Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 18, 2013

T. Tsou
Huawei Technologies (USA)
J. Schoenwaelder, Ed.
Jacobs University Bremen
Y. Shi
T. Taylor, Ed.
Huawei Technologies
G. Yang
China Telecom
July 17, 2012

# Problem Statement for the Automated Configuration of Large IP Networks
## draft-ietf-opsawg-automated-network-configuration-04

Abstract

This memo discusses the steps required to bring a large number of devices into service in IP networks in an automated fashion. The goal of this document is to list known solutions where they exist, to point out approaches proven to be problematic, and to identify gaps that require further specifications.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 18, 2013.

Table of Contents

## 1. Introduction

Many large IP networks are being deployed that entail the installation of tens of thousands of new network devices. To keep costs down, it is desirable to automate the establishment of such networks to the maximum extent possible. This naturally raises the question how new devices can pick up the configuration information they need to operate properly in an automated fashion. The goal of this document is to list known solutions where they exist, to point out approaches proven to be problematic, and to identify gaps that require further specifications.

The document primarily targets (a) network operators (in the generic sense) who are facing the challenge to roll out a large number of new devices and think about how to implement things properly, (b) network equipment vendors who like to add features to their products that make the roll out of lots of new devices simpler for their customers, and (c) people active in the IETF by identifying gaps where further standards may be useful to develop. The aim of the document is to provide guidance to actors who have not already experienced success in this area by informing about the trade-offs of different approaches.

A certain basic amount of configuration information must be pre-configured by the vendor or network operator before the devices are physically deployed. This pre-provisioned configuration can either be stored directly on the device itself or it can be provided to the device during the deployment operation via pluggable memory cards or near field communication technologies. Further device configuration information is best delivered after startup, to ensure that it is consistent with the physical deployment and the desired network configuration.

One example where automated configuration is important are new service provider networks. 3GPP work in progress describes requirements [TS_32_500] and an architectural specification [TS_36_300] for the self-configuration of edge node entities called eNodeBs. (The expansion of eNodeB is too unwieldy to spell out.) Specifically, procedures are specified for establishing transport connections to and for exchanging configuration data with control entities called MMEs (Mobility Management Entities) and with neighbouring eNodeBs. [TS_36_300] currently assumes as a starting precondition that the eNodeB knows its own IP address and knows IP address endpoints for the target MMEs and neighbouring eNodeBs.

The Broadband Forum has defined a CPE WAN Management Protocol (running over SOAP/HTTP/TLS) to manage customer premise equipment (CPE) terminating broadband access networks (typically DSL access

networks) [TR_069]. CPE devices locate and connect to an Auto-Configuration Server (ACS), which provides configuration data and software/firmware images and modules. The ACS also performs status and performance monitoring and diagnostic functions. CPE devices use DHCP to locate an ACS and since both peers, the ACS and CPE, can initiate connections, the protocol can work across network address translators (NATs). The DHCP exchange uses vendor-specific options defined by the Broadband Forum (number 3561 in the IANA Enterprise Numbers registry).

Next to service provider networks, many large enterprise networks face the same challenge to roll out a large number of network devices, which often connect to a 3rd party network provider. The current development of IP-based home automation and utility monitoring technologies might carry the problem to roll out large numbers of devices that need to automatically configure themselves to private households.

IETF work on automated configuration goes back to BOOTP [RFC0951], followed eight years later by DHCP [RFC1541] and successors. The years since have seen a steady growth in the number of DHCP options. The Simple Network Management Protocol (SNMP) [RFC3410] was designed to convey management information between SNMP entities such as managers and agents. The number of SNMP MIB modules grew steadily, but SNMP has historically seen only limited use for configuration [RFC3535]. For a period, IETF configuration efforts were focussed on the distribution of policy information in the network. [RFC3139] provides a good insight into this period. More recently, the network configuration protocol NETCONF [RFC6241] was devised as an alternative to SNMP, but the development of standard NETCONF configuration data models is just beginning.

Recent IETF work closest in spirit to the 3GPP self-organizing network effort cited above is embodied in CAPWAP [RFC5415]. Like the 3GPP work, CAPWAP focusses on the configuration of edge nodes, in a Wi-Fi rather than cellular network. The CAPWAP work goes beyond that of 3GPP by specifying the process of Access Controller (AC) discovery rather than leaving discovery out of scope. A CAPWAP Wireless Termination Point (WTP) may use broadcasts and multicasts to discover local ACs, it may use CAPWAP DHCP options [RFC5417] to obtain IP addresses of ACs, or it may utilize CAPWAP DNS SRV records if a domain name is known. With regard to the configuration process itself, CAPWAP provides for the download of new images to the WTP (Wireless Termination Point). In contrast, [TS_32_500] assumes that this has already been completed for the eNodeB.

