Study of Tools for Network Discovery and Network Mapping

Sergei Bantseev and Isabelle Labbé
Communications Research Centre

Prepared by:

Communications Research Centre
3701 Carling Ave, Ottawa ON K2H 8S2

Contract number: A1410FE965
Contract Scientific Authority: Joanne Treurniet

Defence R&D Canada – Ottawa

Contract Report

DRDC Ottawa CR 2006-302

November 2003

Study of Tools for Network Discovery and Network Mapping

Sergei Bantseev
Isabelle Labbé

Communications Research Centre
Ottawa, Canada

November 14, 2003

Contacts

DRDC-O Responsible Manager

Joanne Treurniet
NIO Section
DRDC Ottawa
3701 Carling Ave.
Ottawa, ON K1A 0Z4
Phone: (613) 990-7096
Email: Joanne.Treurniet@drdc-rddc.gc.ca

DRDC-O Alternate POC

Marc Grégoire
NIO Section
DRDC Ottawa
3701 Carling Ave.
Ottawa, ON K1A 0Z4
Phone: (613) 998-2113
Email: Marc.Gregoire@drdc-rddc.gc.ca

Technical Point of contact

Isabelle Labbé
Network Systems Division
Communications Research Centre
3701 Carling Ave.
Ottawa, ON K1A 0Z4
Phone: (613) 998-1410
Fax: (613) 998-9648
Email: isabelle.labbe@crc.ca

CRC Responsible Manager

Frederic Massicotte
Network Systems Division
Communications Research Centre
3701 Carling Ave.
Ottawa, ON K1A 0Z4
Phone: (613) 998-2843
Fax: (613) 998-9648
Email: frederic.massicotte@crc.ca

Executive Summary

The work presented in this report is related to the DRDC Joint Network Management and Defence System (JNMDS) Technology Demonstrator Project. This work is an investigation of the currently available tools that are capable of performing automatic network discovery in an IP-based network. For the proposed system, it is required to provide an assessment of the existing tool's capabilities in identifying: the network topologies (map of physical links and logical links), the network resources (network elements and the configuration information) and the network services (network applications and system support). In particular, the study addresses the following issues: how is automatic network discovery achieved by the existing tools, and what is discovered.

A number of tools from the commercial sector (COTS), the open-source community and the research/academic community were identified. Based on the main auto-discovery techniques that are implemented by the tools, four tool categories were defined. These are: the Active SNMP-based, the Active Hybrid, the Passive only and the Inventory & Audit tool category. For each of the four categories, a number of tools from the three sources were selected for further study. For the selected tools, a documentation-based evaluation of their auto-discovery capabilities was performed. The report presents the detailed evaluation of seventeen tools. The results of the evaluation are summarized in two characteristic tables.

One thing that is apparent when looking at the outcome of the study is that although some tools present good capabilities, they all have their strengths and weaknesses. Within the scope of interest, the "one tool does it all" solution does not exist. It is reasonable to expect that for the Technology Demonstrator Project system, the solution is likely to consist of an integrated suite of tools where functionality of each tool will be combined to achieve the desired capability.

Table of Contents

List of Figures

List of Tables

1 Introduction

The work presented in this report is related to the DRDC Joint Network Management and Defence System (JNMDS) Technology Demonstrator Project. The proposed system will use a network map to display information required by network administrators, analysts and operators, as well as commanders using the networks to plan missions. Network information and status, mission information, defensive posture, security events and potentially course of action will be overlaid on this view. In brief, the tools and techniques that can create this view must be investigated.

This report is a study of the techniques used by these tools to perform network auto-discovery and network mapping. A survey of commercial off the shelf and open source products that are capable of automatically discovering network hosts and possibly their interconnections is presented. Products that clearly offered the desirable characteristics were selected for a detailed documentation review. For the selected tools, a documentation-based evaluation of the auto-discovery capability was performed.

This report is structured as follows: Section 2 gives the scope of the study. Section 3 presents the methodology. Section 4 presents a review of the main auto-discovery techniques. Section 5 provides a short description of the tools by category. Section 6 explains how the tool selection was conducted. Section 7 describes the evaluation criteria against which the selected tools were evaluated. Section 8 presents the evaluation of each selected tool along with some general observations. Section 9 is a general discussion on various elements to consider when performing network discovery. Section 10 addresses emerging trends in the network discovery and network mapping fields for the upcoming years. Finally, Section 11 presents a conclusion to the study.

2 Scope

This work is an investigation of the currently available tools that are capable of performing automatic network discovery in an IP-based network. The field of automatic network discovery encompasses a broad range of aspects. In the context of the Technology Demonstrator, it is required to provide an assessment of the existing tool's capabilities in identifying: the network topologies (map of physical links and logical links), the network resources (network elements and the configuration information) and the network services (network applications and system support). In particular, the study addresses the following issues: how is automatic network discovery achieved by the existing tools, and what is discovered. Automated display of the network information obtained in the discovery stage is often included as part of a tool, however this attribute is not considered to be a critical evaluation criteria.

The report is not a product comparison. In some cases, in addition to network discovery, the tool presented other capabilities (e.g. network management). In such cases, only the tool's network discovery feature was evaluated.

Finally, it should be mentioned that claims of the efficiency, accuracy or completeness of the tool were not verified. This would have required running the tool in an actual testbed. This was beyond the scope of the study.

3 Methodology

As mentioned above, the tools were not actually deployed nor evaluated in a testbed network. As such, the study is limited to the information that could be gathered from the following sources:

- documentation taken off the Web (tool's official site)
- email exchanges with the companies
- phone conversation with technical people of the company
- tool's user guide (when available)
- in-house experience of tool usage (e.g. HPOV, SNMPc, IBM Tivoli)

It is important to mention that for many commercial products, the information included in the report is limited by the technical content that the company agreed to release.

4 Auto-discovery techniques background

Inherent properties of computer-networks allow for two approaches to network discovery. The first approach uses "ask and tell" methodology, and is referred to in this report as *active*. The second approach uses "listening" methodology and is referred to in this report as *passive*.

In active techniques, the discovery process actively sends traffic to the network devices to stimulate a response whereas in passive techniques, the discovery process does not introduce any traffic of its own on the network, it strictly listens.

The following sections present an overview of active and passive techniques.

4.1 Active Techniques Overview

Active techniques generate certain "probe" packets that stimulate response from network nodes. Data from resulting responses and behaviour of the responses are used to derive discovery information. The probe packets are generated using legitimate networking protocols, including protocols such as: SNMP, ARP, ICMP, TCP, DNS, NetBIOS, etc. Of the mentioned protocols, SNMP has the capability to provide the most information for the discovery process and as such, is the most widely used by tools that offer active auto-discovery feature. The discovery information obtained via probe messages represents the information at a discrete point in time, i.e. probe messages generate a "snap shot" view of

the information. Therefore in order to maintain a current network view, probe messages need to be generated periodically. Active techniques provide fast discovery of information at the expense of network bandwidth resources. Active techniques also provide the ability to discover remote networks. Active techniques may, however, be limited by filtering devices such as firewalls (e.g. a firewall installed on a machine may block "ping" requests).

The following section presents an overview of the most commonly used network protocols by active techniques. Within the scope of this report, from the very early survey results, SNMP's presence was strongly felt. Today a very large proportion of commercial tools that offer network discovery either partially or fully depend on SNMP to deliver their functionalities. Therefore, protocols used by active techniques were divided into SNMP and non-SNMP categories and are presented in the following subsections.

4.1.1 SNMP Overview

The Simple Network Management Protocol (SNMP) is part of the TCP/IP suite of protocols and was designed to facilitate exchange of management information between network devices. SNMP is an application layer protocol that was designed to operate over UDP. Today, there are three versions of SNMP: SNMPv1, SNMPv2, and SNMPv3.

A typical network managed via SNMP is depicted in Figure 1 and consists of four main elements: managed devices, agents, Network Management System (NMS), and Management Information Base (MIB).

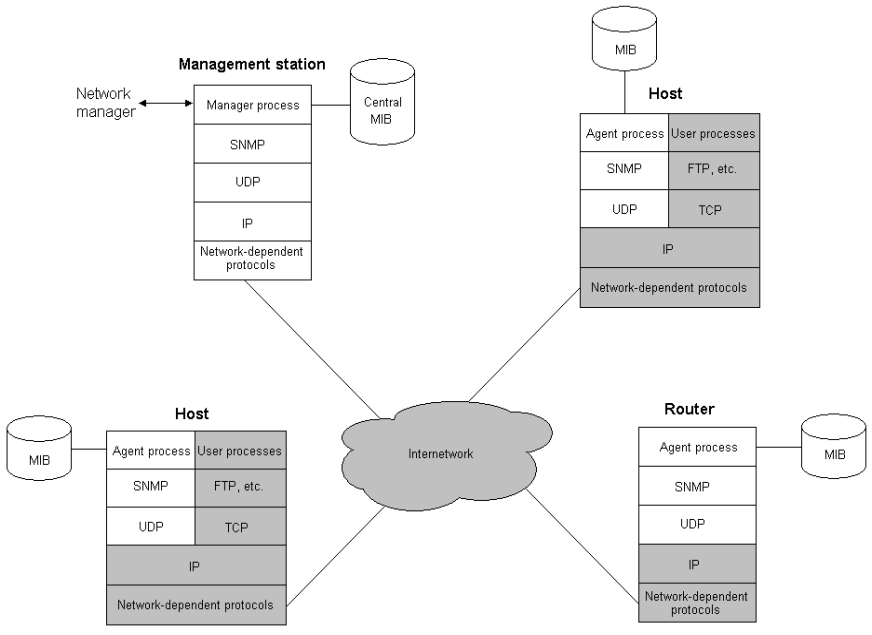

Figure 1: SNMP-managed network

A *managed device* is any network node that hosts an SNMP agent and a MIB. Examples of managed devices are: personal computer, printer, router, switch or a bridge, etc.

An *agent* is a software module that runs on a managed device. An agent's functionality includes populating and maintaining the local MIB and to communicate with the NMS via SNMP.

The *NMS* provides capability and user interface for monitoring and controlling managed devices. It also populates and maintains the central MIB. At least one NMS must exist on any SNMP-managed network.

A *Management Information Base* (MIB) is a hierarchical collection of well-defined information objects, called managed objects. MIB objects are accessed via SNMP for monitoring and configuration purposes. Managed objects within a MIB represent various attributes of a managed device and are identified by unique object IDs. Every object within a MIB is defined in a formal way. The Abstract Syntax Notation One (ASN.1) is used to define each individual object and also to define the entire MIB structure. There exist three main types of MIBs:

1. The standard MIBs

Standard MIBs are created by the IETF and documented in various RFCs. A number of standard MIBs exist. The most well known MIBs are the MIB-I (rfc 1156) and the MIB-II (rfc 1213). MIB-II is actually a superset of MIB-I. MIB-II defines configuration and performance information for products that have TCP/IP interfaces. MIB-II consists of a basic set of 9 groups. For a detailed description of those groups and the list of supported values, please refer to RFC 1213. These groups are the basic unit of conformance. If the semantics of a group is applicable to an implementation, then it must implement all objects in that group. In addition of the basic set, there exist a number of MIB-II Extensions. It is not mandatory that Extensions be implemented by vendors.

2. Experimental MIBs

Typically, experimental MIBs are MIBs that have been developed as part of research projects. They are constructed for a particular application. They are not intended for operational use. They are often not yet mature. As they gain in popularity, they may be submitted to become a standard.

3. Proprietary vendor MIBs/private MIBs

The Internet Standard and Experimental MIBs do not cover the entire range of statistical, state, configuration and control information that may be available in a network element. This information is, nevertheless, extremely useful. Vendors of network devices generally have developed MIB extensions that cover this information. These MIB extensions are called Vendor Specific MIBs.

4.1.1.1 SNMPv1

SNMPv1 is widely used in the Internet community and functions within the specifications of the Structure of Management Information (SMI). SNMPv1 is defined in RFC 1157 and SMI is defined in RFC 1155. The SNMPv1 SMI specifies that all managed objects have a certain subset of ASN.1 data type associated with them.

SNMPv1 is a simple protocol of requests and responses. The Network Management Station (NMS) issues requests to devices of interest and managed devices return appropriate responses. This functionality is provided via four protocol operations: Get, GetNext, Set, and Trap. The Get and GetNext operations are used by the NMS to monitor an agent's MIB values, the Set operation is used by the NMS to set certain values in the agent's MIB for configuration purposes, and the Trap operation is used by agents to inform the NMS of an event or a reached threshold condition.

Two different text-based passwords are used to authenticate the management station, one password for Get operations and the other password for Set operations. The passwords are transmitted across the network in the clear, i.e. not encrypted, and can be easily obtained by an unauthorized person.

4.1.1.2 SNMPv2

In addition to the four protocol operations in SNMPv1, SNMPv2 offered two new operations: GetBulk and Inform. The GetBulk operation was designed for a more efficient retrieval of large blocks of data from an agent's MIBs by the NMS. If the agent responding to the GetBulk operation is not able to provide all the requested values, it will provide partial results. The Inform operation was designed to allow sending of trap information from one NMS to another NMS and to receive a response. In addition to the new protocol operations, SNMPv2 SMI introduced changes to bit strings, network address, and counter data types. Detailed information on the SNMPv2 SMI can be found in RFC 1902.

The authentication method of management station and managed devices remained the same as in SNMPv1.

4.1.1.3 SNMPv3

SNMP version 3 was introduced in 1997 to overcome the lack of security in SNMPv1/v2. SNMPv3 addressed many of the security problems that versions 1 and 2 did not. SNMPv3 is defined in a set of specifications. The SNMPv3 specifications were approved by the Internet Engineering Steering Group (IESG) as full Internet Standard in March 2002. The SNMPv3 set of specifications can be found in [1]. The SNMPv3 standard uses a different paradigm with new terminology to model and describe the exchange and processing of network management information. However, the underlying SNMP

mechanisms for version 3 are very similar as those found in version1 and 2. SNMPv3 does not redefine the SNMP protocol, but instead uses what is found in version 2 and adds a layer of security. It is often expressed as follows: "SNMPv3 is SNMPv2 plus security and administration". The SNMPv3 protocol data unit format reflects this approach. A SNMPv3 message is comprised of a SNMPv2 message, SNMPv3 security header, and SNMPv3 overhead header information [2].

With SNMPv3 it is still possible to use a non-encrypted community name or password for authentication just as in SNMPv1 and v2. However, there are provisions within SNMPv3 to encrypt not just the password but also the entire SNMP protocol data unit.

SNMPv3 uses operator specified values for the following:

- Username: Account name used to access MIB data
- Auth: Essentially the password used along with the Username value. If specified, MD5 or SHA encryption can be used.
- Privacy Password: used to encrypt and un-encrypt the SNMP PDU

In summary, SNMPV3 adds secret-key based authentication and encryption services to the functionality of SNMPv2.
The authentication mechanism in SNMPv3 employs HMAC-MD5 and HMAC-SHA algorithms and ensures the authenticity of the originator of the received message. In addition, this mechanism also ensures that that the received message was not altered while in transit.
The encryption mechanism in SNMPv3 is based on the data encryption standard (DES-56).

4.1.1.4 Strengths and Weaknesses of SNMP

The strength of SNMP is indeed its simplicity. Today, virtually all vendors of networking equipment provide SNMPv1 and v2 agent software for their equipment. SNMPv3 support is growing but it is not as common as SNMPv1/v2.

Despite SNMP's wide availability, the main drawback of both SNMPv1 and SNMPv2 is their lack of security. With the "community names" being sent across networks as plain text, they can be easily learned by an unauthorized user (e.g. via the use for example of a packet "sniffer" program). As such, the use of SNMPv1/v2 creates security vulnerability. With the proper "GET community name", an unauthorized user can obtain/read the configuration information of all the SNMP-enabled devices of the network. Moreover, with the proper "SET community name", an unauthorized user can use SNMP protocol messages to potentially misconfigure SNMP-enabled devices of the network (e.g. change routing table entries of a router). Because of this drawback, many deployed SNMPv1/v2 management systems suppress SNMP's configuration capability and simply use SNMP for monitoring purposes only. On some networks with high security requirements, SNMP-based network management is not permitted even for monitoring purposes.

4.1.2 Non-SNMP

A number of communications protocols other than SNMP are also used to discover network devices. This section summarizes the most commonly used.

4.1.2.1 ICMP

The Internet Control Message Protocol (ICMP) uses several message types that assist TCP/IP communications. For example, it can be used to determine whether a machine is alive on the network. An ICMP echo request is sent to an IP address (e.g., through *ping*), and if there is a machine at that address, it will send back an ICMP echo reply. The *traceroute* tool also uses ICMP to reveal a particular hop-by-hop path between two hosts. As well, the third-party tool Xprobe [3-4] takes advantage of minor variations in ICMP implementations to determine what operating system is running on a networked host, based on specific replies to ICMP probes. In brief, the ICMP protocol can be used to find:

- The device status (host up/down)
- The network mask
- The device type (e.g. router)
- The OS
- The hop counts (topology info)

4.1.2.2 TCP

The Transmission Control Protocol (TCP) was designed to provide reliable, connection-oriented communication over the connection-less and unreliable IP layer. It implements control and signalling messages to establish, maintain and terminate a connection between two nodes. Active discovery techniques leverage TCP signalling messages to discover information. Different implementations of the TCP/IP stack may respond slightly differently to network probes. The small differences in response to these probes can indicate what specific Operating System is running on a network computer. This technique is sometimes called “OS Fingerprinting”. As well, tests can be performed on ports to discover what network services respond. This reveals what services are running and which ports are opened or closed. In brief, the TCP protocol can indicate:

- The OS
- The device status (up/down)
- The services
- The opened/closed TCP ports
- The MTU size
- The System uptime (OS dependant, done via the TCP “timestamp” option field)

4.1.2.3 ARP

The Address Resolution Protocol (ARP) was designed to provide mapping between IP and MAC addresses. MAC address is required for every transmitted packet. When a host doesn't know the MAC address of the next-hop, it broadcasts an ARP request asking for the MAC address associated with the IP address of interest. The machine with the IP address of interest will reply to the ARP request and provide its MAC address. Therefore, ARP protocol is used by active techniques to learn MAC addresses for the discovered IP nodes. ARP can be used to find:

- The device status (up/down)
- IP and MAC relationship
- The OS (very limited number of signatures)

4.1.2.4 DNS

The Domain Name Service (DNS) provides a mapping between numeric IP addresses and textual hostnames, via a distributed database. Active techniques use DNS services to find the name of a host from its IP address or vice-versa, by contacting the name server. This allows providing hostname labels for the discovered nodes. The DNS protocol can be used to find:

- The IP and Name relationship
- The list of possible active computers
- Possibly the OS (DNS includes a field for this value, however it is usually unused)

4.1.2.5 NetBIOS

The Network Basic Input Output System (NetBIOS) is a protocol for supporting services over local area networks. It was designed to support inter-application communications and data transfer such as file sharing. NetBIOS can be used to discover resources offered by a network server. Many systems use NetBIOS including Microsoft's Windows systems, IBM's OS/2, Novell NetWare and Unix-like system via the SAMBA package. By making use of the various queries/replies implemented by the protocol, the following information can be found:

- The name (different than the DNS name)
- The OS type (limited to Windows and system running a SAMBA server)
- A list of the shared drives
- The users
- The services (limited to a Windows network)

4.2 Passive Techniques Overview

Unlike active techniques, passive techniques employ listening devices called "sniffers" to obtain their discovery information. Sniffers derive discovery information by parsing protocol data units (PDUs) as well as by observing behaviours of certain protocols (e.g. TCP). They do not generate any data packets on the network for the purpose of discovery. They are thus not limited by filtering devices such as firewall. They may however generate data traffic to communicate their findings to a central entity when distributed sniffers are deployed. Since a single sniffer device can only "hear" data traffic on its network segment, passive discovery techniques often deploy distributed sniffer devices to cover the target network. Passive techniques present a real-time view of the network (a movie as opposed to an active scanner's slide show). However nodes that do not transmit or receive data will not be discovered by passive techniques. Passive techniques can detect all active services, not just those on well-known ports. They can also detect all active network devices, not just IP-based hosts. They employ no host-based elements to deploy and/or manage and/or update. Finally, they work in heterogeneous environments where host technologies do not exist. Because of the way they operate, the passive methods will generally accumulate the information over a longer period of time.

