Security in Network Management

Security in distributed and remote network management protocols

Jeremy Rauch <jrauch@cadre.org>
Network Management

- What is it?
- Why do we need it?
- What are our options with regard to selecting a network management scheme?
- What are the security flaws it can introduce
- What can be done to minimize the risk of these security flaws?
Network Management: What is it?

- **Hardware**
  - Switches, routers, firewalls, WAP’s, hosts, printers
  - Just about anything on the network

- **Software**

- **Protocols**

- **Allows for remote management of the network from convenient, centralized sites**
Network Management: Why is it needed?

- Lowers costs by eliminating the need for many administrators at multiple locations performing the same function
- Makes network administration and monitoring easier and more convenient
- Coherent presentation of data
Major NM Options

- SNMPv1
- SNMPv2c
- SNMPv3
- Vendor proprietary solutions
- Quite a few options that never panned out…
  - DCE
  - REAL SNMPv2
  - CMIP
SNMP Flaws…

◆ The Protocols
  ◆ SNMPv1
  ◆ SNMPv2
  ◆ SNMPv2c
  ◆ SNMPv3
  ◆ RMON/RMON2

◆ The Implementations
  ◆ Default communities
  ◆ Buffer overflows
  ◆ Design + Logic errors
  ◆ Miscellaneous
SNMPv1 History

- Why was it created?
  - RFC 1067, 1988
- RFC 1155, 1158, 1990: Original specification of the MIBII
SNMPv1 Overview

- Information to be stored laid out in the Management Information Base (MIB)
- Specification of fields to be collected, data types, formatting, access controls
- Written in ASN.1
  - Easy to read
  - Not so fun to write
  - Basically akin to a Db schema
- Data encoded using BER
SNMP sample output

[1:38pm manager] **snmpwalk agent public system**

```plaintext
system.sysDescr.0 = Sun SNMP Agent, SPARCstation-20
system.sysObjectID.0 = OID: enterprises.42.2.1.1
system.sysUpTime.0 = Timeticks: (619954285) 71 days, 18:05:42.85
system.sysContact.0 = manager@cadre.org
system.sysName.0 = agent
system.sysLocation.0 = Under my desk
system.sysServices.0 = 72
```

[1:39 manager ] **snmpwalk agent public .1.3.6.1.2.1.4.22.1.4**

```plaintext
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.1.10.1.98.1 = other(1)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.1.10.1.98.2 = dynamic(3)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.2.10.1.98.36 = dynamic(3)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.2.10.1.98.37 = other(1)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.3.10.1.97.1 = other(1)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.3.10.1.97.101 = other(1)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.3.10.1.97.254 = dynamic(3)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.4.10.1.98.41 = dynamic(3)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.4.10.1.98.45 = other(1)
ip.ipNetToMediaTable.ipNetToMediaEntry.ipNetToMediaType.7.10.1.96.1 = other(1)
```
SNMPv1 Protocol

- Five Simple Messages:
  - get-request
  - get-next-request
  - get-response
  - set-request
  - trap
SNMPv1 Protocol continued...

Manager

get_request

get_next_request

set_request

port 162

Agent

get_request

get_response

get_response

get_response

trap

port 161

port 161

port 161

port 161
SNMPv1 Protocol continued...

- UDP Transport Mechanism
- Community: Shared “password” between agent and manager
- PDU: Specifies request type
- Request ID
- Error Status
- Error Index
## SNMPv1 Packet Format

<table>
<thead>
<tr>
<th>UDP Header</th>
<th>Version</th>
<th>Community</th>
<th>PDU Type</th>
<th>Request ID</th>
<th>Error Status</th>
<th>Error Index</th>
<th>name</th>
<th>value</th>
<th>name</th>
<th>...</th>
</tr>
</thead>
</table>

The table above represents the fields within an SNMPv1 packet format, including UDP header information, version details, community settings, PDU type, request ID, error status, error index, and additional values and names.
SNMPv1 Security Flaws

- Transport Mechanism
  - Data manipulation
  - Denial of Service
  - Replay

- Authentication
  - Host Based
  - Community Based

- Information Disclosure
SNMPv1 Transport Mechanism Flaws

- UDP Based
- Unreliable - packets may or may not be received
- Easily forged - trivial to forge source of packets
SNMPv1 Authentication Flaws

- Host Based
  - Fails due to UDP transport
  - DNS cache poisoning

- Community Based
  - Cleartext community
  - Community name prediction/brute forcing
  - Default communities
SNMPv1 Information Disclosure

- Routing tables
- Network topology
- Network traffic patterns
- Filter rules
- Vendor proprietary information + invocation
  - Execute arbitrary programs, etc
SNMPv1 Security Flaw Implications

- Altering/Manipulation of network by unauthorized individuals
- Denial of Service on whole networks
- Modification of ACL’s & configurations
- Clear topology of network behind router
- Makes creation of more sophisticated host based attacks easier
SNMPv2 History

- Written to address security and feature deficiencies in SNMPv1
SNMPv2 Protocol

- Extension to SNMPv1
- Provided security model
- 2 new commands
  - get-bulk-request
  - inform-request
    - Acknowledged trap
- A big, big failure
SNMPv2 Protocol continued...