As can seen, standards for the automated configuration of devices in IP networks have so far been primarily developed for specific network

access technologies (3GPP, Broadband, 802.11 WLANs) and the various solutions make different assumptions about the services that are available and they are designed to support a configuration protocol that is specific to a certain access technology. The aim of this document is to analyse the various phases of an automated configuration process and to identify gaps that are currently not covered in standard and general purpose configuration management protocols of the IETF.

## 2. Intra-domain and Inter-domain Scenarios

There are two different scenarios to consider. In the first scenario, called the Intra-domain Scenario, the new network device N is attached to the network operated by the service provider which is also operating the new device. In the second scenario, called the Inter-domain Scenario, the new device N is attached to a third party network providing connectivity to the network of the service provider operating the new device.

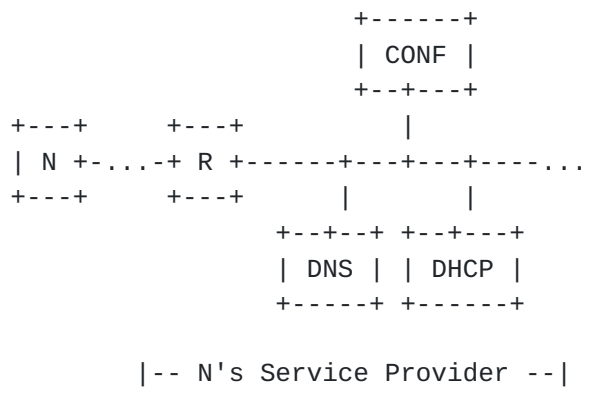


Figure 1: Intra-domain Scenario

Figure 1 depicts the Intra-domain Scenario. We assume that the new decive N attaches to a link connected to router R. Furthermore, we assume that the service provider provides a Domain Name System (DNS) server, a reachable DHCP server, and a Configuration Server (CONF). Overall, this scenario does not differ much from conventional network scenarios.

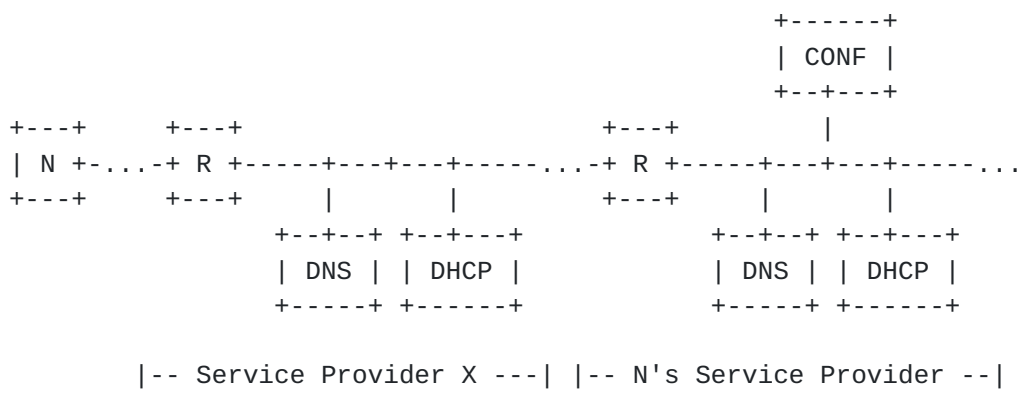


Figure 2: Inter-domain Scenario

Figure 2 depicts the Inter-domain Scenario where the new device N attaches to a router R owned by a different service provider X. The service provider X might offer its own DNS service and a reachable DHCP service. We assume that the service provider X has connectivity to the service provider planning to operate the new device.

It should be noted that handing out DHCP options specific to N's service provider via X's DHCP service requires some close coordination between the two parties involved. This might be difficult in practice. A more general alternative might be to have X's service provider establish a tunnel such that the new device logically appears to be part of N's service provider network.

