CLONED IDENTITY THREATS IN PACKETCABLE™

Alexander Medvinsky, Jay Strater
Motorola Broadband

Abstract

MTA clones might lead to service theft, breach of privacy, and denial of service. This paper proposes techniques that may be utilized by an IP Telephony service provider to detect and disable cloned MTAs and investigates what MTA configurations make sense for tamperproof hardware. It also considers techniques involving CMTS and DHCP server configuration and filtering options for limiting the geographic distribution of MTA clones.

INTRODUCTION

PacketCable [2] provides a set of specifications for VoIP services layered on top of DOCSIS-based HFC networks [7]. Denial of service threats that could disrupt an IP Telephony network, phone service theft and user privacy issues were all considered in the PacketCable security design. The PacketCable security specification [1] provides cryptographic protection that addresses these threats at a protocol level. But is protocol-level security enough to address these threats?

This paper considers a particular class of threats due to illegal duplication of the PacketCable client (MTA) identities. Since PacketCable provides cryptographic security, in order to duplicate an MTA identity one would need to make a copy of the MTA private keys and certificates in addition to copying the MTA host name, IP address, and MAC address. Let us say that an owner of a legitimately purchased (or leased) MTA proceeds to duplicate its identity into a number of illegal clones. What kind of threat does it pose to IP Telephony service providers? The paper discusses scenarios where the use of MTA clones might lead to service theft.

The PacketCable security team also took these cloning scenarios into consideration and the PacketCable security specification includes a discussion of these threats. Two main techniques that could be used to prevent clones are Fraud Detection/Prevention services and tamperproof hardware inside the MTA that would make duplication of cryptographic keys difficult. Because these techniques do not require inter-operability and because both of them affect either the cost of running an IP Telephony network or the cost of the MTAs, PacketCable does not provide specific requirements in this area.

This paper proposes techniques that may be utilized by an IP Telephony service provider to detect and disable cloned MTAs and investigates what MTA configurations make sense for tamperproof hardware. The proposed fraud detection and disabling techniques are based on particular properties of the Kerberos/PKINIT protocol that is utilized by PacketCable to distribute cryptographic keys to the MTAs. Fraud management puts an additional burden on the operator and if improperly administered could result in an uncomfortably large number of false alarms. Therefore, it is desirable to complement fraud management with tamper-resistant key storage in the MTAs.

Cost effectiveness of tamper-resistant storage in the MTA seems to largely depend on what other services are provided by that MTA. If it is a stand-alone MTA device that provides nothing but interactive VoIP

services, secure storage can only be used to protect PacketCable cryptographic keys. If it is an MTA that is integrated with a settop box, it is resident on a platform that already has a very high motivation for secure key storage in order to prevent theft of broadcast video. In such an environment, an MTA benefits from secure key storage that is already present on that platform at little or no added hardware cost. Other integrated MTA platforms are also considered in this paper along with the corresponding motivation to utilize secure key storage.

This paper also addresses an MTA cloning threat that is better addressed with cloning prevention at the DOCSIS level rather than at the PacketCable application level. The paper discusses a denial of service scenario where a malicious adversary using a false identity is able to fool a CMTS into dropping valid downstream packets destined for some MTA. This threat is based on the fact that a CMTS will allocate a gate for each phone call that is authorized for a specific quality of service and has a specific bandwidth limit. If an MTA were to receive packets at too high of a rate, the CMTS would be forced to drop some of them. In order to orchestrate such an attack, this adversary need not know any of the cryptographic keys of another MTA. This paper proposes a solution to the problem that involves the CMTS with a cost of some administrative burden.

Finally, this paper considers techniques involving CMTS and DHCP server configuration and filtering options to severely limit the geographic distribution of MTA clones and, therefore, the viability of MTA cloning operations.

PACKETCABLE™ ARCHITECTURE OVERVIEW

PacketCable is a project conducted by CableLabs. The project goal is to identify and define standards used to implement packet based voice, video, and other real time multimedia services over cable systems. PacketCable products are a family of products designed to deliver enhanced communication services using packetized data transmission technology over the HFC data network using the DOCSIS protocol. PacketCable products overlay the 2-way data ready, broadband cable access network. Initial offerings are packet voice. Packet video and other multimedia are longer term goals that are just now starting to be addressed by the PacketCable MultiMedia project.

