Network Working Group
Request for Comments: 4686
Category: Informational

J. Fenton
Cisco Systems, Inc.

September 2006

Analysis of Threats Motivating DomainKeys Identified Mail (DKIM)

Status of This Memo

Copyright Notice

Abstract

This document provides an analysis of some threats against Internet mail that are intended to be addressed by signature-based mail authentication, in particular DomainKeys Identified Mail. It discusses the nature and location of the bad actors, what their capabilities are, and what they intend to accomplish via their attacks.

Table of Contents

1. Introduction

The DomainKeys Identified Mail (DKIM) protocol is being specified by the IETF DKIM Working Group. The DKIM protocol defines a mechanism by which email messages can be cryptographically signed, permitting a signing domain to claim responsibility for the use of a given email address. Message recipients can verify the signature by querying the signer's domain directly to retrieve the appropriate public key, and thereby confirm that the message was attested to by a party in possession of the private key for the signing domain. This document addresses threats relative to two works in progress by the DKIM Working Group, the DKIM signature specification [DKIM-BASE] and DKIM Sender Signing Practices [DKIM-SSP].

Once the attesting party or parties have been established, the recipient may evaluate the message in the context of additional information such as locally-maintained whitelists, shared reputation services, and/or third-party accreditation. The description of these mechanisms is outside the scope of the IETF DKIM Working Group effort. By applying a signature, a good player enables a verifier to associate a positive reputation with the message, in hopes that it will receive preferential treatment by the recipient.

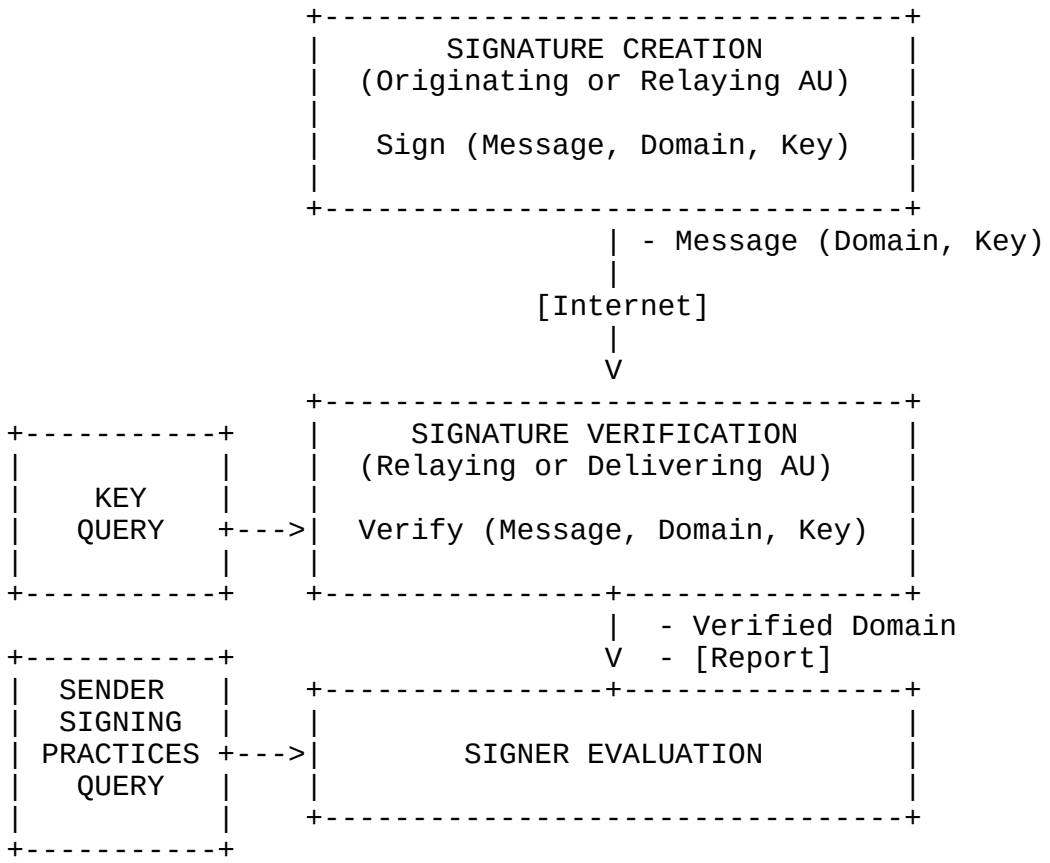
This effort is not intended to address threats associated with message confidentiality nor does it intend to provide a long-term archival signature.

1.1. Terminology and Model

An administrative unit (AU) is the portion of the path of an email message that is under common administration. The originator and recipient typically develop trust relationships with the administrative units that send and receive their email, respectively, to perform the signing and verification of their messages.

The origin address is the address on an email message, typically the RFC 2822 From: address, which is associated with the alleged author of the message and is displayed by the recipient's Mail User Agent (MUA) as the source of the message.

The following diagram illustrates a typical usage flowchart for DKIM:



DKIM operates entirely on the content (body and selected header fields) of the message, as defined in RFC 2822 [RFC2822]. The transmission of messages via SMTP, defined in RFC 2821 [RFC2821], and such elements as the envelope-from and envelope-to addresses and the HELO domain are not relevant to DKIM verification. This is an intentional decision made to allow verification of messages via protocols other than SMTP, such as POP [RFC1939] and IMAP [RFC3501] which an MUA acting as a verifier might use.

