INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

MD5 の安全性の限界に関する調査研究報告書

2008 年 7 月

独立行政法人　情報処理推進機構

目　　次

１．調査研究概要

１.１．背景

電子的な情報の信頼性を確保するための電子署名等の目的のためにメッセージダイジェストあるいはハッシュ値といったものが広く用いられている。

このハッシュ値を生成するハッシュ関数は、元となる情報に対してある演算を施して短い数字列を生成（圧縮）する関数であり、この短い数字列への変換は一方向性のものであって、変換された短い数字列から元の情報を逆算できない（ほとんど不可能）こと、同じハッシュ値をもつ別の情報を見つけることは非常に困難であるという性質を持つ。そのことから、公開鍵暗号基盤（PKI）や電子署名等に用いられる。そのため、ハッシュ値を生成するハッシュ関数は重要なセキュリティ機能の一つである。

1991 年に開発され、現在でも広く用いられてきている MD5 (Message Digest 5) と呼ばれるハッシュ関数は、2004 年 8 月に異なる 2 つの元情報に対して同一の MD5 ハッシュ値を生成できる（衝突が起きる）ことが示された。

さらに、2007 年 4 月には、情報の一部を固定しながら同一の MD5 ハッシュ値を生成する計算法が発見され、その結果短時間で元の情報を特定できることが示された。この結果、例えば電子メールのパスワードの確認等のために MD5 を用いている APOP (Authenticated Post Office Protocol) において、MD5 のハッシュ値を解読することで元のパスワードを推定できることから、電子メールを盗聴される危険性がでてきている。

この MD5 は電子メールのパスワード秘匿や電子文書の信頼性を担保する電子署名等に広く用いられている技術であることから、当該技術を用いるシステム利用者、開発者に対し早急な対応を促し、盗聴等による被害が発生、拡大することを未然に防ぐことが急務である。

１.２．目的

本調査研究では、MD5 のハッシュ値から元の情報を特定する手法に関する再確認並びに実環境での検証を行い、MD5 を利用する上での限界を明らかにすることを目的とする。