The following diagram shows the PacketCable reference architecture (also see [2]).

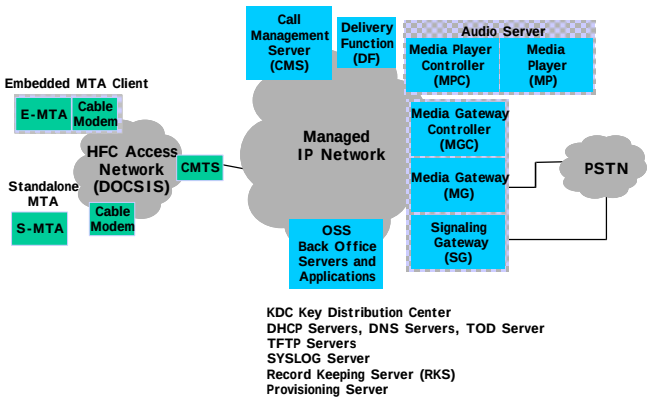

Figure 1. PacketCable Reference Architecture

For the purpose of subsequent MTA security discussion, key elements in this architecture are the MTA, CMTS, CMS, MTA Provisioning Server, and KDC. The PacketCable signaling [3], bearer [5], and management protocols [4] between these

components are shown in the following figure.

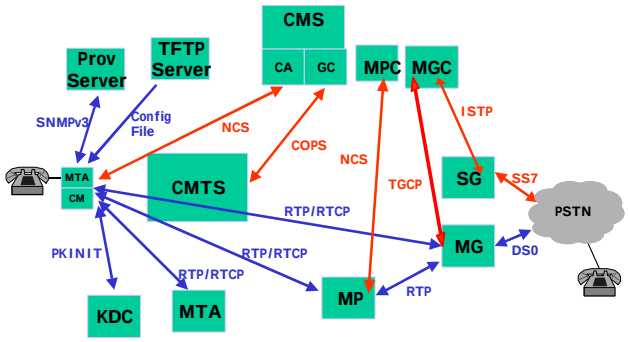

Figure 2. PacketCable Signaling, Bearer, and Management Interfaces

Following are a listing of the security protocol and key management techniques on the MTA interfaces (see [1] for details):

- Public key enabled Kerberos protocol between MTA and KDC
- Kerberized SNMPv3 on SNMP protocol between MTA and Provisioning Sever
- Hash and encryption of MTA configuration file retrieved from TFTP server
- Kerberized IPsec on NCS protocol between MTA and CA
- Cipher + MAC (Message Authentication Code) with NCS key distribution on RTP and RTCP protocol between MTA and MG, MP, and other MTA

IP TELEPHONY THREATS OVERVIEW

The IP Telephony system threats fall into three general categories:

1. **Service Theft**. An adversary manipulates the IP Telephony system in order to gain some financial benefit. For example, an adversary impersonates a valid VoIP subscriber and is able to make free long distance phone calls and charge them to the victim's account.

2. **Breach of Privacy**. An adversary is able to snoop on IP Telephony traffic (either signaling, management, or bearer channel) without a proper authorization.

3. **Denial of Service**. An adversary disrupts IP Telephony service, making the network completely non-functional, decreasing Quality of Service (QoS) below an acceptable level, or corrupting MTA configuration content.

PacketCable™ security addresses all known theft of service threats on IP Telephony system interfaces that are within the scope of the PacketCable™ project. Similarly, PacketCable™ security addresses breach of privacy threats on PacketCable™ interfaces that require privacy. Major denial of service threats resulting from unauthorized protocol manipulation are also addressed.

However, protocol and interface manipulation are not the only means by which an adversary may attack an IP Telephony system. Hacking into an IP Telephony server and disabling it would be an attack that should be prevented using various techniques such as firewalls, local access control, etc. However, since these local security measures do not require interoperability, they would fall out of the scope of protocol specifications such as PacketCable™. Similarly, identity cloning threats, where secret cryptographic keys are illegally extracted from a VoIP client and then distributed to other VoIP clients, should be addressed but are normally not covered by protocol specifications. The rest of the paper specifically addresses the cloning threats and how they may be prevented.

