Operations and Management Area Work Group
Internet-Draft
Intended status: Informational
Expires: March 27, 2017

S. Rich
Cisco Systems

T. Dahm
Google

27 September 2017

MUD Lifecyle: A Network Operator's Perspective
draft-srich-opsawg-mud-net-lifecycle-01.txt

Abstract

This memo describes the lifecycle of MUD as seen from the perspective of a network operator. It is informational and intended to help provide perspective around the operation of a network which connects MUD-supporting devices and uses MUD-supporting network infrastructure. All phases of network operation that involves or affects MUD will be described. Considerations specific to device manufacturers will be described elsewhere. Considerations relevant to network equipment manufacturers and networking software authors will be described where appropriate where MUD behavior is affected.

1. MUD Introduction

Network architects and operators have the goal of designing and operating networks so that they are reliable, secure, and operate correctly. Making them do so requires that the network permit traffic which is intended to be allowed on the network while rejecting or blocking traffic which is not. Both goals are met with a combination of policies and configurations which promote efficient routing of packets for certain classes of traffic and which rate limit or even block (possibly by black-holing) other classes of unwanted or lower-priority traffic.

A common assumption is that devices on the inside of the network can have relatively unrestricted access to other parts of the network and to the local network segment. This is reasonable for devices which themselves have certain configurations which will naturally govern which network access they require. For example, a printer will usually be configured to accept connections from hosts which wish to print to it. The printer itself may not tend to initiate outbound connections and thus does not require a complex set of custom ACLs. If the printer needs external connectivity, the usual scenario is to allow the printer to make outbound connections while still preventing inbound connections using a stateful firewall rule or similar. However, there are often no rules preventing the printer from making arbitrary connections within network delineated by the firewall.

Other devices such as general-purpose end-user hosts (PCs, Mac, etc.) might need unrestricted access, at least in the outbound direction, because, contrary to the printer example, end-user hosts are generally expected to make outbound connections to an unpredictable number of hosts. Even if outbound restrictions to

certain ports (such as 80, 443, 22, 25, etc.) are enforced, the destination address may be unrestricted. As stated above, restrictions from internal hosts to internal addresses may be even more lax.

Enter into this situation IoT devices which may be introduced to the network in the thousands and which may have unspecified or unclear requirements for network access. For example, IoT light bulbs may need to talk to DNS, NTP, LLDP, DHCP, and a controller on the local network and nothing else. An IoT thermostat may need to talk to DNS, NTP, LLDP, DHCP, and its cloud-based controller, but nothing else. For both of these cases, while their specific requirements vary, knowing each one's requirements would allow a tight set of ACLs to be imposed, all the way to the port level, to limit what connectivity is afforded to each individual instance.

Recent examples of IoT-based malware campaigns will not be repeated here and the benefits of providing such security will no doubt be obvious to network operators. What has not been available before MUD is an ability to automatically retrieve configuration policy and then automatically apply it for each device. This document will describe the ''lifecycle'' of MUD from the perspective of a network operator. The details of the protocol and contents of the MUD file itself are described in [LEAR2017], and familiarity with it is assumed for this document.

2. Terminology

This document will use some terms and abbreviations which will be listed and described in this section.

MPD
: "MUD-Protected Device" - While this is a possibly tedious use of a three-letter acronym, repeated use of "MUD-protected device" or similar is equally tedious

AAA Server
: "Authentication, Authorization, and Accounting Server" - A network service which processes AAA requests

ACL
: "Access Control List" - In the context of this document, an ACL will refer specifically to those which are specified in a MUD file and which get applied at some point in the network to enforce the security policy needed by a device. These ACLs may be configured down the port into which the device the is plugged, and they may be applied "dynamically" in the sense that they appear in response

to the MUD request as opposed to a static configuration. They will not be dynamic in the sense that they change frequently. The actual implementation by any particular vendor is left up to that vendor and thus may differ from the examples given in this document.

3. MUD Lifecycle Description for Network Operators

The totality of what network operators must do to build, operate, and maintain networks will not be described in exhaustive detail in this document. Instead, we will describe what additional or different things are necessary or recommended when establishing MUD support within the network. Some of the steps discussed will presuppose that networking equipment vendors will have added MUD support to their products.