The Sender Signing Practices Query referred to in the diagram above is a means by which the verifier can query the alleged author's domain to determine their practices for signing messages, which in turn may influence their evaluation of the message. If, for example, a message arrives without any valid signatures, and the alleged author's domain advertises that they sign all messages, the verifier might handle that message differently than if a signature was not necessarily to be expected.

1.2. Document Structure

The remainder of this document describes the problems that DKIM might be expected to address, and the extent to which it may be successful in so doing. These are described in terms of the potential bad actors, their capabilities and location in the network, and the bad acts that they might wish to commit.

This is followed by a description of postulated attacks on DKIM message signing and on the use of Sender Signing Practices to assist in the treatment of unsigned messages. A list of derived requirements is also presented, which is intended to guide the DKIM design and review process.

The sections dealing with attacks on DKIM each begin with a table summarizing the postulated attacks in each category along with their expected impact and likelihood. The following definitions were used as rough criteria for scoring the attacks:

Impact:

- High: Affects the verification of messages from an entire domain or multiple domains
- Medium: Affects the verification of messages from specific users, Mail Transfer Agents (MTAs), and/or bounded time periods
- Low: Affects the verification of isolated individual messages only

Likelihood:

- High: All email users should expect this attack on a frequent basis
- Medium: Email users should expect this attack occasionally; frequently for a few users
- Low: Attack is expected to be rare and/or very infrequent

2. The Bad Actors

2.1. Characteristics

The problem space being addressed by DKIM is characterized by a wide range of attackers in terms of motivation, sophistication, and capabilities.

At the low end of the spectrum are bad actors who may simply send email, perhaps using one of many commercially available tools, that the recipient does not want to receive. These tools typically allow one to falsify the origin address of messages, and may, in the future, be capable of generating message signatures as well.

At the next tier are what would be considered "professional" senders of unwanted email. These attackers would deploy specific infrastructure, including Mail Transfer Agents (MTAs), registered domains and networks of compromised computers ("zombies") to send messages, and in some cases to harvest addresses to which to send. These senders often operate as commercial enterprises and send messages on behalf of third parties.

The most sophisticated and financially-motivated senders of messages are those who stand to receive substantial financial benefit, such as from an email-based fraud scheme. These attackers can be expected to employ all of the above mechanisms and additionally may attack the Internet infrastructure itself, including DNS cache-poisoning attacks and IP routing attacks.

2.2. Capabilities

In general, the bad actors described above should be expected to have access to the following:

1. An extensive corpus of messages from domains they might wish to impersonate

2. Knowledge of the business aims and model for domains they might wish to impersonate

3. Access to public keys and associated authorization records associated with the domain

and the ability to do at least some of the following:

1. Submit messages to MTAs and Message Submission Agents (MSAs) at multiple locations in the Internet

2. Construct arbitrary message header fields, including those claiming to be mailing lists, resenders, and other mail agents

3. Sign messages on behalf of domains under their control

4. Generate substantial numbers of either unsigned or apparently-signed messages that might be used to attempt a denial-of-service attack

5. Resend messages that may have been previously signed by the domain

6. Transmit messages using any envelope information desired

7. Act as an authorized submitter for messages from a compromised computer

As noted above, certain classes of bad actors may have substantial financial motivation for their activities, and therefore should be expected to have more capabilities at their disposal. These include:

1. Manipulation of IP routing. This could be used to submit messages from specific IP addresses or difficult-to-trace addresses, or to cause diversion of messages to a specific domain.

2. Limited influence over portions of DNS using mechanisms such as cache poisoning. This might be used to influence message routing or to falsify advertisements of DNS-based keys or signing practices.

3. Access to significant computing resources, for example, through the conscription of worm-infected "zombie" computers. This could allow the bad actor to perform various types of brute-force attacks.

4. Ability to eavesdrop on existing traffic, perhaps from a wireless network.

Either of the first two of these mechanisms could be used to allow the bad actor to function as a man-in-the-middle between author and recipient, if that attack is useful.

2.3. Location

Bad actors or their proxies can be located anywhere in the Internet. Certain attacks are possible primarily within the administrative unit of the claimed originator and/or recipient domain have capabilities beyond those elsewhere, as described in the below sections. Bad actors can also collude by acting from multiple locations (a "distributed bad actor").

It should also be noted that with the use of "zombies" and other proxies, externally-located bad actors may gain some of the capabilities of being located within the claimed originator's or recipient's administrative unit. This emphasizes the importance of appropriate security measures, such as authenticated submission of messages, even within administrative units.

2.3.1. Externally-Located Bad Actors

DKIM focuses primarily on bad actors located outside of the administrative units of the claimed originator and the recipient. These administrative units frequently correspond to the protected portions of the network adjacent to the originator and recipient. It is in this area that the trust relationships required for authenticated message submission do not exist and do not scale adequately to be practical. Conversely, within these administrative units, there are other mechanisms such as authenticated message submission that are easier to deploy and more likely to be used than DKIM.

External bad actors are usually attempting to exploit the "any to any" nature of email that motivates most recipient MTAs to accept messages from anywhere for delivery to their local domain. They may generate messages without signatures, with incorrect signatures, or with correct signatures from domains with little traceability. They may also pose as mailing lists, greeting cards, or other agents that legitimately send or resend messages on behalf of others.

