exfiltrate sensitive data.

Historically, platform-specific API's have ameliorated this problem by facilitating communication between one system and another. But in modern IT environments, this approach doesn't scale. Today, the number of platforms with information to share amongst other platforms is simply too great; businesses can't realistically implement single-purpose APIs in order to link each tool to every other tool in its IT environment.

In the first place, maintaining dozens of single-purpose APIs and retesting all of them for minor software updates would be an unmanageable task. And even if this was feasible, businesses would quickly be overwhelmed with information. APIs work on a basic polling model, so having 20 systems sending repeated requests for information at the same time would be incredibly problematic from a performance and scalability perspective.

Additionally, APIs are traditionally insecure. They rely on relatively weak username-and-password authentication and do not offer authorization capabilities over how that data is used. Once an API opens a doorway into a system, other platforms can access that information and do what they want with it. This presents a significant security risk.

Therefore, businesses are left with a difficult security operations challenge. Wouldn't it be better if all the diverse tools in the IT environment could talk to each other? What if platform vendors could draw contextual information and capabilities from the other systems in the environment and give the operations staff everything they need to solve real world problems and respond to threats faster? That's exactly what Cisco pxGrid provides.

Introducing pxGrid

Cisco Platform Exchange Grid (pxGrid) is a framework based on a publish-subscribe-query architecture that uses Cisco Identity Services Engine (ISE) as its foundation for publishing session information for ecosystem partners to consume. Ecosystem partners will connect, register and subscribe to ISE topics such as Session Directory, which provides more meaningful information and context around the security event. These session attributes are shown in Fig.1

Figure 1. User Context Information

```
Session={ip=[192.168.1.15], Audit Session Id=0A0000010000000170001B0AB, UserName=jeppich,
ADUserDNSDomain=lab10.com, ADUserNetBIOSName=LAB10,
ADUserResolvedIdentities=jeppich@lab10.com, ADUserResolvedDNs=CN=John
Eppich,CN=Users,DC=lab10,DC=com, MacAddresses=[00:50:56:86:C9:92], State=STARTED,
ANCstatus=ANC_Quarantine, SecurityGroup=Quarantined_Systems, EndpointProfile=VMWare-
Device, NAS IP=192.168.1.3, NAS Port=GigabitEthernet1/0/11, RADIUSAVPairs=[ Acct-Session-
Id=0000002E], Posture Status=null, Posture Timestamp=, LastUpdateTime=Sat Jan 21 11:49:04 EST
2017, Session attributeName=Authorization_Profiles, Session attributeValue=Quarantined_Systems,
Providers=[None], EndpointCheckResult=none, IdentitySourceFirstPort=0, IdentitySourcePortStart=0,
```

In addition, ecosystem partners can take an Automated Threat Response, or trigger Adaptive Network Control (ANC) mitigation actions on the endpoint if the endpoint is deemed in violation of the organization's security policy. ANC mitigation actions include quarantining the endpoint, limiting or restricting network access, and assigning Cisco Security Group Tags (SGT).

With Cisco Platform Exchange Grid (pxGrid), inter-system communications can happen automatically, without manual intervention. Instead of relying on dozens of single-purpose APIs to share contextual information from platform to platform, all of the systems in the IT environment can be integrated with pxGrid to facilitate the flow of information.

More specifically, IT and security vendors can leverage pxGrid to share context bi-directionally with Cisco platforms and ecosystem partners, eliminating the need for platform-specific APIs. To this end, pxGrid enables multivendor, cross-platform system collaboration among multiple parts of the IT infrastructure, including security monitoring and detection systems, network-policy platforms, asset and configuration management, identity and access management platforms, and virtually any other IT operations platform.

Additionally, pxGrid is fully secured and customizable. Through pxGrid's publish-subscribe-query architecture, partners can select which information they want to publish (share) and subscribe to relevant information from other platforms on the grid. This architecture provides the scalability that single-purpose APIs do not. Furthermore, pxGrid enables ecosystem partner platforms to execute actions with Cisco network infrastructure. This allows security operations teams to gather relevant threat information faster and take responsive action immediately.

pxGrid can work with a variety of data types to suit a wide range of use cases because it is information model and data format agnostic. Ultimately, these context-sharing and network capabilities make it possible for IT infrastructure providers to address more use cases, undertake their functions more effectively, and extend their reach deeper into the network infrastructure.

Capabilities and Benefits

pxGrid provides:

- **A single framework for multiple systems to share context:** With pxGrid, any partner platform can connect with other platforms in the IT environment (including both Cisco and third-party platforms that use pxGrid) to share relevant context. This can include real-time operation status, historical event information, operational telemetry, usage statistics, or any other information that an IT platform may need to share or consume.
- **Total control over what context is shared, and with which platforms:** Because pxGrid is fully customizable, partners can specify the contextual information they want to 'publish' and control which partner platforms that information gets shared with.
- **Bidirectional, many-to-many context sharing:** pxGrid enables platforms to both share and consume context with other connected platforms, with all communication orchestrated and secured centrally by the grid and delivered to each platform in its native data format.
- **Scalable, simultaneous connectivity with multiple platforms:** pxGrid enables platforms to publish only the context data relevant to partner platforms. Numerous 'topics' can be customized for a variety of partner platforms, yet always shared through the same reusable pxGrid framework. Furthermore, by sharing only relevant data, platforms that are both publishing and subscribing can easily scale their sharing by eliminating irrelevant data.
- **Integration with Cisco platforms:** pxGrid provides a unified method of publishing and subscribing to relevant context with a growing number of Cisco platforms that use pxGrid for third-party integrations.
- **Automated network threat response:** With pxGrid's network instrumentation, pxGrid-enabled platforms can take network threat-response actions by simply making a call to pxGrid – even if the platform making the call itself has no network topology or control awareness.
- **Ability to share ISE context with IT infrastructure:** ISE provides user/identity, device, and network context with ecosystem partner products and platforms. This allows customers to have a detailed understanding of the 'who-what-where' associated with security events, and

to implement this context in formulating policies.

- **Functionality for classifying IOT devices within an organization:** Ecosystem platforms can publish IOT asset information into ISE for device classification. This enables customers to define network access for IOT devices, based on their organization's security policy.

A Superior Mechanism for Cross-Platform Communication

pxGrid provides a much better way to share information than conventional APIs for several reasons:

First, its ability to implement "many to many" communication among diverse network platforms means that it's innately scalable – more so than the basic polling models that operate conventional APIs.