The following high-level tasks are required to support the automatic network configuration aspects of MUD devices on the network:

1. Network Segmentation Considerations and Design

2. Install and/or enable a MUD Policy Server

3. Configure network devices so that they will receive and enforce ACLs generated by the MUD Policy Server

4. Test and verify functionality by confirming that MUD files are retrieved and ACLs are applied to the appropriate ports and that those ACLs are removed when the port goes down

The MUD Policy Server may support caching retrieved MUD files. If it does, then the operator may choose to enable, tune, test, and monitor this functionality as well. Details about caching MUD files as well as each task above will be covered later in this document.

The network equipment to which MPDs connect must be capable of accepting and enabling dynamic ACLs which can preferrably be scoped to a port. While it is conceivable that the ACLs be combined and applied at a point in network that is multiple hops away from the switch to which the MPD connects, the tightest security controls are possible when enforcement can happen directly on the port. This eliminates the possibility that a MPD can talk to other devices on the same switch unless explicitly permitted. The remainder of this document will only discuss the case of using ACLs.

3.1. Network Segmentation Considerations and Design

A well-designed network is one which includes the use of segmentation which keeps different parts of the network isolated from each other to the optimimum degree. For example, groups of machines which need to communicate frequently and at high speed most likely should be on the same LAN. Different groups of machines which rarely communicate together can be separated into different routed networks, and depending upon security requirements, may even be guarded by ACLs or other mechanisms.

Different network segments may be designed with different expectations of security. Inner-bastion networks may contain sensitive systems which are isolated from all but the most trusted systems. Segments which allow guest users or devices which are less trusted may be relegated to segments which have also been protected with ACLs, but the focus can be on limiting what the devices in the segment can access rather than worrying about what external devices can access inside the segment itself.

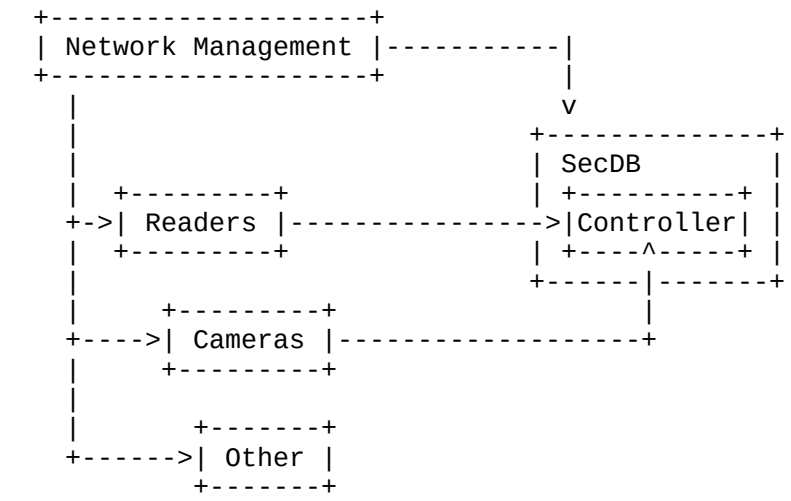
The goal of MUD is to enable the near-automatic management of device segmentation for the class of devices which have MUD support. To be maximally effective, though, the network designer should take advantage of pre-defining segments into which MUD-capable devices can be grouped by function and by required access. An optimal middle ground (for a large network with many types of MUD-enabled devices) would comprise some device-class-specific segments, some vendor-specific segments, the essential set of network segments (required regardless of MUD for the normal operation), and perhaps a ‘‘default network’’ into which untrusted devices are placed which get no internal network access and severely limited internet access.

Ideally, with full MUD support in devices deployed in a network, there would be no need for the so-called ‘‘default network’’ segment (except perhaps as a ‘‘guest’’ network) since MUD profiles would result in a properly-segmented and protected devices. Until MUD is ubiquitously supported, though, it is wise to consider the option.

To make these ideas more clear, an example network will be described (at a high level) with various segments defined. The use of each segment by MUD will then be described. These are segments within a larger network which will not be described to avoid cluttering the diagram.

Segment Name	Segment Description	MUD
SecDB	Recorders, IDs	N
Readers	Badge Scanners	Y
Cameras	Security Cameras	Y
Other	Other IoT devices	Y
NetMgmt	Network Management	N

There are five segments. Two of them will have no MUD-enabled devices in them, whereas the other three will. Of those three, one is a non-classed MUD network (i.e., one in which MUD-enabled devices which do not belong to specifically-configured classes will be placed). The connectivity of the network looks like:



The SecDB segment will contain senstive systems as well as a controller to which some of the other devices will need to communicate. The Readers will be a segment in which all badge/card readers will be grouped. The Cameras segment will contain all of the security cameras. Finally, all other MUD-enabled devices will be placed in the Other segment. Devices placed into any of these segments as a result of MUD will still have applicable ACLs applied. In addition, any static access control restrictions given to each segment will be enforced per the network designers' intentions.