図　１.２-１に本調査研究の概略イメージを示す。

本調査研究の課題としては、

・実環境での実装の確認：手法の再確認、検証

・なりすましの認知：メールクライアント側での認知度合いの確認

等がある。

本研究による成果としては、

・利用者への注意喚起による被害発生、拡大の防止

・適用システムへの自動チェック機能実装促進

・新手法への移行促進

等がある。

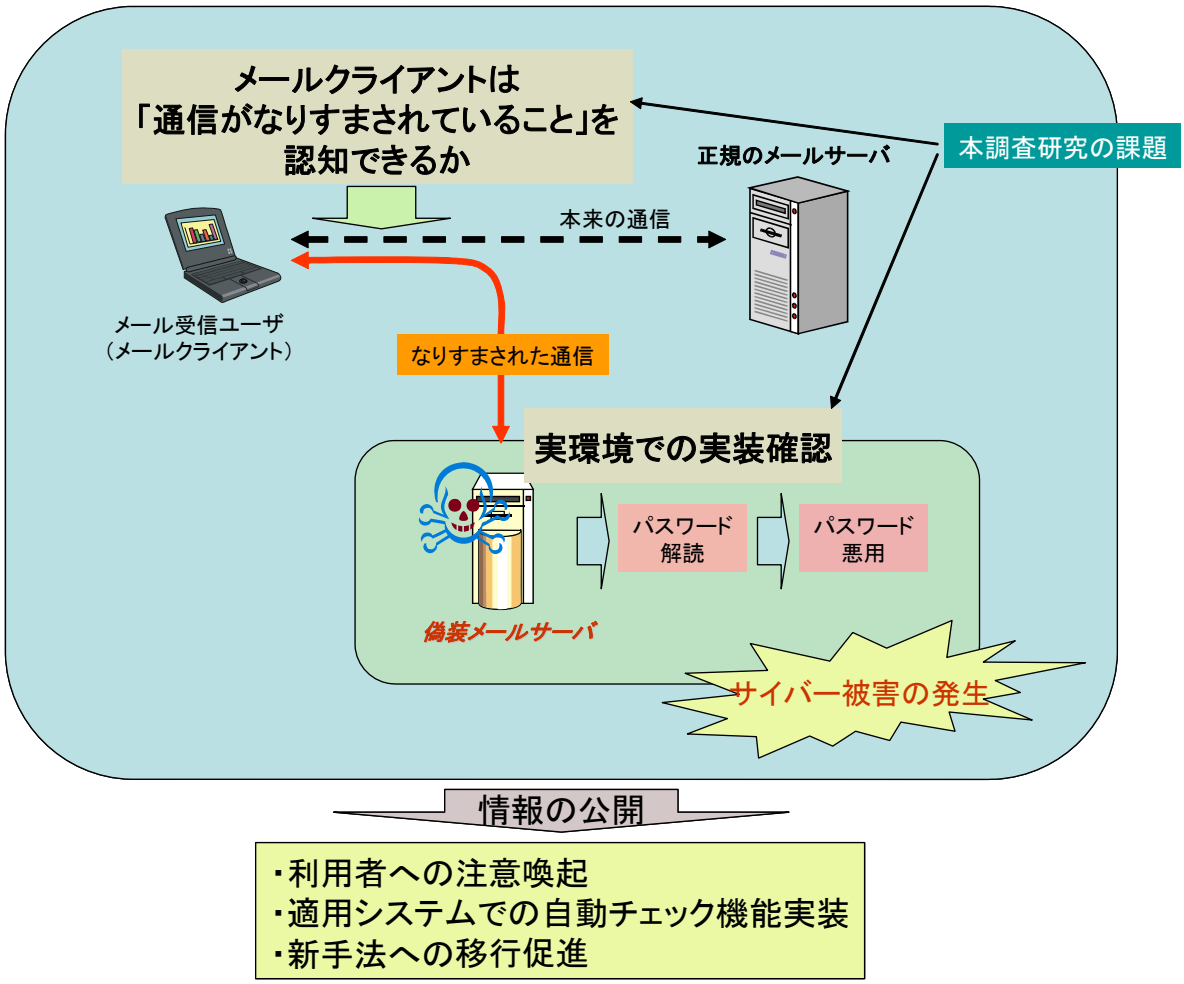

図　1.2-1　本調査研究の概略イメージ

１.３. 調査研究内容

本調査研究では、MD5 の安全性限界を明らかにするため、以下の 2 つの調査研究を行った。

(1) MD5 の脆弱性に関する調査研究

2007 年 4 月に明らかにされた MD5 に係る解読手法の正当性を追試･検証を行い､MD5 脆弱性に係る影響度の分析並びに当面の対応策の研究を行った。

具体的には、当該解読手法を明らかにした電気通信大学の太田和夫教授、國廣 昇准教授から指導・指示に基づき、MD5 に係る解読手法の正当性の追試・検証の検討、MD5 脆弱性に係る影響度の分析並びに当面の対応策の検討を行った。

(2) MD5 解読手法の実ネットワーク環境での実証調査

(1)で検討した結果を受けて、MD5 解読手法を実ネットワーク環境に実装し、動作実験を行うことでその挙動、影響度等に関するデータを収集した。また、対応策の実装を行うことで、当面の対策の有効性の確認等を行った。

２．MD5 の脆弱性に関する調査研究

２.１．攻撃アルゴリズムの確認

２.１.１．MD5 脆弱性に関する公開情報の調査

(1) ハッシュ関数の基礎

ここでは、本報告書で必要となるハッシュ関数に関して必要な事柄を解説する。

暗号学的ハッシュ関数、もしくは暗号学的に安全なハッシュ関数とは、次の性質を持つハッシュ関数である[Handbook04]。

・入力：任意長の系列

・出力：長さの短い系列（128 ビット、160 ビットなど）

以下の性質を満たす。

・一方向性：y が与えられたときに、$y=h(x)$ となる x を求めることが困難である

・第二原像困難性：x が与えられ時に、$h(x)=h(x')$ となり、x とは異なる x' を求めることが困難である

・衝突困難性：$h(x)=h(x')$ となる異なる x と x' を求めることが困難である

衝突困難性は、攻撃者から見れば一番容易な攻撃であり、一方向性は一番困難な攻撃である。すなわち、一方向性を破ることが出来れば、容易に衝突困難性も破ることができる。衝突困難性ですら破れないことをハッシュ関数の最低限の安全性としては要請されている。