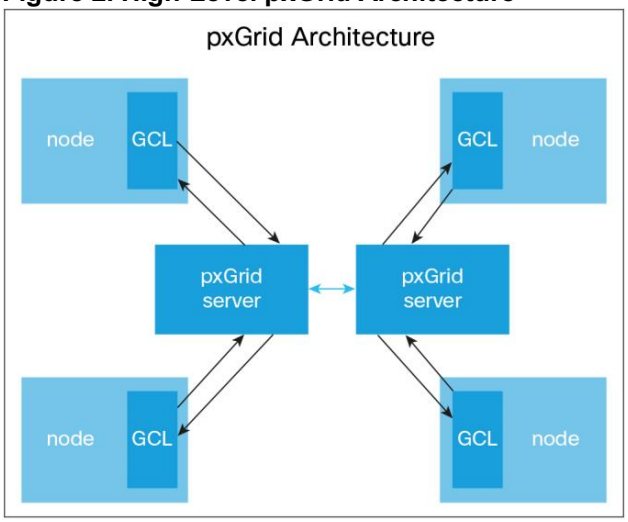
Second, it allows for improved, highly detailed customization. Unlike APIs – which don't allow for much customization in *how* or *what* the systems are communicating – pxGrid enables each connected system to consume only the specific information it needs from other systems on the grid and share only the information that's relevant to the other systems they're communicating with.

Finally, pxGrid provides stronger security and control. It preserves the integrity of each system's data far more effectively than APIs by providing tightly-controlled access, authentication, and authorization for each system on the grid. Just as users can be authorized on the network to access some resources but not others, IT vendors can authorize systems connected to the pxGrid to access the information they need from their platform, but nothing else.

The pxGrid Architecture: How pxGrid Works

At the heart of pxGrid is a controller and participating nodes, as shown in **Figure 2**. In a typical customer deployment, nodes reside on separate hosts but within the same network. However, they may also be federated across multiple customer environments. Each node goes through authentication, registration, and authorization to communicate over pxGrid and can establish itself as a provider or consumer of topics for sharing information. The pxGrid server provides message routing and control based on the contextual data being shared and a participating node's authorization. It supports queries, notifications, and bulk downloads of context data. Depending on context, pxGrid can establish an out-of-band channel for a bulk download.

Figure 2. High-Level pxGrid Architecture

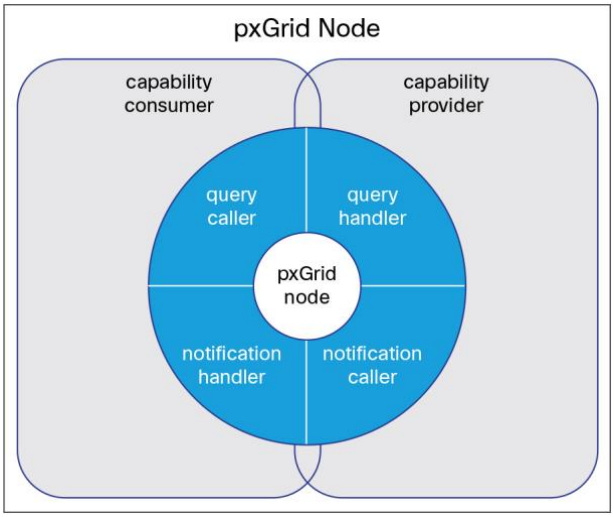


Nodes don't communicate directly with pxGrid. Instead, nodes make programmatic calls to the Grid Client Library (GCL), which in turn connects and communicates with pxGrid. Depending on the use case, one deployment may have only a few nodes, while others may have thousands. pxGrid is designed to scale upward.

Bidirectional Communications

pxGrid enables bidirectional communication between pxGrid nodes. Nodes can be both providers and consumers of capabilities, assuming they are authorized by the pxGrid controller for both functions (**Figure 3**). As the provider of a capability, a node handles queries, generates notifications, or both. As the consumer of a capability, a node initiates queries, receives notifications, or both. (Note: "Capability" here refers to information channels or topics for sharing contextual information. pxGrid uses information models to define the context data, interfaces, or operations for sharing contextual information. For instance, an Identity model could include a Session Directory capability consisting of interfaces for consuming information related to session logins in the network.

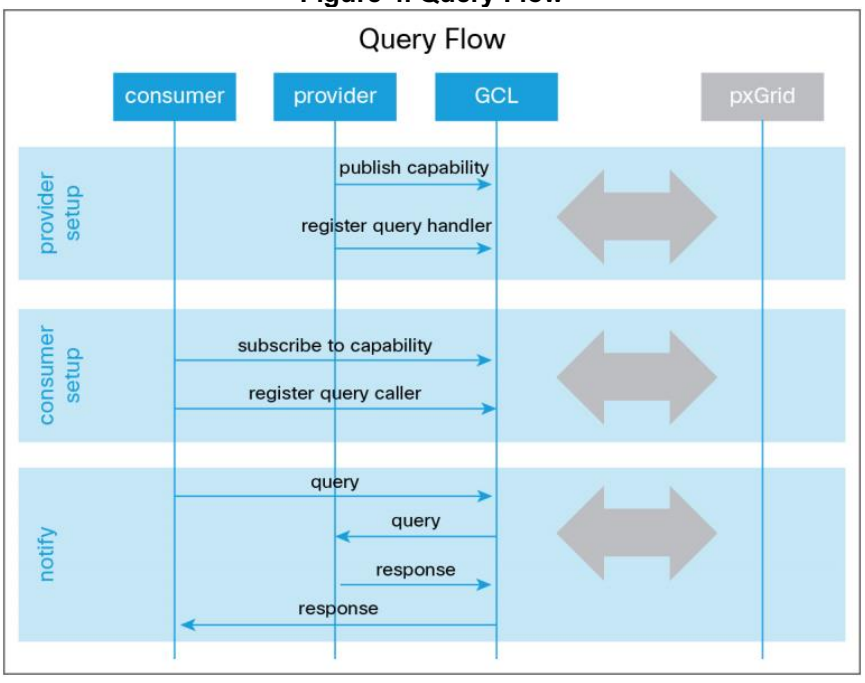
Figure 3. Bidirectional Communications in a Node



Queries

A query is a synchronous call initiated by a consumer and serviced by a provider (**Figure 4**). Using the Grid Client Library (GCL), the consumer utilizes a query caller to initiate the communication. The provider implements a query handler to programmatically process requests and generate responses. pxGrid passes the request through the consumer GCL and into the provider GCL, likely on another node in the network. Using custom code implemented by the developer in a query handler, the provider generates a response that pxGrid then sends back to the consumer. The consumer waits until the response is received.