MTA CLONING THREATS

An MTA clone is a copy of an original, legally configured MTA that possesses the original MTA's identity (e.g., MAC address) as well as the original MTA's secret cryptographic keys. This enables the MTA clone to falsify its identity as if it were the original MTA.

In order to better understand the MTA cloning threats, it is helpful to identify the use of the various MTA identities within PacketCable™:

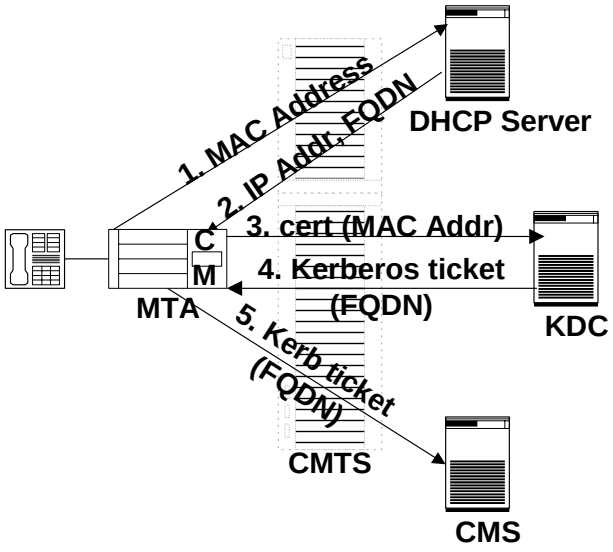

Figure 3: Usage of MTA Identities

Figure 3 shows that an MTA is authenticated by the CMS using a Kerberos ticket issued to this MTA, where the ticket contains the MTA Fully Qualified Domain Name (FQDN). The MTA FQDN along with the MTA port number identifies a specific VoIP subscriber.

The ticket certifying MTA FQDN is obtained from the Kerberos Key Distribution Center (KDC). The MTA provides its digital certificate with the MTA MAC address to the KDC and the KDC verifies the mapping of the MTA MAC address to its FQDN by performing a lookup into a subscriber database. (PacketCable™ defines a secure interface from the KDC to a subscriber database in order to perform this lookup.) Once the KDC verifies the mapping between the MTA MAC address and the FQDN, it returns a ticket to the MTA.

In addition to the MAC address and the FQDN, the other MTA identity used within PacketCable is its IP address. The DHCP server assigns an MTA an IP address based on its MAC address. Also, figure 3 shows that there is a CMTS located in the middle of all of the MTA's interfaces to an IP network. The CMTS has the visibility of the MTA's MAC address and its IP address inside the frame and packet headers respectively. The MTA is shown to be an embedded MTA, where the Cable Modem (CM) is integrated with the MTA as a single device. For the purpose of this paper, this is only an example – the same cloning threats and prevention measures apply for a standalone MTA that is not integrated with the CM.

An MTA clone in this architecture would:

- Somehow obtain a copy of the original MTA's device certificate and RSA private key.
- Use this certificate and private key to sign a ticket request for the KDC and the KDC would map the MAC address in the certificate to the original MTA's FQDN.
- Use the ticket with the original MTA's FQDN to establish security associations with the CMS.

As a result, the cloned MTA would make phone calls using the original MTA's subscriber account.

MTA clones in an IP Telephony system present the following threats:

- A subscriber authorized only for subscription services such as local calls and unlimited long distance minutes within a limited area can freely share the

account with the clones. This subscriber could be a pirate that makes money from selling clones.

- A pirate might sign up using a false subscriber account with a stolen credit card number and then allow clones to make long distance calls. The pirate in this case has no intent to pay the phone bill. Eventually, the pirate account would be closed and the pirate might try to open another one with another false identity.
- In the case of a soft MTA implemented on a PC platform, the private key might be stolen by hackers on the Internet and then used to create clones. These clones could be used to charge phone calls to the victim's account or to disrupt the telephony network operation.

