

IPWorks Application Components Hardening Guide

USER GUIDE

Copyright

© Ericsson AB 2017, 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Target Groups	1
1.2	Prerequisites	1
1.3	Related Information	1
2	Overview	3
3	Hardening DNS (BIND) Server	5
3.1	Domain Namespace	5
3.2	CPU Requirement on the DNS	5
3.3	Zone Information	6
3.4	DNS Resource Records	6
3.5	Unnecessary DNS Resource Records (RR)	8
3.6	Named ACL	8
3.7	Views	9
3.8	TSIG Requirement for DNS	9
3.9	Using the Security-related Options Clauses in the named.conf	10
3.10	Packet Filtering for the Incoming DNS Queries	11
4	Hardening Active Select DNS	13
5	Hardening AAA Server	15
5.1	Hardening Radius	15
5.2	Hardening Diameter	16
6	Hardening DHCP Server	19
6.1	IP Address Models	19
6.2	Adding Security to DHCP	19
6.3	DHCP Multi-Server Non-Overlapping Scopes	20
7	Communication between the DNS and DHCP Servers	21
7.1	System Configuration	21
8	Hardening ENUM server	23
8.1	ENUM Access Control	23



9	Hardening MySQL Server	25
9.1	Protecting the MySQL Files	25
9.2	Removing Test Database	25
9.3	Disabling local-infile	25
9.4	Lowering System Privilege	26
9.5	Lowering Database Privilege	27
9.6	Removing History	28
10	Hardening Storage Server	29
11	Hardening SNMP	31
11.1	Disabling SNMPv1 and SNMPv2	31
12	Appendix	33
12.1	Appendix A: TSIG Configuration Example	33
	Reference List	35



1 Introduction

The document provides guidelines for hardening the security of the IPWorks application components to a generally agreeable level.

Issues dealt within this document are summarized as below, for example:

- Configuring the application components securely.
- Configuring IP filtering.
- Managing user accounts and permissions.

This guideline also includes configuration instructions such as for setting up encrypted connections for sensitive management data.

1.1 Target Groups

This document is intended for personnel that perform security hardening for IPWorks application components.

1.2 Prerequisites

The security of the application components in IPWorks depends largely on the following:

- Hardening of Operating System is completed.
 - For Linux Distribution Extensions (LDE), refer to *IPWorks OS Hardening Guide*.
 - For SuSE Linux Enterprise Server (SLES) 12 SP2, refer to *IPWorks OS Hardening Guide for KVM*
- The network communications security in all layers (layer 2, 3 and 4-7) between the different nodes of IPWorks and between IPWorks and the O&M center.

It is ineffective to apply the application specific hardening guidelines, if the above two criteria are not met. For a separate hardening guideline for the SuSE operating system, refer to *IPWorks OS Hardening Guide*.

1.3 Related Information

Trademark information, typographic conventions, and definition and explanation of abbreviations and terminology can be found in the following documents:



- *Trademark Information*
- *Typographic Conventions*
- *Glossary of Terms and Acronyms*



2 Overview

Domain Name System (DNS) (BIND) Server

DNS maps:

- A domain name to an IP address or vice versa.
- A telephone number to a domain name (ENUM) based on the information stored in the databases of the regular DNS and the ENUM server.

The DNS service must provide data origin authentication and integrity for the queries, responses, dynamic updates and zone transfers. Both IPv4 and IPv6 addresses are supported.

Active Select DNS (ASDNS)

ASDNS functions can be divided into two main groups:

- Monitoring: used to determine the state/load of a resource and report it back to DNS servers.
- Resolving: used by a DNS server to interpret reports received from monitoring and response the queries by filtering and/or re-ordering the answer.

AAA Server

IPWorks AAA server supports the authentication, authorization functionalities based on the RADIUS protocol.

It also supports the Ericsson Gi interface to interwork with the GGSN nodes in M-PBN solutions.

AAA provides the PAP and CHAP for authentication and allows the GGSN to obtain the APN configuration parameters for the specified user through the rule based authorization.

Diameter protocol is supported for EPC AAA.

Dynamic Host Configuration Protocol (DHCP) Server (NACF)

DHCP leases IP addresses. A DHCP server maps a client's Media Access Control (MAC) address to an available IP address stored in the address pool database in the DHCP server. DHCP servers co-operate with the DNS servers on the Dynamic DNS (DDNS) update. IPWorks DHCP server can act as a Network Access Configuration Function (NACF) server, used in IMS broadband access solution.



E.164 telephone Number Mapping (ENUM) Server

Standard ENUM is based on DNS and uses DNS Naming Authority Pointer (NAPTR) resource records to return information to the DNS client. The ENUM server supports standard ENUM queries and returns appropriate information in NAPTR resource records in accordance with RFC 3761.

The External Resolution Handler (ERH) is an extension to the ENUM server to assist lookups in external database. IPWorks ENUM server retrieves the Number Portability (NP) or Toll Free information from external SS7 databases.