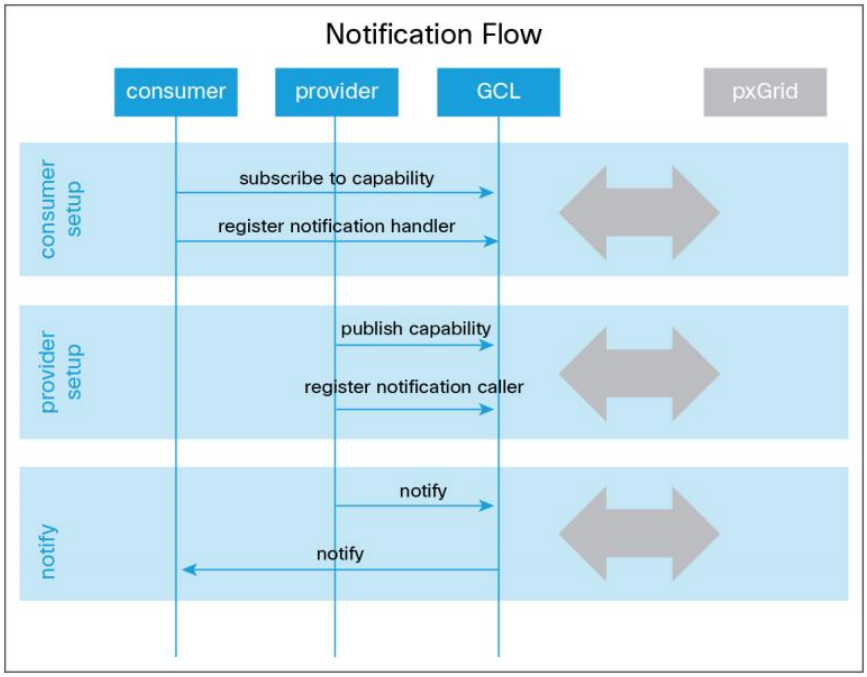
Figure 4. Query Flow



Notifications

A notification is an asynchronous message generated by a provider and received by a consumer. The consumer must first register interest in an information topic. Using messaging terminology, consumers subscribe to a topic and providers publish to the topic. The GCL handles communication with pxGrid, so the consumers and providers can focus on writing code to consume and provide the information. Consumers do not wait for information as they do in a query flow. The GCL uses a separate threat to invoke a notification handler supplied by the consumer. **Figure 5** details the flow.

Figure 5: Notification



pxGrid Dynamic Topics: context-sharing with ecosystem partners

The basic architecture of pxGrid comprises a central pxGrid controller with multiple systems (nodes) connected to a client library. The pxGrid controller is analogous to a switchboard operator. It communicates with each node's agent to allow each connected system to share and consume authorized context information with other nodes.

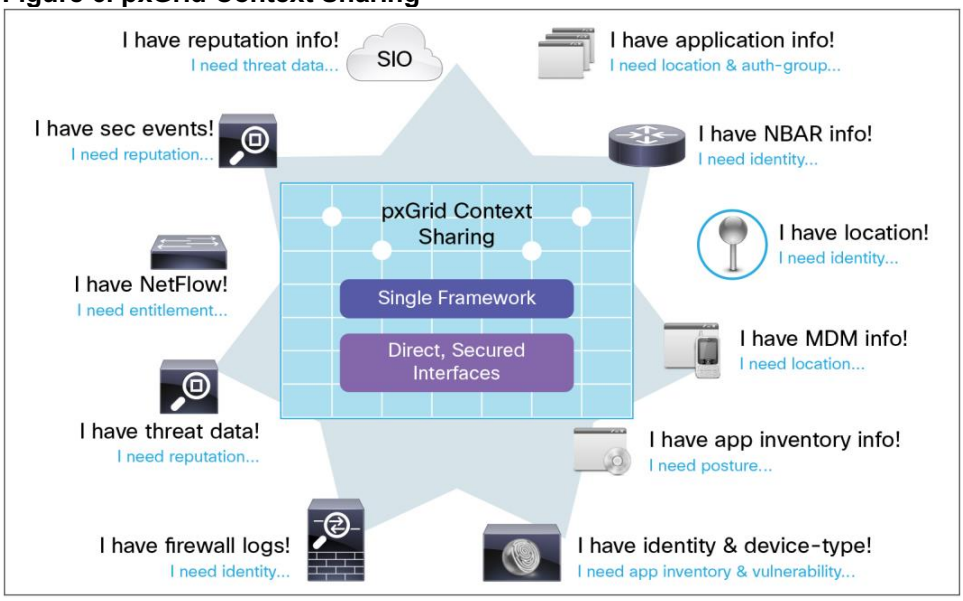
Figure 6 (page 7) shows a basic overview of pxGrid in action. Here, you can see the diversity of platforms involved in security operations for a typical business. Each has an important role to play, but each needs information from other systems to do its job effectively.

As a simple example, imagine a bring-your-own-device (BYOD) use case, where a business wants to implement different levels of access privileges based on the devices its employees are using and the locations of the users. Doing this requires information from three components: (1) an identity and access management (IAM) platform to provide application permissions, (2) a mobile device management (MDM) platform to check the registration status of the device, and (3) a security information and event management (SIEM) platform to assess and inform about the threat risk associated with an employee.

Here's how pxGrid implements inter-platform communication to simplify this process:

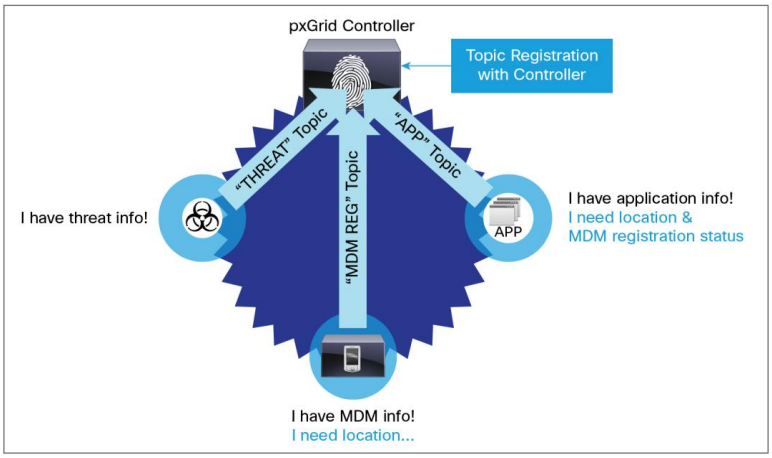
First, each platform independently authenticates with pxGrid. pxGrid also handles the authorizations, controlling which platforms can publish, which can subscribe, and which can query, as well as which specific context information may be shared with other platforms on the grid. This is done through a publish/subscribe ("pub/sub") model. Each connected platform publishes specific "topics" – sets of data – to the grid, and/or subscribes to topics that are relevant to the use cases it handles. Each topic is registered in the pxGrid topic directory so that it can be catalogued and found by platforms interested in that topic. Platforms may subscribe to all real-time updates to a topic, query for specific attributes on demand, or do bulk downloads of information from that topic.