MTA CLONE DETECTION AND DISABLEMENT

Kerberos-Based Clone Detection and Disablement

The characteristics of the Kerberos key management protocol can be utilized to detect and disable MTA clones. Clone detection can be performed either by the KDC or by the CMS as explained in subsequent sections.

Clone Detection and Disablement by the KDC

A PacketCable™ KDC delivers a Kerberos session key to an MTA clone using a method called Diffie-Hellman key agreement. The Diffie-Hellman key agreement is part of a PKINIT extension to Kerberos that allows KDC clients to authenticate to the KDC using public key cryptography. The Diffie-Hellman key agreement is illustrated in the following figure:

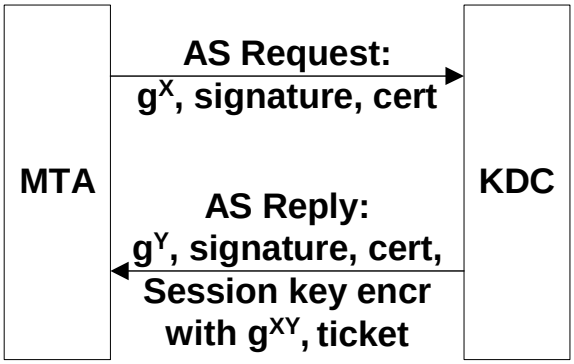

Figure 4: PKINIT with Diffie-Hellman Exchange

During a Diffie-Hellman key agreement, the MTA generates a secret value X, computes g^X and sends it to the KDC in the AS Request. It is not feasible to compute X from g^X within a reasonable amount of time.

The KDC in turn generates a secret value Y and computes both g^Y and $(g^X)^Y = g^{XY}$. The KDC then generates a unique session key and a ticket for this client and encrypts the session key with g^{XY}. The encrypted session key, ticket and g^Y are all sent back to the client in the AS Reply message.

After receiving the AS Reply, the client computes $(g^Y)^X = g^{XY}$ and decrypts the session key. A snooper that doesn't know the value of X or Y cannot figure out g^{XY} and thus cannot decrypt the session key. An MTA clone does not know X because X is generated on the fly for each ticket request. Therefore, MTA clones cannot snoop on the AS Reply message and determine the session key that was received by the original, legally authorized MTA.

In order for MTA clones to obtain their own tickets they each have to send their own AS Request and obtain their own unique session key. This makes the clones detectable at the KDC. When a KDC issues a ticket, it puts in a lifetime that specifies

when this ticket is no longer valid. A MTA should keep reusing this same ticket until it expires. The following occurs when multiple MTA clones attempt to obtain tickets:

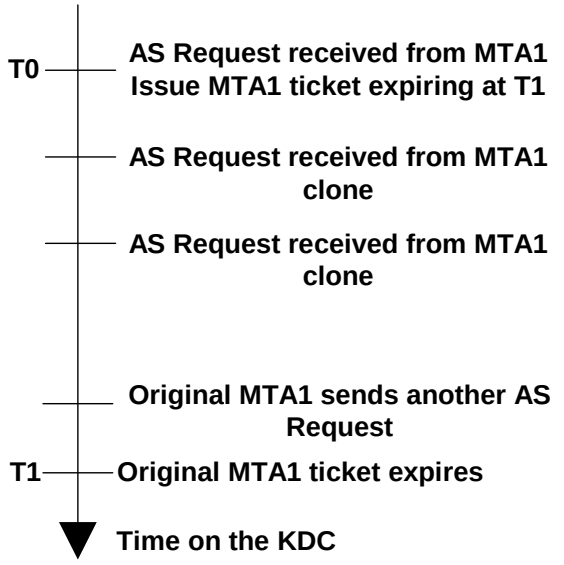

Figure 5: Clone Detection by the KDC

After issuing a ticket to an MTA, the KDC could save the expiration time of that ticket. Normally, the KDC would not expect the MTA to request another ticket until the original ticket is almost expired. If the KDC receives AS Requests with the same MTA identity too early, it could be one of the following situations:

1. MTA clones are obtaining tickets
2. The MTA somehow lost its ticket and had to get another one.
3. Since PacketCable™ uses Kerberos over UDP, the AS Reply packet can be lost and the MTA would time out and retry.