The most commonly parsed protocols include: ARP, IP, RIP, OSPF, EGP, ICMP, TCP (DNS, FTP, SMTP, TELNET, HTTP,…) and UDP. Given the ability to parse these PDUs, passive techniques can provide information on the devices and the network. Examples of the information they can provide are:

- Device status
- Device type
- Device IP address
- Device MAC address
- Device OS
- Device hostname
- Services (associated to ports/protocols info)

To provide details on how the above information is actually obtained using passive techniques is beyond the scope of this report. A number of documents have been written on the subject. A few references [5], [6], [7], [8] are given at the end of the report.

5 Tool categories

The main auto-discovery techniques were briefly presented in the previous section. Based on these existing techniques, a number of tool categories were defined. The categories are explained below:

5.1 Active SNMP-based

This category includes the tools that strongly rely on SNMP to perform discovery and collect the information. As such, these tools require that a majority of the network devices be SNMP-enabled. When that is not the case, most of these tools will be able to discover little information of interest. Typically, the discovered information will be limited to the IP address and status of the node.

Since SNMP is the only widely deployed and supported network management standard, most commercial network discovery products (COTS) belong to this category. The tools in this category are largely tied to the network management area.

5.2 Active Hybrid

This category includes the tools that perform discovery by making use of multiple active techniques and/or combined active/passive techniques.
A tool may include SNMP as part of its active discovery protocols but it does not mainly rely on SNMP for its operation. The tool is capable of providing sufficient interesting information even if SNMP is not supported on the network devices.

Typically the tools that form this category implement active discovery techniques other than SNMP or in parallel to SNMP. They may (and often will) also combine passive discovery by monitoring Internet protocols.

5.3 Passive only

The tools in this category discover the information by strictly monitoring the network traffic. Their capabilities are generally more limited, but they offer some unique advantages (ref. Section 4).

A large proportion of these tools is open-source. They are mainly developed by individuals or a small community of people.

Often, these tools have the characteristic of being very focused i.e. they tend to address a specific discovery area (e.g. OS discovery, application flow tracking,…) as opposed to covering network discovery in general. In particular, a number of tools that use passive methods were found to specialize in the performance area. Many of them were developed to provide network traffic statistics and application performance information only. Because of their limited functionality, these specialized performance tools were not included in the study[1].

---

[1] Examples of such tools are: NetScout, Network Associates, HP Netmatrix, ntop, MRTG,…

Most of the tools in this category are tied to or issued from the network security community area.

5.4 Inventory & Audit tools

This category includes specialized tools which functions are to perform a hardware and software inventory of the devices on a network. These tools do not automatically discover network hosts but they have the ability to provide very detailed information about the hosts.

Usually, proprietary software agents need to be installed on each computer. The agent will often have access privileges and be tailored to the specific host OS (all well-known OS such as Windows, Linux, MAC and Solaris are usually supported). This allows the agent to collect detailed information about the host. The agent regularly forwards the information back to a central server.

These tools provide node configuration information only (no network information). The information they can provide is often impossible to obtain via traditional methods, for example the hardware used on a host, or software installed on a host. Therefore, they can constitute a useful complement to the network discovery tools. Provided the accessibility of the data is not an obstacle, an integrated solution could be developed, where the functionality of both tools could be combined.

6 Tools selection

A survey to identify the various network discovery tool candidates was conducted. Three main sources were considered: the commercial sector (COTS), the open-source community and the research/academic community. The commercial sector provides the current state-of-the-art, the open-source offers customization and features that may not be found in COTS whereas the research/academic community is a good indicator of the trends and potential innovative solutions.

The list of tools that were identified are shown in Annex A. For each of the four categories (as defined in Section 5), a number of tools from the three sources were selected for further study. The selection was based according to the following criteria:

1- **Trusted source**:
   The tool has been developed by a well established/known company or organization. In the case of open-source, the tool is widely used, has a good reputation and/or is being developed/maintained by a community of people (not only by an isolated individual).

2- **Completeness of information**:
The tool is able to present enough information of interest with regards to network discovery.

3- Level of support/compatibility with other software:
The tool presents a good level of extensibility to allow/ease potential integration with other software.

4- Unique features of interest:
The tool may not exactly satisfy the first three criteria stated above but it presents interesting unique feature(s) that show capabilities not found in other tools.

The SNMP-based category constitutes by far the largest category. In this category, many COTS products are direct competitors to one another and they all present very similar functionality. When evaluated against the selection criteria, many of them could have been equally selected. It was decided to select a subset that best covers all the functionality/capabilities provided by these tools and be well representative of that group.

A number of open-source tools that belong to the two categories "active hybrid" or "passive only" were identified[2]. Many of these tools, although related to network discovery, do not meet the selection criteria. Many of them only address a very narrow aspect of network discovery (lack completeness of information) and/or although the code can be examined, such tools have been developed by an individual (not a trusted source). These tools are often not mature and lack support and trust. As a result, it was decided that only a few well-known open-source tools would be selected for these two categories[3].

Only one commercial product that met the selection criteria could be identified in the "passive only" category. The tools that were selected in each category are shown in Table 1. Each tool is presented in Section 8. Note that time limited the number of prototype tools that could be identified for the research category. Many research papers address the field of network discovery in some way or another. A good survey of all the techniques that have been published on the subject could not be performed in the allotted time.

[2] Ref. Annex A, open-source tools table.
[3] For the reasons stated, many open-source tools were not evaluated in this report. Nevertheless, it is desirable to keep aware of the existence of these tools and monitor their evolution as new capabilities and improved features may come out of them.

Table 1: The tools selected for this study

Category	COTS	Open Source	Research
Active SNMP-based	HPOV Node Manager IBM Tivoli NetView CastleRock SNMPc BMC Patrol Visualis Micromuse NetCool Precision	OpenNMS Nomad	
Active Hybrid	Fluke OptiView FoundStone	Nmap Cheop-Ng Big Sister	CRC network mapping tool
Passive only	IPSum Route Dynamics	RNA/Snort	
Inventory & Audit	LANDesk LanAuditor iInventory		

7 Evaluation criteria

In the context of the Technology Demonstrator, a network auto-discovery tool should have the capability to: provide a comprehensive map that spans several segments (i.e. host separated by routers), give detailed levels of node inter-connectivity and identify every network components along with their state and configuration information (software, services,…). The tool should also provide dynamic updates of this information (real-time updates). In addition, the data collected by the tool should be accessible.

Based on the above desired capabilities, two characteristic tables were constructed: a network characteristic table and a node attributes table. The tables are presented in Section 8. The network table lists the evaluation criteria that address capabilities related to "how" network discovery is performed. The node attributes table lists the evaluation criteria that address capabilities related to "what" information is discovered (namely node configuration information).

This Section presents the evaluation criteria of each table. Each item included in the network table is presented in Section 7.1 whereas each item included in the node attributes table is presented in Section 7.2.

7.1 Network table

Each item listed in the network table and against which the selected tools were evaluated is explained below.

7.1.1 Visual map

A tool that supports visual mapping has the capability to present a visual display of the network topology, showing the discovered hosts and their interconnectivity. Host interconnectivity can generally be shown at two levels:

Level 3 or network layer:
A layer 3 map is characterized by its subnets. It displays devices that operate at level 3 such as routers, PCs, printers, IP phones, etc.
A layer 3 map provides the logical connectivity of devices. It will typically show the routers and the host clusters (subnets) connected to the routers' interfaces.

Level 2 or data link layer:
A layer 2 map is more revealing than a layer 3 map. It is concerned with the link-layer connectivity of devices in a switched network. A layer 2 map will illustrate the switches' interconnectivity as well as how the hosts are distributed with regard to the switches.

In this report, only the ability of the tool to support visual mapping and the level at which it can generate the map was evaluated. No evaluation was made on the actual accuracy and completeness of the displayed map as this was beyond the scope of this study.

7.1.2 Auto-discovery of devices

The auto-discovery is the capability to automatically discover network components. It generally lies at two levels:

Level 3 or network layer:
This level includes the discovery of devices that is concerned with OSI layer 3. This includes all devices that have at least one IP address.

Level 2 or data link layer:
This level includes the discovery of devices that is concerned with OSI layer 2 (e.g. switches). Typical Level 2 information is: hardware MAC address (which all network devices have), the port/MAC address association (of a switch) and VLANs information.

7.1.3 Scope of Auto-discovery

The scope of auto-discovery relates to the type/structure of the network the tool is capable of covering. The auto-discovery process can span across LANs and WANs:

LAN: capability to discover nodes in the local broadcast domain i.e. up to the local gateway.

WAN: capability to discover nodes beyond local broadcast domain, including local and remote routers. Discovery tools offer configuration option to define the limits of how far into the WAN the auto-discovery will proceed, since if no limit is given, the tool may attempt to auto-discover nodes in the entire Internet. For example, this limit can be specified by explicitly providing network ranges of interest.

7.1.4 Technique of Auto-discovery

The two main techniques for performing auto-discovery have been presented in Section 4. These are:

Active: The discovery process actively sends traffic to the network devices to stimulate a response.

Passive: The discovery process does not introduce any traffic of its own on the network, it strictly listens.

7.1.5 Mode of Operation

The mode of operation relates to the high level architecture of the tool i.e. the approach the tool uses to perform its active and/or passive network discovery.
There appears to be two dominating modes of operation that tools implement: “agent-less” and “agent-dependent”.

An agent is a piece of software that runs on a network device. The functionality of the agent is to provide discovery information about that device, either network related information or node attributes specific. The agent responds and/or reports to a central entity.

a. Agent-less mode:

In the “agent-less” mode, the tool does not require that separate agents (SNMP agents or others) be running on the network devices for the discovery process to succeed to a satisfactory level. To achieve monitoring and data collection, the tool does not use/rely on any proprietary agents or standard agents such as SNMP agents.
Multiple instances of the tool or some of its components may be distributed in the network.

b. Agent-dependent mode:

In the “agent-dependent mode” the tool requires that agents (either proprietary agents or standard agents such as SNMP agents) be installed and run on the network devices for the discovery process to succeed to a satisfactory level.

In some cases, the nodes that are automatically found, will be limited to those hosting the agents. This is typically the case for the tools of the Inventory & Audit category.

Agents certainly provide a higher level of monitoring. They can provide, at times, very detailed and critical information but they can also be cumbersome to install, difficult to administer and sometimes impossible to get installed on certain machines. An agent-less approach may be best suited to an heterogeneous environment (likely to work on more devices) than an agent approach.

7.1.6 Supported Architecture

The architecture can be defined by the way the various components of a tool are deployed and configured in order to achieve discovery.
In general, the types of architectures supported by the network discovery tools can be categorized as follows:

a) Central

In the central architecture, the entire functionality of the tool is installed on one platform i.e. a single node executes all the discovery, monitoring, data collection, administration and display functions that are supported by the tool. The node performs these functions for the entire network domain of interest.

The main advantage of a central architecture resides in its simplicity of deployment. Since all the tool's components are localized in one place, it eases the configuration and the management effort. This type of architecture is well suited for small local area networks.

On the other hand, the central architecture presents a number of issues. Firstly, it can generate a lot of traffic on the network[4], affecting the performance of the network and the well functioning of other applications. Secondly, it does not scale well. Obviously, it is an approach that can handle a limited number of nodes because of the burden imposed by the amount of information to process. Finally, it is not a very robust solution as it offers a single point of failure.

The network size and the network layout will often dictate the type of architecture that would be desirable to deploy. A distributed architecture, for example, offers a better approach to large networks. It presents a much more efficient and scalable approach. The processing of the information is divided among multiple platforms which allows parallel handling of a greater number of nodes.

As well, if remote networks need to be discovered, a distributed architecture will be preferable. Access to remote networks is usually performed over much lower bandwidth

[4] This applies to tools that use active techniques. This problem is generally experienced during the initial discovery phase.

links (e.g. ISDN, satellite links). Such links are slow and generally expensive to use. The deployment of a distributed architecture will prevent congestion of the links by largely decreasing the amount of data that must travel over the links.

The various types of distributed architectures are described below.

b) Distributed Client-Server

The client server architecture is depicted in Figure 2.

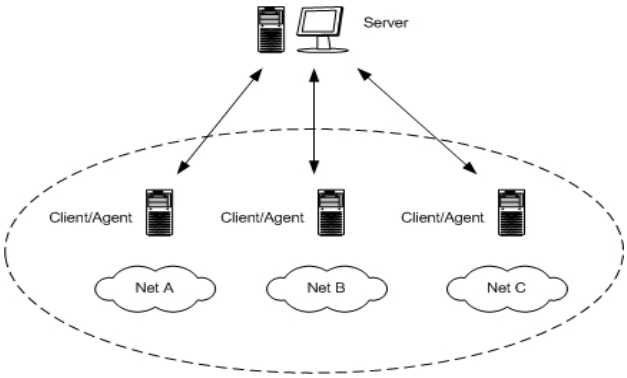

Figure 2: Distributed Client-Server architecture

In the Client-Server architecture, the tool's functionality is split in two distinct entities. The clients (usually proprietary software although some solutions come in the form of hardware appliances) are distributed throughout the network. Each client is responsible for a well-defined portion of the entire network of interest. The client's functionality is limited to the "data gathering" i.e. its role strictly consists in performing the discovery and monitoring functions on a portion of the network. It usually does not present any display of the information. Each client reports back the collected information to a server. The distributed clients do not have knowledge of the global network.

One machine acts as a server. The server collects information from the associated clients. The server processes the data, summarizes and correlates the information. It is also responsible for presenting the information to the user (configuration, tables, graphs,…) and displays a visual map when supported. The server has knowledge of the global network.

The communication and data exchange between the client and the server will, in most cases, be done using a proprietary TCP/IP protocol. In some cases, standard protocols such as Remote Procedure Call (RPC) or HTTP (Web-based) are used.

c) Distributed Hierarchical

The distributed hierarchical architecture is depicted in Figure 3.

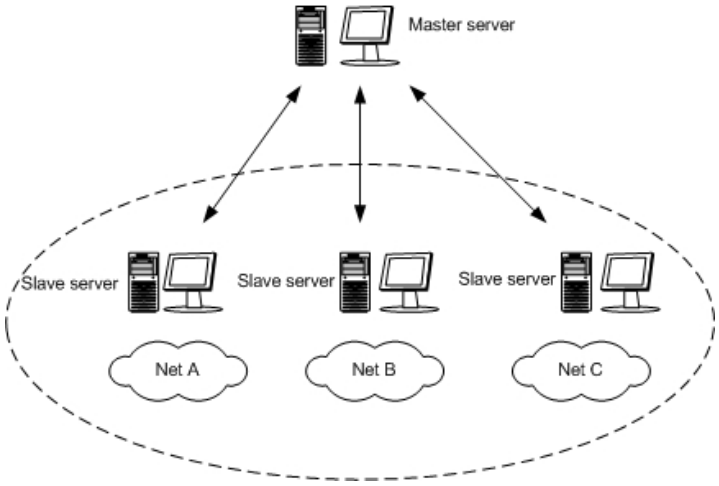

Figure 3: Distributed hierarchical architecture

In this architecture, the tool is installed on multiple nodes throughout the network. Each node is responsible for a sub-network of the entire network of interest and the node's action is restricted to this portion of the network. Each node implements the entire tool's functionality i.e. discovery, monitoring, data correlation, display of the information (visual map if supported),… Each node is independent and acts as a server. The hierarchy is reflected by the fact that nodes can be configured to be "slaves" or "masters". Each slave feeds its information back to a "master" machine. The master machine combines the knowledge of all slaves/sub-networks.

This architecture is typically deployed when remote networks are involved. Although each remote network may be locally managed, a central group (e.g. a Network Operating Centre) may require the have a global "view".

A hierarchical architecture has the advantage of offering a scalable approach. It clearly accommodates networks that may be geographically dispersed or too large to monitor from a single location. Also, a hierarchical approach can significantly reduce traffic induced by the discovery tool.

d) Distributed Peer-to-Peer

The distributed peer-to-peer architecture is depicted in Figure 4.

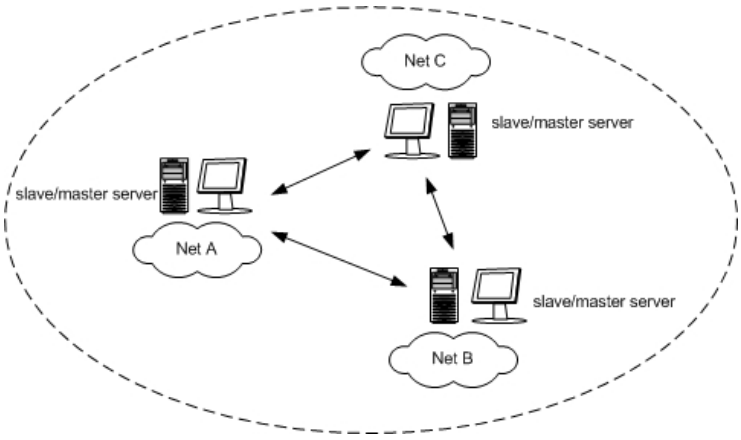

Figure 4: Distributed peer-to-peer architecture

The fully distributed peer-to-peer architecture is very similar to the distributed hierarchical architecture previously described. In addition, each server node can be configured to act simultaneously as both "slave" and "master". Each node has now a complete knowledge of all networks (they all exchange their views). A peer relationship exists.

7.1.7 Protocols Used for Active Auto-discovery

This item is a list of the network protocols that are used by the tool when performing active auto-discovery. The most commonly used are:

SNMP version 1, 2 and 3
ICMP
DNS
ARP

These protocols have been presented in Section 4.

7.1.8 Performance monitoring

Performance monitoring is defined here as the tool's capability to present to the user measurements related to network performance and/or application statistics. Examples are: network throughput, network utilization, application bandwidth, etc.

When initially conducting the tool survey, many tools that are specialized in the performance monitoring field were encountered. However, as already mentioned, it was decided to exclude these tools from the study as it was felt that their focus was too narrow. Instead, it was decided to include performance monitoring as a tool's feature.

Therefore, it is considered a tool's asset if, in addition of network discovery, a tool is also capable of reporting on network performance/application statistics.

7.1.9 Extensibility

Extensibility is defined as a built-in capability of the tool, which promotes/allows further extension of the tool's functionality.
Extensibility is especially important when the tool may be used, for example, in conjunction with other systems.

The extensibility of each tool was evaluated by verifying mainly two items:

1. APIs:

APIs are software Application Programmable Interfaces. The integration with other tools may only be possible via supplied programmable interfaces or plug-ins development. This is generally the case for commercial products. Since the source code is not available, APIs must exist in order to allow extension of the tool's functionality.

As opposed to COTS, the source code for the open-source tools is accessible. This fact in itself promotes the extensibility. Furthermore, if the developers of the tool have provided good APIs that hide the underlying complexity of the code, the tool's functionality will be extended with lesser effort.