Figure 6. pxGrid Context Sharing



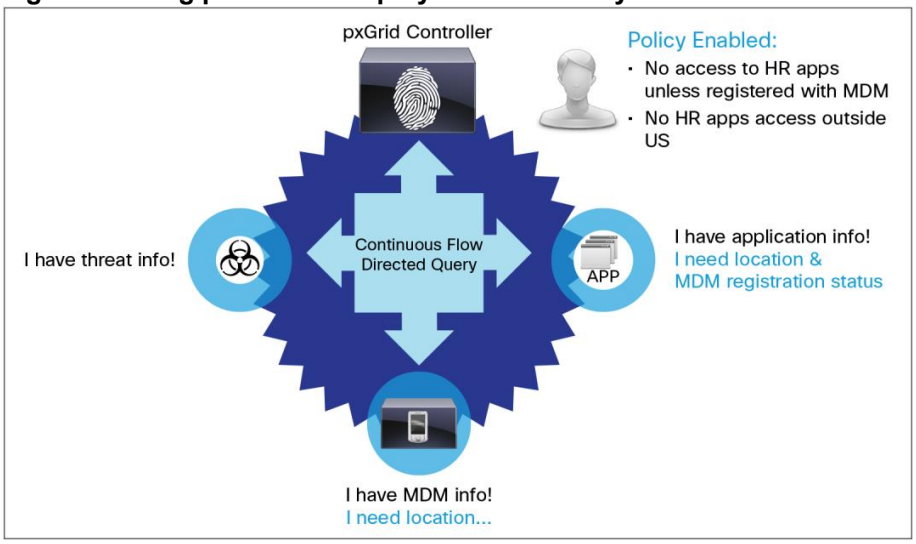
So, in the BYOD case, the IAM platform publishes an Applications topic, providing application permissions for users. The MDM platform publishes an MDM Registration topic, providing information about the registration status of a given device. The SIEM publishes an Employee Threat Risk topic, providing the threat risk associated with a given user. (See **Figure 7**, page 8.)

Figure 7. pxGrid Context Sharing for BYOD and Threat Defense



At the same time, each platform on the grid subscribes to the topics that are relevant to its specific operations. For example, if the IAM platform needs information on MDM Registration and Employee Threat Risk, it can subscribe to these topics. Likewise, the MDM platform can subscribe to the Application topic. Meanwhile, the SIEM publishes Employee Threat Risk information for consumption by both the MDM and IAM platforms. Once this framework is established, each platform can continually pull or ad-hoc query the information it needs from other platforms on the grid, in the appropriate data format to fulfill its role in allowing or denying access to BYOD users. (See **Figure 8)**.

Figure 8. Using pxGrid to Simplify BYOD Security

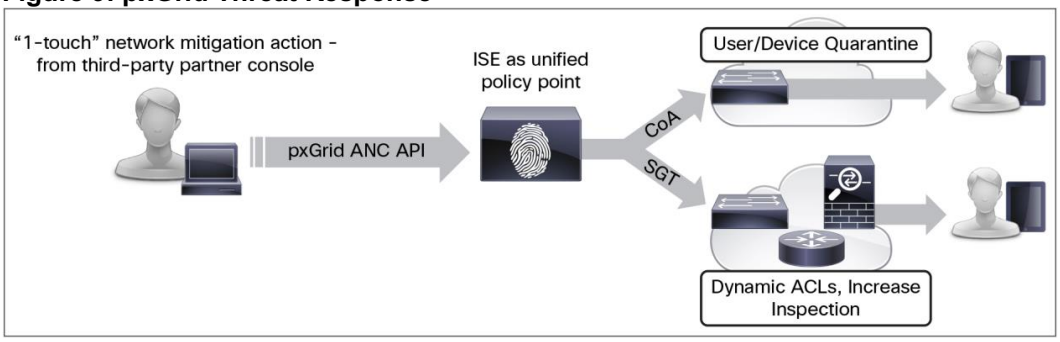


Automated Network Threat Response

pxGrid also provides pxGrid-enabled platforms with the capability to take network threat-response actions on users or endpoints directly from the partner platform. For example, in a SIEM platform an operator may quarantine users or devices or take investigative actions by rerouting traffic. With pxGrid, all this can be done from the SIEM console, using the same pxGrid framework used for sharing and consuming context. In this scenario, the system calls a threat-response API within the pxGrid framework. It can then use threat-response capabilities that are already instrumented in platforms like the Cisco Identity Services Engine to carry out the response action. With this architecture, pxGrid-enabled platforms don't need to understand how to take a response action. They just make a call through pxGrid to a platform, like the Identity Services Engine, that understands the network and has the instrumentation to carry out the requested action.

Figure 9 outlines the Adaptive Network Control (ANC) capabilities on the Identity Services Engine, as invoked by a pxGrid-enabled platform making a call to the threat-response API. The engine carries out a change of authorization and issues a security group tag.

Figure 9: pxGrid Threat Response



Cisco pxGrid Context-In: Ecosystem publishes context information for ISE to consume

Internet of Things (IOT) devices are considered 'anything' connected to the 'network' at 'any time.' Manufacturing tools and medical devices, such as infusion pumps and MRI machines, are common examples in a growing network of IOT infrastructure. With the proliferation of IOT devices, many of which are unpatched and unmonitored, organizations face an increasing security problem. As these new technologies connect to the enterprise's network infrastructure, the biggest challenge is classifying these devices and providing them the appropriate network access based on the organization's security policy for IOT devices.

For example, medical devices require specific access to areas of the medical facility and to the appropriate medical personnel. Are these devices allowed to connect via wired, wireless, or VPN? These are some considerations that define an IOT Security Policy.

As new technologies are being developed, there are safeguards in place to protect them from vulnerabilities that may exist in their code and from possible malware attacks. If these devices get compromised from changing the new technology vendors attributes defined in an ISE profiled policy, they will not be allowed network access.