If the MTA has to retry because of UDP unreliability, the retries will be allowed only within a few seconds of the original request. Furthermore, the MTA can use an identical value of g^X while sending the retries. Therefore, the KDC should be able to distinguish the UDP retries from other unexpected AS Requests.

The MTA should not accidentally lose tickets very often if at all. Even if this were the case with a bad MTA implementation, such implementations should not be deployed until the software bugs are fixed. Therefore, if a KDC receives an early AS Request before the original MTA ticket expired and this request is not a retry, there is a very high probability that this is a request from an MTA clone. After one or two such early requests it is probably safe to flag this MTA as a clone threat and deny it any further tickets.

Clone Detection and Disablement by the CMS

Within the PacketCable™ architecture, when the CMS receives an MTA ticket, it uses it to establish IPsec security associations with that MTA but does not need to save this ticket. If the CMS were to save at least the session key and the ticket expiration time, that would enable the CMS to perform clone detection.

The CMS would normally not expect the same MTA to send a different ticket with a different session key until the old ticket is either already expired or is close to its expiration time. When the CMS notices that the MTA changed its session key too early, it can be one of the following:

1. MTA clones are alternating at establishing security associations with the CMS in order for each to make a phone call.
2. The MTA somehow lost its ticket and had to get another one.
3. MTA had to retry due to the unreliability of the UDP transport.

The cloning detection at the CMS is analogous to that at the KDC. UDP retries

would be limited to a short period after the original message and can be identified by a common attribute such as an IPsec SPI (Security Parameters Index). Also, a reasonable MTA implementation should not be accidentally losing tickets. So, after one or two such early session key changes, a CMS can assume that a particular MTA has been cloned and flag it in its database.

The limitation of this cloning detection method is that there could be many CMSs on the same IP Telephony network and when each clone is assigned a different CMS, cloning will not be detected. Since in general there are fewer KDCs than there are CMSs, it is easier to catch clones at the KDC. However, if the KDC does not support this functionality, it may still be useful to perform clone detection at the CMS.

Tamper-Resistant Key Storage in the MTA

Although it is possible to detect MTA clones at the KDC or CMS, cloning detection is not a PacketCable™ requirement and may not be available in a particular CMS or KDC brand. Cloning detection would also add cost and complexity to a server. It must be properly implemented and tuned to avoid false alarms and does not guarantee 100% detection. Therefore, other anti-cloning measures should also be considered.

Another way to prevent MTA cloning is to build an MTA where the cryptographic keys are protected inside tamper-resistant hardware. A secure hardware module must not expose protected keys on its external interfaces, which means that the module must internally implement all cryptographic operations associated with the protected keys.

This protected key storage does come at some cost. This cost can be significantly reduced if the cryptographic module does not have to support internal generation of public keys. Public/private key pairs as well as the corresponding digital certificates can be generated and installed into the secure hardware module during the manufacturing process.

There are a number of integrated client platforms where an MTA may be integrated with other functions in the same device that may also have a need for secure hardware. In such cases, the cost of secure hardware may be a lesser factor or maybe not a factor at all if a particular integrated platform is already required to include secure hardware for functions other than the MTA. These integrated platforms are discussed in the following subsections.

Advanced Digital Television Set-Top

Most digital television set-tops available today already include some form of secure hardware. This secure hardware may take the form of a Smart Card, PCMCIA card or an embedded secure co-processor.

The reason for this is that, in the case of broadcast television, set-tops are not required to send any upstream messages and therefore clones are not detectable in the network. Furthermore, premium television programming has a significantly large revenue stream that attracts pirates.

Some of the more recent advanced set-top models support not only digital broadcast television, but also Internet connectivity and email services with an integrated DOCSIS cable modem. In addition, integrated set-tops may also support MTA functionality to provide VoIP services.

For this type of integrated platform, it makes sense to take advantage of the already available secure hardware modules to protect PacketCable™ MTA keys, including the 1024-bit RSA key as well as the Kerberos session keys.

There are some additional short-lived keys that could also be protected inside the secure hardware module, although there is less risk in losing the shorter-lived ones.