APOP に対する攻撃では、ハッシュ関数が Merkle-Damgard 構成法を基にしていることに強く依存している。Merkle-Damgard 型のハッシュ関数の場合、その性質から、途中の値が一旦衝突を起こしてしまうと、それ以降の入力値が共通であれば、常に衝突が起きてしまう。これ以降の入力値が、未知でも構わないことが攻撃にとって重要である。

(2) MD5 自身の脆弱性に関する歴史

以下、MD5 自身の提案、解読の歴史を簡単に振り返る。

1991 年 Ron Rivest により、MD4（MD5 のもととなる方式）が提案される[Rivest91]。

1992 年 Ron Rivest により、ハッシュ関数 MD5 が提案される[Rivest92]。

1993 年 B. den Boer らにより、異なる IV に対して衝突を発見する方法が提案される[BB93]。すなわち、MD5(IV;X)=MD5(IV';X')となる IV, IV', X, X'の生成法が提案される。MD5 は、厳格に IV が定義されているため、これは、実際の脅威ではない。

1996 年 Dobbertin により、仕様と異なる IV に対して、衝突が発見される[Dobbertin96-1,

Dobbertin96-2]。すなわち、MD5(IV';X)=MD5(IV';X')となる IV'、X、X'を求めることに成功した。仕様とは異なる初期ベクトル IV に対する攻撃であるため、これもまた、実際の脅威ではない。

2004 年 8 月 Wang らにより、2^{39}回の MD5 の計算で、衝突を見つける方法が提案された[WL04]。これは、完全な実現可能な脅威である。それとともに、MD5 の衝突困難性は保証されなくなった。

2005 年 5 月 Wang らにより、詳細な衝突発見アルゴリズムが学会で発表される[WY05]。

2005 年 5 月 Klima により、2^{33}回の MD5 の計算で衝突が見つけられるアルゴリズムが提案される[Klima05]。

2005 年 11 月 Sasaki らにより、2^{30}回の MD5 の計算で衝突が見つけられるアルゴリズムが提案される[SNKO05]。

上記二つのアルゴリズムは、いずれも Wang らのアルゴリズムの拡張に当たる。また、上記拡張で MD5 に対する衝突の発見は、もはや現実的な時間で実行可能である。これ以降は、SHA-0、SHA-1 などのより困難なハッシュ関数の解析に研究の中心が移ることになった。

(3) ハッシュ関数の衝突を用いた実際のシステムへの攻撃

例え、衝突困難性が破られたとしても、実際のアプリケーションの安全性が低下することを必ずしも意味しない。そのため、衝突発見のアルゴリズムの改善が進められるとともに、衝突困難性が破られた時の実際のアプリケーションレベルでの脅威に関する研究が進められた。