MySQL NDB Cluster

MySQL is an open source database that runs on a wide variety of platforms including SuSE OS.

MySQL storage engine stores information about network configurations and the users in its database.

The NDB cluster is located on SC boards.

Storage Server (SS)

SS is a centralized server that coordinates the permanent storage of the DNS and ENUM configuration information for the network. It is also the central point of communication for all IPWorks components.



3 Hardening DNS (BIND) Server

In IPWorks, the regular DNS server (ISC BIND 9) no longer has the external interface, used the conventional UDP and TCP ports 53. Instead, these ports are now connected to the EnumIf (parser) component of the ENUM server, which forwards all the regular DNS queries to BIND using the port 5300 on the loop-back address 127.0.0.1 (IPv4) and ::1 (IPv6).

Whenever ENUM server forwards a query to DNS, it forwards original client source IP address information to DNS, using an internal protocol. Also the related responses go via the EnumIf component. The ENUM server resolves all the (extended) ENUM queries using the co-located MySQL NDB Cluster database.

All the communication originated by the local DNS server towards an external DNS server (For example, zone transfer) still bypasses the EnumIf parser.

3.1 Domain Namespace

The domain namespace is organized as a hierarchical tree. A root domain is on the top of the tree and the branches are spread below. Each domain branch gets a label on the tree. The label information is identified by a domain name in the DNS.

3.2 CPU Requirement on the DNS

It is important for BIND that the CPUs are effective enough to:

- Serve static zones without caching
- Process dynamic updates
- Process security of the zones
- Serve thousands of queries per second.

The memory of the server must be large enough to fit the cache and zones loaded off the disk. A good way to recognize a sufficient capacity of memory is to monitor the DNS traffic during a couple of weeks.

3.2.1 Split DNS Architecture

Extra protection is obtained for the name server by using a physically or logically split DNS architecture. The splitting means that the internal name servers (iDNS) hide the domains that are not allowed to be visible to the external network. The iDNS resolve only the queries from internal hosts. The external



name servers (eDNS) handle all queries to and from the external domains. The eDNS are located on the demilitarized zone (DMZ) in a firewall/security gateway. The iDNS forward all requests that they cannot resolve to the eDNSs. The redundant DNS servers should be placed on different sites if possible, but at least on different network segments (subnets).

With this architecture iDNS servers are more secure than if connected directly to external networks, reducing the risk of external attacks and impacts on the node due to external network failures. The eDNS name servers will do the resolution for all external queries. They will face the same threats as any name servers connected to a public DNS infrastructure like the Internet DNS infrastructure.

This is already part of the planned deployment of IPWorks. Refer to the Section *Configuring DNS in Sample Network 1* in *Configure DNS and ENUM*.

3.3 Zone Information

Since zone transfers are restricted, the attackers will get only minimal information about the target network. The zone data should include only the minimum required information. This information consists of resource records (RR) that shall be defined by the zone administrator.

Zone transfers are intended for the slaves that need the update information from the master located in the same zone. This communication must be guaranteed and protected between the nodes.

To set up a master zone or slave zone, follow the instructions in the sections, *Configuring Master Zones for Sample Network 1* and *Configuring Slave Zones for Sample Network 1*, in *Configure DNS and ENUM*.

3.4 DNS Resource Records

Resource Records (RR) are basic elements used in DNS to represent data that will be distributed by the DNS protocol. For information on specific RRs, such as DNS RRs, Managing PTR Records, RR Data, and SOA RRs, refer to the *DNS Resource Records* section in *IPWorks DNS, ASDNS, ENUM Parameter Description*.

3.4.1 Example: Resource Records CNAME and SOA

Canonical Names (CNAME) maps an alias to its canonical name. Start of Authority (SOA) points to the primary master name server and the contact person of the concerned zone.

In the front of a SOA Resource Record (RR) there may be an at-sign (@) that refers back to the domain name (tele.com. in the example below). This is declared in the `/etc/ipworks/<hostname>/dns/named.conf` file (the default Origin).



The last dot (.) must be specified to declare an absolute domain name. Fully Qualified Domain Name (FQDN) is a unique and complete (ends with a period) domain name, for example, like ns.ericsson.se.

Table 1 Resource Records CNAME and SOA Example

@	IN SOA tea.tele.com
	IN NS tea.tele.com
	IN NS tea6.tele.com
tea	IN A 192.168.1.1
dns	IN CNAME tea ; an alias for dns
tea6	IN AAAA 2001:db8:4008::1

Select concerned dnsserver and zone objects, and make the following additions (DNS hardening) to the associated `/etc/ipworks/<hostname>/dns/named.conf` file by using the following IPWorks CLI commands (and *update operation*):

- Restrict zone transfers:

```
IPWorks> select dnsserver <dnsserver name>
```

```
IPWorks> modify -add option="allow-transfer  
{nameserver1;nameserver2;}";
```

- Hide the BIND version:

```
IPWorks> select dnsserver <dnsserver name>
```

```
IPWorks> modify -add option="version \"Not available\"";
```

- Allow queries only from specific subnets except for a dedicated telecom.com zone:

```
IPWorks> modify -add option="allow-query {subnet1;subne  
t2;localhost;}";
```

```
IPWorks> select masterzone telecom.com
```

```
IPWorks> modify -add option="allow-query {any;}";
```

It shows the following message when the configuration successes.

```
IPWorks> modify -add option="allow-transfer  
{192.168.0.10;}";
```

Working on 1 object(s).