In both scenarios, the new device N is either directly reachable or it may be behind a middlebox such as a Network Address Translator (NAT) or a firewall. Middleboxes may impose restrictions on which party is able to initiate communication. As detailed in [I-D.kwatsen-reverse-ssh], it is often desirable to allow device-initiated connections.

## 3. Model of the Automated Configuration Process

We introduce a model of the configuration process in order to identify the parts that have well-known solutions. The remainder may be worth studying to see if the industry can agree on a solution.

Some basic terminology is needed for the discussion. Depending on the implementation, let us agree that "configuration data" consist of software and sets of configured parameters in some combination. This includes firmware, licenses, certificates, and other configuration data. Also, the system that provides the configuration data is called the "configuration server". Finally, the term "joining device" is used to denote a network device that is in the process of

being incorporated into the network.

Broadly speaking, the configuration process can be broken into five phases:

1. Pre-configuration: configuration carried out either by the vendor or by the service provider prior to physical installation. One possible example is the pre-configuration of certificates or licenses or specific firmware.

2. Bootstrapping: the portion of the process from the time that physical installation is complete until a secure connection is established between the joining device and the configuration server.

3. Initial configuration: downloading of the configuration data that the joining device needs to carry out its function in the network.

4. Configuration auditing: tracking image versions and configuration parameters for each network device and verifying that the installed configuration data matches the physical installation, the network plan, and the records of what data was downloaded. It is possible that an initial audit of the physical installation is done before initial configuration, so that the validity of the intended download can be verified.

5. Configuration update: transferring configuration data to a fully configured and operating device from time to time as the need arises.

## 4. Phase 1: Pre-configuration

This memo identifies a specific requirement for pre-configuration of an invariant device identity and authentication-related material in the form of pre-shared secrets or certificates. There is, as one alternative, also a requirement for pre-configuration of information that permits the joining device to discover the address of the configuration server.

Note that pre-configuration may be carried out on the joining device itself or it may be provided to the joining device during the deployment process via pluggable memory cards or nearfield communication.

## 5. Phase 2: Bootstrapping

[I-D.sarikaya-core-sbootstrapping] deals with the process of security bootstrapping, with particular emphasis on the requirements for highly resource-constrained devices. The document makes a distinction between a data channel, which is used during network operation, and a control channel, which is used during bootstrapping. While both channels can be the same physical channel, they can also be different (e.g., a wireless access point using an infrared control channel to receive bootstrapping information). The draft discusses a number of possible security bootstrapping protocols for resource constrained devices that can be executed in several bootstrapping rounds and can be adapted to the specific contexts in terms of the resources available within individual devices and for the network as a whole.

For network devices in service provider networks or large enterprise networks, bootstrapping consists of several stages:

1. establishment of link layer connectivity with neighbouring nodes;

2. acquisition of IP addresses and basic routing information;

3. discovery of the configuration server;

4. establishment of a secure channel to the configuration server.

Each of these stages is further discussed below.

### 5.1. Establishment of Link Layer Connectivity

The protocol aspects of this phase are out of scope, since it involves non-IETF protocols only. While some link-layer technologies may provide authentication and access control, this cannot be assumed to be available in the general case.

### 5.2. Acquisition of IP Addresses and Basic Routing Information

For IPv4, DHCPv4 [RFC2131] is widely deployed and the usual way to obtain an IPv4 address, the IPv4 address of a link-local router and the IPv4 address of a DNS server. For IPv6, a choice has to be made between stateful DHCPv6 [RFC3315] versus stateless DHCPv6 [RFC3736] combined with stateless address autoconfiguration [RFC4862]. In the latter case, DHCPv6 is needed to configure parameters such as DNS server addresses. A routing advertisement option to configure the IPv6 address of a DNS server as part of the stateless address autoconfiguration is defined in [RFC6106].

Some security protection is provided in this stage by using DHCP authentication [RFC3118]. However, security of the configuration process as a whole has to be assured by other means. This is discussed further below.

Currently the lack of a stable identifier for use in DHCPv6 messaging is an impediment to authentication of the joining device. [RFC6355] discusses the problems with the current DHCPv6 identifiers (DUIDs) and proposes a new form that could be a more stable alternative.

A joining device can also choose to use a pre-configured IP address, a pre-configured link-local router address and a pre-configured DNS server address. This pre-configuration may be hard wired into the device or provided by a pluggable memory card or nearfield communication. However, a static pre-configuration hard-wires assumption about the network a devices operates in and is therefore brittle and not recommended.