まず、衝突困難性だけでは実際の脅威とならない例を示す。ここでは、簡単なパスワード認証を例にとり説明をする（図 ２.１-１）。

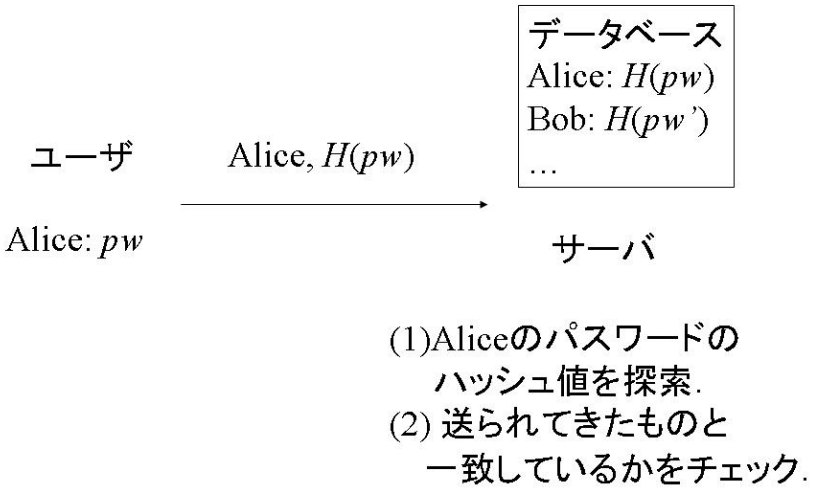

図 2.1-1　ハッシュ値を用いたパスワード認証

ユーザ Alice がサービスの提供を得るために、サーバにログインする状況を考える。ユーザのパスワードを pw とする。サーバのデータベースは、ユーザのパスワード自身ではなく、パスワードのハッシュ値を保管している。ユーザは、ログインするとき、パスワードをそのまま送るのではなく，パスワードのハッシュ値を取った値を自身の ID とともに送る。サーバは、ID をもとに、データベースからユーザに対応したハッシュ値を探し、送られて来た値と一致するかをチェックする。一致していれば正しいユーザであると判定する。

この例の場合は、たとえ衝突が発見できるとしても、攻撃者はその能力を有効に利用することができない。パスワードを入手するためには、ハッシュ値から元の系列を求める必要があるが、それは、ハッシュ関数の一方向性を破る必要がある。これは、衝突困難性よりもはるかに困難であり、現在のところ、この一方向性を破る方法は、MD5 より下位レベルの MD4 においてさえ、提案されていない。以上より、衝突困難性を破るだけでは、安全性が低下されたとは必ずしも言えないことがわかる。

次に、署名の偽造を例に取る。異なるメッセージ *m1*、*m2* に対して、*H*(*m1*)=*H*(*m2*) とすることができたとしよう。このとき、*m1* に対して有効な署名は、*m2* に対しても有効な署名となる。そのため、署名のすり替えが可能であり、これは現実の脅威となる。しかしながら、*m1*, *m2* ともに意味のあるメッセージでなおかつ、衝突を発見する方法は提案されていない。さらに、*m1* を固定したときに、*m2* を求めることは、第二原像衝突困難性を破ることになり、これも現在の技術では不可能である。

衝突を見つけることができることを利用した攻撃法の最初の例は、X.509 証明書の偽造である[LWW2005]。この攻撃は、2005 年 3 月 1 日に公表された。この報告では、正しい CA 署名を持つ 2 種類の X.509 証明書が作成可能であることを示している。この 2 種類の証明書を悪用すれば、否認防止性が崩壊し、現実の脅威となる。この攻撃では、二つの証明者の所有者が同一のものしか証明書の偽造が出来なかった。Stevens らは、この制約を取り払い、所有者が異なっていても、有効な証明書の偽造に成功している[SLW2007]。

次に提案されたのは、Postscript で記述された文書のすり替え攻撃である[DL05]。これは、ps ファイルの構造を強く利用した攻撃である。ハッシュの衝突を計算することにより、同じハッシュ値を持ちながら、任意に選んだメッセージを「表示」させることができる ps ファイルが可能である。

(4) APOP に対する攻撃の歴史

ついで提案されたのは、電子メールにおけるクライアント認証のプロトコルの一つ APOP 方式に対する攻撃である[Leurent07][SYA07]。

POP3 は、メールのサーバ － クライアント間のパスワード認証の代表的かつ最も簡単な方法である。これは、パスワードを平文のまま通信路を通して、ID とパスワードの対応があっているかの判定をすることにより認証を行っている。しかしながら、盗聴することにより簡単にパスワードの取得が可能であり、安全性が保証されていない通信路上では使用すべきではない。そのような経緯で、パスワードをそのまま使うのではなく、ハッシュ値によりパスワード自身は隠す方式が提案され、広く使われている。図 2.1-2 は APOP 認証を示したものである[Myers96]。