Home Gateway

A home gateway platform would be located in a consumer's home and would sit between the HFC network and a subscriber home network. A home gateway may, for example, obtain entertainment content from the Internet and then distribute the content over a home network, subject to protections provided by Digital Rights Management (DRM).

A DRM system generally includes local enforcement of content usage rules. For example, content copying outside of the home network may be prohibited, or the content may be downloaded for only a limited time period after which it must be erased from the home network. Commonly, DRM is enforced by encrypting the stored content and allowing a client to access a decryption key only when content usage rules are satisfied.

Since the evaluation of content usage rules is performed locally inside a home gateway (and inside other home network devices), the home gateway may already contain secure hardware for enforcing Digital Rights Management. An integrated home gateway may also include an MTA and provide VoIP services. In this case, it again makes sense to share the secure hardware element for protection of both the DRM keys and the PacketCable™ MTA keys.

Soft MTA

An MTA can also be implemented in software, running on a PC that is connected to the Internet via a cable modem. This PC may also be running other unrelated software, or may be downloading software from the Internet and would therefore be a potential target for hackers on the Internet.

It is therefore conceivable for the hackers on the Internet to extract MTA keys without the owner's knowledge and then install them into clones. It would therefore be prudent for a soft MTA to store its keys inside a secure hardware module such as a USB token or a Smart Card.

IP ADDRESS CLONING THREATS AND THEIR PREVENTION

Up to this point, the paper described MTA cloning threats in which the subscriber's identity is impersonated, where the subscriber is linked to an MTA FQDN. In order to impersonate a subscriber, an MTA FQDN, an associated set of cryptographic keys, an MTA MAC address and possibly an MTA IP address are copied from a legitimate MTA into a clone. In addition, cloning of only an MTA's IP address can lead to denial of service. Two such threats are explained in the following subsections.

Loss of QoS at an MTA

Media stream (RTP) packets for a voice conversation between two MTAs (or between an MTA and a PSTN Gateway) may only be authenticated end-to-end. PacketCable™ provides an optional MMH MAC that can be added to each RTP packet to verify that it came from a legitimate source and was not modified in transit.

Because the MMH MAC is verified by a VoIP endpoint (MTA), the CMTS cannot distinguish between good and bad downstream RTP packets and will pass them all through to an MTA. At the same time, the CMTS enforces a rate limit for each MTA and will start dropping downstream

packets if the allocated bandwidth for a particular MTA is exceeded. The same applies to the upstream packets, although the CMTS limits the upstream packets to a particular MAC address domain associated with a specific CMTS line card.

Also, the PacketCable™ DQoS specification requires the CMTS to pass only those VoIP packets that are associated with a particular VoIP QoS gate, where a gate is associated with a specific source MTA IP address and a specific destination MTA IP address. In order for a CMTS to forward a downstream VoIP packet to an MTA, the source IP address must correspond to a previously allocated gate.

A possible attack would be where an adversary:

1. Determines the IP addresses of the two MTAs (or MTA and PSTN Gateway) that have a current voice conversation.
2. Impersonates the IP address of a PSTN Gateway or of one of the MTAs.
3. Starts sending garbage packets to the other MTA.

In this case, the CMTS would match the garbage packets against one of the gates and pass them through to the MTA. But once a rate limit for that MTA is reached, the CMTS will start dropping packets – both good and bad.

This attack can be addressed by making it difficult for VoIP clients to falsify an IP address. Assuming that the adversary is located on an HFC network, the following prevention steps can be taken:

1. The CMTS verifies that the HFC MAC address and IP address match for each upstream packet. This forces an adversary to have to impersonate both the IP address and MAC address at the same time.

2. The CMTS matches up an MTA MAC address against a Cable Modem MAC address. (Even an embedded MTA is required to have a separate MAC and IP addresses.) This check forces an adversary to impersonate the Cable Modem MAC address as well.

3. The Cable Modem provides physical security for the BPI+ keys. BPI+ provides Cable Modem authentication. By physically securing the Cable Modem keys, it makes the impersonation of the Cable Modem MAC address very difficult.