2. Accessible output:

This item relates to the capability of the tool to export the collected data into an accessible format/standard format.

Data export to an industry standard database is probably most desirable. Databases will allow the handling of a large amount of data without deploying too much effort. There are, however, some issues that may be encountered with this approach. These are:

i. Granularity/break down of the information

There are no standards that define the break down of the information i.e. the format of the tables into which the information is inserted is implementation-dependent. The requests that may be placed to a database are therefore limited to the table formats that are supported by the tool. This directly impacts the level of granularity of the information that will be possible to obtain from a given tool. Let's illustrate the issue with a simple example.

Let's assume that two tools export to a database table all the discovered hosts' OS information. The first tool produces a table entry as follows:

Hosts	OS	Version
192.168.1.2	Windows	2000 sp4
…	…	…

Whereas the second tool produces the following table:

Host	OS	Release	Version
192.168.1.2	Windows	2000	sp4
…	…	…	…

The information given by the two tools is equivalent but the break down differs. The second tool offers a finer granularity of the information. The information regarding the service pack level can be directly retrieved from the second table whereas parsing of the version entry will be required for the first table.

Thus depending on the need, access to the information of interest may require different levels of effort.

ii. Conversion of the raw data.

There may be cases when direct access to the raw data is not possible. Although the tool collects the raw data, the information made accessible in the databases has been converted by the tool into its own format. For example, instead of providing the pair MAC address/switch interface information as one would expect, the Fluke OptiView tool provides the pair host ID/switch interface information, where the host ID is an internal host entry that the tool uses. A relational database needs to be accessed in order to learn the association of the host ID with the actual MAC address.

Similar issues may happen for statistics data. In some cases, only the statistic values that were computed by the tool will be made available in the database tables. The collected raw numbers from which the statistics were computed will not be available. This may prevent to user from performing other calculations of interest. The user is then restricted to the statistics that are supplied by the tool.

iii. Abstraction of information

The tool may perform some abstraction of the collected information i.e. the tool may make accessible only part of its knowledge. For example, The FlukeOptiView tool will discover all the MAC address entries that a switch sees. However, the tool will only list in the database the MAC addresses of the hosts that are currently connected to the switch.

iv. Accessibility of historical data and event data

In general, network discovery tools keep a history of the collected data. As well, tools are often capable of generating events. The information provided by these two items may

be of interest and should be accessible. However, the export of such information into databases is not straightforward. It is currently not clear whether this data can easily be made available to be processed by external tools.

In summary, it should be noted that export support may imply that:

- data may not be accessible in the required format (granularity/break down)
- data may not be presented at the desired level (converted data)
- although known by the tool, some raw information may be hidden (abstraction of data)
- historical data/events data may not be accessible

7.1.10 Dependencies

Dependency was evaluated as follows: the tool requires the installation/presence of a 3rd party software (commercial or open-source) to provide some of its basic functionality.

It should be noted that for open-source tools, the dependency on 3rd party tools represents less of an issue than for the COTS. When an open-source tool is dependant on a 3rd party software, this software will also be an open-source software and freely available.

7.2 Node attributes table

This table lists node information/attributes that can be provided by network discovery tools. The attributes have been divided in two main categories: the SNMP and the non-SNMP category. The categories are explained below, with an explanation of node attributes for each.

7.2.1 The SNMP category

This category includes all of the node attributes and information that can be extracted using the SNMP protocol. This implies that nodes are SNMP-enabled i.e. they host a SNMP agent which accesses the internal MIB values and provides replies to SNMP requests.

A MIB, or management information base, is a hierarchy of information used to define managed objects in a network device. The MIB structure is based on a tree structure, which defines a grouping of objects into related sets. Each object in the MIB is associated with an object identifier (OID), which names the object. The "leaf" in the tree structure is the actual managed object instance, which represents a resource, event, or activity that occurs in the network device. The three main types of MIBs have been presented in Section 4.

The information that can be extracted on an SNMP-enable host depends on the MIBs implemented by that host. Every network device that claims to be SNMP compliant will,

at minimum, implement the MIB-II basic group values that apply to the device. The complete list of all the basic group values supported by MIB-II can be found in [9]. A device may also implement other standard MIBs. A complete list of existing standard MIBs can be found on the IETF site.

Since the Internet Standard MIBs do not cover the entire range of statistical, state, configuration and control information that may be available in a network element, vendors of network devices generally have developed MIB extensions that cover this information. The SNMP-enable tools will be able to extract the values provided by these vendor-specific MIBs as long as the tool implements a MIB compiler. The compiler will allow the tool to learn the proprietary MIB format. The support of an integrate MIB compiler is a good asset for a network discovery tool. The tool's capability can then be extended to retrieve more values and by doing so, increase its device/network knowledge.

Many MIB values are expressed using a string syntax. It may be interesting to note that there is no standard to specify the actual string value format. Because of this lack of standardization, data correlation that would be performed using string values may require a greater level of effort.

7.2.2 The non-SNMP category

This category includes node attributes that are extracted using methods other than SNMP. The list provided in the table is not exhaustive but rather shows the node attributes that are of particular interest and more desirable to obtain[5]. Each tool was evaluated on its ability to provide this information (without using SNMP to obtain it). Each item is explained below.

- Device status: this indicates if the device is up or down.
- CPU: this provides information on the device's CPU such as processor type, speed, load,...
- Memory: this provides information on the device's memory such as memory capacity, amount of free memory, ...
- Disk: this provides information on the device's hard disk such as total disk capacity, free disk space,...
- MAC address: this provides the device's physical hardware (MAC) address(es)
- IP address: this provides the device's network IP address(es)
- OS version: this provides the type of Operating System and version running on the device

[5] Based on a list that was provided by DRDC NIO Section

- OS patch level: this provides information on the various OS patches that may have been installed on the device.

- Port detection: this provides a list of the TCP and UDP port numbers that are opened on the device

- Application status: this indicates the specific network applications that are running on the device. Examples of applications are: web server, mail server, DNS, etc… The identification of network applications is usually derived from the port detection information.

- Device type: this provides a functional description of the devices. Examples are: router, PC, switch, ...

- Device location: this provides information on where the device is physically located.

- Interface status: provides a list of the device's physical interfaces and their associated status (up or down)

- Interface speed: this provides the nominal speed information for each of the device's interface.

8 Tools study

As shown in the tool selection table of Section 6, seventeen tools were selected for further investigation. This section presents the results of the evaluation. The section is divided in three parts. First, the characteristic tables are presented. Second, for each tool category, a number of general observations are made. Finally, for each of the evaluated tool, a brief highlight is given.

8.1 Characteristic tables

Each tool was evaluated against the criteria listed in the network table and the node attributes table. The results of the evaluation are summarized in the following tables.

Table 2: Network Characteristic Table, Active SNMP-Based Category

	Product Name	Visual map		Auto-discovery of Devices		Scope of auto-discovery		Technique of auto-discovery		Mode of operation	
		Layer 2	Layer 3	Layer 2	Layer 3	LAN	WAN	passive	active	Agent-less	Agent-dependent
COTS	HPOV NNM	yes	yes	yes	yes	yes	yes	no	yes	no	Yes SNMP agents
	IBM Tivoli NetView	no	yes	no	yes	yes	yes	no	yes	no	Yes SNMP agents
	Castle Rock SNMPc	no	yes	no	yes	yes	yes	no	yes	no	Yes SNMP agents
	BMC Patrol Visualis	yes	yes	yes	yes	yes	yes	no	yes	no	Yes SNMP agents
	Micromuse NetCool Precision	yes	yes	yes	yes	yes	yes	Arp broadcast only	yes	no	Yes SNMP agents
Open Source	OpenNMS	no	no	no	yes	yes	yes	no	yes	no	Yes SNMP agents (not for service monitoring)
	Nomad	no	yes	no	yes	yes	yes	no	yes	no	Yes SNMP agents

Table 2: Network Characteristics Table (cont'd), Active SNMP-Based Category

	Product name	Supported architectures				Protocols used for active discovery							Performance monitoring	extensibility		dependency on 3rd party tools
		central	Distributed			SNMP			icmp	dns	arp	others		APIs	Accessible output	
			Peer-to-peer	Hierar-chical	Client/ server	V1	V2	V3								
COTS	HPOV nnm	yes	yes	yes	yes	y	y	Via 3rd party	y	y	Via SNMP		no	Yes see AnnexC	Relational DB	no
	IBM Tivoli Netview	yes	no	yes	yes	y	y	n	y	y	Via SNMP		no	Yes See AnnexC	Export to SQL	no
	Castle Rock SNMPc	yes	yes	yes	yes	y	y	y	y	y	yes		Limited See AnnexC	C/C++ WinSNMP DDE	ODBC Text file	no
	BMC Patrol Visualis	yes	no	no	no	y	y	y	y	n	n		Strong data flow & net performance monitoring	??	Oracle8i CIM	no
	Micromuse NetCool Precision	yes	yes	yes	no	y	y	y	y	y	yes		no	Java Perl	Oracle8I Oracle, Sybase, Remedy See AnnexC	no
Open Source	OpenNMS	yes	no	no	yes	y	y	n	Yes see Annex C				Services status, SNMP performance data	no	Very limited XML,CSV file	no
	Nomad	yes	no	no	no	y	y	y	yes	no	no		no	no	Very limited	no

Table 3: Network Characteristics Table, Active Hybrid Category

	Product Name	Visual map		Auto-discovery of Devices		Scope of auto-discovery		Technique of auto-discovery		Mode of operation	
		Layer 2	Layer 3	Layer 2	Layer 3	LAN	WAN	passive	active	Agent-less	Agent-dependent
COTS	Fluke OptiView	yes	yes	yes	yes	yes	yes	yes	yes	yes	No but will use SNMP agents if available
COTS	FoundStone	no	yes	no	yes	yes	yes	yes	yes	yes	no
Open Source	Nmap	no	no	no	yes	yes	yes	no	yes	yes	no
Open Source	Cheops-ng	no	yes	no	yes	yes	yes	no	yes	yes	no
Open Source	Big Sister	no	no	no	no	n/a	n/a	See Annex C		no	Yes Proprietary agents

Table 3: Network Characteristics Table (cont'd), Active Hybrid Category

	Product name	Supported architectures				Protocols used for active discovery							Performance monitoring	extensibility		dependency on 3rd party tools
		central	Distributed			SNMP			icmp	dns	arp	others		APIs	Accessible output	
			Peer-to-peer	Hierar-chical	Client/ server	V1	V2	V3								
COTS	Fluke OptiView	yes	no	yes	no	y	y	n	y	y	yes	yes	yes	no	SQL compliant DB	yes
COTS	FoundStone	yes	no	yes	yes	n	n	n	y	y	no	yes	no	FSL	SQL compliant DB	yes
Open Source	Nmap	yes	no	no	no	n	n	n	y	y	no	tcp/udp	no	no	text	no
Open Source	Cheops-ng	yes	no	no	no	n	n	n	y	y	no	tcp/udp	no	no	Text file	no
Open Source	Big Sister	no	no	no	yes	y	y	n	See Annex C				Status monitoring of various elements	no	CSV file	Require software modules to be installed

Table 4: Network Characteristics Table, Inventory&Audit, Passive only, Research Categories

Product Name			Visual map		Auto-discovery of Devices		Scope of auto-discovery		Technique of auto-discovery		Mode of operation	
			Layer 2	Layer 3	Layer 2	Layer 3	LAN	WAN	passive	active	Agent-less	Agent-dependent
Inventory		iInventory	no	no	no	no	n/a	n/a	No See Annex C		no	Yes proprietary agents
Inventory		LANDesk	no	no	no	yes	yes	yes	no	yes	no	Yes Proprietary agents
Passive	COTS	IPSum Route Dynamics	no	yes	no	Yes routing devices only	yes	yes	yes	no	yes	no
Passive	Open-scr	RNA/Snort	No?	No?	yes	yes	yes	yes	yes	no	yes	no
Research		CRC Network mapping tool	yes	yes	yes	yes	yes	no	yes	yes	Yes but limited SNMP agent dependance	no

Table 4: Network Characteristics Table (cont'd) , Inventory&Audit, Passive only, Research Categories

Product name			Supported architectures				Protocols used for active discovery							Performance monitoring	extensibility		dependency on 3rd party tools
			central	Distributed			SNMP			icmp	dns	arp	others		APIs	Accessible output	
				Peer-to-peer	Hierar-chical	Client/ server	V1	V2	V3								
Inventory		iInventory	yes	no	no	yes	n	n	n	n	n	n	n	no	no	SQL compliant DB	no
Inventory		LANDesk	yes	yes	no	yes	n	n	n	n	n	n	tcp/ip	Yes see Annex C	no	ODBC	no
Passive	COTS	IPSum Route Dynamics	no	no	no	yes	n	n	n	N/A passive monitoring of routing protocols (ospf, bgp)				Appl path tracking, Routing stats	Java SDK	Export to MS Excel, No DB support	no
Passive	Open-scr	RNA/Snort	no	no	no	yes	n	n	n	N/A passive monitoring				no	??	??	no
Research		CRC network mapping tool	yes	no	no	no	y	y	n	See Annex C				no	no	no	Require software modules to be installed

Table 5: Node Attributes Table, Active SNMP-Based Category

Product Name		SNMP	Non-SNMP							
			Dev status	CPU	Mem	Disk	MAC addr	IP addr	OS version	OS patch level
COTS	HPOV NNM	yes	no	no	no	no	no	very limited	no	no
	IBM Tivoli NetView	yes	yes	no	no	no	no	yes	no	no
	Castle Rock SNMPc	yes	yes	no	no	no	no	yes	no	no
	BMC Patrol Visualis	yes	yes	no	no	no	no	yes	no	no
	Micromuse NetCool Precision	yes	yes	no	no	no	yes/ARP	yes	no	no
Open Source	OpenNMS	yes	yes	no	no	no	no	yes	no	no
	Nomad	yes	yes	no	no	no	no	yes	no	no

Table 5: Node Attributes Table (cont'd), Active SNMP-Based Category

	Product Name	Non-SNMP					
		Port detection	Appl. status	Dev type	Dev location	Interface status	Interface speed
COTS	HPOV NNM	no	no	no	no	no	no
	IBM Tivoli NetView	no	no	no	no	no	no
	Castle Rock SNMPc	no	yes limited	no	no	no	no
	BMC Patrol Visualis	no	no	no	no	no	no
	Micromuse NetCool Precision	no	no	no	no	no	no
Open Source	OpenNMS	yes Associated to the service	yes Predefined in config file	no	no	no	no
	Nomad	no	no	no	no	no	no

Table 6: Node Attributes Table, Active Hybrid Category

	Product Name	SNMP	Non-SNMP							
			Dev status	CPU	Mem	Disk	MAC addr	IP addr	OS version	OS patch level
COTS	Fluke OptiView	yes	yes	no	no	no	yes	yes	no	no
COTS	FoundStone	No	yes	no	no	no	no	yes	yes	yes
Open Source	Nmap	no	yes	no	no	no	no	yes	yes	no
Open Source	Cheops-ng	no	yes	no	no	no	no	yes	yes	no
Open Source	Big Sister	yes	yes	yes	yes	yes	yes	yes	no	no

Table 6: Node Attributes Table (cont'd), Active Hybrid Category

	Product Name	Non-SNMP					
		Port detection	Appl. status	Dev type	Dev location	Interface status	Interface speed
COTS	Fluke OptiView	no	yes	yes	no	no	no
	FoundStone	yes	yes	yes	no	no	no
Open Source	Nmap	yes	yes	Yes See Annex C	no	no	no
	Cheops-ng	yes	yes	yes	no	no	no
	Big Sister	no	Yes Pre-defined in config file	no	no	yes	No??

Table 7: Node Attributes Table, Inventory&Audit, Passive only and Research Categories

Product Name			SNMP	Non-SNMP							
				Dev status	CPU	Mem	Disk	MAC addr	IP addr	OS version	OS patch level
Inventory		iInventory	no	yes	yes	yes	yes	yes	yes	yes	no
		LANDesk	no	yes	yes	yes	yes	yes	yes	yes	yes/current built
Passive	COTS	IPSum Route Dynamics	no	no	no	no	no	no	Yes Routing devices	no	no
	Open Source	RNA/Snort	no	Yes Passive only	no	no	no	yes	yes	Yes OS fingerprinting	no
Research		CRC Network mapping tool	no	yes	no	no	no	yes	yes	Yes OS fingerprinting	no

Table 7: Node Attributes Table (cont'd), Inventory&Audit, Passive only and Research Categories

Product Name			Non-SNMP					
			Port detection	Appl. status	Dev type	Dev location	Interface status	Interface speed
Inventory		iInventory	no	no	no	no	no	no
		LANDesk	no	no	no	no	no	no
Passive	COTS	IPSum Route Dynamics	no	Yes Appl flow tracking	no	no	no	no
	Open	RNA/Snort	Yes Passive only	Yes Limited passive	no	no	no	no
Research		CRC Network Mapping tool	yes	no	Yes limited	no	no	no

8.2 General Observations

A summary of each tool's capabilities was presented in the previous section. In this section, general observations (per tool category) are given.

8.2.1 The Active SNMP-based category

The SNMP-based category constitutes by far the largest category. This clearly shows that the use of the SNMP protocol to perform network discovery is dominant.

The SNMP-based tools generally offer layer 2 and 3 discovery. Similarly, they also offer a visual map display of the network topology. It is interesting to note that in all cases of level 2 discovery, the associated level 2 mapping is supported. This is not necessarily the case for the level 3 discovery.

Since the discovery technique used by these tools is purely active, it should be expected that significant traffic be produced at times[6]. Fortunately, most of them support at least one type of distributed architecture. Depending on the structure of the network, this type of architecture may be desirable to deploy (ref. Section 7) and therefore, definitely represents an interesting asset for this category of tools.

Many of the tools in this category offer SNMPv3 support (this is generally true of every commercial tool that has a significant share of the market). SNMPv3 is definitely a feature of importance if security is a concern. The drawback is that SNMPv3 is not yet widely supported in the various vendor network devices and is definitely not supported in older devices. It is also more complex to configure.

Most commercial products in this category support data export to at least one type of standard database (ref. Section 7 for a discussion on some potential issues with this approach). This does not appear to be the case for the open-source tools. Most SNMP-based open-source tools seem to offer only very limited data export capability.

Finally, because of the use of SNMP, the tools in this category will, in general, be able to provide more information then the tools of the "active hybrid" and "passive only" categories. For example, the SNMP-based tools may be able to provide information which otherwise would be very difficult or even impossible to obtain. This information includes namely: level 2 interconnectivity, VLANs discovery, interfaces information (speed, type), accurate device type and device location information. This latter capability is, however, dependent on SNMP agents running on the network devices.

[6] This problem generally occurs during the initial discovery phase i.e. when the auto-discovery process is initiated for the first time.

8.2.2 The Active Hybrid category

There exist few tools that do not use SNMP as their main active network discovery protocol. Only a few tools could be identified in the commercial sector. More tools could be identified in the open-source community. But as already mentioned in Section 6, most of them only address a very narrow aspect of network discovery. For this reason, many of them were not further investigated in this report.

Tools in this category that also implement SNMP (in addition of the other discovery techniques) provide capabilities that are very similar to the ones described in the previous category. An advantage that these tool offer over the others is that even if SNMP is not present on the network devices, they are still able to provide enough information of interest. Of course, the information they can discover will be more limited. The Fluke OptiView tool for example, will in such case, typically provide the following information:

Host IP address
Host MAC address
DNS name
NetBios name
Manufacturer of Ethernet (via the MAC address info)
Device type such as server, router, switch (via monitoring of protocols)
Classification into IP network domains (IP, IPX, NetBios)
Detection of possible mis-configured devices (based on monitored traffic)

For these tools, SNMP is basically an additional asset that allows more complete discovery if enabled.