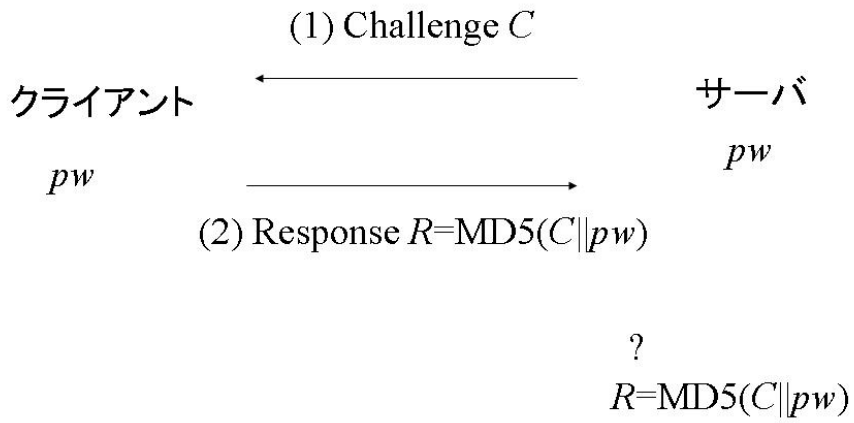

図 2.1-2 APOP 方式のパスワード認証

APOP は、challenge & response 認証の一つである。すなわち、サーバが、毎回変わるチャレンジを生成し、クライアントは、チャレンジと自分の持っているパスワードを連接した文字列に対して MD5 計算をし、その値をレスポンスとして、サーバに送る。受け取ったサーバは、自分が生成したチャレンジと保持しているクライアントのパスワードを元に、同様にレスポンスを生成し、同じ値が送られてきたならば、パスワードを保持している（すなわち、正しいユーザである）と判定し、通常のメールの配信作業を行う。通信路上には生の文字列としてパスワードが流れていないため、安全であると考えられてきた。また、盗聴をしてレスポンスの値を保持したとしても、毎回変わるチャレンジに対してレスポンスを生成しなくてはならないため、リプレイ攻撃にも対処できると考えられてきた。

2007 年 3 月、ルクセンブルグで開かれた国際会議 Fast Software Encryption（略称 FSE）で、フランスの Leurent は、APOP に対する攻撃を提案している[Leurent07]。この論文では、攻撃者がサーバになりすますことにより、3 文字までのパスワードを復元することが可能であることを示している。ほぼ同時期に、Sasaki（電通大），Yamamoto（NTT）、Aoki（NTT）は、同じアイデアに基づき、同じ結果を得ている[SYA07]。この二つの方式は、ともに、Wang らの MD5 の攻撃に基づいている。しかしながら、Wang らの方式に基づく限り、3 文字の限界を超えることは原理上不可能である。それに対して、Sasaki, Wang, Ohta, Kunihiro の電通大の研究グループは、3 文字という壁を突破し、実用上は 31 文字まで、理論上は、61 文字のパスワードを復元する方法を、同会議のランプセッション（採録された論文ではなく、研究速報を発表するセッション）で報告している[SWOK07]。この結果は、4 月 19 日の読売新聞第二面にも紹介され、社会的なインパクトがある結果である。なお、彼らの手法は、2008 年 1 月国内シンポジウム SCIS で概要が発表され[SWOK08-1]、2008 年サンフランシスコで開催される、RSA Conference Cryptographer's track で詳細が発表される予定である[SWOK08-2]。

２．１．２．APOP に対する攻撃アルゴリズムに関する調査

(1) 攻撃アルゴリズム

[SWOK08-1]に基づいて、彼らの手法を簡単に説明する。ここで、正しいパスワードを系列 pw とする。

0. 攻撃者は、メールサーバになりすます。
1. 攻撃者はパスワードの一文字目を推測する。推測したパスワードを $p1$ とする。$MD5(C|p1)=MD5(C'|p1)$となる(C, C')の組を生成する。
2. クライアントからの接続要求に対して、攻撃者（偽装したメールサーバ）は、チャレンジ C を送る。