How do the cameras get into the Cameras segment, and how do the card readers get into the Readers segment? The specifics will depend on the MUD Controller implemenation and the network devices used, but the gist is that the network administrator defines policies which map a MUD file's ''manufacturer'' and ''model'' to the appropriate network segment assignment policy. If no specific mapping is available for a device, then the MUD-enabled device will be placed into a default segment per the operation of the MUD Controller in use.

Another consideration is what to do with devices which have no MUD profile at all. This was the case for all device before MUD was defined and may continue to be the case for certain classes of devices. The solution again lies with the definition of network policies. It is up to the network designer to choose which segment or segments devices which have no MUD support are placed by default. Theoretically, the placement could be influenced by the MAC address, the port into which the device is plugged, etc.

The bottom line is that MUD is not responsible for fully describing the network configuration policy. It is very helpful to automatically limit the access that MUD-enabled devices are afforded to only what they need, but the network operator must insure that the network design is complete.

3.2. Installing and/or Enabling a MUD Controller

MUD Policy Servers can conceivably take on many forms, including stand-alone appliances, software modules installed on a switch or a router, a software package installed and integrated with a DHCP server, etc. The key requirements for MUD Policy Servers are:

1. Able to "see" a MUD URI

2. Able to retrieve a MUD file

For a MUD Policy Server to ''see a MUD URI'', it must either be able to see the DHCP or equivalent requests from MPDs directly or it must be otherwise connected to the service which does get to see these types of requests. For example the MUD Policy Server could be implemented as a plugin to a RADIUS server which is receiving requests from a switch which is handling DHCP requests by generating corresponding RADIUS AAA requests.

For a MUD Policy Server to be able to retrieve a MUD file, it must have network access permissive enough to retrieve files which are served from arbitrary locations on the internet.

Finally, to have any useful effect, the MUD Policy Server must be able to, having parsed a MUD file, generate ACLs which are to be applied to the appropriate port of the appropriate network device (i.e., a dynamic configuration must be generated and applied which reflects the MUD policy). The specifics of how the generated ACLs get back to the NAS and get applied to the proper port will depend on the design of the network.

At the time of this document's preparation, MUD is still a new protocol and is under development. Therefore, descriptions of how it is integrated will be subject to adjustment according to the progression of actual implementations.

3.3. Network Device Configuration

There are two distinct "network configuration" concepts involved in the deployment of MUD:

1. Configuration of the network infrastructure such that the MUD controller is "in the loop" and able to issue configurations for devices as they appear on the network

2. The per-device dynamic configuration that is generated through the behavior of MUD itself

This document discusses both concepts where applicable. To avoid confusion, when a reference is made to "configuring a device" or similar, we will be referring to setting up the network infrastructure to include the MUD Policy Server into operations. The actions of the MUD infrastructure and network infrastructure to effect changes to network configurations persuant to MUD-advised policies will be referred to as "applying device policy" or (when it is more clear to do so) "applying the dynamic device configuration". The key word in the latter is dynamic and may be used when describing the specific steps being taken by the devices to apply the policies.

As previously mentioned, the ideal point for the application of MUD-based access restrictions is the port into which a device is directly plugged since this results in the most finely-grained application of access control and insures that devices are not able to talk even to neighbors on the same shared media without MUD authorization. For this to happen, the switches which connect to MUD-enabled devices must be configured to allow ACLs to be applied to each port. If the switch is stand-alone, then it will have to be configured to allow something like RADIUS or similar so that a controller device can send ACLs to the switch via an

authorization transaction once the MUD profile has been processed.

For MUD to work properly, the switches MUST remove any dynamic configuration applied to a port when the connection on that port is dropped (such as when the cable to the port is disconnected). Once reconnected, a device will again issue a DHCP or similar request and the MUD behavior will begin again.