1 object(s) were updated.

3.5 Unnecessary DNS Resource Records (RR)

The zone data provides a plethora of data for attackers. Certain resource records (RRs) give many informational details about the domain, how to plan and do an intrusion or even worse a DoS attack. This is the reason why certain RRs should be removed.

The following list contains the most common resource records that involve potential vulnerabilities for the DNS system. Do not provide zone data with these resource records unless they are absolutely necessary for the DNS application:

- HINFO record that identifies the hardware and operating system
- MB record that identifies a factor in the mail system
- MD record that identifies a factor in the mail system
- MF record that identifies a factor in the mail system
- MG record that identifies a factor in the mail system
- MINFO record that identifies a factor in the mail system
- MR record that identifies a factor in the mail system
- MAILA record that identifies a factor in the mail system
- MAILB record that identifies a factor in the mail system
- “wildcard” record that identifies all available records in a (sub)zone
- RP record that identifies Responsible Person of the name server in a zone
- RT record that identifies the route that is used between the name servers in the network
- TXT record identifies any textual information.

Note: The (extended) ENUM service makes use of the TXT RRs for certain functions like call gapping, IN triggers and handling incomplete digit strings.

3.6 Named ACL

Named Access Control List (ACL) provides a mechanism that allows a user to define a list in only one place and then refer to that list name.

An ACL simplifies the DNS system management since the database (For example, zone data) changes are propagated regularly and automatically to all of the DNS servers around the specific zone.



Follow the instructions for Named ACLs and Address Match Lists in the *Configuring Named ACLs for Sample Network 2* section in *Configure DNS and ENUM*.

3.7 Views

Views allow a user to support multiple definitions of zones within the DNS server. When the logically split DNS (namespace) is implemented, it is based on views. Four parameters are used for the views: name, area, option, and rank.

For more information, refer to the section *Configuring Views for Sample Network 2* in *Configure DNS and ENUM*.

3.8 TSIG Requirement for DNS

Split DNS system needs appropriate protection between the name servers on the network. Transaction Signatures (TSIG) is a good alternative protection mechanism for this purpose. Refer to Secret Key Transaction Authentication for DNS, Reference [11], for more information. TSIG is used to provide data origin authentication and data integrity using a shared secret with the HMAC-MD5 integrity algorithm. If TSIG is used, the TSIG's shared secret key must be hidden from unauthorized users. This means that the named and the secret key files must not be readable by anyone other than named process or the user running nsupdate.

Periodic key renewal is recommended to avoid compromising the secret key.

TSIG and Access Control List (ACL) strengthen DNS server's operations and the service. Usage of ACLs and TSIG is recommended to prevent unauthorized transactions, thereby providing additional security. DNS ACLs are used to allow DNS transactions (like queries and zone transfers) only from certain hosts (co-operating DNS servers and DNS clients). The zone transfers, queries and responses can be transferred confidentially under this protection, if the number of involved entities is rather small.

An ACL is a list of IP addresses and optionally TSIG keys. Instructions on how to set up TSIG can be found in the section *Configuring TSIGKeys for Sample Network 2* in *Configure DNS and ENUM*.

Operator's security policy and network topology defines whether TSIG is needed (For example, between sites when two DNS servers make zone transfers).

An example of TSIG configuration files is provided in Section 12.1 on page 33.

3.9 Using the Security-related Options Clauses in the named.conf

The configuration file `/etc/ipworks/<hostname>/dns/named.conf` should not be edited manually, but only through the CLI. An `address_match_list` is a list of one or more IP addresses, IP prefixes, or TSIG key IDs when applicable.

Check the `/etc/ipworks/<hostname>/dns/named.conf` file has not written access for the world.

```
#chmod 644 /etc/ipworks/<hostname>/dns/named.conf
```