2.3.2. Within Claimed Originator's Administrative Unit

Bad actors in the form of rogue or unauthorized users or malware-infected computers can exist within the administrative unit corresponding to a message's origin address. Since the submission of messages in this area generally occurs prior to the application of a message signature, DKIM is not directly effective against these bad actors. Defense against these bad actors is dependent upon other means, such as proper use of firewalls, and Message Submission Agents that are configured to authenticate the author.

In the special case where the administrative unit is non-contiguous (e.g., a company that communicates between branches over the external Internet), DKIM signatures can be used to distinguish between legitimate externally-originated messages and attempts to spoof addresses in the local domain.

2.3.3. Within Recipient's Administrative Unit

Bad actors may also exist within the administrative unit of the message recipient. These bad actors may attempt to exploit the trust relationships that exist within the unit. Since messages will typically only have undergone DKIM verification at the administrative unit boundary, DKIM is not effective against messages submitted in this area.

For example, the bad actor may attempt to spoof a header field indicating the results of verification. This header field would normally be added by the verifier, which would also detect spoofed header fields on messages it was attempting to verify. This could be used to falsely indicate that the message was authenticated successfully.

As in the originator case, these bad actors can be dealt with by controlling the submission of messages within the administrative unit. Since DKIM permits verification to occur anywhere within the recipient's administrative unit, these threats can also be minimized by moving verification closer to the recipient, such as at the Mail Delivery Agent (MDA), or on the recipient's MUA itself.

3. Representative Bad Acts

One of the most fundamental bad acts being attempted is the delivery of messages that are not intended to have been sent by the alleged originating domain. As described above, these messages might merely be unwanted by the recipient, or might be part of a confidence scheme or a delivery vector for malware.

3.1. Use of Arbitrary Identities

This class of bad acts includes the sending of messages that aim to obscure the identity of the actual author. In some cases, the actual sender might be the bad actor, or in other cases might be a third-party under the control of the bad actor (e.g., a compromised computer).

Particularly when coupled with sender signing practices that indicate the domain owner signs all messages, DKIM can be effective in mitigating against the abuse of addresses not controlled by bad

actors. DKIM is not effective against the use of addresses controlled by bad actors. In other words, the presence of a valid DKIM signature does not guarantee that the signer is not a bad actor. It also does not guarantee the accountability of the signer, since DKIM does not attempt to identify the signer individually, but rather identifies the domain that they control. Accreditation and reputation systems and locally-maintained whitelists and blacklists can be used to enhance the accountability of DKIM-verified addresses and/or the likelihood that signed messages are desirable.

3.2. Use of Specific Identities

A second major class of bad acts involves the assertion of specific identities in email.

Note that some bad acts involving specific identities can sometimes be accomplished, although perhaps less effectively, with similar looking identities that mislead some recipients. For example, if the bad actor is able to control the domain "examp1e.com" (note the "one" between the p and e), they might be able to convince some recipients that a message from admin@examp1e.com is really from admin@example.com. Similar types of attacks using internationalized domain names have been hypothesized where it could be very difficult to see character differences in popular typefaces. Similarly, if example2.com was controlled by a bad actor, the bad actor could sign messages from bigbank.example2.com, which might also mislead some recipients. To the extent that these domains are controlled by bad actors, DKIM is not effective against these attacks, although it could support the ability of reputation and/or accreditation systems to aid the user in identifying them.

DKIM is effective against the use of specific identities only when there is an expectation that such messages will, in fact, be signed. The primary means for establishing this is the use of Sender Signing Practices (SSP), which will be specified by the IETF DKIM Working Group.

3.2.1. Exploitation of Social Relationships

One reason for asserting a specific origin address is to encourage a recipient to read and act on particular email messages by appearing to be an acquaintance or previous correspondent that the recipient might trust. This tactic has been used by email-propagated malware that mail themselves to addresses in the infected host’s address book. In this case, however, the author’s address may not be falsified, so DKIM would not be effective in defending against this act.

It is also possible for address books to be harvested and used by an attacker to post messages from elsewhere. DKIM could be effective in mitigating these acts by limiting the scope of origin addresses for which a valid signature can be obtained when sending the messages from other locations.

3.2.2. Identity-Related Fraud

Bad acts related to email-based fraud often, but not always, involve the transmission of messages using specific origin addresses of other entities as part of the fraud scheme. The use of a specific address of origin sometimes contributes to the success of the fraud by helping convince the recipient that the message was actually sent by the alleged author.

To the extent that the success of the fraud depends on or is enhanced by the use of a specific origin address, the bad actor may have significant financial motivation and resources to circumvent any measures taken to protect specific addresses from unauthorized use.

When signatures are verified by or for the recipient, DKIM is effective in defending against the fraudulent use of origin addresses on signed messages. When the published sender signing practices of the origin address indicate that all messages from that address should be signed, DKIM further mitigates against the attempted fraudulent use of the origin address on unsigned messages.

3.2.3. Reputation Attacks

Another motivation for using a specific origin address in a message is to harm the reputation of another, commonly referred to as a "joe-job". For example, a commercial entity might wish to harm the reputation of a competitor, perhaps by sending unsolicited bulk email on behalf of that competitor. It is for this reason that reputation systems must be based on an identity that is, in practice, fairly reliable.