### 5.3. Finding the Configuration Server

Four alternatives are available for finding the configuration server:

- pre-configuration;
- DHCP configuration;
- Service Location Protocol [RFC2608]; or
- DNS service discovery using DNS SRV records [RFC2782].

Pre-configuration of an IP address is brittle and not recommended unless the IP address is used as an anycast address. In the case of an IP anycast address, the routing system will select one out of an anycast cluster of configuration servers the devices connects to. For this to work well, all configuration servers in the anycast cluster should provide the same configuration data.

The pre-configuration of a Uniform Resource Identifier (URI) or fully qualified domain name (FQDN) is a slightly better approach than pre-configuring non-anycast IP addresses since this allows for a limited dynamic mapping of the name to an IP address. One variant that has been suggested is to burn the URI of a vendor server into the device's firmware along with a device identifier, and have that server redirect to the URI of the service provider's configuration server based on the device identity. Such an approach requires that the device vendor's redirection server is always reachable, that the device vendor offers such a redirection service for the lifetime of their devices and that service providers are able to update the URI

of the service provider's redirection server. Furthermore, this approach can lead to problems if certificates are used to authenticate the involved parties if a service provider tries to prevent the usage of a vendor's redirection service. Finally, this approach also requires a trust relationship between the vendor and the service provider and agreement on a protocol to update the redirect information on the vendor's server. As a consequence of these considerations, using this approach is not recommended.

DHCP configuration can use the usual DHCP options and is technically straightforward since DHCP is widely used by end user devices to obtain basic configuration information. There is, however, no standardized DHCP option to communicate the address of a configuration server.

The Service Location Protocol (SLP) has seen some usage to locate services such as printers or file system shares. Usage of SLP to locate configuration servers requires to define a new service template [RFC2609].

The use of DNS SRV records requires the joining device to obtain the correct domain suffix first, presumably from DHCP or via Routing Advertisements in the case of IPv6 or pre-configuration. A service type for the desired configuration protocol would have to be defined in the DNS for the purpose. See Section 3.3 of [RFC5415] for a discussion of the corresponding discovery process for CAPWAP.

The Inter-domain Scenario requires that the DHCP server or the SLP server of service provider X's network is able to provide the correct information to the joining devices. To accomplish this, the discovery servers need to be able to match a device identification against a list of possible configuration servers. Furthermore, there needs to be a mechanism for the service provider operating the joining device to provision the configuration server's address, e.g., by using an extension of the Extensible Provisioning Protocol (EPP) [RFC5730]. However, if the joining device has pre-configured information about the name of the service provider's network, DNS SRV records may be queried after obtaining IP connectivity, avoiding the need to provision information in service provider X's network.

## 5.4. Establishing a Secure Channel to the Configuration Server

It is essential that the configuration server and the joining device authenticate themselves to each other, since the steps leading up to this point in the process may not be fully secure. This raises two issues: how the joining device identifies itself, and how authentication takes place.

It seems best if the device has an invariant identity built in and accessible to whatever operating system is running on it. [RFC6355] provides such an identity in the form of a Universally Unique IDentifier (UUID). The vendor should make that identity available in a form that can be read and transferred into a database accessible to the configuration server along with the associated configuration data in advance of the bootstrapping stage (e.g., in bar-coded format on the device packaging).

Serial numbers may be used for identification purposes if UUIDs are not available. However, serial numbers often encode information such as model-numbers or manufacturing dates. Hence, it is not recommended to pass serial-numbers in the clear for security reasons. Similar precautions apply to Common Language Equipment Identifier (CLEI) codes that encode information about properties of the device.

This leaves the mutual authentication process itself. This has two aspects: the security protocol used to perform authentication, and initial keying methodology. The security protocol is tied together with the choice of configuration data transport, but the basic choices are:

- IP Security (IPsec) [RFC4301];
- Transport Layer Security (TLS) [RFC5246];
- Datagram Transport Layer Security (DTLS) [RFC6347];
- Secure Shell (SSH) [RFC4251], [RFC4252], [RFC4253], and [RFC4254]; and
- SNMPv3's User-based Security Model (USM) [RFC3414].

For initial keying methodology, the two basic choices are between pre-shared secrets and certificates. All of the security protocols listed above except USM support both methods. USM supports pre-shared secrets only.