How it Works

Cisco pxGrid Context-In augments device classification by using Cisco Identity Services Engine as the network enforcement and network authorization point. Classified endpoint access is determined when technology partners publish their IOT asset information to the pxGrid endpoint Asset Topic. The ISE pxGrid node subscribes to this topic. (This process is shown in **Figure 10.)** The IOT Asset ISE profiling policy is defined by the IOT Assets as in **Figure 11.** The logical profile consists of the defined ISE profiling policy (**see Figure 12**) to be used in an ISE Authorization policy for classified network access (**see Figure 13**)

in pxGrid and defining these assets in an ISE profiling policy. The ISE pxGrid node becomes the subscriber to this topic and enforces and authorizes the technology partners connection. The technology partner is also considered a pxGrid client.

Figure 10: ISE ecosystem partner publishes attributes to Endpoint Asset Topic

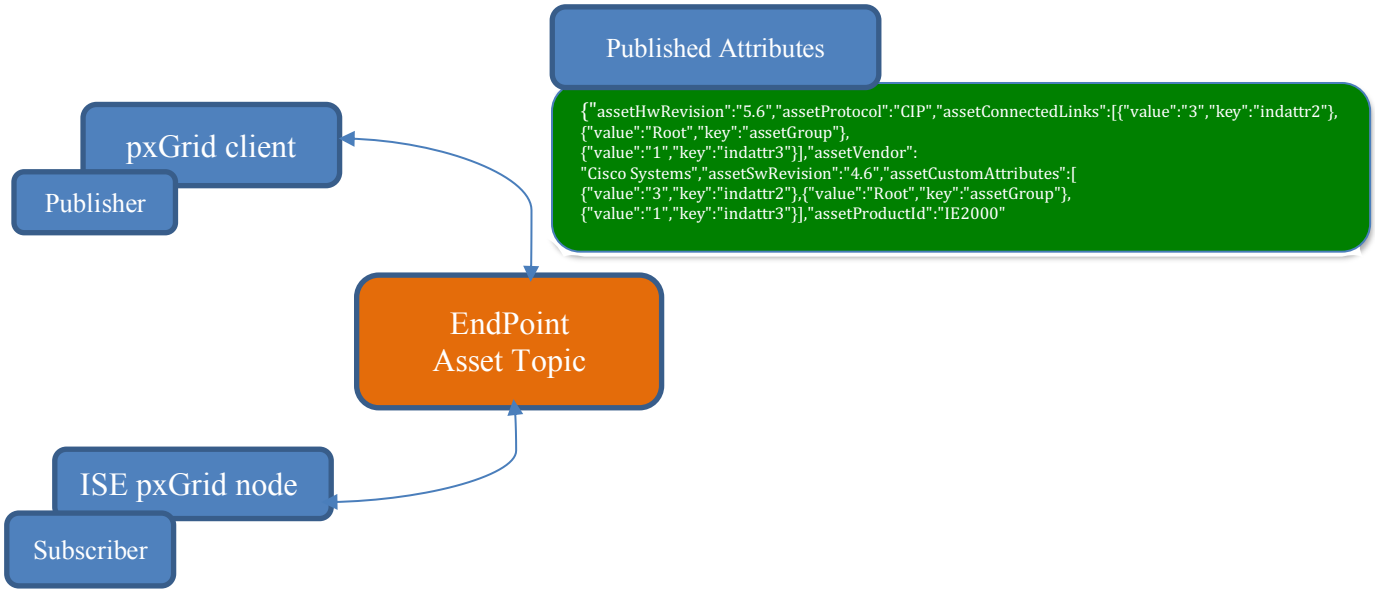


Figure 11: The IOT Asset Profiling Policy is created

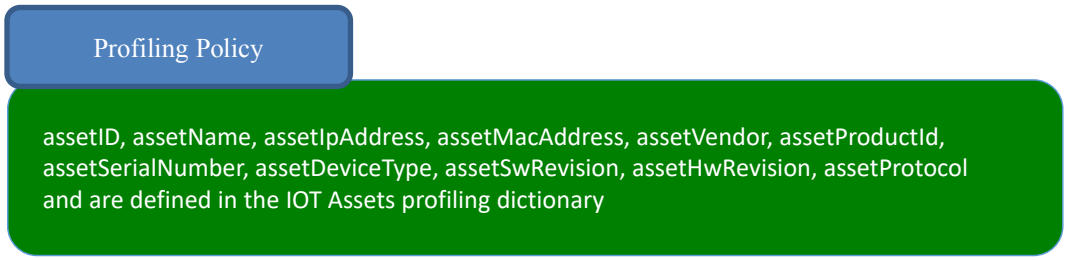


Figure 12: The Profiling Policy is assigned to a Logical Profile

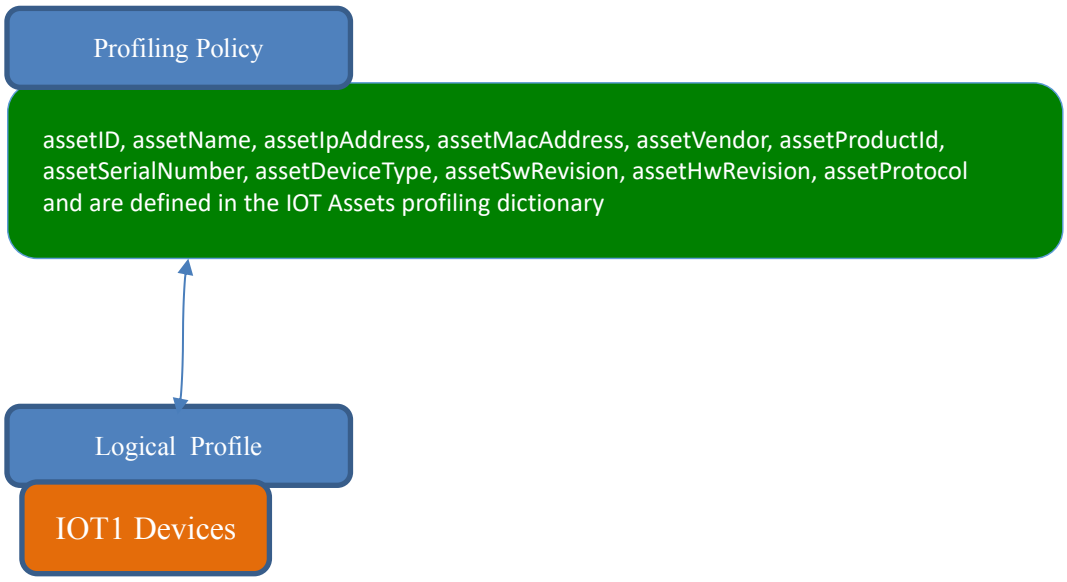
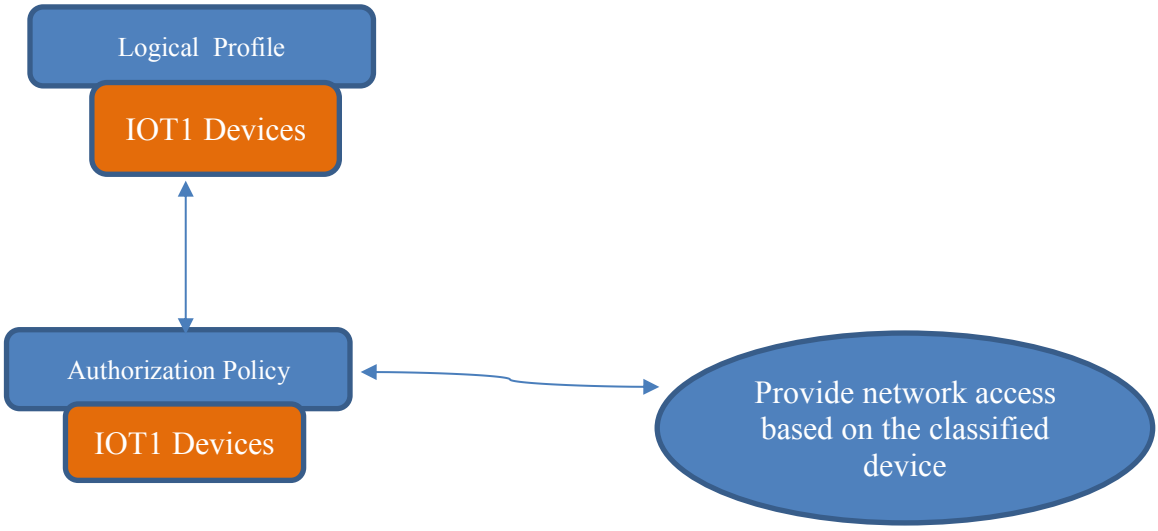