3.2.4. Reflection Attacks

A commonly-used tactic by some bad actors is the indirect transmission of messages by intentionally mis-addressing the message and causing it to be "bounced", or sent to the return address (RFC 2821 envelope-from address) on the message. In this case, the specific identity asserted in the email is that of the actual target of the message, to whom the message is "returned".

DKIM does not, in general, attempt to validate the RFC2821.mailfrom return address on messages, either directly (noting that the mailfrom

address is an element of the SMTP protocol, and not the message content on which DKIM operates), or via the optional Return-Path header field. Furthermore, as is noted in Section 4.4 of RFC 2821 [RFC2821], it is common and useful practice for a message's return path not to correspond to the origin address. For these reasons, DKIM is not effective against reflection attacks.

4. Attacks on Message Signing

Bad actors can be expected to exploit all of the limitations of message authentication systems. They are also likely to be motivated to degrade the usefulness of message authentication systems in order to hinder their deployment. Both the signature mechanism itself and declarations made regarding use of message signatures (referred to here as Sender Signing Practices or SSP) can be expected to be the target of attacks.

4.1. Attacks against Message Signatures

The following is a summary of postulated attacks against DKIM signatures:

Attack Name	Impact	Likelihood
Theft of private key for domain	High	Low
Theft of delegated private key	Medium	Medium
Private key recovery via side channel attack	High	Low
Chosen message replay	Low	M/H
Signed message replay	Low	High
Denial-of-service attack against verifier	High	Medium
Denial-of-service attack against key service	High	Medium
Canonicalization abuse	Low	Medium
Body length limit abuse	Medium	Medium
Use of revoked key	Medium	Low
Compromise of key server	High	Low
Falsification of key service replies	Medium	Medium
Publication of malformed key records and/or signatures	High	Low
Cryptographic weaknesses in signature generation	High	Low
Display name abuse	Medium	High
Compromised system within originator's network	High	Medium
Verification probe attack	Medium	Medium
Key publication by higher-level domain	High	Low

4.1.1. Theft of Private Key for Domain

Message signing technologies such as DKIM are vulnerable to theft of the private keys used to sign messages. This includes "out-of-band" means for this theft, such as burglary, bribery, extortion, and the like, as well as electronic means for such theft, such as a compromise of network and host security around the place where a private key is stored.

Keys that are valid for all addresses in a domain typically reside in MTAs that should be located in well-protected sites, such as data centers. Various means should be employed for minimizing access to private keys, such as non-existence of commands for displaying their value, although ultimately memory dumps and the like will probably contain the keys. Due to the unattended nature of MTAs, some countermeasures, such as the use of a pass phrase to "unlock" a key, are not practical to use. Other mechanisms, such as the use of dedicated hardware devices that contain the private key and perform the cryptographic signature operation, would be very effective in denying export of the private key to those without physical access to the device. Such devices would almost certainly make the theft of the key visible, so that appropriate action (revocation of the corresponding public key) can be taken should that happen.

4.1.2. Theft of Delegated Private Key

There are several circumstances where a domain owner will want to delegate the ability to sign messages for the domain to an individual user or a third party associated with an outsourced activity such as a corporate benefits administrator or a marketing campaign. Since these keys may exist on less well-protected devices than the domain’s own MTAs, they will in many cases be more susceptible to compromise.

In order to mitigate this exposure, keys used to sign such messages can be restricted by the domain owner to be valid for signing messages only on behalf of specific addresses in the domain. This maintains protection for the majority of addresses in the domain.

A related threat is the exploitation of weaknesses in the delegation process itself. This threat can be mitigated through the use of customary precautions against the theft of private keys and the falsification of public keys in transit. For example, the exposure to theft can be minimized if the delegate generates the keypair to be used, and sends the public key to the domain owner. The exposure to falsification (substitution of a different public key) can be reduced if this transmission is signed by the delegate and verified by the domain owner.

4.1.3. Private Key Recovery via Side Channel Attack

All popular digital signature algorithms are subject to a variety of side channel attacks. The most well-known of these are timing channels [Kocher96], power analysis [Kocher99], and cache timing analysis [Bernstein04]. Most of these attacks require either physical access to the machine or the ability to run processes directly on the target machine. Defending against these attacks is out of scope for DKIM.

However, remote timing analysis (at least on local area networks) is known to be feasible [Boneh03], particularly in server-type platforms where the attacker can inject traffic that will immediately be subject to the cryptographic operation in question. With enough samples, these techniques can be used to extract private keys even in the face of modest amounts of noise in the timing measurements.

The three commonly proposed countermeasures against timing analysis are:

1. Make the operation run in constant time. This turns out in practice to be rather difficult.

2. Make the time independent of the input data. This can be difficult, but see [Boneh03] for more details.

3. Use blinding. This is generally considered the best current practice countermeasure, and while not proved generally secure is a countermeasure against known timing attacks. It adds about 2-10% to the cost of the operation and is implemented in many common cryptographic libraries. Unfortunately, Digital Signature Algorithm (DSA) and Elliptic Curve DSA (ECDSA) do not have standard methods though some defenses may exist.

Note that adding random delays to the operation is only a partial countermeasure. Because the noise is generally uniformly distributed, a large enough number of samples can be used to average it out and extract an accurate timing signal.

4.1.4. Chosen Message Replay

Chosen message replay refers to the scenario where the attacker creates a message and obtains a signature for it by sending it through an MTA authorized by the originating domain to himself/herself or an accomplice. They then "replay" the signed message by sending it, using different envelope addresses, to a (typically large) number of other recipients.