Table 2 Security-related Options

Description	Option	Example
Change the version name to improve the protection of the DNS system	<code>options {version "Not available";};</code>	<code>modify dnsserver dns1 -add option="version \"Not available\""</code>
Specify a list of addresses that the server will not accept queries from or that are used to make a query. Queries from these addresses are not responded.	<code>options {blackhole {address_match_list};};</code>	<code>modify dnsserver dns1 -add option="blackhole {1.2.3.4;}"</code>
Restrict zone transfers. Use the allow-transfer directive to specify the hosts (slaves) allowed performing zone transfers:	<code>options {allow-transfer {address_match_list};};</code>	<code>modify dnsserver dns1 -add option="allow-trans fer {10.0.0.0/24;150.0. 0.0/24;}"</code>
Specify which hosts are allowed to notify this slave server of zone changes in addition to the zone masters:	<code>options {allow-notify {address_match_list};};</code>	<code>modify dnsserver dns1 -add option="allow-notif y {10.0.0.0/24;150.0.0. 0/24;}"</code>
Restricted recursion improves performance and partially prevents a machine from an attack known as DNS cache poisoning. An eDNS server must not allow any recursive queries from external networks. The appropriate option is:	<code>options {allow-recursion {address_match_list};};</code>	<code>modify dnsserver dns1 -add option="allow-recur sion {5.6.7.8;}"</code>
Regarding server-to-server (UDP based) communication the source server can be configured to use fixed source port 1053 (rather than a dynamic high port), with the option:	<code>options {query-source address * port 1053;};</code> <code>options {query-source-v6 address * port 1053;};</code>	<code>modify dnsserver dns1 -add option="query-sourc e address * port 1053"</code> <code>modify dnsserver dns1 -add option="query-so urce-v6 address * port 1053"</code>
BIND can be configured to only allow resolver queries from explicit hosts or networks via the option:	<code>options {allow-query {address_match_list};};</code>	<code>modify dnsserver dns1 -add option="allow-query {10.0.0.0/24;150.0.0.0/ 24;}"</code>
Internal DNS servers do not need to have any external access except to the own external DNS servers. They must forward irresolvable queries to the own external DNS server(s), with the forwarders :	<code>options {forwarders { ip_addr port ip_port ; ... };};</code>	<code>modify dnsserver dns1 -add option="forwarders {5.6.7.8 port 333;}"</code>



Description	Option	Example
Likewise, queries to internal DNS servers must be restricted to valid internal (intranet) addresses via the "allow-query" command:	<code>options {allow-query {address_match_list};};</code>	<code>modify dnsserver dns1 -add option="allow-query {10.0.0.0/24;150.0.0.0/ 24;}"</code>
If the DNS queries are to be served only in certain interfaces (IP-addresses) the listening port can be started in configured IP-addresses with the listen-on parameter:	<code>options { listen-on port 53 { address_match_list }; listen-on-v6 port 53 { address_match_list };};</code>	<code>modify dnsserver dns1 -add option="listen-on port 53 {1.2.3.4;}","1 listen-on-v6 port 53 {any;}"</code>

Note: Dynamic update is enabled on a zone-by-zone basis, by including an allow-update or update-policy clause in the zone statement.

For more details on the command of modifying DNS server options, refer to *Configure DNS and ENUM*.

3.10 Packet Filtering for the Incoming DNS Queries

When the ENUM server passes a regular DNS query to the BIND server the query source IP address will be changed to the address of the ENUM server, because the response must be first sent to the ENUM server. The original query source IP address will be appended to the query using a proprietary protocol between the two servers. This way all the BIND functionality based on the query source IP address (For example, Views, Access control lists) will be still supported in the new IPWorks configuration. The BIND server will accept the usage of the internal protocol only if the regular query source IP address is the configured ENUM server IP address. If there's no configured ENUM server IP address on the BIND server, or there is an address mismatch, BIND will merely discard the query. The ENUM server, in turn, will discard all the incoming queries using the internal protocol.

To prevent rogue DNS clients from using the internal protocol to circumvent the access control on the BIND server (access allowed to the host according to the appended IP address), a proper IP filtering action must be taken by IPTables. The concerned filtering rule must discard all the incoming queries where the packet's source IP address is that of the ENUM server.

For more information about IPTables, refer to *IPWorks IPTables Service Configuration*.

It is recommended to configure firewall rules to accept DNS queries only in those interfaces that are configured for listening on the queries. Filtering rules for both IPv4 and IPv6 addresses shall be configured.

Special care must be taken in the (uncommon) case where the BIND and ENUM servers are installed on different machines, because such a filtering rule would block the communication between the two servers if the filtering action would be taken by the IPTables on the BIND server machine. In this case it's feasible to do the filtering only in the site firewall and only in the network configuration where both the ENUM and the BIND server machines are located



behind the same site firewall, that is, their mutual communication does not traverse the site firewall doing the filtering.

IPWorks with split DNS architecture should be configured as follows:

- ENUM in iDNS forwards queries for certain number ranges to BIND, and BIND forwards those queries to eDNS. Security configurations in IPTables must allow the flow between iDNS, BIND and eDNS.
- ENUM in eDNS may forward queries to external NP database depending on configuration. Security configurations in IPTables must allow the flow between eDNS and the external NP database.

For eDNS, the site firewall is a proper place to do the filtering of incoming DNS queries from external sources; however, for iDNS IPTables should be used to prevent internal attacks not traversing the site firewall.



4 Hardening Active Select DNS

There is no integrity protection for the ICMP and SNMP interfaces used for monitoring. If the monitored node supports IPsec, it is recommended to use that for integrity protection.