Figure 13: The Logical Profile is Assigned to an ISE Authorization Policy



pxGrid 1.0: How XMPP is Used in the Grid

pxGrid 1.0 uses the Extensible Messaging and Presence Protocol (XMPP) as the foundation protocol for exchanging security data between systems across the grid. Based on XML, XMPP uses a decentralized client-server architecture, where clients safely connect to servers, and the messages between the clients are routed through the XMPP servers deployed within the cluster. XMPP has been used extensively for publish-subscribe systems in a wide range of file transfer, video, Internet of Things, and other collaboration and social networking applications.

XMPP offers several important advantages for exchanging security data in pxGrid. It provides:

- **An open and standards-based communication framework**, with a decentralized and federated architecture that has no single point of failure
- **Strong security** with support for highly secure domain segregation and federation using Simple Authentication and Security Layer (SASL) and Transport Layer Security (TLS) mechanisms
- **Real-time information exchange and event management** using pub-sub notifications
- **Flexibility and extensibility** as an XML-based framework that can easily adapt to new use cases and support custom functionality
- **Support for multiple information-exchange mechanisms** between participating clients
- **Support for both on-demand and directed queries between clients**, communicated through the XMPP server
- **Support for out-of-band file transfers and direct communication** between participating clients
- **Bidirectional communication**, eliminating the need for firewall tunneling or opening up a new connection in each direction between client and server
- **Scalability**, with support for cluster mode deployments with fan-out message routing, and peer-to-peer communications
- **Easy deployment** through a straightforward XMPP framework for nodes to detect the presence, availability, and service capabilities of other participating nodes in the system.

To simplify the integration with diverse partner platforms, pxGrid defines an infrastructure protocol that hides the nuances of XMPP data plane protocol and makes information-sharing models extensible with simple and intuitive APIs. pxGrid nodes connect to the grid using this pxGrid protocol, which uses the XMPP transport protocol and introduces an application-layer protocol that uses XML and XMPP extensions. Partners providing platforms for the grid can extend the pxGrid protocol infrastructure model and define capability-specific models and schemas. And they can take advantage of a clean separation between infrastructure and the capabilities that can run on that infrastructure.

pxGrid 2.0: How WebSockets and Representational State Transfer (REST) API Model are Used on the Grid

pxGrid 2.0 uses both WebSockets and REST over the Simple Text Oriented Messaging Protocol (STOMP) for the development platform. Both WebSockets and REST are heavily supported and provide industry- standards for application-to-application communications. In addition, WebSockets provides quick and scalable bi-directional data transfer. Using WebSockets, the client can transfer as much data as it through a series of frames containing the data or payload, which is otherwise known as frame-based messaging. Using this frame-based messaging system helps to reduce the amount of non-payload data that is transferred, leading to significant reductions in latency. pxGrid will use port 8910 on ISE for pxGrid-related REST and WebSockets communication and the sever will play a WebSockets ping-pong game to detect network failures, as well as when pxGrid clients are offline, slow, or faulty. Messages over WebSockets will be sent and received in binary format and conform to the Simple Text Oriented Messaging Protocol (STOMP).

The following STOMP commands are supported in pxGrid:

- CONNECT
- DISCONNECT
- SUBSCRIBE
- UNSUBSCRIBE
- SEND
- MESSAGE
- ERROR

pxGrid consumers will typically implement SUBSCRIBE while providers will implement SEND. (**Note!** Consumers can and should subscribe to multiple pxGrid services using the same connection.)

WebSockets and REST API offer the following advantages over XMPP:

- **"JAVA" and "C" SDKs are no longer required.** The development platform is no longer limited to SDKs. Python, JAVA, Golang and other development platforms using WebSockets and STOMP are supported.
- **More scalable, there is no limitation on the number of pxGrid nodes**. Using pxGrid 1.0, there can be only 2 nodes, in an Active Standby configuration. pxGrid 2.0 now supports Active-Active Fail-Over

Transitioning to pxGrid 2.0

pxGrid 1.0 is upwards compatible with pxGrid 2.0, in that there will be a bridging component to port over the existing pxGrid 1.0 code. Development will still be performed using the SDK. Session filtering will not work in pxGrid 2.0, but in everything else there will be feature parity.

pxGrid 2.0 is not backwards compatible with pxGrid 1.0, if the developer desires backwards compatibility with existing versions of ISE, they must also develop using the SDK available on Cisco Devnet. pxGrid 2.0 is officially supported in ISE 2.4 and higher.

Figure 14 enumerates the differences between Cisco pxGrid 1.0 and 2.0.