As an example, if a Layer-2 switch is used which can process DHCP requests by issuing RADIUS AAA requests to complete the port-level authorization, MUD process can occur by:

1. The switch adds the MUD URI to the RADIUS request (see [WEIS2017])

2. The RADIUS server passes the MUD URI to a MUD Controller

3. The returned MUD file is processed and the appropriate ACLs generated

4. The ACLs are encoded into the RADIUS Authorization response and returned to the switch

5. The switch receives the RADIUS Authorization, matches it to the port being provisioned, and applies the ACLs

3.4. Testing and Verification

In addition to the normal activities of validating through monitoring commands that ACLs have been applied as expected, the following items are suggested:

- If one wants to understand what ACLs will be applied during a test of a particular device, one can read the MUD file to understand what access requirements it has and thus compare that with what ACLs get applied during the operation of the MUD protocol

- The devices with MPDs attached to them should be checked to confirm the application of the expected ACLs and they are scoped to the appropriate ports

- An ideal test would be to connect a MUD-enabled test client which will issue an appropriate network access negotiation via DHCP or whatever is appropriate for the NAS in use so that a full MUD File retrieval is triggered. The test client should then be used to try to both confirm connectivity to its explicity provisioned destination(s) while also verifying that it is not possible to reach sites outside the stipulated ACLs.

- The MPD should be disconnected from the switch and the switch checked to verify that the ACLs are removed (which may not occur until another device is plugged into the same port)

3.5. Caching MUD Files

MUD Files may be cached by the MUD Controller. The MUD File itself indicates the minimum time between re-retrievals of a MUD File via the ‘‘cache-validity’’ attribute. When the MUD Controller is asked for a MUD File, if the URIs match a cached MUD File which is recent enough to be used, then that cached MUD File should be used. If not, then a valid MUD File MUST be retrieved by using the URI as a URL.

Note, however, that MUD files are very small. Additionally, MPDs will likely be installed into networks and then left running for long periods of time such that the number of MUD file requests will likely be small. Given those considerations, the value in caching MUD files, at least in the near term, is expected to be low.

4. Security Considerations

The bulk of this document describes the use of MUD to increase the security of a network. However, it is possible to compromise the effectiveness of MUD by attacking its behavior directly. This section discusses the known attacks and describes possible mitigations (all from the network operator’s perspective). This section also attempts to clarify the limits to which MUD is expected to perform in terms of increasing security.

The use of MUD is intended to increase the level of security in the network relative to its current state. If the network has no security protections in place, then MUD may improve the situation by limiting access to MUD-enabled devices, but the network may already be too permissively accessible to be secure. A common comment about MUD is that a compromised MUD File can allow a MUD-enabled device to access arbitrary parts of the network or to allow arbitrary access to the device. If the network had had no security to begin with, then the compromised MUD File will not have reduce the security in any meaningful way.

To put this another way, any network SHOULD be properly designed such that the minimum required access is granted to all parties involved. If this is done, then a bad MUD File can only result in too permissive access to and from a single device in the network.

Although MUD is still a new protocol, it is conceivable that an "ecosystem" around it will grow that will enable a level of security validation that is much more difficult without it. In particular, the published MUD Files could be analyzed by third parties to assess their contents and to make users aware of anomalies. Additionally, deviations in successive versions of MUD Files can be audited to detect surprising changes.

Another commonly-mentioned attack scenario is tampering with the MUD URI during device bring-up to cause a different MUD File to be fetched and applied in place of the correct, manufacturer-supplied file. The ramifications of such an attack are no different than that of a compromised MUD File. The mitigation against the attack is insure the use of secure means of receiving and processing the device's advertisement of the MUD URI.

One other intriguing attack scenario is the spurious introduction of something akin to a "phantom" DHCP request with a MUD URI intended to coax the network infrastructure into fetching and acting on a MUD File, possibly without an actual device being present (or the "device" actually being a rogue software element running on a real device). In addition to mitigations already mentioned, port-level security should be used whenever possible with strict security policies to enable the detection of these rogue DHCP or other advertisements.

5. IANA Considerations

This document has no actions for IANA.

6. Normative References

[LEAR2017]
Lear, E., "Manufacturer Usage Description Specification", draft-ietf-opsawg-mud-03, January 05, 2017

[WEIS2017]
Weis, B., "RADIUS Extensions for Manufacturer Usage Description", draft-weis-radext-mud-00, October 25, 2016

7. Informative References

[RFC2882]
Mitton, D., "Network Access Servers Requirements: Extended RADIUS

Practices", RFC2882, July 2000

Authors' Addresses

Steven Rich
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

Email: srich@cisco.com

Thorsten Dahm
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043

Email: thorstendlux@google.com