5 Hardening AAA Server

AAA Server is deployed in fixed and mobile network to provide the basic AAA functions as well as application specific functions. The current realization of AAA Server in IPWorks supports the RADIUS protocol, mainly RFC 2865 and RFC 2866 and Diameter protocol (RFC3588)..

Apart from supported RFCs, IPWorks AAA server also supports the PLMN interworking with external Packet Data Networks (PDN), as specified in the 3GPP TS 29.061. For detailed description of AAA server components, refer to Section List of Sub-functions in *IPWorks Security Management*.

The AAA Server is managed through IPWorks CLI and ECLI. These interfaces are secured utilizing SSH. The IPWorks schema includes the managed objects for the AAA functionality. The AAA server has support for SNMP alarm and statistics modules.

5.1 Hardening Radius

RADIUS must be configured to run under a dedicated non-privileged user and group account, and utilize dedicated log files. All logins should be logged, but the password information must be excluded. User passwords should be protected using strong and secure one-way hash algorithm regardless the storage is a flat file, database or directory service.

The RADIUS configuration files must be owned by root with full access privileges, whereas read-only access is set for RADIUS user and group and for the rest of the users no access.

The network access should be restricted to minimum to just those networks or systems that require access. This can be achieved by IPTables, refer to *IPWorks IPTables Service Configuration*.

Each RADIUS Network Access System/Server (NAS) or Client should be identified by their IP address, not by a Network ID (hostname) to prevent DNS attacks.

Each RADIUS NAS or Client should be configured by a unique shared secret key, so that compromising one does not necessarily mean the compromising of the whole system. If a compromise occurs, it must be possible to change the shared secrets by just reloading the RADIUS stack, not the whole RADIUS server.

The shared secrets scheme used by RADIUS introduces many problems. While the secrets are stored on the RADIUS servers they must be protected against disclosure. Also the distribution of shared secrets must be organized so as not to reveal the secrets to unauthorized parties. The best way to protect the

shared secrets is to calculate the hash value for them and encrypt the data. The difficulties are emphasized with the large number of secrets to protect, as there must be own secret for each hop in a proxy chain, as the authentication scheme in RADIUS is hop-by-hop.

The recommended shared secret should be at least 22 bytes. The shared secret must not use dictionary words, guessable patterns or varying words. It should include a variety of special characters, numbers and alphabetic characters. It is recommended to use robust Random Key Generator to generate the shared secrets.

The Access-Request packet contains a 16 octet Request Authenticator in the authenticator field. The poor quality of the random number at the Request Authenticator may also cause security problems. If the same request authenticator and shared secret combination occurs more than once, an attacker may use it for replay attacks. Thus the random numbers of the Request Authenticator should be unique over the lifetime of the shared secret in use. The Request Authenticator should be both globally and temporally unique. Besides being unique, the Request Authenticator should also be unpredictable. It is recommended to use robust Random Key Generator to generate unique request authenticators.

The standard AVPs are supported in IPWorks, which are AVPs for authentication, authorization and accounting. The core part of IPWorks AAA will examine the incoming AVP and dispatch the request to different functional plug-in according to the configuration file. The functional plug-in can be PAP authentication, CHAP authentication and/or rule-based-authorization.

PAP and CHAP are considered weak authentication protocols because they use MD5 hashing. These protocols are not recommended in the long run but are used by IPWorks for now. It is also recommended to have at least 128 bit key length for encryption. If the rule base authorization is used it will not only check the incoming AVP, but also provides some logical calculation ability and can be used to specify the return code and policy reply-list.

RADIUS uses UDP as transport protocol; therefore RADIUS realizations have to manage retransmission timers themselves, as UDP provides no retransmission strategy.

Failover mechanism should have other triggering as Server Status. RADIUS server should not response to Server Status request as it is considered useless and possible harmful (DoS). Disable the Server Status altogether if possible. In IPWorks the proxy functionality is used and thus the “proxy directive” cannot be disabled to increase security.

5.2 Hardening Diameter

Diameter application must be configured to run under a dedicated non-privileged user and group account, and utilize dedicated log files. All logins should be logged, but the password information must be excluded User passwords should



be protected using strong and secure one-way hash algorithm regardless the storage is a flat file, database or directory service.

The Diameter configuration files must be owned by root with full access privileges, whereas read-only access is set for Diameter user and group and for the rest of the users no access.

The network access should be restricted to minimum to just those networks or systems that require access. This can be achieved by IP filtering, and for the applicable port numbers refer to Section *IPWorks Security for External Interfaces* in *IPWorks Security Management*.

Each Diameter client should be identified by their IP address, not by a Network ID (hostname) to prevent DNS attacks.





6 Hardening DHCP Server

DHCPv4 server uses the UDP ports 67 and 68 for the DHCP protocol, and the TCP ports 647 and 847 for the failover protocol run between two DHCP servers.

6.1 IP Address Models

Split IP address pool helps the system administrator to make observations while monitoring the IP traffic, whether it is a conventional or malicious traffic which causes problem to the end users on the network. In this model the malicious traffic is easier to explore.