Figure 14: The Differences between pxGrid 1.0 and 2.0

	pxGrid 1.0	pxGrid 2.0
Consumer	• Requires SDK • Java, C GCL client • XMPP queries • XMPP pubsub subscriber	• No SDK • Any language (Java, C, Python, C#...) • REST API calls • STOMP/WebSocket subscriber
Provider	• Requires SDK • Java, C GCL client • XMPP Discovery/Authz API • XMPP authentication • XMPP query handlers • XMPP pubsub publisher	• No SDK • Any library, any language • REST Discovery/Authz API • Webapp authentication provider • REST API handlers • STOMP/WebSocket publisher
Pubsub	• XMPP pubsub • XML parsing • Single instance • Dynamic topics support	• WebSockets • Data is opaque • Horizontal scaling with multiple active instances • Dynamic topics support
Control Plane	• XMPP Discovery, Authc, Authz • XMPP component • Clients require SDK	• REST + STOMP • Written as a Webapp • No SDK required
Topics	• ISE topics are published and available both on pxGrid 1.0 & pxGrid 2.0 • Dynamic topics created on pxGrid 1.0 are available for pxGrid 1.0 clients only • pxGrid 1.0 clients can subscribe to ISE pxGrid 1.0 topics • pxGrid 2.0 clients can subscribe to ISE pxGrid 1.0 or ISE pxGrid 2.0 topics	• Topics created on pxGrid 2.0 are available to pxGrid 2.0 clients only

Enabling Secure Information Exchange

pxGrid enables secure information exchange between nodes on the grid in three ways. First, it goes beyond the basic username and password to provide strong public key infrastructure (PKI) and certificate-based authentication for every platform that is sharing or consuming information on the grid.

Second, pxGrid provides a detailed authorization framework to control what each connected platform can and can't do on the grid. For example, operators can specify which individual topics a platform can share or consume, authorize some nodes to subscribe to topics but not publish (or vice versa), or specify that certain platforms can no bulk downloads but others cannot.

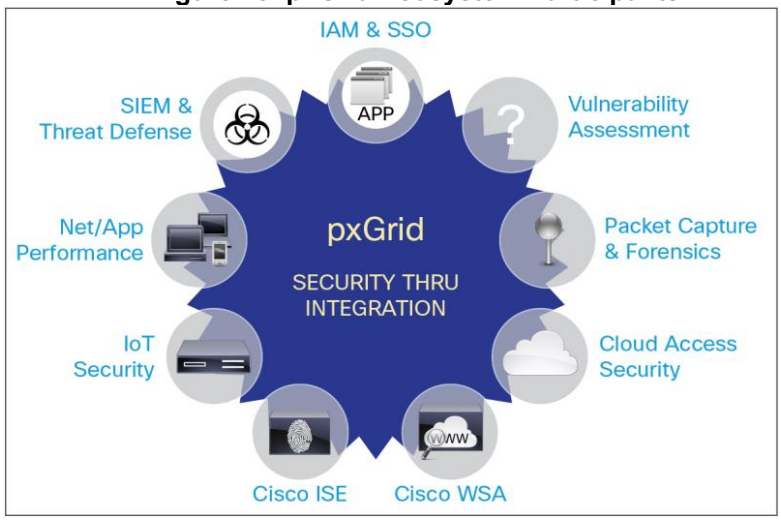
Finally, all data shared over pxGrid is encrypted. Communications are always private and protected from being intercepted by a "man in the middle" attack. Together, these tools allow for much stronger security and control in cross-platform communications that APIs do.

The pxGrid Ecosystem in Action

pxGrid can provide a powerful framework to allow cross-platform communication in IT environments with more security and scalability than anything that businesses have used in the past. But the grid is only as valuable as the platforms that take advantage of it. How much can you actually do with pxGrid today? In fact, quite a lot.

A large and growing number of IT and security vendors are already building pxGrid integration into their solutions. Current ecosystem partners include providers of industry-leading solutions for SIEM and threat defense, network ad application performance acceleration, cloud and IOT security, and many others (**Figure 15**).

Figure 15: pxGrid Ecosystem Participants



How to Integrate with pxGrid

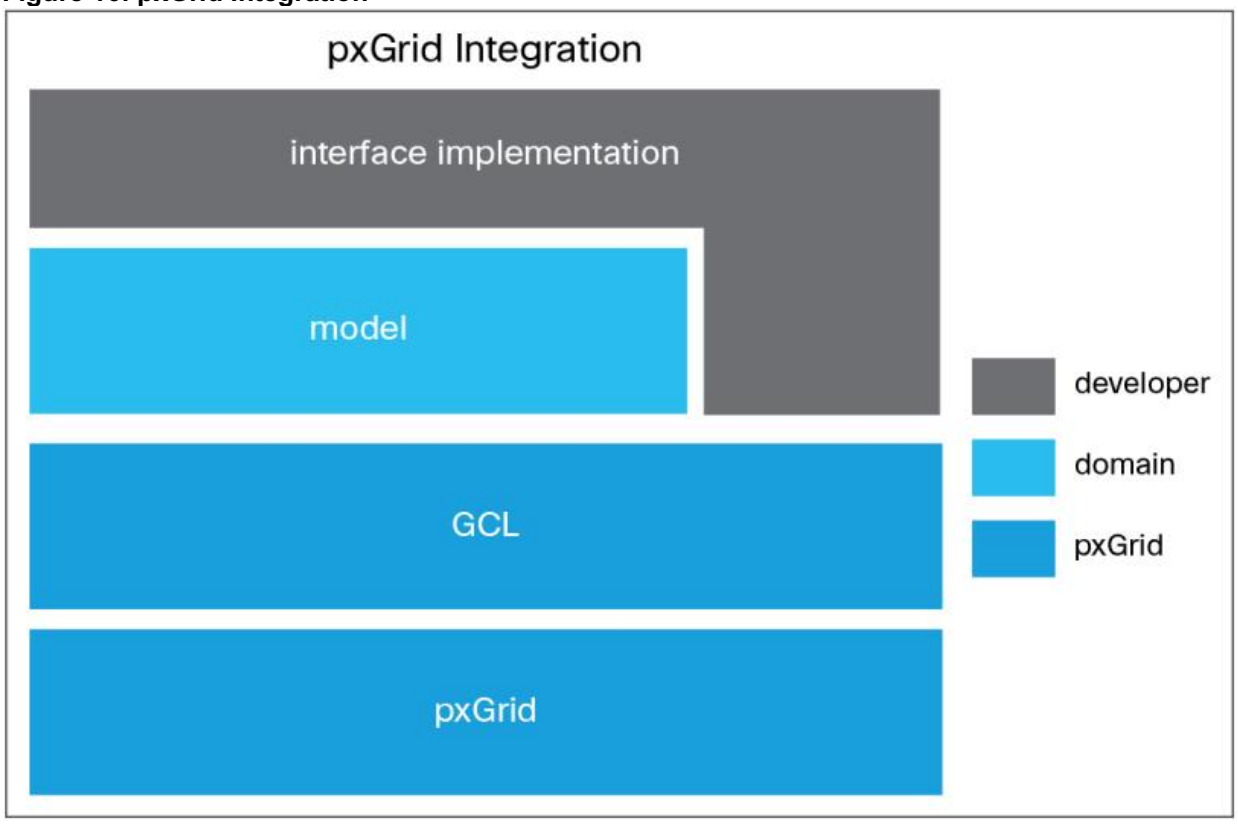
Cisco makes it easy for partners to integrate their solutions with pxGrid and capitalize on bidirectional communication with other security platforms to bring new capabilities and business benefits to their customers. We provide a complete technical overview and tutorial on Cisco DevNet.