<table>
<thead>
<tr>
<th>privDst</th>
<th>authInfo</th>
<th>dstParty</th>
<th>srcParty</th>
<th>context</th>
<th>PDU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nonsecure Message</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>privDst</th>
<th>digest</th>
<th>dstTime</th>
<th>srcTime</th>
<th>dstParty</th>
<th>srcParty</th>
<th>context</th>
<th>PDU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authenticated, not encrypted</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>privDst</th>
<th>0-length OCTET STRING</th>
<th>dstParty</th>
<th>srcParty</th>
<th>context</th>
<th>PDU</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>privDst</th>
<th>digest</th>
<th>dstTime</th>
<th>srcTime</th>
<th>dstParty</th>
<th>srcParty</th>
<th>context</th>
<th>PDU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Private, not authenticated</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>privDst</th>
<th>digest</th>
<th>dstTime</th>
<th>srcTime</th>
<th>dstParty</th>
<th>srcParty</th>
<th>context</th>
<th>PDU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Private and authenticated</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SNMPv2 Security Flaws

◆ Replay

◆ 4 types of time error conditions
  ◆ manager’s version of agent’s clock greater than agent’s actual clock
    ◆ Collect packets for future replay to agent
  ◆ manager’s clock greater than agent’s version of manager’s clock
  ◆ agent’s clock greater than manager’s version of agent’s clock
  ◆ agent’s version of manager’s clock greater than the manager’s version of the manager’s clock

◆ No unique nonce to prevent replay within window
SNMPv2 Security flaws...

- Replay attacks possible via complex clock attacks
  - Clock sync is NOT part of SNMPv2
    - Dependence on external protocols opens vulnerabilities (NTP)
  - Behavior for clock skew forward + back is ill defined
SNMPv2 Security Flaws Attacks against DES

- Duplication of privDst in dstPty allows for known plaintext attacks
- 16 character, user defined DES pass phrase
- Allows easy dictionary attacks
SNMPv2 Security Flaws MD5 attacks

- Again, user defined
- 16 character secret
- Dictionary attackable
SNMPv2 Security

- Still uses UDP transport
- SNMPv1 Compatibility can compromise security
- Default DES and MD5 phrases
- Does not prevent D.O.S or traffic analysis
SNMPv2 Downfall

- Marginal security
- Complex implementation
- Devices were a whole lot slower and lacking in ram
SNMPv2C

- What is it?
- Why does it exist
SNMPv2C Protocol

- SNMPv2 additional PDU types
- SNMPv1 Community based authentication
- UDP transport
- All the features of SNMPv2 with the security of SNMPv1
SNMPv3 History

- RFC 3412, 2002: “Message Processing and Dispatching”
- RFC 3413, 2002: “SNMP Applications”
- RFC 3415, 2002: “View-based Access Control Model”
- RFC 3417, 2002: “Transport Mappings”
- RFC 2576, 2578, 2579, 2580…
- Written to address the failures of the original SNMPv2 security model
Protocol

- Designed to be implementable and secure
  - Based on the original SNMPv2 work (SNMPv2u and SNMPv2*)
- Uses SNMPv2 PDU format + types
  - No new PDU types specified
- UDP transport
- Strong (enough) encryption and authentication
- New User-based Security Model
- New View-based Access Control (enhanced MIB view concept)
- Starting to catch on (kinda sorta)
Packet Format

- msgVersion
- msgID
- msgMaxSize
- msgFlags
- msgSecurityModel
- msgSecurityParameters
- contextEngineID
- contextName
- PDU

Packet Format: User-Based Security Model

- **msgVersion**
- **msgID**
- **msgMaxSize**
- **msgFlags**
- **msgSecurityModel**
- **msgSecurityParameters**
- **contextEngineID**
- **contextName**