The usual concern with pre-shared secrets is scalability. In the bootstrapping case, the scale of operation required is linear with the number of devices to be configured, so it would definitely be a feasible approach if connection to the configuration system were the only consideration. The most likely procedure would be for the secret to be configured in the device during pre-configuration and also captured in a database along with the device identity, for use by the configuration server.

The problem with the use of pre-shared secrets is that the device

needs to authenticate itself at an earlier stage, while it is establishing communications with its neighbours and acquiring IP addresses. It seems undesirable to use the same secret that is used to authenticate the device to the configuration server for that purpose as well, on the basic principle of limiting the potential damage from disclosure of a particular key.

This need for additional pre-shared secrets argues for consideration of certificates as an alternative. One issue for certificates is where the trust anchor resides. It seems logical that it should reside with the service provider rather than the vendor, to make it easy to install equipment from multiple vendors. On that basis, pre-configuration requires service provider input. On the other hand, if devices are drop-shipped to the destination from the vendor, having the trust anchor reside with the vendor might be acceptable as well.

CAPWAP (Section 2.4.4.3 of [RFC5415]) makes use of the Extended Key Usage (EKU) certificate extension [RFC5280] to distinguish certificates identifying the Access Controllers (i.e., the configuration servers in the CAPWAP case) from the Wireless Transfer Points (the configured devices in the CAPWAP case). Thought should be given to whether such distinctions are required in the general case of network device configuration.

CAPWAP (Section 12.8 of [RFC5415]) also discusses the use of the Common Name rather than SubjectAltName field of the certificate to carry device identity, due to lack of a Uniform Resource Name (URN) specification allowing the use of SubjectAltName to carry MAC addresses. This encoding of device identifiers in certifications needs to be investigated further if a new form of device unique identity is used, as discussed above.

Middleboxes such as NATs or firewalls may impose restriction on which party is able to initiate communication. In the common case of NATs in IPv4 access networks, communication can only be established from the device to the configuration server. Not all secure transports, in particular those where authentication is not symmetric, support this "call home" mode of operation. A recent proposal to reverse the establishment of the TCP connection for SSH can be found in [I-D.kwatsen-reverse-ssh].

## 6. Phase 3: Initial Configuration

As mentioned at the beginning, the configuration data being downloaded may be a combination of software/firmware and configuration parameters. Some of the data will be vendor-specific and not subject to standardization. It appears that there is a

continuing debate on whether the configuration data should be pushed to the joining device or whether the device should pull the configuration data from the configuration server. In the latter case, the device needs to know about the existence of the data and the path to reach it before it can act. One way to acquire this information is through DHCP. DHCPv4 has provided the necessary options from its beginnings, inheriting them from BOOTP. They have been recently added to DHCPv6 [RFC5970].

Protocols that can transport configuration data can be classified as follows: The first class consists of generic file transfer protocols that can carry configuration data serialized into configuration files. The second class consists of protocols that manipulate structured configuration data directly. The structure of the configuration data is defined by some data model.

In the first class, we find the following file transfer protocols:

- The File Transfer Protocol (FTP) [RFC0959] can be used to move files containing configuration data. It can be secured by running FTP over TLS [RFC4217].

- The Trivial File Transfer Protocol (TFTP) [RFC1350] has been used extensively to load boot images over the network. However, it does not provide security and the only option is to rely on IP layer security (IPsec).

- The Hypertext Transfer Protocol (HTTP) [RFC2616] can be used to transfer documents containing configuration data. It is commonly secured by running HTTP over TLS [RFC2817] [RFC2818].

- The SSH File Transfer Protocol (SFTP) [I-D.ietf-secsh-filexfer] provides roughly the same services as FTP but runs over SSH and thus utilizes the security services provided by SSH.

- UNIX utilities to transfer files such as RCP and SCP provide limited flexibility and they differ in their degree of integration with SSH.

- The Control And Provisioning of Wireless Access Points (CAPWAP) protocol [RFC5415] can be used to control the download of images. CAPWAP can be secured by running CAPWAP over DTLS.

In the second class, we find the following configuration protocols:

- Version 3 of the Simple Network Management Protocol (SNMPv3) [RFC3411] can be used to manipulate MIB objects and to carry event notifications. SNMPv3 has its own security protocol (USM)

[RFC3414] but can also run over the secure transports SSH [RFC5592], TLS, or DTLS [RFC6353].