pxGrid 1.0

For pxGrid 1.0 users, there is a full software development kit (SDK) with all the necessary Java and C client libraries for developers to integrate their nodes with the pxGrid 1.0 server. The SDK includes a full testing environment, as well as all the necessary instrumentation to quickly integrate with the grid and begin publishing and subscribing to topics and taking network actions.]

Figure 16 shows the high-level relationship between pxGrid, the GCL, the model for a particular domain, and custom code written by the developer. The model, as discussed earlier, consists of entities and interfaces common to the domain (security, for example). "Interface implementation" refers to the actual functionality for how a node will behave. The developer extends this model's interfaces and uses existing classes in the GCL to implement this functionality.

Figure 16: pxGrid Integration



To download the SDK, visit http://cisco.com/go/pxgrid. The SDK includes:

- pxGrid technical overview, tutorial, configuration and testing guide, and other relevant documents
- The GCL in Java and C that will be integrated with the platform connecting to pxGrid
- Identity Services Engine virtual machine and setup documentation that can be used as a pxGrid controller for testing pxGrid client implementations

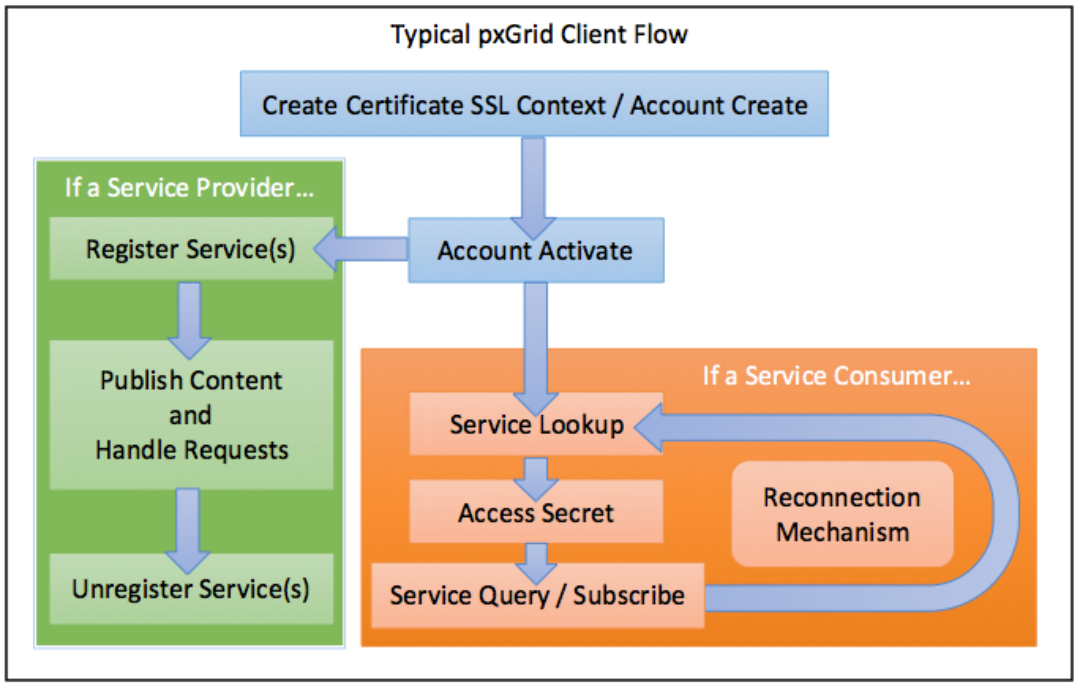
- Information on how to access the cloud-hosted pxGrid test bed to test your integration if you prefer to not install a local testing instance
- Sample pxGrid session output to test a system's capability to consume pxGrid data
- RADIUS session emulator to generate live user and device session data to test pxGrid-integration,

pxGrid 2.0

For pxGrid 2.0 users, there is no development kit to download. You can refer to: https://github.com/cisco-pxgrid/pxgrid-rest-ws/wiki for a complete list Cisco WebSockets and REST API services and topics. Before you do, it is best to review the Testing and Configuration Guide for Cisco Platform Exchange Grid (pxGrid) 2.0 from https://cisco.com/go/pxgrid. This provides details on pxGrid 2.0 fundamentals and sample code, this also includes pxGrid Context-in.

Figure 17 below represents the typical pxGrid 2.0 client flow:

Figure 17: Typical pxGrid Client Flow



Typical Client Workflow:

- **Create Certificate SSL Context / Account Create:** All clients must authenticate to the ISE pxGrid controller either via certificate-based SSL authentication or username-password authentication.
- **Account Activate:** All clients request to activate their accounts on the pxGrid server which is handled by the REST API.
- **Register/Unregister Service:** Service providers will use these APIs to provide and update the necessary information (i.e. resource URLs) from which their services are accessible for other pxGrid clients.

- **Service Lookup:** All clients can use this API to dynamically discover all available provider services and their locations.
- **Access Secret:** For every service returned that interests a particular client, that client must also query the pxGrid controller for an access secret in order to obtain the information provided by the service.
- **Service Query / Subscribe:** With the access secret and service location information in hand, client can then perform REST-based queries or build WebSocket connections to receive information.

Conclusion

As network threats grow more sophisticated and harder to detect, speed will become an increasingly critical element in effective threat response. Swiveling from one security platform to another or patching together dozens platform-specific APIs will no longer suffice.

Cisco pxGrid provides the framework to accelerate and automate complex security process in the modern enterprise. Any security platform can share information with any other system in the environment in real time, in a highly scalable and tightly controlled manner without relying on platform-specific APIs. For security vendors, pxGrid provides an easy-to-implement framework to extend and enhance the capabilities of their solutions and deliver more value to their customers.

For More Information

To learn more about Cisco pxGrid, visit http://www.cisco.com/go/pxgrid.