Due to the requirement to get an attacker-generated message signed, chosen message replay would most commonly be experienced by consumer ISPs or others offering email accounts to clients, particularly where there is little or no accountability to the account holder (the attacker in this case). One approach to solving this problem is for the domain to only sign email for clients that have passed a vetting process to provide traceability to the message originator in the event of abuse. At present, the low cost of email accounts (zero) does not make it practical for any vetting to occur. It remains to be seen whether this will be the model with signed mail as well, or whether a higher level of trust will be required to obtain an email signature.

A variation on this attack involves the attacker sending a message with the intent of obtaining a signed reply containing their original message. The reply might come from an innocent user or might be an automatic response such as a "user unknown" bounce message. In some cases, this signed reply message might accomplish the attacker's objectives if replayed. This variation on chosen message replay can be mitigated by limiting the extent to which the original content is quoted in automatic replies, and by the use of complementary mechanisms such as egress content filtering.

Revocation of the signature or the associated key is a potential countermeasure. However, the rapid pace at which the message might be replayed (especially with an army of "zombie" computers), compared with the time required to detect the attack and implement the revocation, is likely to be problematic. A related problem is the likelihood that domains will use a small number of signing keys for a large number of customers, which is beneficial from a caching standpoint but is likely to result in a great deal of collateral damage (in the form of signature verification failures) should a key be revoked suddenly.

Signature revocation addresses the collateral damage problem at the expense of significant scaling requirements. At the extreme, verifiers could be required to check for revocation of each signature verified, which would result in very significant transaction rates. An alternative, "revocation identifiers", has been proposed, which would permit revocation on an intermediate level of granularity, perhaps on a per-account basis. Messages containing these identifiers would result in a query to a revocation database, which might be represented in DNS.

Further study is needed to determine if the benefits from revocation (given the potential speed of a replay attack) outweigh the transactional cost of querying a revocation database.

4.1.5. Signed Message Replay

Signed message replay refers to the retransmission of already-signed messages to additional recipients beyond those intended by the author or the original poster of the message. The attacker arranges to receive a message from the victim, and then retransmits it intact but with different envelope addresses. This might be done, for example, to make it look like a legitimate sender of messages is sending a large amount of spam. When reputation services are deployed, this could damage the author's reputation or that of the author's domain.

A larger number of domains are potential victims of signed message replay than chosen message replay because the former does not require the ability for the attacker to send messages from the victim domain. However, the capabilities of the attacker are lower. Unless coupled with another attack such as body length limit abuse, it isn't possible for the attacker to use this, for example, for advertising.

Many mailing lists, especially those that do not modify the content of the message and signed header fields and hence do not invalidate the signature, engage in a form of signed message replay. The use of body length limits and other mechanisms to enhance the survivability of messages effectively enhances the ability to do so. The only things that distinguish this case from undesirable forms of signed message replay is the intent of the replayer, which cannot be determined by the network.

4.1.6. Denial-of-Service Attack against Verifier

While it takes some computing resources to sign and verify a signature, it takes negligible computing resources to generate an invalid signature. An attacker could therefore construct a "make work" attack against a verifier, by sending a large number of incorrectly-signed messages to a given verifier, perhaps with multiple signatures each. The motivation might be to make it too expensive to verify messages.

While this attack is feasible, it can be greatly mitigated by the manner in which the verifier operates. For example, it might decide to accept only a certain number of signatures per message, limit the maximum key size it will accept (to prevent outrageously large signatures from causing unneeded work), and verify signatures in a particular order. The verifier could also maintain state representing the current signature verification failure rate and adopt a defensive posture when attacks may be under way.

4.1.7. Denial-of-Service Attack against Key Service

An attacker might also attempt to degrade the availability of an originator's key service, in order to cause that originator's messages to be unverifiable. One way to do this might be to quickly send a large number of messages with signatures that reference a particular key, thereby creating a heavy load on the key server. Other types of DoS attacks on the key server or the network infrastructure serving it are also possible.

The best defense against this attack is to provide redundant key servers, preferably on geographically-separate parts of the Internet. Caching also helps a great deal, by decreasing the load on authoritative key servers when there are many simultaneous key requests. The use of a key service protocol that minimizes the transactional cost of key lookups is also beneficial. It is noted that the Domain Name System has all these characteristics.

4.1.8. Canonicalization Abuse

Canonicalization algorithms represent a tradeoff between the survival of the validity of a message signature and the desire not to allow the message to be altered inappropriately. In the past, canonicalization algorithms have been proposed that would have permitted attackers, in some cases, to alter the meaning of a message.

Message signatures that support multiple canonicalization algorithms give the signer the ability to decide the relative importance of signature survivability and immutability of the signed content. If an unexpected vulnerability appears in a canonicalization algorithm in general use, new algorithms can be deployed, although it will be a slow process because the signer can never be sure which algorithm(s) the verifier supports. For this reason, canonicalization algorithms, like cryptographic algorithms, should undergo a wide and careful review process.

4.1.9. Body Length Limit Abuse

A body length limit is an optional indication from the signer of how much content has been signed. The verifier can either ignore the limit, verify the specified portion of the message, or truncate the message to the specified portion and verify it. The motivation for this feature is the behavior of many mailing lists that add a trailer, perhaps identifying the list, at the end of messages.

When body length limits are used, there is the potential for an attacker to add content to the message. It has been shown that this content, although at the end, can cover desirable content, especially in the case of HTML messages.