- The Common Open Policy Service for Policy Provisioning protocol (COPS-PR) [RFC3084] was designed to provision structured policy information from a Policy Decision Point (PDP) to a Policy Enforcement Point (PEP). The COPS protocol [RFC2748] provides an integrity object that can achieve authentication, message integrity, and replay prevention. Optionally, COPS and COPS-PR can run over TLS.

- The NETCONF protocol [RFC6241] provides mechanisms to install, manipulate, and delete the configuration of network devices. A protocol extension provides an asynchronous event notification delivery mechanism [RFC5277]. NETCONF by default runs over SSH but can also run over transports secured by TLS.

- The Control And Provisioning of Wireless Access Points protocol (CAPWAP) [RFC5415] supports the discovery of so called Access Controller (AC) by Wireless Termination Points (WTPs) and the configuration of WTPs by an AC. While CAPWAP can be extended to configure other devices, its main focus are WTPs. The CAPWAP protocol is protected by using DTLS after the discovery phase.

Table 1 lists the protocols plus their basic properties while Table 2 lists the security options available for each protocol.

| Transport | Data Transfer Model |
|---|---|
| FTP | Push or pull of (configuration) files |
| TFTP | Push or pull of (configuration) files |
| HTTP | Push or pull of (configuration) files |
| SFTP | Push or pull of (configuration) files |
| RCP | Push or pull of (configuration) files |
| SCP | Push or pull of (configuration) files |
| CAPWAP | AC pushes configuration parameters, WTP pulls software |
| SNMPv3 | Push of structured configuration parameters, event notifications |
| COPS-PR | Push of structured policy information |
| NETCONF | Push of structured configuration data, event notifications |

Table 1: Protocols for transporting configuration data

| Transport | IPsec | TLS | DTLS | SSH | Other |
|---|---|---|---|---|---|
| FTP | + | + | | | |
| TFTP | + | | | | |
| HTTP | + | + | | | |
| SFTP | + | | | + | |
| RCP | + | | | | |
| SCP | + | | | + | |
| CAPWAP | + | | + | | |
| SNMPv3 | + | + | + | + | USM |
| COPS-PR | + | + | | | |
| NETCONF | + | + | | + | |

Table 2: Security options for configuration transport protocols

SNMPv3, NETCONF, and COPS-PR carry structured data specified in pre-defined data models. SNMPv3 and COPS-PR have size limitations on the data objects and thus make the transport of larger software images difficult. NETCONF does not suffer from hard size restrictions and can in principle carry software images inline. However, there is currently no work in progress to standardize the transfer of software images over NETCONF. CAPWAP combines the functions of configuration parameter transport and software download. The parameter transport aspect lacks the generality offered by SNMP, NETCONF, and COPS-PR, since the parameters are specified within the protocol specification itself. The remaining transports are independent of the nature of the information being transferred.

## 7. Phase 4: Configuration Auditing

To complete the process, it must be possible to audit the configuration status of the device in some detail. This is likely to begin even before all the configuration data has been downloaded. For instance, configuration management may wish to collect basic information such as the MAC addresses of the device's interfaces, the link-local addresses assigned to them, and similar information for the neighbours of the joining device.

SNMP and SNMP MIB modules are obviously one way to collect this information. NETCONF [RFC6241] is an alternative, but the necessary data models have to be defined. YANG modules for NETCONF [RFC6020] can be generated from existing SNMP MIB modules by translating the SNMP modules into YANG modules [RFC6643].

Another important auditing activity is the analysis of system events.

The SYSLOG protocol [RFC5424] is widely used for this purpose but SNMPv3 and NETCONF can ship event notifications as well. Translations of SNMP notifications into structured SYSLOG messages and vice versa do exist [RFC5675] [RFC5676]. NETCONF can carry SYSLOG content as well [RFC5277].

NETCONF provides generic notifications that help with tracking configuration changes [RFC6470]. Similar standardized configuration change notifications do not exist for SNMP or SYSLOG.

## 8. Phase 5: Configuration Update

Configuration updates can in principle be handled with the same protocol that delivered the initial configuration. However, in some deployments, the mechanism used for initial configuration might be different.

An advantage of NETCONF over SNMPv3 and CAPWAP in the context of configuration updates is the support of concurrent updates through explicit locking mechanisms and the support of network wide configuration change transactions through the confirmed commit capability.