The configuration of IP addresses is meaningful issue when the IP addresses are shared between the ISP operators. These shared addresses increase the network protection, integrity and confidentiality from security point view.

Four different models are possible to define when a client asks an IP address from the DHCP server.

- All IP addresses in the address pool are shared freely to every client
- Some IP addresses in the address pool are reserved for specific clients in accordance with their MAC addresses and the rests of the IP addresses are freely available for the clients.
- All IP addresses in the address pool are reserved for the clients in accordance with their MAC addresses
- IP addresses in the address pool are divided into specific IP address ranges

6.2 Adding Security to DHCP

IPWorks supports all the functionalities in DHCP Relay Agent Information Option, Reference [12], which can be used to prevent:

- IP spoofing
- Client identifier spoofing
- MAC address spoofing
- DHCP address exhaustion

This option is targeted towards environments in which the network infrastructure -- the relay agent, the DHCP server, and the entire network in which those two devices reside -- is trusted and secure.

Furthermore to minimize risks the recommended solution for preventing communication with unauthorized DHCP servers and clients is to use following techniques:

- Control carefully the physical access to the network.
- In the authentication implemented in IPWorks, the administrator needs to configure each client with a unique authentication key and then configure all these keys on the server enabling delayed authentication mechanism (RFC 3118) in DHCPv4.

Since DHCP runs over UDP and IP, IPsec could be used to provide data integrity and also origin authentication.

6.3 DHCP Multi-Server Non-Overlapping Scopes

From security point of view overlapping must be planned and handled in a controlled way. Usage of multiple DHCP servers is recommended for redundancy reasons. The administrator sets the servers different and non-overlapping scope assignments. Alternatively, the same scope can be given to each server with the definition to exclude the addresses the other server is leasing.

For instance, the entire IP address range is 192.168.56.1 to 192.168.56.254. The DHCP servers (A and B) have been assigned non-overlapping scopes to ensure that they do not conflict. Server A has server B's address range excluded and B has A's range excluded. The range of IP address 192.168.56.1 to 192.168.56.19 is reserved for the network routers, printers and other servers on the network.

The non-overlapping setting is defined as follows:

Server A excludes B's range that is 192.168.56.200 to 192.168.56.254.

Server B excludes A's range that is 192.168.56.20 to 192.168.56.199.

The main advantages of this are:

- When one server goes down the administrator can quickly remove the exclusion and let the running server access all the addresses.
- When one server runs out of addresses while another has plenty of them available, the allocations can be easily shifted.
- Misconfiguration of IP addresses pool is an unlikely situation, but possible. From the security point of view this is a risk.

Note that overlapping is acceptable within the single server, for example, in the server A.



7 Communication between the DNS and DHCP Servers

This section provides information on communication between the DNS server and DHCP server.

7.1 System Configuration

The following sections explain shortly the rules for configuring the usage of DDNS and the communication between the DNS and DHCP servers. Note that the ENUM server does not support DDNS.

7.1.1 Configuration for DHCP and DNS

It is an evident security risk if the DNS and DHCP have configurations that don't follow an agreed network plan.

DNS server should allow dynamic updates only from specific IP addresses. This can be defined in `/etc/ipworks/<hostname>/dns/named.conf` by using the *allow-update* directive or the more powerful *update-policy* statement.

7.1.2 Authenticated dynamic update

The DHCP and DNS servers (BIND) must be configured to use TSIG for data origin authentication and data integrity. The DHCP server is configured to update the appropriate in-addr.arpa zone each time a lease is let or modified.





8 Hardening ENUM server

If the queries are ENUM queries, the ENUM server tries to resolve them, and replies with the proper NAPTR RRs if resolution is successful. If the queries are regular DNS queries, the ENUM server only forwards them to the BIND (DNS) server for resolution. The External Resolution Handler (ERH) will interact with the SS7 databases like HLR, FNR, SCP, and other NPDB to get number portability (NP) information. The connection to SS7 databases will be done using SIGTRAN or over conventional circuit switched connections. For a description of ENUM management, refer to the Section *ENUM Management* in *IPWorks Configuration Management*. For information on how to configure ENUM, refer to *Configure DNS and ENUM*.

8.1 ENUM Access Control

It is not uncommon that operators' ENUM zone data is more or less private information. Especially in case of eDNS servers it's normally required to not expose their entire ENUM database to just anybody. To this end the ENUM server provides an access control mechanism that is based on Views and associated Access Control Lists (ACL). As the eDNS server will query all the ENUM information from the iDNS server, ENUM ACLs must be defined in the eDNS to restrict the access to the ENUM information.

Every ENUM zone is included at least in one View, which is connected to an ACL containing the IP addresses of the clients authorized to access the View and the included ENUM zone data. Every View is configured with a rank defining the order where the Views are searched for a queried ENUM zone that may be included in several Views.

If no View is configured for an ENUM zone, it will be included in the Default View without any access control (no ACL is applied). ENUM Views support the Split Namespace feature. In ENUM server, it is possible to have different configurations for the same number range within an ENUM Zone in different ENUM Views.