If the body length isn't specified, or if the verifier decides to ignore the limit, body length limits are moot. If the verifier or recipient truncates the message at the signed content, there is no opportunity for the attacker to add anything.

If the verifier observes body length limits when present, there is the potential that an attacker can make undesired content visible to the recipient. The size of the appended content makes little difference, because it can simply be a URL reference pointing to the actual content. Receiving MUAs can mitigate this threat by, at a minimum, identifying the unsigned content in the message.

4.1.10. Use of Revoked Key

The benefits obtained by caching of key records opens the possibility that keys that have been revoked may be used for some period of time after their revocation. The best examples of this occur when a holder of a key delegated by the domain administrator must be unexpectedly deauthorized from sending mail on behalf of one or more addresses in the domain.

The caching of key records is normally short-lived, on the order of hours to days. In many cases, this threat can be mitigated simply by setting a short time-to-live (TTL) for keys not under the domain administrator's direct control (assuming, of course, that control of the TTL value may be specified for each record, as it can with DNS). In some cases, such as the recovery following a stolen private key belonging to one of the domain's MTAs, the possibility of theft and the effort required to revoke the key authorization must be considered when choosing a TTL. The chosen TTL must be long enough to mitigate denial-of-service attacks and provide reasonable transaction efficiency, and no longer.

4.1.11. Compromise of Key Server

Rather than by attempting to obtain a private key, an attacker might instead focus efforts on the server used to publish public keys for a domain. As in the key theft case, the motive might be to allow the attacker to sign messages on behalf of the domain. This attack provides the attacker with the additional capability to remove legitimate keys from publication, thereby denying the domain the ability for the signatures on its mail to verify correctly.

In order to limit the ability to sign a message to entities authorized by the owner of a signing domain, a relationship must be established between the signing address and the location from which a public key is obtained to verify the message. DKIM does this by publishing either the public key or a reference to it within the DNS hierarchy of the signing domain. The verifier derives the location from which to retrieve the public key from the signing address or domain. The security of the verification process is therefore dependent on the security of the DNS hierarchy for the signing domain.

An attacker might successfully compromise the host that is the primary key server for the signing domain, such as the domain's DNS master server. Another approach might be to compromise a higher-level DNS server and change the delegation of name servers for the signing domain to others under the control of the attacker.

This attack can be mitigated somewhat by independent monitoring to audit the key service. Such auditing of the key service should occur by means of zone transfers rather than queries to the zone's primary server, so that the addition of records to the zone can be detected.

4.1.12. Falsification of Key Service Replies

Replies from the key service may also be spoofed by a suitably positioned attacker. For DNS, one such way to do this is "cache poisoning", in which the attacker provides unnecessary (and incorrect) additional information in DNS replies, which is cached.

DNSSEC [RFC4033] is the preferred means of mitigating this threat, but the current uptake rate for DNSSEC is slow enough that one would not like to create a dependency on its deployment. In the case of a cache poisoning attack, the vulnerabilities created by this attack are both localized and of limited duration, although records with relatively long TTL may persist beyond the attack itself.

4.1.13. Publication of Malformed Key Records and/or Signatures

In this attack, the attacker publishes suitably crafted key records or sends mail with intentionally malformed signatures, in an attempt to confuse the verifier and perhaps disable verification altogether. This attack is really a characteristic of an implementation vulnerability, a buffer overflow or lack of bounds checking, for example, rather than a vulnerability of the signature mechanism itself. This threat is best mitigated by careful implementation and creation of test suites that challenge the verification process.

4.1.14. Cryptographic Weaknesses in Signature Generation

The cryptographic algorithms used to generate mail signatures, specifically the hash algorithm and digital signature generation and verification operations, may over time be subject to mathematical techniques that degrade their security. At this writing, the SHA-1 hash algorithm is the subject of extensive mathematical analysis that has considerably lowered the time required to create two messages with the same hash value. This trend can be expected to continue.

One consequence of a weakness in the hash algorithm is a hash collision attack. Hash collision attacks in message signing systems involve the same person creating two different messages that have the same hash value, where only one of the two messages would normally be signed. The attack is based on the second message inheriting the signature of the first. For DKIM, this means that a sender might create a "good" message and a "bad" message, where some filter at the signing party's site would sign the good message but not the bad message. The attacker gets the good message signed, and then incorporates that signature in the bad message. This scenario is not common, but could happen, for example, at a site that does content analysis on messages before signing them.

Current known attacks against SHA-1 make this attack extremely difficult to mount, but as attacks improve and computing power becomes more readily available, such an attack could become achievable.

The message signature system must be designed to support multiple signature and hash algorithms, and the signing domain must be able to specify which algorithms it uses to sign messages. The choice of algorithms must be published in key records, and not only in the signature itself, to ensure that an attacker is not able to create signatures using algorithms weaker than the domain wishes to permit.

Because the signer and verifier of email do not, in general, communicate directly, negotiation of the algorithms used for signing cannot occur. In other words, a signer has no way of knowing which algorithm(s) a verifier supports or (due to mail forwarding) where the verifier is. For this reason, it is expected that once message signing is widely deployed, algorithm change will occur slowly, and legacy algorithms will need to be supported for a considerable period. Algorithms used for message signatures therefore need to be secure against expected cryptographic developments several years into the future.