A simplification of steps 1 and 2 can also be applied in what is known as the "DHCP authority" function. This function has the DHCP server only assign long-term IP addresses to CM and MTA with provisioned CM and MTA MAC addresses respectively. Furthermore, it has the CMTS store IP address to CM MAC address associations based on DHCP requests/acknowledgements. In this case the CMTS acts as a DHCP relay agent for CM and MTA, allowing it to sniff and direct DHCP packets passing through. With the DHCP authority function, upstream packets must match IP and CM MAC address associations or be dropped. This applies to CM IP address to CM MAC address mapping as well as MTA IP address to CM MAC address mapping.

An adversary that is sending bad VoIP packets can also be located somewhere else on the Internet where the upstream packets do not go through a CMTS. In that case, an impersonation of a legitimate MTA's IP address can be prevented as follows:

1. The operator of the managed IP backbone would have some Edge

Router that connects to the Internet at-large, knowing that MTAs don't connect through that interface.
2. The Edge Router receiving packets from the general Internet would mark the TOS (Type-Of-Service) byte in the IP header to distinguish them from other packets.
3. When a CMTS gets incoming packets with this TOS byte value, it knows they didn't come from an MTA and would therefore not allow any such packets to match any of the gates, regardless of the IP address values.

Loss of IPsec Security Associations

IPsec keys are normally associated with specific IP addresses. A CMS keeps a list of IPsec Security Associations, where each one has a different MTA IP address.

An adversary could:

1. Take a certified PacketCable™ MTA (MTA-A) and spoof an IP address of another legitimate MTA (MTA-B).
2. MTA-A with MTA-B's IP address sends MTA-A's ticket to the CMS to establish new IPsec SAs.
3. The CMS replaces IPsec SAs of MTA-B's IP address with new ones, based on the session key in MTA A's CMS ticket. Since the real MTA-B did not initiate this key management transaction, it will no longer share IPsec keys with the CMS and will temporarily lose service.

This attack can be addressed by having the CMS conduct a DNS query of the IP address corresponding to the MTA FQDN in the AP Request CMS ticket. If the IP address from this interaction differs from the IP address of the AP Request the request is dropped. Unfortunately, this approach may be CMS processing intensive.

A better approach of mitigating this attack is to have the KDC place the MTA's IP address into a ticket. In this case the CMS would not accept a ticket if the IP address inside the ticket doesn't match the address in an AP Request IP header. If a KDC client falsifies its IP address during a ticket request, usually the KDC will not be able to route a ticket back to that client. So, it would be difficult for an adversary to falsify the IP address inside the ticket. This protection could be strengthened further by verifying the IP address to MAC address mapping at the CMTS. Alternatively it could be strengthened by having the KDC determine the MTA IP address via DNS lookup using the MTA's FQDN, based on MTA MAC address and returned by the provisioning server.

LIMITS TO MTA CLONING

MTA subscriber cloning consists of extracting the MTA FQDN, device certificate and private key and copying them into clones. A cloned MTA could have its own MAC address, since in general the KDC looks at the MAC address in the MTA device certificate and does not know if it is the same as the MAC that the CMTS encountered in the MAC frame header. Current PacketCable specifications do not require the KDC to check the MTA IP address, so each MTA clone could also have its own IP address.

Such threats can be mitigated through clone detection and/or with tamper-resistant key storage in an MTA. But what happens if these approaches are not feasible? Can a cloning threat still be mitigated? The answer is "possibly", if the clones can be restricted to a small portion of a cable plant and forced to operate under operating conditions inconvenient to the pirate.

If MTA clones and their associated CM can be restricted to the same CMTS

upstream, then they will be limited to a small cloning population and will be forced to have only one clone operate at a time. In the latter case the MTA's associated CM would also have to be cloned (MAC and BPI key included) so that more than one CM clone would experience conflicting upstream synchronization messages. CM cloning could be forced through use of the "DHCP authority" function as described previously.

Still, how can an MTA/CM clone be forced onto a single CMTS upstream? First, MTA clones cannot be allowed to use their own IP address. As already mentioned in this paper, the KDC can map the MAC address in the MTA device certificate to the IP address that was previously assigned by the DHCP server and then verify that it is the same as the source IP address in the Kerberos AS Request message. Alternatively, the KDC can first map the MAC address to an MTA FQDN and then to an IP address. Either way, all MTA clones would be forced to share the IP address of the original MTA.