The tools in this category that do not support SNMP offer more limited capabilities. Most do not support a visual map display. If they do, it is only at the layer 3. They are more limited in the information they are able to automatically discover. They will typically be able to automatically discover: the IP address, the OS version, the open ports (active scanning) and possibly some application status. Their auto-discovery capability at level 2 is practically non-existent or very limited. Typically, the only level 2 information that will be discovered is the MAC address.

The open-source tools of this category also offer only very limited data export support (mainly text-base export of some data).

8.2.3 Passive only category

Tools that conduct network auto-discovery by strictly using passive techniques are not widely available. Only one commercial tool having enough capabilities could be identified. The tools in this category generally operate at the level 3 only. The information that they will generally be able to discover is: the IP address, the OS version, limited port detection and limited application status, status device (when the device

generates traffic on the network). When a tool is able to discover information at level 2, it will be very limited (typically MAC address only).

The inherent characteristics of passive monitoring will naturally be found in these tools (ref. auto-discovery techniques of Section 4). The tools in this category will discover the information over a longer period of time. The discovered information may not always be accurate or complete (in OS detection for example). On the other hand, they are less intrusive and provide of a near real-time view of the network (constant monitoring as opposed to a snapshot view).

These tools all support a distributed architecture and usually make use of proprietary agents deployed in the network.

8.2.4 The Inventory & Audit category

The tools in this category differ from the others in the way they operate. In fact, they offer no auto-discovery of device support and no map display support. They usually do not implement any active nor passive technique. They actually do not provide any network information. Their capability resides in host configuration discovery. The strength and interest of these tools currently lies in the fact that they have the ability to provide host information that may not be obtainable otherwise. The host configuration information they can provide is currently more detailed than the information that can be provided by the SNMP-based tools. SNMP was developed to manage the network environment. This is well reflected in the type of information that SNMP provides which is mainly network-related information. The standard MIBs do not cover very much on end system configuration information. MIB-II basic set for example, includes most if its end system configuration data in the "system" group. The system group is meant at providing general information about the managed system and offers very limited information [9].
The Inventory & Audit tools will generally provide most of the host configuration information carried in standard MIBs and more. They will provide information such as: detailed hardware inventory, precise Operating System data, list of software applications and attributes, networking configuration data and network services, etc… A sample list of the host information that can be provided by such tools is given as an example in Annex C.

These tools all make use of proprietary agents. These agents need to be installed on each monitored system. The deployment of such tools implies a higher level of configuration and management effort.

There exist open-source equivalent software but for this study, only commercial tools were studied. Open-source tools most likely offer similar capabilities, possibly in exception of export support. Most commercial tools provide data export support to standard databases. This is not the case for many open-source tools.

8.3 Tool highlights

For each of the evaluated tools, a brief highlight is given. The actual detailed description of each tool's capabilities, architecture, discovery techniques, extensibility, unique features as well as strengths and limitations is presented in Annex C.

8.3.1 HPOV Network Node Manager

Network Node Manager (NNM) is part of the well-known HP OpenView suite of network management tools. NNM provides automatic network discovery at layer 2 and layer 3 and presents a visual map of the discovered network. The discovery process is highly SNMP-dependent and offers support for SNMPv1/v2. Support for SNMPv3 is currently being offered via a 3rd party plug-in module. NNMs architecture scales very well to large and complex networks and is one of the tools strengths. NNM exports its data either into a build-in relational database or into an external data base such as Oracle and SQL Server. NNM integrates well with other OpenView suite products for additional functionality and features.

8.3.2 IBM Tivoli NetView

NetView is part of IBM Tivoli suite of network management tools. NetView offers automatic network auto-discovery at layer 3 and presents a visual map of the discovered network. The discovery process is highly SNMP-based and currently supports SNMPv1/v2. NetView supports central and distributed modes of operation that allow it to scale for a variety of network configurations and provides a web-based interface. NetView provides capability to export its network-topology data into SQL database.

8.3.3 BMC Patrol Visualis

BMC Patrol Visualis heavily relies on SNMP to perform its auto-discovery. It supports SNMPv3. Visual display is one of the tool's strengths. It can perform 3D graphics display of the network topology at layer 2 and layer 3. It claims to have a very efficient and accurate topology resolution engine. It is capable of discovering VLANs. It performs real-time application and network performance monitoring. It supports export to Oracle 8i relational database as well as exploits the Common Information Model (CIM) standard [10] to represent objects.

8.3.4 Micromuse NetCool Precision for IP

NetCool Precision heavily relies on SNMP to perform its auto-discovery. It supports SNMPv3. It can automatically discover layer 2 and layer 3 network devices as well as produce the associated map. It is capable of discovering VLANs. It has multiple built-in MIBs (pre-compiled) and therefore can extract information on a variety of protocols (via

MIB support in devices). It can be deployed in a number of architectures, namely distributed hierarchical and peer-to-peer. It offers a variety of APIs as well as what seems to be a very good proprietary solution for data export.

8.3.5 Castle Rock SNMPc

SNMPc heavily relies on SNMP to perform its auto-discovery. It supports SNMPv3. It discovers and maps the network at level 3 only. It offers a good flexibility for deployments (it can be configured for central architecture or 3 types of distributed architectures). It allows export of some of its data to ODBC databases and offers a number of APIs for tool customization.

8.3.6 OpenNMS

OpenNMS is an open-source management software that provides an alternative to expensive COTS and complex solutions. Although it mainly relies on SNMP to perform discovery, service discovery is done via active port polling based on a pre-defined list of services. It discovers devices at level 3 only. It does not provide a map display. It was designed to be highly configurable. It makes a strong usage of XML for its configuration files. It has limited MIB import capabilities (no MIB compiler) and has a very limited export capability.

8.3.7 Nomad

Nomad is an open-source product that is available with its source-code from the Internet. Nomad aims to provide SNMP-based network discovery and network mapping services. The discovery process relies heavily on SNMP and is based on another open-source software called Net-SNMP. Currently the program discovers layer 3 devices and claims to support SNMPv1, SNMPv2, and SNMPv3. The tool's strength lies in its ability to create and present a visual network map from information gathered via SNMP.

8.3.8 Fluke OptiView

The OptiView Console provides the ability to monitor network performance and to generate a variety of reports. As part of it's functionality it offers automatic network discovery of layer 2 and layer 3 devices and presents a visual map of the discovered network. Discovery scope for a single OptiView Console is typically limited to a LAN however OptiView agents can be distributed and integrated with other Fluke Networks hardware and software agents to extend discovery scope beyond the LAN, number of distributed agents is limited by the license purchased. OptiView requires a separate installation of Microsoft Visio 2000, support for which has been recently discontinued by Microsoft, for the actual map drawing. OptiView's discovery process uses a combination of SNMP, active non-SNMP, and passive techniques to achieve its discovery. OptiView

supports central and distributed architectures and as such has the ability to scale for larger networks. OptiView supports SQL server database for data storage.

8.3.9 FoundStone

FoundStone Enterprise was designed to assess and address security vulnerabilities within a network. The product uses the company's own proprietary scripting language for vulnerability assessment and claims to have the lowest percentage of "false positives" in the industry. As part of it's functionality it provides automatic network discovery of layer 3 devices and offers a visual map of the discovered network. The discovery process uses a combination of active non-SNMP techniques including TCP scanning and traceroutes. The product supports central and distributed architectures and scales well for larger networks. In the distributed mode of operation, FoundStone uses SOAP/XML for communication between its components. FoundStone uses an SQL-compliant database for its data storage.

8.3.10 Nmap

Nmap is open-source software that can determine existence, services and OS types. The discovery process uses a combination of active non-SNMP techniques such as: ICMP-ping scanning (to determine which hosts on a network are up), several TCP and UDP port scanning techniques, and operating system identification via TCP/IP fingerprinting. Nmap does not provide visual network mapping capability. Nmap supports Unix, Windows, and several handheld-based platforms in both GUI and command-line modes. Nmap supports central architecture.

8.3.11 Cheops-Ng

Cheops-ng is open-source software that offers layer 3 network auto-discovery and network mapping functionality. The discovery process uses a combination of active non-SNMP techniques. From the Cheops-ng source code, dependence on Nmap source code can be observed. Cheops-ng supports central architecture. The information that could be obtained for this tool was very limited.

8.3.12 Big Sister

Big Sister is an open-source network monitoring tool. It does not automatically discover devices on a network. Instead, it automatically gathers information and constantly monitors pre-specified hosts. Its mode of operation closely resembles the one of an inventory software tool. Every monitored system is monitored by a Big Sister agent. Agents operate based on information read from configuration files. The configuration files specify the checks to perform and the hosts on which to perform the checks. The tool offers the advantage of being able to retrieve information which may not be

accessible via standard methods (such as SNMP). On the other hand, it requires significant manual configuration prior to its operation.

8.3.13 CRC Network Mapping Tool

The Communications Research Centre has developed a network mapping tool prototype. The tool automatically discovers and maps devices at layer 2 and 3 in a LAN network. The tool combines active and passive techniques to achieve discovery. It implements unique features such as: the ability to create a level 2 map of a remote LAN, the ability to discover virtual machines (VMWare), the ability to provide accurate and complete results based on its correlation technique. Work is in progress to migrate the tool towards a distributed passive only system. Correlation algorithms are being investigated to combine network information with security events.

8.3.14 IPSumnetworks Route Dynamics

Route Dynamics' uniqueness resides in its ability to provide constant monitoring of IP routing and IP application flow paths. It is implemented in the form of appliances (IP Listeners). The appliances are distributed throughout the network. The monitoring and auto-discovery is performed entirely passively. The tool operates by passively monitoring Internet routing protocols. OSPF is the only protocol currently supported. BGP is being developed. The map includes only the devices that participate in the routing as well as their logical relationship to subnets where end-hosts reside. Its export capability is currently very limited.

8.3.15 SourceFire Real-Time Network Awareness (RNA)

RNA is passive sensing technology and analysis tool that discovers and monitors network assets. RNA is built on top of Snort. RNA's uniqueness lies in its ability to provide an integrated security monitoring system, i.e. it attempts to integrate the network information with security event information, conducts behavioral profiling and vulnerability analysis. RNA sensors are distributed in the network. The sensors extract as much information as possible. Their discovery capabilities are, of course, limited by the inherent capabilities of passive monitoring. On the other hand, the tool benefits from all the advantages of this approach. At the time of writing, RNA has not yet been released.

8.3.16 LANDesk Management Suite

LANDesk Management Suite was designed to offer software and hardware inventory auditing, software and OS distribution and migration, and remote configuration/control services. To achieve it's functionality it requires the use of LANDesk agent software on all hosts to be managed. LANDesk offers limited network auto-discovery, i.e. only nodes hosting agents are discovered, however it does not offer a visual network map.

LANDesk supports Windows, Macintosh, NetWare, Linux, Unix and handheld operating systems. For data storage, LANDesk supports MS SQL Server and Oracle databases.

8.3.17 LanAuditor iInventory

iInventory software was designed to keep track of hardware and software resources on Windows, Mac, and Linux computers. It provides a comprehensive list of a node's hardware and software attributes. However, iInventory does not offer automatic network discovery or a visual network map. To audit a particular host, iInventory is used to generate an agent executable file for the particular host platform. The agent file is then executed on the host with administrative privileges. Agents produce an audit output file that is imported into the iInventory database for analysis and generation of reports. iInventory supports MS Access 2000 and MS SQL Server databases.

9 Other elements for consideration

Variation in network size, network complexity, and network configuration will have an impact on performance and effectiveness of network-discovery techniques. Presence of such network elements as firewalls, Network Address Translation (NAT) devices, low-bandwidth links, and mobile nodes should be considered when selecting an auto-discovery technique. This section highlights effects on the network-discovery process when the above-mentioned elements are present in the network.

9.1 Low-Bandwidth Links

Networks that span multiple geographical locations often rely on low-bandwidth wide area network (WAN) links such as leased ISDN lines, dial-up connections, or wireless links for interconnections between remote sites. An example scenario with two remote sites, Head Office and a Satellite Office, is shown in Figure 5 below. For such networks, the choice of active auto-discovery technique deployed in a central architecture running from the Head Office location may introduce enough data-traffic to interfere with other application data flows across the WAN link. The overall network performance will be affected. To auto-discover such networks with the least impact on the WAN link, a distributed technique with a "discovery module" per remote site should be used. The inter-communication protocol and information exchanged between the sites should be optimized to avoid congesting the low bandwidth links.

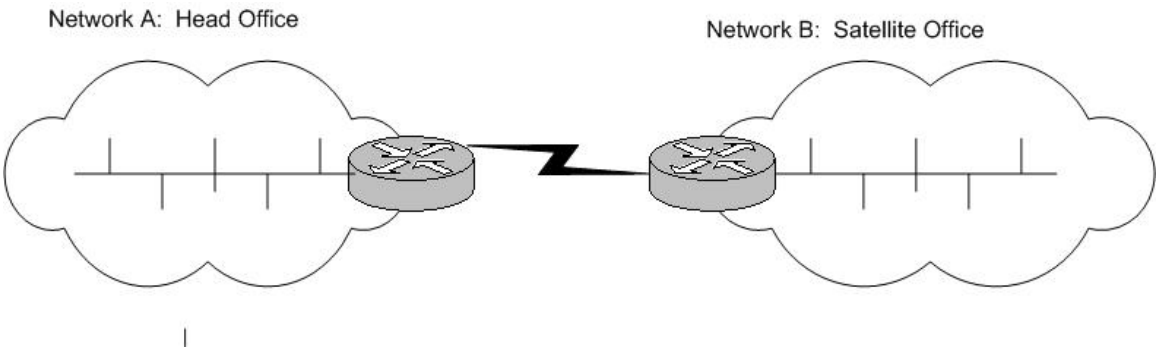

Figure 5: Low-bandwidth wireless WAN link.

9.2 Node Mobility and DHCP

When a network supports mobile users that connect to the network from different places such network will often provide services of the dynamic host configuration protocol (DHCP). DHCP was designed to simplify TCP/IP client configuration and manages a set of IP addresses that are assigned to nodes on a temporary basis, after the node disconnects from the network the IP address is returned back the "pool" of available addresses and may later be assigned to a different node. Therefore with DHCP, a given IP address may have several MAC addresses associated with it over a period of time. Such scenario may present challenges to an auto-discovery tool as it may interpret such situation as misconfiguration and generate unnecessary events. For instance, HP OpenView Network Node Manager (NNM) requires that the address range used by the DHCP be explicitly specified. Otherwise, NNM will generate unnecessary alarms.

9.3 NAT devices

Network Address Translation (NAT) was designed to address the shortage of globally unique IP addresses by providing mapping between "private" non-unique IP addresses to "public" unique IP addresses (defined in RFC 1631) and vise versa. In addition to addressing shortage of IP addresses, other benefits of deploying NAT include security and administration.

Many nodes on the private side (or behind) of the NAT router or a gateway can be represented by a single IP address to the network outside of the NAT as shown in Figure 6 .

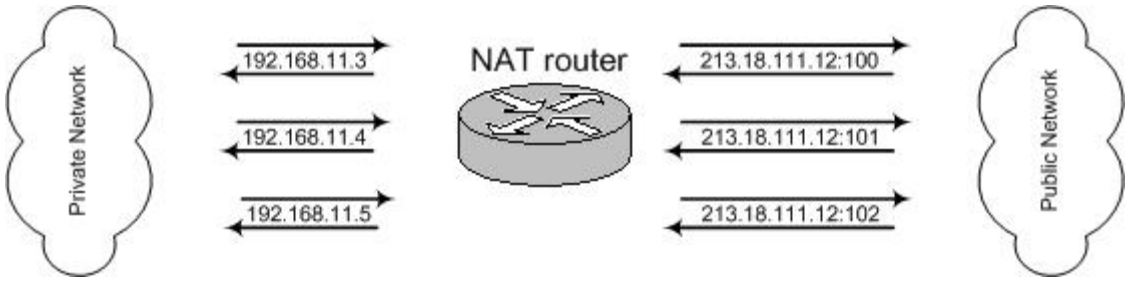

Figure 6: Private networks with NAT

Networks with private network segments behind NAT devices pose challenges for discovery techniques since the private network nodes to the outside are seen as a single IP host. To effectively discover nodes behind NATs distributed discovery technique can be utilized by placing a discovery module on the private side of the NAT. Methods have been investigated for discovery of networks behind NAT [16].

9.4 Firewalls

Firewalls were designed to provide intelligent control of data flows between two or more networks for security purposes. Firewalls provide the ability to block unwanted data traffic from entering or leaving the network. Firewalls can also be installed on end-system devices to filter traffic. For example, a firewall can be configured on a PC to block all ICMP ping packets, in which case, discovery using ICMP ping will not succeed. If the scope of discovery includes networks that are behind a firewall, specific knowledge of the protocols used by the discovery techniques becomes important. Clearly, if the firewall is configured to block all protocol data packets that the discovery techniques relies-on to perform the discovery, the network nodes behind such firewall will not be discovered. In the case of distributed architecture, the firewall may need to be configured to allow the distributed discovery module to communicate with its central entity, i.e. specific TCP ports will have to be left "open" on the firewall.

9.5 Third party networks

If the discovery process transits over a 3rd party network (e.g. a service provider network) no physical access to the 3rd party network is usually allowed. The transit network appears to the user as a "link". The characteristics of the link (e.g. bandwidth and availability) will generally be negotiated between the user and the provider and be captured in the service level contract agreement. For this reason, there is no need for the user to know about the exact topology of the transit network.
The common use of firewalls will, in such cases, limit the discovery performed by active techniques to finding just the 3rd party edge nodes.

10 Trends in network discovery technology

Like much of computer-related technologies, the area of network auto-discovery is on a rapid path of evolvement. While reviewing auto-discovery techniques used in COTS and open-source tools a number of trends were observed. The following sections present trends related to network auto-discovery.

10.1 Evolvement of SNMPv3

The SNMPv3 specifications were approved by the Internet Engineering Steering Group (IESG) as full Internet Standard in March 2002. Today a number of vendors and research departments have implemented SNMPv3. Although current deployment of SNMPv3 remains limited, the list of vendors given in [1] that support SNMPv3 clearly indicates growth in SNMPv3 deployment since March 2002.

SNMPv3 greatly improved security of communicating management information by adding authentication and encryption services to SNMPv2. The SNMPv3 specification documents define the use of HMAC-MD5-96 and HMAC-SHA-96 as the authentication protocols and the use of 56-bit CBC-DES as the encryption protocol. However, the specification also defines extensibility techniques for both authentication and encryption so that new authentication and encryption protocols can be used with SNMPv3.

Because the 56-bit DES algorithm is theoretically breakable within reasonable amount of time using current computing technology, there is interest in adding new security protocols to SNMPv3. Currently, efforts are being made within the IETF community to add two encryption algorithms to SNMPv3: 128-bid Advanced Encryption Standard (AES) and 168-bit key triple-DES. Several network equipment vendors have implemented these extended security options in their SNMPv3 implementations. Vendors with triple-DES implementations include: SNMP Research, Juniper, and Marconi. Vendors with AES implementations include: SNMP Research, Juniper, MGSoft, and NetSNMP.

At this point in time it is expected that SNMPv3 will continue to grow in popularity, especially with traditional SNMP-based network management systems. However, inherent limitations of SNMP framework will continue to fuel research and development of competing technologies, such as web-based management approaches seen with XML.