- **msgAuthoritativeEngineID**
- **msgAuthoritativeEngineBoots**
- **msgAuthoritativeEngineTime**
- **msgUserName**
- **msgAuthParams**
- **msgPrivacyParams**

SNMPv3 User-based Control Model

- Encryption
  - DES
    - CBC mode
- Authentication
  - HMAC
    - SHA-1
    - MD5
- Timeliness mechanism
SNMPv3 Flaws

- Encryption
  - CBC mode depends on 64 bit IV
  - IV is created by taking last 8 octets of 16 octet privKey (pre-IV)
  - 8 octet salt value is xored with the pre-IV to create the IV
  - Only the salt value is transmitted, in msgPrivacyParameters field
  - Problem: Salt generation is left as an exercise to the implementor
  - Brute force of bad passwords
    - Slowed by password to key mechanism
SNMPv3 Flaws

- **Authentication**
  - Handled via HMAC-{SHA-1, MD5}
  - Output truncated to 12 octets
  - **MD5**
    - 16 octet auth key
  - **SHA-1**
    - 20 octet auth key
  - Stored in msgAuthParameters
  - Actually, HMAC is an excellent authentication mechanism
  - Short auth password can be brute forced
    - Password to key mechanism slows down attack
    - Harder due to collisions due to truncated output
SNMPv3 Flaws

✦ Timeliness mechanism
  ✦ Uses boot count + time since last reboot of agent
  ✦ Transmitted via a 2-step synch mechanism + stored
    ✦ snmpEngineBoots
    ✦ snmpEngineTime
    ✦ latestReceivedEngineTime
      ✦ Can prevent replay attacks within window
  ✦ 150 second skew allowed
    ✦ Skew depends on authoritative v. non-authoritative recipient
SNMPv3 Realized…

- Pretty cool protocol
- Still susceptible to denial of service
  - But what isn’t?
- Forgery possible, but difficult to use
- Brute forcing possible, but tough + slow
- Time based attacks may be possible
  - Immediate replay of packets MAY allow action invocation attacks
- Traffic analysis
RMON and RMON2 Security

- SNMP’s flaws
- additional hazards by introducing “action invocation” objects
- collects extensive info on subnet
- packet captures
Implementation Vulns

- Defaults
- MIB designs
- Buffer Overflows + parsing
- Design + logic errors
- Miscellaneous
Default Communities

- public
- private
- write
- "all private" (sun)
- monitor (3com)
- manager (3com)
- security (3com)
- OrigEquipMfr (brocade)
- "Secret C0de" (brocade)
- secret
- cable-docsis
- xyzzy, agent_steal, freekevin, and fubar (!?)

- admin
- default
- password
- tivoli
- openview
- community
- snmp
- snmpd
- system (aix, others)
- And so on...
Hidden Communities

- An obscene percentage of managed devices contain hidden communities
- Often fully read/write privileged

```bash
for I in `dz < xxx.bin | strings`
do
echo $I; snmpget -c $I host system.sysDescr.0
Done
```
Too much info!

D-Link password disclosure
- enterprises.937.2.1.2.2.0
- Similar problems affect all “toy” routers

Cisco VACM community disclosure
- snmpVcmMIB.vcmMIBObjects.vcmAccessTable

A quick perusal of interesting keywords at www.mibdepot.com reveals hundreds of potential vulns
OULU PROTOS evaluation
- Identified hundreds of test cases for evaluating SNMP protocol implementations
  - Invalid BER length fields
  - Long strings
  - Format strings
- Found dozens of implementation flaws
  - Most implementations derived from CMU/UCD/Net-SNMP

Real world examples abound
- IRIX snmpd overflow
All sorts of “conveniences”

- Cisco CONFIG-COPY.mib & CISCO-FLASH.mib

Management stations not without own problems

- Tivoli Netview - execute arbitrary commands with a well formed trap under custom configs

net-snmp has had client tool + agent flaws

- Most recent one patched about 3 weeks ago…
Securing existing implementations

- Risk assessment
- Minimization of use
- Allow get-*’s only, no remote setting
- Eliminate defaults
- Filtering EVERYWHERE
  - Marginally useful at best
- Management network
Sources you need to check out…

- Multiple SNMP RFC’s (mentioned throughout talk)
- *TCP/IP Illustrated Volume 1*, Richard Stevens (ISBN
- www.mibdepot.com
- Simple Times (www.simple-times.org)
- OULU PROTOS (http://www.ee.oulu.fi/research/ouspg/protos/index.html)
- www.securityfocus.com
  - Vulnerability DB
  - Bugtraq
- Net-snmp (www.net-snmp.org)
Questions?