The sharing of an IP address requires some out-of-band coordination between MTA clones since they will not be able to make phone calls at the same time without interfering with each other. This already creates inconvenient operating conditions for a pirate.

Once we know that all clones of the same MTA have to share the same IP address, the subnet component of the IP address could be utilized to restrict the geographical location of the MTA clones. This requires that a CMTS, as part of its DHCP relay operation, convey the IP subnet of an upstream interface associated with a CM or MTA in the "giaddr" field of their DHCP discover messages (see [6]). It also requires that the DHCP server that is configured with the MAC address of the CM and MTA sending an offer message (per "DHCP authority" function) have these MAC addresses assigned to the IP address pool corresponding to the subnet of the CMTS upstream channel in which the CM and MTA are located.

Such restrictions would come at the expense of added DHCP and CMTS configuration complexity. It would also come with the restriction that CM and MTA locations be known for the provisioning. However this restriction could be avoided if the CMTS were to record the upstream interface on the first CM or MTA registration, and make sure that this interface does not change without the customer calling a CSR.

SUMMARY

MTA Cloning attacks could potentially result in loss of revenue and disruption of IP Telephony service. In order to fully address cloning, one needs to fully understand the PacketCable™ architecture and in particular the use of multiple MTA identities that include an MTA MAC address, IP address and its FQDN. Different cloning attacks may be based on the duplication of a different MTA identity.

While most MTA cloning attacks are detectable, cloning detection still has to be built into the IP Telephony network and would potentially affect server performance and complexity. Cloning detection has to be carefully implemented so as not to cause false alarms.

In addition to cloning detection and disablement, it is also possible to protect cryptographic keys with the secure key storage inside MTAs. These two anti-cloning measures can be complementary to each other. The use of secure key storage is particularly attractive on integrated client platforms where it is already utilized for other functions such as decryption of

broadcast television and Digital Rights Management.

Cloning may also be effectively mitigated by forcing them to operate under a single CMTS upstream channel. This requires DHCP and CMTS configuration and filtering options as well as DHCP exchange measures. The approach comes at the expense of added configuration complexity and location knowledge, but may be attractive when cloning detection and/or secure key storage is not feasible.

REFERENCES

[1] PacketCable Security Specification, *PKT-SP-SEC-I05-020116, January 16, 2002, Cable Television Laboratories, Inc.*, *http://www.PacketCable.com/*

[2] PacketCable 1.0 Architecture Framework Technical Report, *PKT-TR-ARCH-V01-991201, December 1, 1999, Cable Television Laboratories, Inc.*, *http://www.PacketCable.com/*

[3] PacketCable Network-Based Call Signaling Protocol Specification, *PKT-SP-EC-MGCP-I04-011221, December 21, 2001, Cable Television Laboratories, Inc.*, *http://www.PacketCable.com/*

[4] PacketCable MTA Device Provisioning Specification, *PKT-SP-PROV-I03-011221, December 21, 2001, Cable Television Laboratories, Inc.*, *http://www.PacketCable.com/*

[5] PacketCable Audio/Video Codecs Specification, *PKT-SP-CODEC-I03-011221, December 21, 2001, Cable Television Laboratories, Inc.*, *http://www.PacketCable.com/*

[6] Dynamic Host Configuration Protocol, *IETF (R. Droms), Internet Informational Standard, RFC 2131, March 1997.*

[7] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, *SP-RFIv1.1-I07-010829, August 29, 2001, Cable Television Laboratories, Inc. http://www.CableLabs.com/*

ACKNOWLEDGEMENTS

This paper is in part based on the work done by the PacketCable™ security team. Several of the cloning threats discussed in this paper were uncovered during discussions within the PacketCable™ security team.

CONTACT INFORMATION

Alexander Medvinsky
Motorola Broadband
6450 Sequence Dr.
San Diego, CA 92121
Tel: (858) 404-2367
smedvinsky@gi.com

Jay Strater
Motorola Broadband
101 Tournament Dr
Horsham, PA 19044
Tel: (215) 323-1362
jstrater@gi.com