The following example shows the configuration for a case where only the DNS clients in the 10.10.0.0/16 subnet are authorized to access the ENUM zone 8.8.e164.arpa that has been previously provisioned and configured with the zone ID 5 in the ENUM servers IDs 3 and 4. The zone is included in the View "internalview" (its View ID is 2), and the associated ACL is called "acl1" (its ACL ID is 1):

```
IPWorks> create enumacl 1 -set aclname=acl1;matchlist="{10
.10.0.0/16;}"
```

```
IPWorks> create enumview 2 -set viewname=internalview;rank
=100;aclid=1;serveridlist="1,2"
```



```
IPWorks> create enumzvrel -set zoneid=5;viewid=2
```

Note: In case IPv6 is supported in the network the filtering rules and ACLs with IPv6 addresses must be configured.



9 Hardening MySQL Server

9.1 Protecting the MySQL Files

The following presents the actions that must be performed to harden a MySQL server:

Check the `*my.conf` files. They should be owned and changeable only by root:

```
# chmod 644 /etc/ipworks/mysql/confs/ipworks_datanode_my.c
onf
```

```
# chmod 644 /etc/ipworks/mysql/confs/ipworks_sqlnode.conf
```

9.2 Removing Test Database

By default, MySQL comes with a database named “test” that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Perform the following to remove the test database:

```
SC-1:~ # /usr/local/mysql/bin/mysql -P 3307 --protocol=tcp
```

```
mysql> DROP DATABASE test;
```

```
mysql> DELETE FROM mysql.db WHERE Db='test' OR
Db='test\\_%';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> exit
```

9.3 Disabling local-infile

A txt file contains different data, for example, table, in the database, and you can use this function to import data to the database. Disable this function to avoid: someone imports `/etc/passwd` to the table in the database to fetch the system OS user password.

1. Open the MySQL configuration file `/etc/ipworks/mysql/confs/ipworks_sqlnode.conf` on SC-1.

```
SC-1:~ # vi /etc/ipworks/mysql/confs/ipworks_sqlnode.c
onf
```



2. Disable the `local-infile` by adding the following information under the element `[mysqld]`.

```
local-infile=0
```

Save and exit the file.

3. Restart SQL node.

```
SC-1:~ # ipw-ctr restart sqlnodemgr SC-1
SC-1:~ # ipw-ctr restart sqlnodemgr SC-2
```

9.4 Lowering System Privilege

To protect the database, do the following:

1. Ensure that only the `mysql` and `root` accounts can access the directory `/var/lib/mysql-cluster`.

For example,

```
SC-1:~ # ls -l /var/lib/mysql-cluster

total 4
-rw-rw---- 1 mysql mysql 6 Feb 15 17:02 sqlnode.pid
```

2. Ensure that only the specific user groups can access the directory `/usr/local`. The specific user groups are defined by the administrator, for example, the user group `ipworks` or `mysql`.

Note: The reason of this action is that IPWorks makes a soft link of the MySQL binary files (saved in `/opt/ipworks/mysql/mysql`) to the directory `/usr/local`.

```
SC-1:~ # ll /opt/ipworks/mysql/mysql/bin

total 423148
-rwxr-xr-x 1 ipworks ipworks 6116146 Jan 22 15:42 innochecksum
-rwxr-xr-x 1 ipworks ipworks 486 Jan 22 15:42 mcc_config.py
-rwxr-xr-x 1 ipworks ipworks 32260 Jan 22 15:42 memclient
-rwxr-xr-x 1 ipworks ipworks 930 Jan 22 15:42 msq2mysql
-rwxr-xr-x 1 ipworks ipworks 9025348 Jan 22 15:42 myisamchk
-rwxr-xr-x 1 ipworks ipworks 8486728 Jan 22 15:42 myisam_ftdump
-rwxr-xr-x 1 ipworks ipworks 7205842 Jan 22 15:42 myisamlog
-rwxr-xr-x 1 ipworks ipworks 8603200 Jan 22 15:42 myisampack
-rwxr-xr-x 1 ipworks ipworks 6049608 Jan 22 15:42 my_print_defaults
```

3. Ensure that only the specific account (for example, `root` or `mysql`) can access the Management Node logs and SQL Node logs.



In IPWorks, MySQL stores the logs in the `/local/ipworks/mysql-cluster/mgmnode/mgm.log` and `/local/ipworks/mysql-cluster/sqlnode/sqlnode.err` files respectively.

For example,

```
SC-1:~ # ll /local/ipworks/mysql-cluster\
/mgmnode/mgm.log
```

```
-rw-r--r-- 1 root root 6440 Feb 26 14:32 /local/ipworks/mysql-cluster/mgmnode/mgm.log
```

9.5 Lowering Database Privilege

Some applications connect MySQL Server with username and password about specific database tables, so you cannot grant the general accounts with all the authorities.