## 9. Gap Analysis

This document discussed the automated configuration of devices in large IP networks. Several gaps were identified requiring further specification:

G1: Definition of a DHCP option to provide the IPv4/IPv6 address of a configuration server. Such an option allows a joining device to pickup the configuration server's address as part of the DHCP exchange. This is particularly interesting for Intra-domain Scenarios.

G2: Definition of DNS SRV records for locating configuration servers. Using SRV records, a joining device can lookup the configuration server's address in the DNS. This is particularly useful in an Inter-domain Scenario.

G3: Definition of a SLP template for discovering configuration servers. Such a template is useful only in environments where SLP is used also for other purposes.

G4: Definition of NETCONF data models to support the download / update of software images through NETCONF.

G5: Definition of NETCONF data models for collecting basic system information and integrity information (e.g., checksums of software images).

G6: Some management protocols lack a mechanisms for devices to initiate a secure communication channel with a management system ("call home").

## 10. Conclusions

This section summarized the previous discussions and provides some concrete recommendations. The following recommendations are given to network operators and equipment vendors who have not yet experienced success in this area:

- Hard-wired non-anycast IP addresses are brittle and not recommended for finding a configuration server. Hard-wired URIs or domain names allow one level of indirection but can still be problematic during the lifetime of a product. Using DHCP to provision the IP address of a configuration server dynamically or using DNS SRV records to query the DNS for a suitable configuration server is preferred over solutions that use hard-wired information.

- For device identification, identifiers are generally preferred that do not carry further semantics about a device, such as UUIDs produced by cryptographic hash functions.

- A number of protocols can be used to transfer the initial configuration (software/firmware and configuration parameters). Selection of a suitable protocol should take into account (i) whether a push or pull model or both are needed (e.g., to support working around middleboxes such as Network Address Translators, NATs) and (ii) how the security options and their key management mechanisms integrate into the target network.

Section 9 identifies gaps where additional standardization work might be useful. The first three (G1-G3) all address the need to locate a configuration server. Out of the three options, G3 seems to be mostly of theoretical value since SLP does not appear to be widely used for this purpose. For G1, however, there is some usage evidence coming from the CPE WAN Management Protocol [TR_069]. The usage of DNS SRV records requires to obtain the domain name via other means first. As such, the it seems that G1 is more meaningful to address

at this point in time.

Addressing G4 and G5 does not seem to be of high priority at this point in time. NETCONF's strength are its operations to make incremental configuration changes on a large number of devices in a robust manner. As such, NETCONF is a good protocol for incremental configuration updates. For transferring files or software images, other protocols work reasonably well. At this point in time, the only benefits of addressing G4 or G5 would be the reuse of the security and authorization mechanisms provided by NETCONF.

Due to the prevalence of middleboxes such as NATs, it is often required that devices establish the management sessions to their management systems. A BoF covering this topic was held at the 64th IETF meeting, which was triggered in part by work in the ISMS working group on alternate secure transports for SNMP. While the ISMS working group, after consultation with security experts, decided to not address this problem, the issue resurfaced later in the NETCONF working group but was not addressed there either. Vendors meanwhile seem to ship proprietary solutions. As such, G6 seems worthwhile to address but it is also known to be a difficult topic, requiring extensive support from the security area.

## 11. Security Considerations

The security of a configuration management solution is of crucial importance. Section 6 discusses the security options of several protocols that might be used. The relevant protocol definitions should be consulted to learn more about the specific security aspects of the various protocols.

It should be noted that some steps in the described process, in particular the bootstrapping phase, may not be secure and it is thus important to verify the identity of the device and the identity of the configuration server when a secure connection to a configuration server is established. Usage of IPsec, which focuses on securing the IP layer, may not be sufficient for this.

During the choice of protocols, the available security mechanisms and the required key management infrastructures may play a major role in the selection of protocols. Easy integration into existing Authentication, Authorization and Accounting (AAA) infrastructures can significantly reduce the operational costs associated with the security management of the configuration system.

While [I-D.sarikaya-core-sbootstrapping] discusses security bootstrapping mechanisms in the context of constrained devices, many

of the mechanisms are also applicable for bootstrapping security in normal devices.

Finally, [RFC6092] discusses security capabilities for customer premises equipment providing residential IPv6 Internet service.

## 12. IANA Considerations

This memo includes no request to IANA.

## 13. Acknowledgements

Thanks to Ronald Bonica, Mehmet Ersue, Wesley George, Yiu Lee, Christopher Liljenstolpe, Kent Watsen, and Cathy Zhou for their comments during the preparation of this memo.