4.1.15. Display Name Abuse

Message signatures only relate to the address-specification portion of an email address, while some MUAs only display (or some recipients only pay attention to) the display name portion of the address. This inconsistency leads to an attack where the attacker uses a From header field such as:

From: "Dudley DoRight" <whiplash@example.org>

In this example, the attacker, whiplash@example.org, can sign the message and still convince some recipients that the message is from Dudley DoRight, who is presumably a trusted individual. Coupled with the use of a throw-away domain or email address, it may be difficult to hold the attacker accountable for using another's display name.

This is an attack that must be dealt with in the recipient's MUA. One approach is to require that the signer's address specification (and not just the display name) be visible to the recipient.

4.1.16. Compromised System within Originator's Network

In many cases, MTAs may be configured to accept and sign messages that originate within the topological boundaries of the originator's network (i.e., within a firewall). The increasing use of compromised systems to send email presents a problem for such policies, because the attacker, using a compromised system as a proxy, can generate signed mail at will.

Several approaches exist for mitigating this attack. The use of authenticated submission, even within the network boundaries, can be used to limit the addresses for which the attacker may obtain a signature. It may also help locate the compromised system that is the source of the messages more quickly. Content analysis of outbound mail to identify undesirable and malicious content, as well as monitoring of the volume of messages being sent by users, may also prevent arbitrary messages from being signed and sent.

4.1.17. Verification Probe Attack

As noted above, bad actors (attackers) can sign messages on behalf of domains they control. Since they may also control the key service (e.g., the authoritative DNS name servers for the _domainkey subdomain), it is possible for them to observe public key lookups, and their source, when messages are verified.

One such attack, which we will refer to as a "verification probe", is to send a message with a DKIM signature to each of many addresses in a mailing list. The messages need not contain valid signatures, and each instance of the message would typically use a different selector. The attacker could then monitor key service requests and determine which selectors had been accessed, and correspondingly which addressees used DKIM verification. This could be used to target future mailings at recipients who do not use DKIM verification, on the premise that these addressees are more likely to act on the message contents.

4.1.18. Key Publication by Higher-Level Domain

In order to support the ability of a domain to sign for subdomains under its administrative control, DKIM permits the domain of a signature (d= tag) to be any higher-level domain than the signature's address (i= or equivalent). However, since there is no mechanism for determining common administrative control of a subdomain, it is possible for a parent to publish keys that are valid for any domain below them in the DNS hierarchy. In other words, mail from the domain example.anytown.ny.us could be signed using keys published by anytown.ny.us, ny.us, or us, in addition to the domain itself.

Operation of a domain always requires a trust relationship with higher-level domains. Higher-level domains already have ultimate power over their subdomains: they could change the name server delegation for the domain or disenfranchise it entirely. So it is unlikely that a higher-level domain would intentionally compromise a subdomain in this manner. However, if higher-level domains send mail on their own behalf, they may wish to publish keys at their own level. Higher-level domains must employ special care in the delegation of keys they publish to ensure that any of their subdomains are not compromised by misuse of such keys.

4.2. Attacks against Message Signing Practices

The following is a summary of postulated attacks against signing practices:

Attack Name	Impact	Likelihood
Look-alike domain names	High	High
Internationalized domain name abuse	High	High
Denial-of-service attack against signing practices	Medium	Medium
Use of multiple From addresses	Low	Medium
Abuse of third-party signatures	Medium	High
Falsification of Sender Signing Practices replies	Medium	Medium

4.2.1. Look-Alike Domain Names

Attackers may attempt to circumvent signing practices of a domain by using a domain name that is close to, but not the same as, the domain with signing practices. For instance, "example.com" might be replaced by "examp1e.com". If the message is not to be signed, DKIM does not require that the domain used actually exist (although other mechanisms may make this a requirement). Services exist to monitor domain registrations to identify potential domain name abuse, but naturally do not identify the use of unregistered domain names.

A related attack is possible when the MUA does not render the domain name in an easily recognizable format. If, for example, a Chinese domain name is rendered in "punycode" as xn--cjsp26b3obxw7f.com, the unfamiliarity of that representation may enable other domains to more easily be mis-recognized as the expected domain.

Users that are unfamiliar with internet naming conventions may also mis-recognize certain names. For example, users may confuse online.example.com with online-example.com, the latter of which may have been registered by an attacker.

4.2.2. Internationalized Domain Name Abuse

Internationalized domain names present a special case of the look-alike domain name attack described above. Due to similarities in the appearance of many Unicode characters, domains (particularly those drawing characters from different groups) may be created that are visually indistinguishable from other, possibly high-value domains. This is discussed in detail in Unicode Technical Report 36 [UTR36].

Surveillance of domain registration records may point out some of these, but there are many such similarities. As in the look-alike domain attack above, this technique may also be used to circumvent sender signing practices of other domains.

4.2.3. Denial-of-Service Attack against Signing Practices

Just as the publication of public keys by a domain can be impacted by an attacker, so can the publication of Sender Signing Practices (SSP) by a domain. In the case of SSP, the transmission of large amounts of unsigned mail purporting to come from the domain can result in a heavy transaction load requesting the SSP record. More general DoS attacks against the servers providing the SSP records are possible as well. This is of particular concern since the default signing practices are "we don't sign everything", which means that SSP failures result in the verifier's failure to heed more stringent signing practices.