10.2 Evolvement of MIBs

Since the specification of MIB-II allows for definition of vendor-specific MIB structures (proprietary MIBs) and MIB-extensions, new MIBs are continually being introduced at the IETF. As new technologies are developed and launched, the standardization process for defining MIB structures for those technologies is expected to follow. For instance currently, there are five MIBs that are being standardized by the IETF (ADSL MIB, AToM MIB, Bridge MIB, Entity MIB, and Ethernet Interfaces and Hub MIB). A good

representation of current MIB-II extensions can be found in [11] . An example of MIBs supported by Cisco equipment can be found in [12].
At this time it is reasonable to expect that MIBs will continue to evolve.

10.3 Use of XML

While performing this survey, a growing interest has been noticed for using Extensible Markup Language (XML) technologies for communicating network management information between nodes and for node configuration purposes. XML was developed by an XML working group formed under the World Wide Web Consortium (W3C) in 1996.

XML is a subset of the Standard Generalized Markup Language (SGML) [ISO 8879]. XML describes data objects called XML documents and the behavior of computer programs that process them [13]. XML is widely accepted and there are many tools available that potentially support the implementation of management functions. Since management data can be represented as XML documents and protocols such as TCP and HTTP can be used to transport XML data, XML can be used to facilitate integration of network and service management applications.

Example of XML technology usage was observed in the FoundStone Enterprise tool and OpenNMS tool. These tools use XML for configuration purposes as well as to communication among their components. They claim it provides an ideal way to integrate with custom applications or development of new services that use Web services model.

At this time it is reasonable to expect that XML-based solutions will continue to gain support among vendors and its deployment will increase. Current support among vendors for XML is very broad and is evident from the membership at W3C. Among the members of W3C are: Adobe Systems Inc., Alcatel, AT&T, Canon Inc., Cisco Systems, IBM Corporation, and many others.

10.4 Network Discovery and Intrusion Detection Systems

Intrusion Detection Systems (IDSs) utilize passive monitoring techniques to detect unauthorized network activity. Upon detection of an intrusion, IDSs generate alarms and notify network administrators of such events. An inherent drawback of many IDSs available today, which limits their efficiency, is the amount of false positives and negatives they generate.

In many instances this inherent issue of IDS can be tackled by providing to the IDS sensors the knowledge of network topology and configuration that they are protecting. A company called SourceFire is actively pursuing this approach. Their Real-Time Network Awareness (RNA) product uses passive network discovery techniques to supply IDS sensors with network topology information. RNA is due for release in November 2003.

Similar approach is also used in a product called NeVO (Network Vulnerability Observer) by Tenable.

Given the unique strengths of passive discovery techniques and the passive nature of IDSs, the two technologies naturally integrate well together. Therefore, it is reasonable to expect to see more IDS-based network and service management and configuration systems in the future.

10.5 Service mapping

The existing auto-discovery techniques focus at finding the physical and logical network connectivity (i.e. node interconnections at layer 2 and 3 of the OSI model). However, on the networks there also exists a logical connectivity at higher layers, which could also be discovered and mapped. For instance, the concept of overlay network is currently being investigated in the research community. "Overlay Networks" is an application-level routing and packet forwarding service that gives end-hosts and applications the ability to take advantage of network paths that traditional Internet routing cannot make use of, thereby improving their reliability and performance [14] [15]. If Overlay Networks technology gains in popularity, discovering and mapping of such networks will then provide valuable information to network administrators.

It is therefore reasonable to expect that future network auto-discovery and mapping techniques will evolve to include such service discovery and mapping in addition to the current actual physical and logical interconnectivity information.

11 CONCLUSION

A study on network auto-discovery tools was performed. A number of tools from the commercial sector (COTS), the open-source community and the research/academic community were selected and their capabilities further evaluated. One thing that is apparent when looking at the outcome of the study is that although some tools present good capabilities, they all have their strengths and weaknesses. Within the scope of interest, the "one tool does it all" solution does not exist. As it was seen in the report, the tools can be characterized by the network discovery technique they implement. Each technique comes with its strengths and limitations. It is reasonable to expect that for the TDP system, the solution is likely to consist of an integrated suite of tools where functionality of each tool will be combined to achieve the desired capability.

Since the tool evaluation was mainly based on documentation review, this study provides a preliminary assessment of current network discovery tools and their capabilities. However, as the requirements for the system are further defined, it is believed that broader information and a deeper knowledge could be gained by performing a subsequent study that would involve deployment of a specific set of tools in an actual test network environment.

12 REFERENCES

[1] SNMPv3 specifications: http://www.ibr.cs.tu-bs.de/projects/snmpv3/

[2] Stallings W. "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2", Third Edition. Addison-Wesley, 1999.

[3] Arkin, Ofir. Xprobe http://www.sys-security.com/html/projects/X.html

[4] Arkin, Ofir. X-Remote ICMP Based Fingerprint Techniques. August 2001. http://www.sys-security.com//archive/papers/X_v1.0.pdf

[5] J. Nazario, Passive System fingerprinting using Network Client Applications, November 27, 2000, available at http://www.crimelabs.net/docs/passive.pdf

[6] SANS's Intrusion Detection FAQ: How can passive techniques be used to audit and discover network vulnerability? available at http://www.sans.org/resources/idfaq/passive_vuln.php

[7] Lance Spitzner, Know Your Enemy:Passive Fingerprinting, available at http://project.honeynet.org/papers/finger/

[8] Treurniet J., "Passive Information Gathering for Network Security", DRDC Technical Memorandum (in preparation).

[9] IETF RFC 1213. "Management Information Base for Network Management of TCP/IP-based internets:MIB-II".

[10] CIM, http://www.dmtf.org/standards/standard_cim.php

[11] SNMP MIB-II and Extensions http://www.ansdell.demon.co.uk/networks/mib2.html

[12] Cisco MIB support ftp://ftp.cisco.com/pub/mibs/supportlists/

[13] World Wide Web Consortium at http://www.w3.org/

[14] The Case for Resilient Overlay Networks. David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Proc. HotOS VIII, Schloss Elmau, Germany, May 2001.

[15] Savage, S., Collins, A., Hoofman, E., Snell J., and Anderson T. "The End-toEnd Effects of Internet Path Selection". Proc of ACM SIGCOMM 1999. pp 298-299

[16] Steven M. Bellovin, A Technique for Counting NATted Hosts, available at http://www.research.att.com/~smb/papers/fnat.pdf

Acronyms

AES	Advanced Encryption Standard
API	Application Programmable Interface
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation One
BGP	Border Gateway Protocol
CIM	Common Information Model
DES	Data Encryption Standard
DNS	Domain Name Server
DRDC	Defence R&D Canada
EGP	External Gateway Protocol
FTP	File Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IESG	Internet Engineering steering Group
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
HTTP	Hyper Text Transport Protocol
JNMDS	Joint Network Management and Defence System
LAN	Local Area Network
MAC	Media Access Control
MD5	Message Digest
MIB	Management Information Base
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NetBIOS	Network Basic Input Output System
NMS	Network Management System
OSI	Open System Interconnect
OSPF	Open Shortest Path First
PDU	Protocol Data Units
RFC	Request For Comments
RIP	Routing Information Protocol
RPC	Remote Procedure Call
SHA	Secure hash Algorithm
SMI	Structure of Management Information
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network
WC3	World Wide Web Consortium
XML	Extended Markup Language

Annex A: Initial Tools Survey Tables

Table 8: COTS tools survey

	Tool Name	Company	Product Link
1	ActiveXperts Network Monitor	ActiveXperts Software	http://www.activxperts.com/
2	Amerigo	Tavve Software Company	http://www.tavve.com/amerigo.html
3	Argent Software Guardian	Argent Software	http://www.argent.com
4	CommView	TamoSoft	http://www.tamos.com/
5	E-Health	Concord Communications	http://www.concord.com
6	Eye of the Storm	Entuity	http://www.entuity.com
7	FoundStone	FoundStome	http://www.foundstone.com
8	Heroix eQ	Heroix	http://www.heroix.com
9	HP OpenView (NNM)	HP	http://www.openview.hp.com/
10	LAN MapShot	Fluke	http://www.flukenetworks.com/
11	LANsurveyor	Neon Software	http://www.neon.com/LSwin.html
12	LANWatch	Sandstorm Enterprises	http://sandstorm.net/products/lanwatch/
13	Link Analyst	Network Instruments	http://www.networkinstruments.co.uk/
14	LoriotPro	LUTEUS	http://www.loriotpro.com/
15	Microsoft Operations Manager 2000	Microsoft	http://www.microsoft.com/
16	MonitorIT	Breakout Software	http://www.breakoutsoft.com/
17	NetCool Precision	Micromuse	http://www.micromuse.com
18	NetCrunch	AdRem Software	http://www.adremsoft.com/netcrunch/
19	NetIntercept	Sandstorm Enterprises	http://sandstorm.net/products/netintercept/
20	NetSight Atlas	NetScout	http://www.netscout.com

Annex A (cont'd)

Table 8: COTS tools survey (cont'd)

	Tool Name	Company	Product Link
21	NetSight Atlas	Enterasys Networks	http://www.enterasys.com/
22	NetStatus	OTO Software	http://www.otosoftware.com/
23	NetVigil	Fidelia Technology	http://www.fidelia.com/products/
24	Network Discovery	Peregrine Systems	http://www.peregrine.com/
25	Network Inspector	Fluke	http://www.flukenetworks.com/
26	NetworksAOK	Interloci Network Management	http://www.interloci.com/
27	NeVO	Tenable security	http://www.tenablesecurity.com/nevo.html
28	NPS	NPS	http://www.npservices.com/
29	Open NerveCenter	Open Service	http://www.open.com/
30	OpManager	AdventNet	http://www.adventnet.com/products/opmanager/
31	Packeteer	Packeteer	http://www.packeteer.com/
32	PATROL Visualis	BMC Software	http://www.bmc.com/
33	Route Dynamics	Ipsum Networks	http://ipsumnetworks.com
34	SNMPc 6.0	Castle Rock Computing	http://www.castlerock.com/products/SNMPc/
35	Solstice Domain Manager	Sun Microsystems	http://store.sun.com/
36	Spectrum	Aprisma	http://www.aprisma.com
37	SysOrb	Evalesco Systems	http://www.evalesco.com/
38	Tivoli NetView for z/OS	IBM	http://www-3.ibm.com/software/tivoli/
39	Ultra Network Sniffer	GJPSoft	http://www.gjpsoft.com/UltraNetSniffer/
40	VisualRoute	Visualware	http://www.visualware.com/
41	VitalSuite	Lucent technologies	http://www.vital.lucent.com
42	WhatsUp Gold	Ipswitch	http://www.ipswitch.com/Products/WhatsUp/

Annex A (cont'd)

Table 9: Open Source tools survey

	Tool Name	Company	Product Link
1	Argus		http://www.qosient.com/argus/index.htm
2	Big Sister		http://bigsister.graeff.com/
3	Cheops		http://www.marko.net/cheops/
4	Cheops-ng		http://cheops-ng.sourceforge.net/
5	DISCO		http://www.altmode.com/disco/
6	Ettercap		http://ettercap.sourceforge.net/
7	GxSNMP		http://www.gxSNMP.org/
8	Internet2 Detective		http://detective.internet2.edu/
9	JFFNMS		http://jffnms.sourceforge.net/index.php
10	Nagios		http://www.nagios.org/
11	NBTScan		http://www.inetcat.org/software/nbtscan.html
12	NET-SNMP		http://net-SNMP.sourceforge.net/
13	NetStat Live		http://www.analogx.com/contents/download/network/nsl.htm
14	Network Probe (Open Source ???)		http://objectplanet.com/Probe/
15	NMAP		http://www.insecure.org/nmap/index.html
16	NTOP		http://www.ntop.org/ntop.html
17	Nomad		http://netmon.ncl.ac.uk/
18	OpenNMS		http://opennms.org/
19	Passifist		http://www.cqure.net/tools.jsp?id=14
20	P0f		http://lcamtuf.coredump.cx/p0f.shtml
21	Ring		http://www.intranode.com/site/techno/techno_articles.htm
22	THC-Amap		http://www.thc.org/releases.php
23	Xprobe		http://www.sys-security.com/html/projects/X.html

Annex A (cont'd)

Table 10: Inventory & Audit tools survey

	Tool Name	Company	Product Link
1	Adminpal Software manager	Adminpal	http;//www.adminpal.com
2	Desktop Inventory	Peregrine Systems	http://www.peregrine.com
3	iInventory	LanAuditor	http://lanauditor.com
4	LANDesk	LanDesk	http://www.landesk.com
5	SynexSys Inventory	SynexSys	http://synexsys.com
6	System Monitoring	Eastbow	http://synexsys.com

Table 11: Research tools survey

	Tool Name	Company	Product Link
1	Network Mapping Tool for Real-Time Security Analysis	CRC Canada	
2	NVisionIP	NCSA, University of Illinois	MILCOM 2003
3	Network Performance Advisor	The National Laboratory for Applied Network Research (NLANR)	http://dast.nlanr.net/Projects/Advisor/

Annex B: Sample Asset Inventory Listing of LANDesk

BIOS - System Serial Number =
BIOS - System Model Number =
BIOS - Manufacturer - (SeqKey:2.1) - Value =
BIOS - Manufacturer - (SeqKey:2.1) - Copyright Notice1 =
BIOS - Manufacturer - (SeqKey:2.1) - Copyright Notice2 =
Scan Type =
Type =
Device Name =
Network - TCPIP - Address =
Network - TCPIP - Host Name =
Network - TCPIP - IP Routing Enabled =
Network - TCPIP - WINS Proxy Enabled =
Network - TCPIP - NetBIOS Resolution Uses DNS =
Network - TCPIP - Bound Adapter - (Number:0) - Description =
Network - TCPIP - Bound Adapter - (Number:0) - Physical Address =
Network - TCPIP - Bound Adapter - (Number:0) - DHCP Enabled =
Network - TCPIP - Bound Adapter - (Number:0) - IP Address =
Network - TCPIP - Bound Adapter - (Number:0) - Subnet Mask =
Network - TCPIP - Bound Adapter - (Number:0) - Default Gateway =
Network - TCPIP - Bound Adapter - (Number:0) - Hidden =
Network - NIC Address =
Last Hardware Scan Date =
Processor - Processor Serial Number =
Processor - Vendor =
Processor - Type =
…
Processor - Features - Virtual Mode Extensions =
…
Coprocessor - Math =
BIOS - Date =
BIOS - System Model =
BIOS - Copyright String =
BIOS - System Serial Number =
BIOS - System Model Number =
BIOS - Monitor Model =
BIOS - Monitor Manufacturer =
Ports - Communications Port - (Name:COM1) - Address =
Ports - Printer Port - (Name:LPT1) - Address =
Bus - Type =
Mouse - Buttons =
Network - NetBIOS - Exists =
Video - Adapter - (Number:0) - Adapter String =
Video - Adapter - (Number:0) - Chip Type =
Video - Adapter - (Number:0) - DAC Type =
Video - Adapter - (Number:0) - Memory =
Video - Resolution =
Video - Colors =
Keyboard - Type =
Keyboard - SubType =
Keyboard - Number of Function keys =
Keyboard - Code Page =
Sound Card - Manufacturer =
Sound Card - Type =
Sound Card - Version =
Memory - Physical - Bytes Total =
Memory - Physical - Bytes Available =
Memory - Page File - Maximum Size =
Memory - Page File - Available =
Mass Storage - Floppy Drive Count =
Mass Storage - Floppy Drive - (Number:0) - Type =
Mass Storage - Floppy Drive - (Number:0) - Cylinders =
Mass Storage - Floppy Drive - (Number:0) - Heads =
Mass Storage - Floppy Drive - (Number:0) - Sectors =
Mass Storage - Fixed Drive - (Number:0) - Cylinders =
Mass Storage - Fixed Drive - (Number:0) - Heads =
Mass Storage - Fixed Drive - (Number:0) - Sectors =
Mass Storage - Fixed Drive - (Number:0) - Bytes Per Sector =
Mass Storage - Fixed Drive - (Number:0) - Total Storage =
Mass Storage - CDROM - (Number:0) - Drive Letter =
Mass Storage - Logical Drive - (Drive Letter:A) - Removable =
Mass Storage - Logical Drive - (Drive Letter:C) - Removable =
Mass Storage - Logical Drive - (Drive Letter:C) - Available Storage =
Mass Storage - Logical Drive - (Drive Letter:C) - Total Storage =
Mass Storage - Logical Drive - (Drive Letter:C) - File System =
Mass Storage - Logical Drive - (Drive Letter:C) - Serial Number =
OS - NT Info - Current Build =
OS - NT Info - Current Type =
OS - NT Info - Current Version =
OS - NT Info - Registered Organization =
OS - NT Info - Registered Owner =
OS - NT Info - System Root =
OS - Name =
OS - NT Info - Service Pack =
OS - NT Info - Install Date =
Environment - Variable - (Name:ALLUSERSPROFILE) - Value =
Environment - Variable - (Name:APPDATA) - Value =
Login Name =
Full Name =
LANDesk Management - Remote Control - Secure =
…
Mass Storage - CDROM - (Number:0) - Drive Letter =
Mass Storage - CDROM - (Number:0) - Description =
Mass Storage - CDROM - (Number:0) - Name =
Mass Storage - CDROM - (Number:0) - Manufacturer =
Mass Storage - CDROM - (Number:0) - Media Type =

Annex B (cont'd)

Ports - USB - Controller - (Number:0) - Name =
Ports - USB - Controller - (Number:0) - Manufacturer =
Ports - USB - Device - (Number:0) - Name =
Ports - USB - Device - (Number:0) - Manufacturer =
Ports - SCSI - SCSI Controller - (Number:0) - Model Name =
…
Printers - Default Printer =
Printers - Printer - (Number:0) - Port =
…
Modems - Modem - (Number:0) - Port =
Modems - Modem - (Number:0) - Manufacturer =
…
Resources - Resource - (Name:00000001) - IRQ =
Resources - Resource - (Name:00000005) - Port =
Resources - Resource - (Name:00000005) - Memory =
…
Network Adapters - Network Adapter - (Number:0) - Vendor =
Network Adapters - Network Adapter - (Number:0) - Description =
Database - ODBC - Driver - (Name:SQL Server) - Path = l
Database - ODBC - Driver - (Name:SQL Server) - ODBC Version =
Database - ODBC - Driver - (Name:SQL Server) - Driver =
Database - ODBC - Driver - (Name:SQL Server) - Date =
Database - ODBC - Driver - (Name:SQL Server) - Description =
Database - ODBC - Driver - (Name:SQL Server) - Version =
PDA - Windows CE - Sync Path =
PDA - Windows CE - Sync Version =
PDA - Windows CE - Device Type =
PDA - Windows CE - Device Processor =
PDA - Windows CE - Device OEM Info =
Custom Data - Registry - LANDesk Custom Fields - Serial Number =
Custom Data - Registry - LANDesk Custom Fields - Machine Type =
LANDesk Management - Server Manager - Installed =
LANDesk Management - Server Manager - Legacy =
_SOFTWARE =
OS - Drivers and Services - Kernel Driver - (Name:Abiosdsk) - Status =
…
OS - Drivers and Services - Service - (Name: Remote Control Service) - Status =
Last Software Scan Date =
Cfg - C:\LDBIOS.TXT =
…
Software - Application Suites - Application Suite - (Name:Adobe Acrobat 4.0) -
Version =
Software - Application Suites - Application Suite - (Name:Adobe Acrobat 4.0) -
Publisher =
Software - Application Suites - Application Suite - (Name:Adobe Acrobat 4.0) -
Product ID =
Software - Application Suites - Application Suite - (Name:Adobe Acrobat 4.0) -
Registered Company =
Software - Application Suites - Application Suite - (Name:Adobe Acrobat 4.0) -
Registered Owner =
…
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) -
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) - Name =
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) - File Size =
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) - File Date =
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) - Attribute Read Only
=
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) - Attribute System =
Software - Package - (Path:C:\LDCLIENT\LDISCN32.EXE) - Attribute Hidden =
…
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - Version =
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - Name =
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - File Size =
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - File Date =
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - Attribute Read Only =
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - Attribute System =
Software - Package - (Path:C:\PROGRAM FILES\MICROSOFT OFFICE\OFFICE\EXCEL.EXE) - Attribute Hidden =
Number of Files =

Annex C: Detailed tools evaluation

HPOV Network Node Manager

COMPANY

HP is a global provider of technology solutions to consumer, businesses and institutions.