Note: Follow this section to set proper authorities when you are creating a new user. Otherwise, do not change it for the existing users.

Check the user authorities:

```
mysql> show grants for <SITE_SPECIFIC username>@<SITE_SPE
CIFIC hostname>;
```

For example,

```
mysql> grant all PRIVILEGES on *.* to john@'10.170.57.76' identified by '123';
Query OK, 0 rows affected (0.00 sec)
```

Define the access authorities for an account according to the specific requirement:

```
mysql> grant select on ipworks.arecord to <SITE_SPECIFIC
username>@< SITE_SPECIFIC hostname>;
```

```
mysql> flush privileges;
```

In this example, the user can only read the arecord data from the table.

Remove some specific authorities:

```
mysql> revoke select on ipworks.arecord \
from <SITE_SPECIFIC username>@< SITE_SPECIFIC hostname>;
```

```
mysql> flush privileges;
```

In this example, the user read authority to the IPWorks arecord is removed.

Remove all authorities of an account for the IPWorks database:



```
mysql> revoke all on ipworks.* from \  
<SITE_SPECIFIC username>@< SITE_SPECIFIC hostname>;
```

9.6 Removing History

MySQL history record is stored in `~/.mysql_history`.

Remove the history record by executing the following command on the SC where the database is located:

For example,

```
SC-1:~ # cat ~/.bash_history
```

You can also disable the history-record function by executing the following command on the SC where the database is located:

For example,

```
SC-1:~ # export MYSQL_HISTFILE=/dev/null
```

Once the function is disabled, the environment variable redirects all logs to `/dev/null`.



10 Hardening Storage Server

The SS machines (For example, SuSE) are hardened according to *IPWorks OS Hardening Guide* or *IPWorks OS Hardening Guide for KVM*.





11 Hardening SNMP

SNMP supports SNMPv1, SNMPv2c and SNMPv3 versions but only the last one (v3) includes reasonable security mechanisms while the first two rely on using clear text Community Strings for access control.

For how to configure SNMP targets, refer to *Create SNMPv1 Target* , *Create SNMPv2C Target*, and *Create SNMPv3 Target*.

SNMPv3 provides a User-Based security Model (USM) and a View-based Access Control Model (VACM) employing DES and AES encryption and MD5 message authentication. However DES algorithm should not be used anymore. Therefore it is strongly recommended that only SNMPv3 is used. Refer to *Create SNMPv3 Target* for detailed instructions on how to configure SNMPv3, and the MO *SnmpTargetV3* in *Managed Object Model (MOM)* for more information about its attributes.

In case of insecure protocol versions, SNMPv1 and SNMPv2c must be used for a good reason, then a secret Community String(s) must be specified. It is possible to obtain the community name of the remote SNMP server. An attacker can use this information to gain more knowledge about the remote host, or to change the configuration of the remote system. Thus incoming UDP packets going to port 161 must to filtered via iptables. For more information, refer to the default configuration and SNMP in the template of Section *IPWorks IPTables Configuration* in *IPWorks IPTables Service Configuration*.

Note: Do not delete what has been configured in iptables.

11.1 Disabling SNMPv1 and SNMPv2

To disable SNMPv1 and SNMPv2 targets, refer to *Disable SNMP Target*.





12 Appendix

12.1 Appendix A: TSIG Configuration Example

TSIG configuration files:

```
// file /etc/ipworks/<hostname>/dns/named.conf
// Configuration for the caching-only name server ns1 using the key
// ns1-ns2-key1.example.com. towards the authoritative name server
// Note: recursion must not be prohibited
//
acl "trusted_hosts" {192.168.0.10;192.168.0.11;}; // two trusted DNS
options {
    allow-query {trusted_hosts;};
    directory "/etc/namedb"; // working directory
    pid-file "named.pid"; // store named process id
    forward only; // do not try to resolve
    forwarders {192.168.0.11;}; // forward all to ns2
};
key ns1-ns2-key1.example.com. { // Sign with this key
    algorithm hmac-md5; // queries forwarded to ns2
    include "secretkey1.conf"; // Located in working dir
server 192.168.0.11 {
    keys {ns1-ns2-key1.example.com.}; // Sign all queries to ns2
}
//

// file /etc/namedb/secretkey1.conf
secret wC0+bOAOeXGjkdhe6Sb6zg==
/ /
```





Reference List

Ericsson Document

- [1] *IPWorks OS Hardening Guide*
- [2] *IPWorks OS Hardening Guide for KVM*
- [3] *Trademark Information*
- [4] *Typographic Conventions*
- [5] *Glossary of Terms and Acronyms*
- [6] *Configure DNS and ENUM*
- [7] *IPWorks DNS, ASDNS, ENUM Parameter Description*
- [8] *IPWorks Configuration Management*
- [9] *IPWorks IPTables Service Configuration*
- [10] *IPWorks Security Management*

Other Reference

- [11] [Secret Key Transaction Authentication for DNS](#)
- [12] [DHCP Relay Agent Information Option](#)