As with defense against DoS attacks for key servers, the best defense against this attack is to provide redundant servers, preferably on geographically-separate parts of the Internet. Caching again helps a great deal, and signing practices should rarely change, so TTL values can be relatively large.

4.2.4. Use of Multiple From Addresses

Although this usage is never seen by most recipients, RFC 2822 [RFC2822] permits the From address to contain multiple address specifications. The lookup of Sender Signing Practices is based on the From address, so if addresses from multiple domains are in the From address, the question arises which signing practices to use. A rule (say, "use the first address") could be specified, but then an attacker could put a throwaway address prior to that of a high-value domain. It is also possible for SSP to look at all addresses, and choose the most restrictive rule. This is an area in need of further study.

4.2.5. Abuse of Third-Party Signatures

In a number of situations, including mailing lists, event invitations, and "send this article to a friend" services, the DKIM signature on a message may not come from the originating address domain. For this reason, "third-party" signatures, those attached by the mailing list, invitation service, or news service, frequently need to be regarded as having some validity. Since this effectively makes it possible for any domain to sign any message, a sending

domain may publish sender signing practices stating that it does not use such services, and accordingly that verifiers should view such signatures with suspicion.

However, the restrictions placed on a domain by publishing "no third-party" signing practices effectively disallows many existing uses of email. For the majority of domains that are unable to adopt these practices, an attacker may with some degree of success sign messages purporting to come from the domain. For this reason, accreditation and reputation services, as well as locally-maintained whitelists and blacklists, will need to play a significant role in evaluating messages that have been signed by third parties.

4.2.6. Falsification of Sender Signing Practices Replies

In an analogous manner to the falsification of key service replies described in Section 4.1.12, replies to sender signing practices queries can also be falsified. One such attack would be to weaken the signing practices to make unsigned messages allegedly from a given domain appear less suspicious. Another attack on a victim domain that is not signing messages could attempt to make the domain's messages look more suspicious, in order to interfere with the victim's ability to send mail.

As with the falsification of key service replies, DNSSEC is the preferred means of mitigating this attack. Even in the absence of DNSSEC, vulnerabilities due to cache poisoning are localized.

4.3. Other Attacks

This section describes attacks against other Internet infrastructure that are enabled by deployment of DKIM. A summary of these postulated attacks is as follows:

Attack Name	Impact	Likelihood
Packet amplification attacks via DNS	N/A	Medium

4.3.1. Packet Amplification Attacks via DNS

Recently, there has been an increase in denial-of-service attacks involving the transmission of spoofed UDP DNS requests to openly-accessible domain name servers [US-CERT-DNS]. To the extent that the response from the name server is larger than the request, the name server functions as an amplifier for such an attack.

DKIM contributes indirectly to this attack by requiring the publication of fairly large DNS records for distributing public keys. The names of these records are also well known, since the record names can be determined by examining properly-signed messages. This attack does not have an impact on DKIM itself. DKIM, however, is not the only application that uses large DNS records, and a DNS-based solution to this problem will likely be required.

5. Derived Requirements

This section lists requirements for DKIM not explicitly stated in the above discussion. These requirements include:

The store for key and SSP records must be capable of utilizing multiple geographically-dispersed servers.

Key and SSP records must be cacheable, either by the verifier requesting them or by other infrastructure.

The cache time-to-live for key records must be specifiable on a per-record basis.

The signature algorithm identifier in the message must be one of the ones listed in a key record for the identified domain.

The algorithm(s) used for message signatures need to be secure against expected cryptographic developments several years in the future.

6. Security Considerations

This document describes the security threat environment in which DomainKeys Identified Mail (DKIM) is expected to provide some benefit, and it presents a number of attacks relevant to its deployment.

7. Informative References

[Bernstein04] Bernstein, D., "Cache Timing Attacks on AES", April 2004.

[Boneh03] Boneh, D. and D. Brumley, "Remote Timing Attacks are Practical", Proc. 12th USENIX Security Symposium, 2003.

[DKIM-BASE] Allman, E., "DomainKeys Identified Mail (DKIM) Signatures", Work in Progress, August 2006.

[DKIM-SSP] Allman, E., "DKIM Sender Signing Practices", Work in Progress, August 2006.

[Kocher96] Kocher, P., "Timing Attacks on Implementations of Diffie-Hellman, RSA, and other Cryptosystems", Advances in Cryptology, pages 104-113, 1996.

[Kocher99] Kocher, P., Joffe, J., and B. Yun, "Differential Power Analysis: Leaking Secrets", Crypto '99, pages 388-397, 1999.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[US-CERT-DNS] US-CERT, "The Continuing Denial of Service Threat Posed by DNS Recursion".

[UTR36] Davis, M. and M. Suignard, "Unicode Technical Report #36: Unicode Security Considerations", UTR 36, July 2005.

Appendix A. Acknowledgements

The author wishes to thank Phillip Hallam-Baker, Eliot Lear, Tony Finch, Dave Crocker, Barry Leiba, Arvel Hathcock, Eric Allman, Jon Callas, Stephen Farrell, Doug Otis, Frank Ellermann, Eric Rescorla, Paul Hoffman, Hector Santos, and numerous others on the ietf-dkim mailing list for valuable suggestions and constructive criticism of earlier versions of this document.

Author's Address

Jim Fenton
Cisco Systems, Inc.
MS SJ-9/2
170 W. Tasman Drive
San Jose, CA 95134-1706
USA

Phone: +1 408 526 5914
EMail: fenton@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).