PRODUCT

Network Node Manager (NNM) is part of the HP OpenView product suite. It provides a robust management for large and complex network environments.

VISUAL MAP

NNM provides a visual map of the discovered network at level 2 and level 3.

ARCHITECTURE AND DISCOVERY

In effort to improve NNM's perform on networks of varying sizes and degrees of complexity, NNM supports central, distributed and hierarchical architectures. By distributing NNM's functionality across several hosts on the network, the following is achieved:

- Remote networks can be equipped with a local management/collection station. This will reduce management traffic across WAN links, that are often relatively low bandwidth and costly.
- The management station's task can be shared among several hosts to optimize system resources at a management station.
- Having several management/collection stations provides stability in the case when one management station fails or goes off-line (maintenance, etc.)

Each distributed component contains the same NNM software but is configured to perform one or a combination of the following roles: management console, collection station, and management station. Management consoles off-load display processing from the management station to a display station without loss of performance. One or more consoles may be connected to a single management station. Management consoles typically reside on the same network segment as the management station. Collection stations serve as a remote collection points for the overall management system. Their functionality includes topology and IP status monitoring, threshold data collection, local event correlation, and event forwarding or handling on behalf of one or more management stations. Management station provides NNMs management functionality to users either directly or via one or more management consoles.

In a central mode of operation the entire functionality of NNM resides on a single station (possible with one or more management consoles). In a distributed mode, two ore more management stations manage their unique and "disjoint" segments of the network. Each management station serves as a collection station for the other. And each management station has a "full management view" of the entire network. In a hierarchical mode of operation, functionality of NNM is distributed among one management station and one or more collection stations. Collection stations (that serve management console/s) have "management view" of their unique network segments and communicate their findings to the remote management station. The management station has a "management view" of the entire network.

For network discovery, NNM uses ICMP and SNMP polling techniques and availability of SNMP agents is critical to a successful network discovery. Upon start up, NNM will discover all layer 3 and layer 2 devices that support any of the three standard MIBS: bridge, repeater, or MAU MIBs. In addition to IP, if NNM is running on a Windows 2000 host, NNM will also perform discovery of IPX devices.

The background process responsible for network discovery is the *netmon* process. *netmon* uses a combination of ICMP pings and SNMP requests sent over UDP and IPX to perform the discovery. In order for *netmon* to find nodes on the network it requires the following:

- management station's default router address
- SNMP community strings from the default router (minimum) and from other routers and nodes on the network
- the management station must run SNMP agent
- nodes must be responding to ping requests
- management station, gateways/routers must have correct subnet mask

The more routers and nodes run SNMP agents, the more effective *netmon* will be.

The *netmon* background process is responsible for periodic ICMP polling of all managed interfaces in the topology database (one of five operational databases). The polling interval is user configurable on a per-address/interface basis and can be set via the SNMP configuration dialog. If a particular interface does not reply after a preconfigured number of pings (default is three pings), *netmon* marks the interface status as down and propagates it through the system. Similarly, if an interface with a down status replies to a ping, *netmon* updates it's status to up.

In addition to interfaces from the topology database, *netmon* also pings "hint" IP addresses. *netmon* learns about hint IP addresses via ARP and routing table entries of other nodes. If a hint interface replies to a ping, *netmon* creates a node entry for it, performs an Initial Configuration SNMP poll, and adds the newly created entry to the topology database. If a hint interface does not reply after 20 pings, the interface is "forgotten" (taken off of the hint list).

After a node is discovered, *netmon* checks if the node supports an ICMP mask request. If mask request is supported, *netmon* periodically sends ICMP mask requests to that node at "configuration checking" interval. Otherwise, no mask requests are ever sent to that

node. If at a later time the node is found to support SNMP, the mask check is permanently disabled for that node.

Because all of the ICMP activity is done through a single kernel-resident ICMP socket, some of *netmon's* ICMP packets may be lost if the are other processes that are using ICMP. The lost packets will be resent. *netmon* also receives ICMP replies to requests that it did not generate. Such replies are used to learn about hint IP interfaces.

All of *netmon's* SNMP polling is done via four operations:

1 Initial Configuration
2 Daily Configuration Check
3 Discovery or AT Check (named after Address Translation table of MIB-I)
4 Demand Poll

Based on information from SNMP polls, *netmon* learns about hint nodes and updates the topology database.

The Initial Configuration operation is performed on new nodes, after a "hint" node has been confirmed. The Daily Configuration Check is done once per day on each node as part of the maintenance procedure. SNMP requests that are generated during both Configuration Checks include:

1 System object identifier (system.sysObjectID)
- topology database is updated
- event is generated if change occurred
- identifies uniques nodes

2 System descriptor (system.sysDescr)
- Stored in database
- Shown in ipmap's attribute box
- Generates event if changed

3 IP address table (ip.ipAddrTable)
- Fields ipAdEntNetAddr, ipAdEntNetMask, and ipAdEntIfIndex are retrieved
- Stored in database
- Entries ipAdEntNetAddr and ipAdEntNetMask are shown in ipmap's attributes box
- *netmon* gets ipAdEntNetAddr for the IP address; this could result in a new interface event
- *netmon* gets ipAdEntIfIndex for matching the IP address with the interface table's entry
- *netmon* gets the ipAdEntNetMask. If the IP address is the first on the network, then this becomes the network mask for all subsequent interfaces on the same network. Otherwise it is compared to the current netmask for that network. If it doesn't match, an event is generated.

4 Interface number (interfaces.ifNumber)
   - Used to limit the number of rows to be requested

5 Interface table (interfaces.ifTable)
   - Fields ifIndex, ifPhysAddress, ifType, ifOperStatus, ifAdminStatus, and ifDescr are retrieved
   - Stored in the database

If a node doesn't respond after a preconfigured number of SNMP sysObjectID requests, that node is marked as not supporting SNMP. However, *netmon* will try issuing SNMP requests every configuration checking interval in case SNMP becomes available. There are additional queries that are performed for HP hubs and bridges (switches). For nodes with HP object Ids, *netmon* also performs two SNMP *set* operations. First operation sets the agent's capabilities to respond to further HP requests. The second set operation adds the management station to the agent's trapDestination table so the traps will be sent to the management station.

The goal of Discovery operation, also called "new node discovery", is to find new hint nodes. It is performed periodically on all SNMP enabled nodes. Polling interval is either fixed or dynamic. For the dynamic case, the interval can range between 1 min and 24 hrs based on the amount of new information acquired from the node on a previous poll. The more new information a node has the more frequently it will be polled and vise versa.

The Discovery operation consists of SNMP polls of the ipNetToMedia table, the default routing table, and the entire routing table (only on nodes with WAN interfaces and only once during the first Discovery operation).

1 default routing entry (ipRouteNextHop.0.0.0.0)
   a. used for discovery of hint nodes
2 IP Net-to-Media table (ipNetToMediaTable) or address translation table (atTable)
   a. ipNetToMediaTable tried first (atTable retrieved only if ipNetToMediaTable not supported)
   b. The objects NetAddress and PhysAddress of the ipNetToMediaTable (or atTable), the ARP cache for Ethernet interfaces, are retrieved.

The Demand Poll operation is a combination of Configuration and Discovery polls. It is done on request from the user. The *nmdemandpoll* process is responsible for execution of Demand polls and works with *netmon* during it's execution. For each of the nodes that the user requests Demand Poll the following is done:

1 *nmdemandpoll* sets up a TCP socket to the system which is the primary management station for the node
2 *nmdemandpoll* sends a trap to *netmon* containing the node to be polled and the connection informantion (host and socket port number)
3 *netmon* received the trap and
   a. opens the socket
   b. marks the node as being "demand polled"

c. schedules pings of the node's interfaces one at a time and a demand poll operation on the node itself
d. prints out polling results to the socket
e. closes the socket once the polling is over

4 *nmdemandpoll* reads from the socket and copies it to stdout

EXTENSIBILITY AND DEPENDENCIES

NNM contains several operational databases that store specific kinds of data for various components or processes of NNM. In addition, NNM also includes a relational database called data warehouse that stores historical information about the network. Although the operational databases are internal to NNM and as such cannot be accessed the data from them can be exported to the data warehouse where it is readily available through the standard database APIs. The data warehouse can either use a build-in relational database or an external database. NNM supports Oracle and SQLServer databases.

The five operational databases that store NNM's internal data are:

- Object database
  Manages al object and field information for HP OenView Windows. Information from this database is not exported into data warehouse.

- Map database
  Each map within NNM environment has a map database associated with it. Map database contains presentation related information such as: symbol labels, placement of a symbol on the map, and correlation between map symbols and their corresponding data objects.

- Topology database
  Contains basic information about nodes and interfaces of the devices on the network, such as: node name, IP addresses, and router interface information. Information from Topology database can be exported to the data warehouse. Note that data warehouse doesn't retain historical topology information, only a current snapshot.

- Trend, Binary, or the SNMPCollect database
  Contains SNMP data as collected by *SNMPCollect* process. This data is used by *xnmgraph* for reporting and data collection features other. Data here is stored in a proprietary database format, but can be exported to the data warehouse using *ovdwtrend* process.

- Event database
  The event database is the repository for SNMP traps/events that are received by NNM. Data here is stored in a proprietary format but can be exported to the data warehouse.

STRENGTHS AND LIMITATIONS

Strengths of HP OpenView NNM include: layer 2 and layer 3 device discovery, visual network map, availability and ease of integration with other HP OpenView suite products for additional functionality, and access to data via relational data base for use by other 3rd party products.

Weaknesses of HP OpenView NNM include: very strong SNMP dependence, no built-in support for SNMPv3. If a node exists on the network that never talks to another SNMP enabled host or a router, such node may never be discovered. However, NNM will allow a ping request be sent to that node "manually" provided the IP address of the node is known and after that the node will be added to the topology.

IBM Tivoli NetView

COMPANY

Tivoli Software is an IBM software brand.

PRODUCT

IBM Tivoli NetView 7.1 offers proactive management of network resources. NetView is part of IBM Tivoli large suite of network management tools. Supported Platforms include: AIX, Linux, Solaris, and Windows NT/2000.

VISUAL MAP

NetView provides a visual map of the discovered network. The map is periodically updated to reflect detected changes in the topology.

ARCHITECTURE AND DISCOVERY

In effort to distribute CPU and memory loads NetView can be configured to act as either a Regional Manager or a Mid-Level Manager (MLM). With this capability, NetView supports central and distributed architectures. In a central architecture, NetView is installed on a single management station from where network discovery and management is performed. In the distributed case, MLM nodes are installed on remote and mutually exclusive subnets with the following responsibilities: network discovery, status monitoring, SNMP data collection, trap filtering and forwarding. Communication between regional manager and MLM nodes occurs either when the regional manager periodically polls MLM nodes for their discovery information or when MLM nodes generate trap messages to the regional manager.

NetView discovers layers 3 devices and relies on SNMP, ICMP, and ARP protocols. The background process responsible for network discovery is called *netmon* and it resides on the management station. With the default settings, *netmon* discovers nodes that are one hop away from the management station. *Netmon* is capable of discovering nodes beyond one hop and can be directed to do so interactively or through a seed file. For discovery, *netmon* relies on SNMP agents being present on the management station and on as many other nodes as possible. Nodes without SNMP may be discovered but cannot be managed by NetView. The ideal network would have SNMP agents running on all nodes. Using SNMP, *netmon* attempts to retrieve ARP entries from each node it discovers and from the management station. Lack of SNMP agents will result in delays and inaccuracies in the discovery process.

Network regions with MLM nodes present are not discovered by *netmon*. Netmon learned about MLM nodes either when it receives trap messages from them or through a configuration file (MLM nodes seed file).

EXTENSIBILITY AND DEPENDENCIES

The NetView background processes control and maintain the following internal databases:

- Map Database: no direct access, contents can be viewed via *ovmapdump* utility
- IP Topology Database: no direct access, data can be exported to an SQL database for use by other applications.
- Generic Topology Database: not accessible
- Object Database: no direct access, contents can be viewed via *ovobjprint* utility
- Events SQL Databases: SQL compatible database.

NetView supports MIB-I and MIB-II.

STRENGTHS AND LIMITATIONS

NetView provides a visual display and automatic discovery of network devices.

NetView is highly SNMP-based, it discovers and manages devices that run SNMP agents. Non-SNMP nodes can be discovered but not managed. NetView does not support SNMPv3.

BMC Patrol Visualis

COMPANY

BMC Software is a leading provider of enterprise management solutions. BMC Software solutions span enterprise systems, applications, databases and service management. BMC Software was founded in 1980. It has offices worldwide.

PRODUCT

Patrol Visualis is an SNMP-based network management tool for switched networks. It performs automatic discovery of the network topology and displays a map of the information system. It makes use of the network topology knowledge to perform application/traffic flow management and generate alarms (fault management).

BMC Patrol Visualis is part of the BMC Patrol suite of product. The suite includes a number of tools such as:

Patrol Dashboard
Patrol Enterprise Manager
Patrol Knowledge Modules

Visualis can be integrated with the above to offer a more complete network management solution.

VISUAL MAP

Visual display is one of the tools' strength. Visualis performs 3D graphics display of the network topology at layer 2 and layer 3. It claims to have a very efficient and accurate topology resolution engine (provided that SNMP is enabled in the network) that makes use of sophisticated algorithms to resolve the topology of the network.

Visualis also displays real-time information on data flows as well as performance indicators.

ARCHITECTURE AND DISCOVERY

The discovery process makes use of a central architecture. A single management station performs the auto-discovery of the devices (routers, switches, servers, workstations,…) and creates the map.
Visualis heavily relies on SNMP to perform its auto-discovery. In the absence of SNMP, only the host and their corresponding ip addresses will be discovered (using active pinging). The tool includes support for SNMP version 1, 2 and 3.
Visualis automatically discovers the network components at layer 2 and 3. It is also capable of discovering VLANs i.e. identify and group devices that belong to the same IP

subnet but that are connected to different physical network segments and maybe geographically separated.

Once devices have been identified, the tool collects configuration and performance data. It specializes in providing very good application flow statistics, network performance information and congestion indications.
As for configuration information, data flow information is extracted via SNMP. It typically makes use of values obtained from RMON, Cisco Netflow MIB values and other SNMP performance MIB values.

EXTENSIBILITY AND DEPENDENCIES

Data collected is stored in an Oracle 8i relational database.

Visualis also exploits the Common Information Model (CIM) standard [10] to represent the various network devices/objects.
The CIM is an object oriented information model that attempts to completely describe the managed network/enterprise environment. The model has been developed by the Distributed Management Task Force (DMTF). The model is intended to complement and unify existing management standards such as SNMP, CMIP, etc, rather than act as a direct replacement.
The CIM gains value over existing standards from its object oriented nature not available in other modeling paradigms (e.g. SNMP tables). The object-oriented design promotes the extensibility.
The CIM consists of a specification (the language and rule set used to describe the model and its interactions with other management standards), and a schema (the model conforming to the specification) which facilitates the accessibility to the data.

STRENGTHS AND LIMITATIONS

Visualis is a network discovery tool that offers very similar features as the ones offered by the other industry leaders SNMP-based tools such as HPOV NNM and IBM Tivoli. Perhaps the feature that differentiates it the most from its competitors is its strong visual capability (3D graphic display of the topology and animated data flow representation) as well as its real-time application performance statistics reporting capability.

It also claims to have one of the most efficient and accurate topology resolution scheme on the market (including discovery of VLANs), although strictly dependant on SNMP to achieve it.

Micromuse NetCool Precision for IP

COMPANY

Micromuse is a global software company that addresses end-to-end management of complex IT and telecommunications infrastructures. The company was founded in 1989 in London, England. Since then, the company has completed four acquisitions: Calvin Alexander Networking (2000), NetOps (2000), RiverSoft plc (2002), and Lumos Technologies (2002). Micromuse's flagship product is the NetCool suite.

PRODUCT

NetCool Precision for IP automatically discovers IP networks, gathers inventory and topology data at layers 2 and 3. It maps the network topology in a visual diagram that shows the connections between devices. It monitors all discovered information for status and continually updates its database with new information as the network changes.
It also makes use of the knowledge it acquires from the network to conduct correlation on data and perform root cause analysis.

NetCool Precision for IP is part of the NetCool Suite. The Suite includes:

- NetCool/OMNIbus
- NetCool/Service Monitors
- NetCool/Reporter
- NetCool/Impact
- NetCool/Visionary
- NetCool/Precision

It can be integrated with the above to offer a more complete network management solution.

VISUAL MAP

NetCool Precision provides a network topology at layer 3 and layer 2, displaying a map which shows port-to-port connectivity between devices.
It is also capable of displaying VLAN logical connectivity.

ARCHITECTURE AND DISCOVERY

NetCool Precision heavily relies on SNMP to perform its discovery process. It initially uses ICMP (ping) to determine the existence of devices. If then uses SNMP to retrieve information from the device (the latter succeeds only if the device is SNMP enabled). The tool supports SNMP version 1, version 2 and version 3.

Assuming SNMP is present on network devices, Precision will discover all layer 2 and 3 devices, interfaces and connectivity. It also has the capability of discovering VLANs,

VLAN membership and VPN membership. This capability is supported via extensive built-in VLAN MIBs knowledge.

As for VLANs, Precision is capable of monitoring and extracting information for a variety of technologies and protocols via its built-in specific MIBs support. For example, it can extract information on the OSPF routing protocol by reading the values off the "standard" OSPF MIB (rfc 1850) supported in a router. The values obtained are then used in the discovery knowledge. In that manner, some of the protocols used for discovery (via built-in MIB support) are:

OSPF
BGP
MPLS
STP
ILMI, PNNI (ATM)
Cisco Discovery Protocol

This built-in MIB knowledge that the tool implements for a number of various technologies differentiates NetCool Precision from other similar tool such as HPOV.

Although heavily SNMP-based, Precision supports a very limited ongoing passive listening capability. The tool implements a sniffer that listens to the ARP broadcast to discover new devices.

Precision supports both central and distributed architectures. In the case of distributed architecture, it supports hierarchical and peer-to-peer. Typically, each server is responsible for discovering and monitoring a local network. Servers can be configured as "slaves". Each slave feeds its information back to a "master server" machine where data is all synthesized.

EXTENSIBILITY AND DEPENDENCIES

NetCool precision provides good export and extensibility support. There are several ways to access the data from Precision:

a. One way to extract data from precision is using a DIST adapter. A DIST adapter is just an application adapter that allows Precision to export data directly to a database. Currently, there exists DIST adapters for Oracle, Sybase and Remedy. These adapters listen to the internal TIBCO bus and push data to one of the supported databases. Almost all data collected by Precision can be sent using a DIST adapter.
   DIST adapters will export the data as it is stored in Precision and may require lists to be parsed.

b. Another way to extract data is by using JAVA or PERL APIs provided with the tool. These APIs allow a user to extract any data from Precision, arrange it into any desired format, write to a file or push it to a database or use it as per the user's requirements.

c. Perhaps the best way to manage the data is via a proprietary solution product called Netcool for Asset Management (NfAM) which can export the data to an Oracle 8i database. This is the most complete, preconfigured tool for gathering information. NfAM normalizes the data and makes it more human readable and easy to query using, for example, standard SQL. Data can quickly be retrieved without having to parse a list (only have to issue a standard database query).