## 14. Informative References

[I-D.ietf-secsh-filexfer]
Galbraith, J. and O. Saarenmaa, "SSH File Transfer Protocol", draft-ietf-secsh-filexfer-13 (work in progress), July 2006.

[I-D.kwatsen-reverse-ssh]
Watsen, K., "Reverse Secure Shell (Reverse SSH)", draft-kwatsen-reverse-ssh-01 (work in progress), June 2011.

[I-D.sarikaya-core-sbootstrapping]
Sarikaya, B., Ohba, Y., Moskowitz, R., Cao, Z., and R. Cragie, "Security Bootstrapping Solution for Resource-Constrained Devices", draft-sarikaya-core-sbootstrapping-05 (work in progress), July 2012.

[RFC0951] Croft, B. and J. Gilmore, "Bootstrap Protocol", RFC 951, September 1985.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[RFC1350] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 1350, July 1992.

[RFC1541] Droms, R., "Dynamic Host Configuration Protocol",

RFC 1541, October 1993.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2608] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[RFC2609] Guttman, E., Perkins, C., and J. Kempf, "Service Templates and Service: Schemes", RFC 2609, June 1999.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2748] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC2817] Khare, R. and S. Lawrence, "Upgrading to TLS Within HTTP/1.1", RFC 2817, May 2000.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3084] Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R., and A. Smith, "COPS Usage for Policy Provisioning (COPS-PR)", RFC 3084, March 2001.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3139] Sanchez, L., McCloghrie, K., and J. Saperia, "Requirements for Configuration Management of IP-based Networks", RFC 3139, June 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4217] Ford-Hutchinson, P., "Securing FTP with TLS", RFC 4217, October 2005.

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4252] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

[RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.

[RFC4254] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Connection Protocol", RFC 4254, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, July 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5415] Calhoun, P., Montemurro, M., and D. Stanley, "Control And

Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, March 2009.

[RFC5417] Calhoun, P., "Control And Provisioning of Wireless Access Points (CAPWAP) Access Controller DHCP Option", RFC 5417, March 2009.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.

[RFC5675] Marinov, V. and J. Schoenwaelder, "Mapping Simple Network Management Protocol (SNMP) Notifications to SYSLOG Messages", RFC 5675, October 2009.

[RFC5676] Schoenwaelder, J., Clemm, A., and A. Karmakar, "Definitions of Managed Objects for Mapping SYSLOG Messages to Simple Network Management Protocol (SNMP) Notifications", RFC 5676, October 2009.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, September 2010.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 6353, July 2011.

[RFC6355] Narten, T. and J. Johnson, "Definition of the UUID-Based DHCPv6 Unique Identifier (DUID-UUID)", RFC 6355, August 2011.

[RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", RFC 6470, February 2012.

[RFC6643] Schoenwaelder, J., "Translation of Structure of Management Information Version 2 (SMIv2) MIB Modules to YANG Modules", RFC 6643, July 2012.

[TR_069] Blackford, J., Ed., Kirksey, H., Ed., and W. Lupton, Ed., "CPE WAN Management Protocol", Broadband Forum TR-069, November 2010.

[TS_32_500] 3rd Generation Partnership Project, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication Management; Self-Organizing Networks (SON); Concepts and requirements (Release 9)", 3GPP TS 32.500, 2010.

[TS_36_300] 3rd Generation Partnership Project, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 9)", 3GPP TS 36.300, 2010.

## Authors' Addresses

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara CA 95050
USA

Email: tina.tsou.zouting@huawei.com

Juergen Schoenwaelder (editor)
Jacobs University Bremen
Campus Ring 1
Bremen  28759
Germany

Email: j.schoenwaelder@jacobs-university.de

Yang Shi
Huawei Technologies
156, Beiqing Road, Zhongguancun, Haidian District
Beijing
P.R. China

Phone: +86 10 60614043
Email: shiyang1@huawei.com

Tom Taylor (editor)
Huawei Technologies
1852 Lorraine Ave.
Ottawa  K1H 6Z8
Canada

Email: tom111.taylor@bell.net

Guoliang Yang
China Telecom
No. 109 Zhongshan Ave. (West), Tianhe District
Guangzhou
P.R. China

Phone: +86 020 38639615
Email: iamyanggl@gmail.com