Of the above, DIST is a good solution for exporting data that has a single return. For example, information such as sysDescr, sysContact and sysName, will have one entry per device and will work great with a DIST adapter. If the information to retrieve is contained in a more complex format (such as interface information for example) than NfAM is a better solution.

Also, the NfAM creates the appropriate database schema automatically, as opposed to the DIST adapter where the user needs to create its own database schema.

STRENGTHS AND LIMITATIONS

NetCool Precision offers similar capabilities as the other well known commercial SNMP -based discovery tools. Perhaps its most distinctive feature is its broad discovery support of network technologies and network vendors through its multiple built-in MIBs. It also seems to offer a good data export capability, however, the best solution being available only via a proprietary product.

Castle Rock SNMPc

COMPANY

Castle Rock was founded in 1987. The company initially was specializing in LAN networking. The company now focuses in providing network management solutions. SNMPc is their only product.

PRODUCT

SNMPc is a network management solution that performs monitoring and control for small-to-midsize IP/IPX networks. It auto-discovers the network, draw visual maps and actively manages the devices using SNMP.

VISUAL MAP

SNMPc provides a level 3 only topology map display.

ARCHITECTURE AND DISCOVERY

SNMPc performs automatic discovery of level 3 devices. The tool uses ARP requests (on local subnet), ICMP (ping) and SNMP polls to achieve discovery. The discovery process can be initiated using two methods: via seed routers and/or specified network ranges. It supports SNMP version 1, 2 and 3.

Once devices have been found, SNMPc has the ability to also perform limited service discovery/application status monitoring. The supported services are TELNET, FTP, HTTP, SMTP and four user-specified TCP ports.

Architecturally, SNMPc consists of separate polling, database servers and console components to enable scalable distributed management. The following architectures are supported:

Central
All the components mentioned above run on a single workstation.

Server/client
Polling agents are distributed in the network. The agents perform discovery of local networks and monitors the status and alarm thresholds for discovered devices (using ICMP and SNMP requests). The collected data is stored in local databases. The information is sent back from the polling agents to server(s).

Hierarchical/Peer-to-peer architecture

Servers can be configured to act as slave and/or master servers. Master servers can import maps and information from slave servers. A full peer-to-peer architecture is also supported where each server can operate both slave and master roles simultaneously.

The various supported modes of operation are shown in Figure 7.

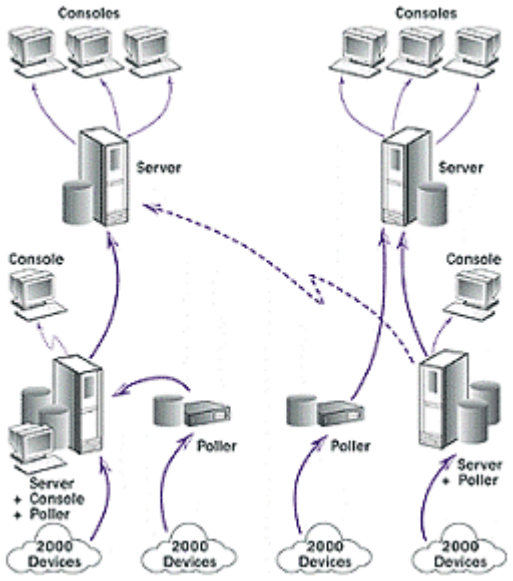

Figure 7: Supported modes of operation of SNMPc

EXTENSIBILITY AND DEPENDENCIES

SNMPc can export the map topology, statistics and events to standard ODBC databases. It also supports some limited text export capabilities.

It has an integrated MIB compiler which allows the tool to include any third party vendor MIBs provided that the MIB follows the standard ASN.1 format.

The tool offers a number of programming interfaces (APIs) to allow customization of menus, graphs, tables,…
The supported programming interfaces are:
- Proprietary object-oriented interface for C/C++ (SNMPc dll APIs to create programs that operate on object classes and run as components.)
- WinSNMP standard API for SNMP based programming under Windows.
- DDE interface. Can be used to create programs to query the map, MIB databases and execute SNMPc commands.

STRENGTHS AND LIMITATIONS

SNMPc is a simple network discovery tool that presents similar capabilities as the other SNMP-based tools. It offers a variety of architectures, a feature that allows scalability and promotes flexibility.

OpenNMS

COMPANY

Opensource software. OpenNMS is developed and maintained by a large group of individuals working together.

PRODUCT

OpenNMS is an open-source enterprise-grade network management software. It provides the traditional SNMP-based functionality and also has the ability to actively monitor services being provided on the network.
It is targeted at providing an alternative to proprietary expensive and complex network management solution.
It differentiates itself from other SNMP-based open source software by the fact that it was designed from the beginning to manage a very large number of devices.
Its main capabilities are:

- automatic node discovery
- network services monitoring,
- operator notification of problems
- service level performance monitoring

VISUAL MAP

OpenNMS does not provide a network map. The original developers of OpenNMS have chosen not to implement map support. The decision was based on their experience indicating that network managers rely more on the event list than the map. Instead, it was decided to develop a Web GUI with performance graphs and various table displays that show status and information classified by categories of devices.

That said, there is a group within OpenNMS that is currently working on developing a mapping capability.

ARCHITECTURE AND DISCOVERY

OpenNMS automatically discovers IP nodes at layer 3 only. The discovery process in OpenNMS consists of two parts: discovering an IP address to monitor and then discovering the services supported by that IP address.

To discover the devices, the tool initially performs an active ping scan on a specified range of IP address. Then for each discovered IP address it performs service discovery by actively polling specific ports. The types of services are specified using a protocol/port association. Supported services need to be defined in a configuration file. If the service

is not listed in the configuration file, the service will not be discovered. The list of services that are currently supported are as follows:

Citrix, DHCP, DNS, Domino IIOP, FTP, HTTP/HTTPS, ICMP,IMAP,LDAP,MS Exchange, POP3, SMB, SMTP, SNMP and TCP (with a user specified port number).

Once all the "protocols" have been tested, for each IP address that was found to support SNMP, more information is collected via the SNMP protocol. OpenNMS, at the moment, does not have a MIB compiler to automatically import OIDs for collection. Therefore, each OID of interest need to be entered in a configuration file (following a particular structure defined by the tool). By default, OpenNMS supports most of the MIBII objects.

OpenNMS currently supports SNMP version 1 and 2 only. Work is in progress to integrate SNMP version 3.

The tool operates using a central or distributed architecture. In the distributed architecture, computers act as "pollers". They are distributed in various location of the network and they are responsible for the discovery and for the polling/monitoring of the services.
A station acts as a "master". It is the machine where the central repository for the information is located. It is as well the centralized point of administration (interface to the users).
The communication and data exchange between the pollers and the master station is done using a proprietary protocol.
In the central architecture, the same machine performs both functions i.e. the machine is simultaneously the master and the poller for the entire network.

The tool's operation and behavior is based on a number of configuration files. These files address various capabilities of the tool such as data collection configuration, notifications configuration, monitoring configuration, services configuration, views configuration, etc…
The extensive number of files has the advantages of providing much freedom and a high level of customization to the user. On the other hand, it may be cumbersome and adds to the complexity of the tool.
All the configuration files are formatted using XML. XML is also used by the tool for internal events specifications.

EXTENSIBILITY AND DEPENDENCIES

OpenNMS export capabilities are limited to transfer of its data values to CSV and XML files.
Information seemed to imply that some limited database queries would be supported. This aspect requires further investigation.

For proper operation, the tool has the following software dependencies:

- Java Virtual machine

- Perl modules
- RRD Tool
- TomCat
- PostgreSQL

STRENGTHS AND LIMITATIONS

OpenNMS has been developed with the purpose of offering a free alternative to the high cost, complex commercial network management solutions. Its capabilities are similar to the other SNMP-based discovery tool although it does not support a network map display.

The tool was designed to be extremely configurable and customizable. It implements a rule-base configuration using the XML technology.

It also implements a notification system that is able to receive/catch SNMP traps generated by a third party tool such as Snort. Snort can be configured to inform and send SNMP traps when it detects unusual network activity. OpenNMS has the ability to receive those traps and automatically generate event alarms via email for example, to notify the security administrators.

Nomad

COMPANY

An open-source community effort led by Paul Coates (Paul.Coates@ncl.ac.uk) at University of Newcastle upon Tyne. Tools web site: http://netmon.ncl.ac.uk/

PRODUCT

Nomad is SNMP-based network discovery/mapping tool with some monitoring capability. It presents visual view of the network as a topology diagram and maintains it with periodic updates. The state of the tool is very much "work in progress" and is typical of the Open Source tools that have been looked at within the scope of this report. The latest version released is 0.3.2 and is based on net-SNMP, a collection of SNMP tools developed by the open-source community (http://net-SNMP.sourceforge.net/).

Protocols Used for Discovery:
SNMP: Net-SNMP supports all three versions of SNMP. Since the latest release of nomad is based on net-SNMP, thus the support for all three versions.

VISUAL MAP

The program presents a visual map of the discovered layer 3 network.

ARCHITECTURE AND DISCOVERY

Relies on SNMP agents being present on all hosts and supports central architecture.

EXTENSIBILITY AND DEPENDENCIES

The tool's source code is freely available under the terms of GNU General Public License as published by the Free Software Foundation.

STRENGHTS AND LIMITATIONS

The development effort for this open-source product has been concentrated around the creation and presentation of a visual map from information gathered via SNMP.

The product can be described as "work in progress" with limited documentation and support.

Fluke OptiView

COMPANY

Former part of Fluke Corporation, Fluke Networks became a separate company in May 2000. Both Fluke Networks and Fluke remain part of the Danaher Corporation.

PRODUCT

The OptiView Console application provides the ability to monitor network performance, generate reports, map network configuration, and generate a notification if problems with network or devices arise. Network scope is limited to LAN. However, OptiView Console agents can be distributed and integrated with other Fluke Network software and hardware gents to extend monitoring scope beyond the LAN and to include the entire enterprise.

OptiView Console runs on Windows NT/2000/XP and requires MS Visio 2000 or later for map drawing.

VISUAL MAP

OptiView Console works in collaboration with MS Visio 2000 to draw visual network map. The map includes layer 2 and layer 3 devices.

ARCHITECTURE AND DISCOVERY

OptiView Console supports central and distributed architectures. OptiView Console consists of the Viewer and the Service Manager that can coexist on the same host or can be installed separately. The Viewer is the main user interface that provides access to all components of the OptiView console. Service Manager is the engine that performs network discovery, data management, data analysis, and provides notification services.

The Service Manager gives you status information and configuration control of the services that are part of the OptiView Console application. These include the Local Agent, Import, Analysis, and Notification Services. The Agent service discovers devices on your network and stores the information in a Microsoft Desktop Engine (MSDE) SQL Server database. The Import service stores information discovered by Hardware Agents in a separate database for each agent. The Analysis service runs algorithms on the information stored in each database and determines network configuration, error conditions, and other network information. The Notification service flags error or performance conditions discovered on your network and can be configured to generate email, email to pager, and SNMP traps to alert network administrators.

Under central architecture, full functionality of OptiView is provided from a single host, i.e. view and service manager reside on the same host. To support distributed architecture, OptiView
Viewer and Service Manager are two main components of OptiView Console application.

Supported architectures include central and distributed-hierarchical.

In a central architecture, the OptiView is installed on a single PC. Under this architecture, network discovery is limited to the broadcast domain of the PC that hosts the OptiView Console. Discovery of nodes beyond one hop is limited to other Fluke devices only.

To support the distributed-hierarchical architecture, OptiView Console has the capability to be installed in any one of the following configurations:

- Complete Master: Viewer and Service Manager are installed on the same PC
- Complete Remote: Viewer and Service Manager are installed on the same PC in a different broadcast domain than the Master. The first time it runs, the software will require IP address of the Master agent.
- Viewer Only: Viewer is installed on a different PC than the master. The first time it runs, the software will require name/IP address of the agent whose database you want to view.
- Agent Only: Service Manager is installed on a PC in the broadcast domain different than the Masters. The first time it runs, it will ask for name/IP address of the Master.

The limit on the number of installations of each configuration is set by the purchased license agreement.

Discovery

The agent uses a combination of passive and active techniques to perform network discovery.

Passive techniques include:

- listening to unicast traffic addressed to the PC where the agent resides
- listening to broadcast and muliticast traffic

By listening to this traffic and extracting source addresses, the agent learns about devices present on its subnet. Passive discovery also includes “protocol analyzer like” filtering/parsing for various routing and service packets, such as: RIP and OSPF. Nodes generating such packets are marked as routers.

Active discovery initially includes sending a set of IP, IPX, and NetBIOS broadcast messages that would cause a reply from nodes on the network. Reply messages are parsed and various attributes are extracted.

IP broadcast messages include ICMP echo request and a broadcast message to UDP echo port. Source addresses are extracted from replies to these messages. For each discovered IP address, the agent sends a DNS query to learn the devices DNS name. Next, the agent sends an SNMP System Group query. If the node replies, the agent generates additional

SNMP Interface and RMON queries. For SNMP-enabled nodes, amount and content of node-specific information is limited by MIB-II. For non-SNMP nodes, information is limited to IP and MAC addresses, NetBIOS names, and services discovered via IPX queries. For nodes that support RMON (Lite, v1, or v2), additional measurements may be taken using a built-in RMON Inspector application.

IPX broadcast messages include Service Advertising Protocol (SAP) discovery requests and Network Control Program requests. Replies to these messages are parsed for identification of File servers, Print servers, Novel Directory Services (NDS) servers, Time Synchronization servers, Net Manage servers, Remote Access, and RIP Router services. For each discovered IPX device, the agent attempts to run userlist or nlist on the Novel server to determine user login name and any services advertised by the device.

A number of queries are used to discover NetBIOS names, domain/workgroup, and to identify any services offered. Services may include: Primary Domain Controller, Backup Domain Controller, and Master Browser.

EXTENSIBILITY AND DEPENDENCIES

Discovery and analysis information is stored in a Microsoft Desktop Engine (MSDE) SQL Server database. The database may be opened using another database tool, however, care must be taken not to modify data in the OptiView's database otherwise OptiView may not work as intended. The documentation suggests copying the database before using it with another application.

OptiView is dependant on MS Visio 2000 for actual drawing of the discovered network.

STRENGTHS AND LIMITATIONS

OptiView strength lies in its discovery approach. It uses a combination of active and passive techniques to achieve its network discovery. Active techniques used include SNMP and non-SNMP methods. OptiView provides a visual map of the discovered network.

OptiView's drawbacks include: dependency on MS Visio for map drawing and it does not offer support for SNMPv3.

FoundStone

COMPANY

Foundation + Cornerstone = Foundstone. Founded in 1999 and based in Mission Viejo, California. The company is being funded by OVP Venture Partners; Riordan, Lewis & Haden (RLH Investors); Banyan Capital Partners; Motorola; Articon-Integralis; Itochu Corporation and Wilson Sonsini Goodrich & Rosati.

PRODUCT

Foundstone 3.0 Enterprise was designed to address security vulnerabilities within a network. It achieves it's task with almost "hacker like" approaches. It first discovers and maps the entire network including systems, routers, firewalls, servers, and custom Web applications and then probes each component for known vulnerabilities.

Runs on Windows 2000 Server platform.

VISUAL MAP

As part of its functionality FoundStone offers a visual map of the discovered network.

ARCHITECTURE AND DISCOVERY

The Foundstone Enterprise 3.0 consists of the following three components:

- Foundstone Enterprise Manager
- FoundScan Engine
- Foundstone Database

Enterprise Manager is a web-based console/GUI that provides the ability to run and manage functionality of Foundstone 3.0. Authorised users may login to Enterprise Manager using their Web browsers. User access is protected via user name and password and communication can be set up to use a Secure Socket Layer (which provides encrypted communication to browsers).

FoundScan Engine is the component that does all the "data gathering" work in Foundstone. It is responsible for scanning the network for information and for transferring collected data into the database. One or more FoundStone Engines can be deployed with the system.

Foundstone Database provides data repository for the Foundstone system and is based on the Microsoft SQL Server.

System components communicate using SOAP/XML.

The minimum recommended architecture for the system is the distributed-hierarchical architecture of two servers: one server hosting Enterprise Manager and the other server hosting FoundScan Engine and the Database. This architecture is said to be suitable for small to medium organizations that will be scanning up to class B networks. For organizations requiring to scan multiple class B and class A networks a minimum of three server architecture is recommended: one server hosting each of the three system components described above. Systems with multiple FoundScan Engines will require dedicated servers for each FoundScan Engine.

FoundScan Engine achieves network and service discovery using a combination of proprietary algorithms. To discover connectivity, identify hosts and services/open ports, the algorithm relies on ICMP and TCP based scanning, traceroutes, OS fingerprinting, and banner grabbing techniques.

EXTENSIBILITY AND DEPENDENCIES

Foundstone's Database is SQL-compliant. This ensures direct access to data for use by other "custom" programs.

Foundstone Scripting Language (FSL) provides an interface for creating custom vulnerability checks.

FoundStone requires MS SQL Database Server and MS Internet Information Services (IIS) Web Server.

STRENGTHS AND LIMITATIONS

FoundStone's security-oriented auto-discovery technique stands out among other COTS tools. It uses a combination of active non-SNMP techniques to achieve its network discovery. It provides security vulnerability assessment. Foundstone provides visual network map.

Nmap

COMPANY

Nmap is free and is available under the terms of the GNU GPL with full source code.

PRODUCT

Nmap is a network scanning utility that detects what hosts are up and what services they are offering.
Nmap runs on following OS types: Linux, Windows (95, 98, NT, ME, 2000, XP) and Unix.

VISUAL MAP

The tools does not offer visual network map.

ARCHITECTURE AND DISCOVERY

Central: Nmap operates from a single host and performs scans of remote hosts.

To perform its' discovery, Nmap has a number of build-in TCP, UDP, and ICMP scans. Some of the scan's provide the same information but vary in their technique. This is needed in presence of filters (firewalls, etc..).
Each scan has it's strengths and limitations and attempts to succeed where the other scan fails. Therefore to gain fuller view of the network, it is sometimes necessary to run a combination of theses scans. For example if a particular site blocks ICMP echo request packets, an ICMP ping scan on that site would be of no effect. However, either a TCP SYN or TCP ACK scans could be successfully used to determine hosts status when ICMP pings fail. The following is a list of Nmap's build-in scans:

- TCP SYN scan: –sS option
- TCP connect() scan: -sT
- ICMP ping scan: -sP
- UDP scan: -sU
- TCP FIN scan: -sF
- TCP Xmas Tree scan: -sX
- TCP Null scan: -sN
- And others …

In addition to discovering hosts and opened ports, Nmap also determines which services are running and the OS type and version.

EXTENSIBILITY

The tool's source code is freely available under the terms of GNU General Public License as published by the Free Software Foundation.

STRENGTHS AND LIMITATIONS

Nmap is a well-known and used open-source tool for network scanning and has reached a certain level of "maturity". It has been reasonably well documented throughout its evolvement. The tool provides fast layer 3 network discovery.

The product does not offer visual network map and discovery is limited to layer 3.

Cheops-Ng

COMPANY

The person behind this Open Source project is Brent C. Priddy. Tool website: http://cheops-ng.sourceforge.net

PRODUCT

Cheops-ng is open-source software that offers layer 3 network auto-discovery and network mapping functionality. The product state is very much "work-in-progress". Documentation is very limited or non-existent. Author of the tool was contacted to answer question and clarify some of the tool's functionality.

ARCHITECTURE AND DISCOVERY

The discovery process uses combination of active non-SNMP techniques. From Cheops-ng source code file, dependence on Nmap source code can be observed. Cheops-ng supports central architecture.

EXTENSIBILITY AND DEPENDENCIES

The tool's source code is freely available under the terms of GNU General Public License as published by the Free Software Foundation.

STRENGTHS AND LIMITATIONS

The tool presents a visual map of the discovered network. The tool uses non-SNMP active discovery techniques very similar to those of Nmap.
The tool is very much work-in-progress with very limited documentation and support.

Big Sister

COMPANY

Open Source software. Big Sister is developed and maintained by a small group of individuals.

PRODUCT

Big Sister is an open-source network monitoring tool. It is based on the Big Brother tool but with significant feature and performance improvements.
It's primary functions are:

- monitor network systems
- provide a simple view of the current network status
- generate alarms on status change
- generate history of status change

VISUAL MAP

Big Sister does not provide any network map. Its GUI mainly displays monitoring status information/events per tested host/service. It also displays alarms and performance data.

ARCHITECTURE AND DISCOVERY

Big Sister does not automatically discover devices on a network. Instead, it automatically gathers information and constantly monitors pre-specified hosts. As such, its mode of operation closely resembles the one of an inventory software tool.

Every monitored system is monitored by a Big Sister agent. Ideally, it requires that Big Sister agents be deployed/installed on the systems to be monitored as some tests the agent performs are only applicable to the system hosting the agent. It is however possible to monitor, via SNMP polling, devices for which Big Sister agents could not be installed (e.g. switches and routers).

Each Big Sister agent operates based on information read from manually pre-configured files. Configuration files provide the agent information on what checks it will run, on what hosts and where the collected data will be sent to (server address). The monitored systems can either be local or remote. However if remote, only a limited set of checks can be done via network tests.

Currently, a Big Sister agent has the following monitoring capabilities (requires correctly pre-configured files):

- machine status via ping
- cpu load

- services status (http, tcp such as pop3, smtp, ftp,printer,…)
- running processes (nfsd, sendmail, lpd,..)
- amount of free disk space
- SNMP polling (to devices that have SNMP support)
- monitoring of SNMP traps
- memory usage (on Win NT and Linux)
- oracle database check
- monitoring of syslog and event log files

The agents send the monitored information to a Big Sister server. The server receives the information collected by the agents and processes the info to build status reports, generate alarms, show graphical display, store the data, etc…

Communication between the agents and the server is performed via a proprietary TCP/IP protocol.

The server does not perform an active discovery of the agents. It listens and accepts communication from the agents. New agents will be discovered as they initiate communication with the server (auto-join feature).

EXTENSIBILITY AND DEPENDENCIES

As it is often the case for open-source, its successful installation and operation is dependant on the presence of other software modules. Big Sister namely requires a Perl interpreter, a Web server and the RRD Tool (for graphical display support).

Big Sister does not seem to have any real export capability. Some of its data can be made available in a CSV file format. There are, however, plans to implement database support in the future.

STRENGTHS AND LIMITATIONS

Big Sister's mode of operation presents both some strengths and limitations. The proprietary agents offer the advantage of being able to retrieve information which would not be accessible via standard methods (such as SNMP).

On the other hand, the tool does not auto-discover the network devices and requires significant manual configuration prior to its operation.

CRC Network mapping tool

ORGANIZATION

Communications Research Centre, Canadian Federal Government

PROTOTYPE

The Research Network Systems (RNS) group of the Communication Research Centre (CRC) has been working for two years on a network mapping tool. The software prototype tool has been developed in-house as part of a research project in network security.
The tool automatically discovers network devices at layer 2 and layer 3 of a LAN, provides both the physical and logical connectivity of the components as well as provides configuration information on each discovered network node.

VISUAL MAP

The tool creates a network map at layer 2 and 3. The physical link-level map (layer 2) includes low-level information such as specific port connections on network devices.

ARCHITECTURE AND DISCOVERY

The tool combines active and passive techniques to achieve discovery. Initial discovery is performed by actively scanning devices within a specified range of IP addresses. Once devices have been identified, the tool attempts to gather configuration information. The configuration information that the tool attempts to find is as follows:

- IP address
- MAC address
- OS type and version
- Device type
- Device name
- Device state
- Manufacturer name (e.g. Cisco, 3COM,…)
- List of services running on the device
- List of open ports
- Interconnectivity at layer 2 and 3

After initial discovery is completed, the tool makes use of passive techniques to discover any new host that would connect to the network (e.g. listens to ARP broadcast and ICMP echo request/reply).
The tool also ensures that the connectivity, state and configuration information is maintained current for all hosts that have been discovered on the network by performing regular polling updates.

At the moment, it supports a central architecture only. All operations are performed from one single workstation. Its scope of discovery is also currently limited to LANs.

EXTENSIBILITY AND DEPENDENCIES

The tool has currently no support to export its data.
All the information that is gathered passively is, however, stored in a MS Access database.
Being a non-commercial tool, any export capability can be added as needed. XML support is currently being investigated.

The tool has a few software dependencies as it integrates some freely-available tools such as Nmap and Xprobe.

STRENGTHS AND LIMITATIONS

Being a research prototype, the tool implements some unique features that distinguish it from other tools (COTS or open-source) that offer similar functionality. The most important unique features are:

- It implements a unique algorithm to create the level 2 network map (physical link level device interconnectivity). The technique relies on SNMP but as opposed to other known techniques, it only requires access to the bridge MIB of the network link layer devices (switches).
It then makes use of artificial intelligence (AI) algorithms to process the information and provide a consistent view of the device interconnectivity.

- It is capable of discovering and creating the level 2 interconnectivity of a remote LAN. In this case, the algorithm only requires access to the bridge MIB of the switches in the remote LAN as well as SNMP access to the LAN border router (default gateway).

- It is capable of discovering VMWare virtual machines (emulated workstations) that may be running on the LAN and their relationship with the system hosting them.

- The tool's most distinctive feature is its ability to provide very accurate results, namely for the level 2 map display, host OS detection and port detection. As opposed to other network mapping tools that often use only one technique to retrieve the information, the CRC research tool combines a number of methods together to gather the information. It then examines the output obtained by each technique and correlates the results to provide a much more accurate and complete information.

WORK IN PROGRESS

Research efforts are currently deployed to extend the tool to display host vulnerabilities along with basic system configuration, link state information and network topology. It is

believed that much benefit can be gained by combining network knowledge with security events. Correlation algorithms are being investigated.

Other efforts are being put in migrating the tool towards a distributed passive only system. A prototype that makes use of passive techniques only to gather network information has been developed. Distributed functionality is being built-in to allow the tool to be deployed in various portions of the network.

IPsumnetworks Route Dynamics

COMPANY

IPsum Networks is an American startup company based in Philadelphia. The company specializes in the analysis and diagnosis of IP networks.

PRODUCT

Route Dynamics is a real-time automated IP fault management tool.
The technology provides passive monitoring of Internet routing protocols. It is implemented in an IP Listener appliance that automatically discovers the logical flows of a network and draws a logical topology map. The map details routing redundancy, path cost and tuning parameters.

Route Dynamics monitors the routing Layer 3 infrastructure in real-time and uses the data to provide real-time notification of routing and flow changes.

VISUAL MAP

Route Dynamics displays a layer 3 topology only. It actually provides a real-time "router's eye view" of the network. The IP map includes only the devices that participate in the routing as well as their logical relationship to subnets where end-hosts reside.

It also displays IP application flows on the map i.e. it identifies the successive sequence of network elements traversed by a specific application.

ARCHITECTURE AND DISCOVERY

Route Dynamics is not a software only solution. It implements a distributed client/server architecture appliances solution.
IP Listeners appliances (the "clients") perform the discovery. IP Listeners need to be deployed in various routing areas of the network. Before initiating the discovery process, IP Listeners first establish an adjacency with a router in a particular OSPF area. Once the adjacency has been established, the Listeners passively start the auto-discovery of the layer-3 network elements. Listeners collect the information by passively monitoring the routing protocols. Currently, the only supported routing protocol is OSPF. The support of BGP is in development and should be integrated in the near future.
The real-time IP state information is forwarded to a central appliance (the 'server") called the Multi-Area Path Appliance (MAP). The central server (the MAP) stores, correlates and synthesizes the information obtained from the multiple Listeners into a cohesive whole and presents the IP logical view of the network.

The communication between the Listeners and the Central appliance (MAP) is performed via a proprietary TCP/IP protocol. Listeners must be manually configured with the

MAP's IP address in order to be able to forward routing tables and updates back to the MAP.

EXTENSIBILITY AND DEPENDENCIES

The export capabilities of the tool are basically limited to pre-formatted reports generated from the GUI. As well, most collected data such as the lists of routers, routes, etc. can be exported to a MS Excel spreadsheet. There is currently no export support to standard databases.

Also, to allow for extensibility, a Software Development Kit (SDK) is available. The Software Development Kit is in fact a module available within Route Dynamics. It provides a structured methodology (APIs) to allow access to the data discovered within a network.

STRENGTHS AND LIMITATIONS

Route Dynamics' uniqueness resides in its ability to provide constant monitoring of IP routing and IP application flow paths. The monitoring and discovery is performed entirely passively, therefore not affecting the actual data flow. It does not rely on any traditional network management protocols such as SNMP although currently depends on the presence of OSPF in the network.

The tool operates by passively monitoring Internet routing protocols. The support of these protocols is, at the moment, rather limited. Route Dynamics only currently supports OSPF. BGP is in development. Any distance-vector based routing protocol such as RIP, IGRP and EIGRP will never be supported[7].

Discovery is limited to layer 3 routing devices only. The tool provides no configuration information other than the routing data and paths flow information.

[7] The company invokes technical issues with these protocols.

Sourcefire Real-time Network Awareness/ Snort

COMPANY

SourceFire was founded by the creators of Snort, one of the most widely deployed Intrusion Detection technology. The company specializes in network security and works at developing a unified security monitoring infrastructure solution.

PRODUCT

SourceFire Real-time Network Awareness (RNA) product is a passive sensing technology and analysis tool that discovers and monitors network assets such as servers, routers, PCs, firewalls, …

RNA builds on top of Snort, the well known open-source intrusion detection system.
At the time of writing, RNA has not yet been released. It is undergoing a beta testing phase. The information the company accepted to release was very limited.
At this point, it is still unclear whether the product will be sold or made available via the open-source community.

There is currently a clear trend in trying to integrate network knowledge with security event information. RNA directly addresses this trend and was developed to be used in combination with Sourcefire network sensors and management console in order to provide an integrated security monitoring system.

VISUAL MAP

Based on the information that could be obtained, it is believed that RNA does not provide any map display. However, this needs to be further verified.

ARCHITECTURE AND DISCOVERY

RNA performs its network discovery using passive monitoring only. The network elements it can discover are:

- MAC address
- OS and version
- Services and versions
- Ports
- Traffic flow, traffic type and traffic volume
- Network info such as hop count, TTL values, MTU parameters

The passive network discovery technique used by RNA naturally calls for a distributed architecture support. RNA sensors are distributed in the network. The sensors monitor the network assets. The number of RNA sensors required and deployed will, of course, vary

based on a number of factors including network topology, network size and the level of detail desired.
Real-time consolidation and analysis of the information collected by the various sensors is performed by a management console. The management console is responsible for data aggregation, consolidated reporting, network visualization, sensor configuration and policy distribution.

EXTENSIBILITY AND DEPENDENCIES

No information could be obtained.

STRENGTHS AND LIMITATIONS

RNA will be one of the rare tools (if not the only one) that performs true passive network discovery. Its discovery capabilities are, of course, limited by the inherent capabilities of the passive method. On the other hand, it also benefits of all the advantages of this approach (ref. auto-discovery techniques in Section 4).

In addition to performing passive network discovery, the tool also conducts behavioral profiling and vulnerability analysis in order to provide an improved visibility of the network environment.

LANDesk

COMPANY

LANDesk Software Inc. founded in 1985 as LANDesk Systems Inc.. In 1991, LANDesk was acquired by Intel Corporation. In 2002 LANDesk Software established as a standalone company.

Corporate Headquarters:
United States & Canada, 698 West 10000 South, Suite 500, South Jordan, Utah 84095
Phone: +1-800-982-2130

PRODUCT

LANDesk Management suite 7 provides proactive management of desktops, servers, and mobile devices in a heterogeneous OS environment. Supported OS's include Windows, Mac, NetWare, Linux, Unix, and PDA OS's. It requires the installation of agent software on all nodes that are to be managed.

Tool's functionality is offered through a single console and includes asset inventory, software distribution, and remote control. It was designed to address the following:

- security patches and virus definition updates
- remote software installation and maintenance
- software license use monitoring and audits
- automatic discovery of hardware and software assets
- multi-user OS migration and updates

VISUAL MAP

The tool does not offer visual map of the network.

ARCHITECTURE AND DISCOVERY

All managed nodes require installation of LANDesk agents.

The following terms are used to describe various components of the LANDesk suite:

- Management Domain: logical entity comprised of the LANDesk Management Suite components and all managed nodes.
- *Core Server*: the center of a management domain. All key files and services for Management Suite are on the core server. A management domain has only one core server.
- *Management Console*: the computer where you conduct management activity such as taking remote control of a management node. A domain can have as many as 30 management consoles.

- Databases: Management Suite requires one database for Software Metering information and another for management information.
- *DataMart*: A database that is optimized for querying. The web console requires a DataMart.
- *Service Center*: servers that host one or more management services. You can install Client Deployment and Certificate Authority service centers to reduce load on the core server. By default, the *core server* contains all services.
- Managed nodes: Workstations or servers in your network that have LANDesk agents installed. A core sever can manage as many as 10,000 nodes. Lager environments require multiple core servers.

The LANDesk Management Suite supports central and hierarchical architectures. From conversation with the company it was clarified that multiple *core servers* may be present within a system. This implies *distributed architecture*.

Under central architecture, the *core server* directly manages all nodes. All management services reside on the *core server*. One or more *management consoles* may be connected to the server.

Under hierarchical architecture, one or more *service centers* hosts some of the management services and thus taking some of the load off of the *core server*. These services are Metering Relay Service, Certificate Authority service, and Client Deployment service.

The automatic node discovery is limited to finding nodes that have the following LANDesk agents installed:

- Remote Control: lets you remotely access and control a client computer
- Common Base Agent (CBA): enables the file transfer service and Intel Ping Discovery Service (PDS).
- Desktop Management Interface (DMI): Enagles management suite to discover clients that have the DMI agent installed. Computer with DMI only cannot be managed.

There are three ways to configure clients or managed nodes:

- Manual configuration: map a drive to the *core server's* LDLogon share and run the client configuration program.
- Login script-based configuration: the configuration is applied to clients as they log in. For Windows NT/2000/XP clients, administrator rights are required.
- Push-based configuration: Desktop Manager is used to push the configuration to the clients. For Windows 95/98 clients, the CBA must be present.

LANDesk supports digital certificate authentication (DCA) to help ensure that only authorized management consoles can access clients.

Communication between system components (i.e. Core Server and Service Centers) makes use of several protocols such as HTTP and RPC and other proprietary TCP/IP

protocols. This implies that firewalls present on the communication path will need to have those ports open.

EXTENSIBILITY AND DEPENDENCIES

Supported databases include:

- Microsoft SQL Server 2000 SP3
- Microsoft MSDE 2000 SP3
- Oracle 8.1.7

No dependencies on 3rd party tools.

STRENGTHS AND LIMITATIONS

LANDesk provides a comprehensive auditing of computer hardware and software. Has a very strong software and OS distribution functionality.

LANDesk offers limited network discovery, i.e. only nodes with LANDesk agents are discovered and managed.

LanAuditor iInventory

COMPANY

iInventory Ltd., a UK-based company. First launched it's product in 1990. A world specialist in hardware and software inventory.

PRODUCT

iInventory is the latest version of LANauditor software. It keeps track of Windows, Mac, and Linux computer hardware and software. Auditing can be done either on a standalone or networked computers. Currently supported OS versions are:

- Windows 95, 98, NT, 2000, Me, Xp
- Apple Mac OS8.1x and up, OS9, OS X
- Linux Mandrake, Red Hat 7.x and SuSe 7.x

VISUAL MAP

The product does not offer visual network map.

ARCHITECTURE AND DISCOVERY

To perform an audit of all computers on a network, the iInventory console needs to be installed on a single Windows PC computer. iInventory is then used to create small and portable inventory agents (an executable) that can be deployed on any target computer without having to install any software. Audit output generated by the agent software is stored in a file on the local host. The audit data generated by agents is imported into the iInventory, which stores it in a MS Access 200 database. iInventory tools, or reporting functions of MS Access, can then be used to manage and share the audit data.

iInventory has no build-in mechanism for automatically distributing agent software to target desktops nor to automatically import audit data to the iInventory console. Therefore, distribution of agent software and importation of audit data requires manual operations to be performed by the IT support person (i.e. agent executable files can be emailed or distributed via a floppy disk and likewise data collected by agents can also be emailed back to be imported into iInventory). However, it is possible to automate distribution and collection process by developing custom scripts and such …

EXTENSIBILITY AND DEPENDENCIES

iInventory v6 provides two database format options. Microsoft Access 2000 (*.mdb) is the default, and does not require you to license or install Access. The alternative is MS SQL Server. This does require you to own a SQL Server license, but it does not have to

be installed on the same PC as iInventory. iInventory will need to connect to the SQL Server machine when required.

It is possible to connect most large database formats (e.g. Oracle, DB2) to iInventory's databases.

No dependencies on 3rd party software.

STRENGTHS AND LIMITATIONS

It provides comprehensive PC hardware and software inventory information.

The tool does not offer automatic network discovery. It does not offer network visual map feature. It requires the use of proprietary agents on each node to be inventoried.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Communications Research Centre
3701 Carling Ave., Ottawa, ON K2H 8S2

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Study of Tools for Network Discovery and Network Mapping (U)

4. AUTHORS (Last name, first name, middle initial)

Bantseev, Sergei and Labbé, Isabelle

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
November 2003	105	16

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contractor report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Network Information Operations
DRDC Ottawa
3701 Carling Ave., Ottawa, ON K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)
15bf28	A1410FE965

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)
DRDC Ottawa CR 2006-302	

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Full unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The work presented in this report is related to the DRDC Joint Network Management and Defence System (JNMDS) Technology Demonstrator Project. This work is an investigation of the currently available tools that are capable of performing automatic network discovery in an IP-based network. For the proposed system, it is required to provide an assessment of the existing tool's capabilities in identifying: the network topologies (map of physical links and logical links), the network resources (network elements and the configuration information) and the network services (network applications and system support). In particular, the study addresses the following issues: how is automatic network discovery achieved by the existing tools, and what is discovered. The report presents the detailed evaluation of seventeen commercial sector, open-source and research/academic sector tools. The results of the evaluation are summarized in two characteristic tables. An outcome of the study is that although some tools present good capabilities, they all have their strengths and weaknesses. Within the scope of interest, the "one tool does it all" solution does not exist. It is reasonable to expect that for the Technology Demonstrator Project system, the solution is likely to consist of an integrated suite of tools where functionality of each tool will be combined to achieve the desired capability.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

network discovery, network management, network mapping