3. クライアントは、プロトコル通り、MD5(*C*|*pw*)を計算し、偽装されたサーバに送る。
4. 攻撃者は、「メールなし」をクライアントに送る。
5. 次のクライアントからの接続要求に対して、攻撃者（偽装したメールサーバ）は、チャレンジ C'を送る。
6. クライアントは、プロトコル通り、MD5(*C'*|*pw*)を計算し、偽装されたサーバに送る。
7. 攻撃者は、クライアントから送られてきた二つの値が等しいかを判定する。つまり、MD5(*C*|*pw*)=MD5(*C'*|*pw*)であるかを判定し、等しければ、*pw* の一文字目は *p*1 であったと断定する。違っている場合は、*p*1 を別のものに置き換えて 1. から繰り返す。
8. *n* 文字までパスワードが確定したとする。確定した部分を *Pn* と書くことにする。*n*+1 文字目を *p* と推測した場合、攻撃者は、MD5(*C*|*Pn*|*p*)=MD5(*C'*|*Pn*|*p*)となる(*C, C'*)を求める。後は、同様のステップを繰り返すことにより、パスワードを全て求める。

衝突を求める際、MD5 の入力の後半部は、攻撃者は自由に設定することができない。これは、推測する値とこれまでに確定した値を入れる必要があるためである。そのため、衝突を求めるアルゴリズムによっては、うまく衝突を求めることができず、攻撃は成功しない。実際、Wang の衝突発見アルゴリズムに基づいた Leurent の攻撃、及び Sasaki -Yamamoto-Aoki の攻撃は、3 文字までしか原理的にパスワードの復元ができない。それに対して、Sasaki-Wang-Ohta-Kunihiro は、IV bridge という技術を新たに導入することにより、3 文字という壁を取り払い、現実的には 31 文字、理論上は 61 文字での復元に成功する攻撃法を提案している。

この攻撃に対する回避法として、Leurent は、送るチャレンジを ASCII 図形文字に限定し、ASCII 図形文字以外を含む文字列が送られてきたときには、クライアントは、その時点で、認証手続きを強制終了するように提案している。

(2) 攻撃アルゴリズムに対する検討事項

Sasaki らの論文では、以下の項目が検討されていない、もしくは検討が不十分であった。

1. 攻撃の大前提として、攻撃者は、メールサーバに成りすましをしなくてはいけない。しかも、今回の攻撃の場合は、瞬間的に盗聴を行えば成功する類の攻撃ではなく、継続的に、成りすましをしなくてはいけない。さらに、このユーザは、アプリケーションとしてメールのみを使っているとは限らず、http などの各種プロ

トコルを使っていると考えるのが妥当である。このような状況で、攻撃者は、クライアントに気づかれることなく、攻撃を継続的に続けることは可能であろうか？また、その逆に、ユーザからみて、攻撃が仕掛けられていることは検出できないであろうか？この点をクリアにする必要がある。そのため、実際にネットワークを構築し、攻撃が成功するかの確認を行う必要がある。

2. ハッシュの衝突を実時間で発見することは可能であろうか？また、どの程度の計算能力のある計算機であれば、遅延なく、衝突を発見し、攻撃に利用できるであろうか？論文中では、可能であると記述されているが、実際に実験を行うことによりこの主張の正当性を検証する。
3. いくつかの APOP に対応した実際のメーラーが、この攻撃に対してどのような振る舞いをするかを検証する必要がある。APOP の仕様は、RFC 1939 により規定されているが、必ずしも、全てのメーラーが仕様を遵守している保証はなく、仕様の曖昧さも少なからずある。そのため、実際のメーラーの挙動、具体的には、チャレンジとして、どの文字が送られてきたときに、エラーが生じるかを実際に検証する。
4. パスワードに対する攻撃として、辞書攻撃が有効であることは良く知られている。Sasaki らの攻撃と辞書攻撃を併用する、すなわち、パスワードの次の文字の推定に辞書を用いる場合の有効性を検証する。さらに、その場合の攻撃に成功するまでの実時間の測定を行う。

以上の項目について、詳細に検討、実証を行うことが重要である。

２.１.３. まとめと今後の課題

以上のように、攻撃者は、ネットワークの構築さえ出来てしまえばユーザに気づかれることなく、APOP のパスワードを入手することが実時間で可能である。そのため、現実の脅威として認識する必要がある。パスワードはいろいろな仕組みで共有している場合（特に、オンラインバンクなど）は、単に、メールの内容を読まれてしまうだけでなく、経済的な被害も生じる。

そのため、早急にこの攻撃に対する対策法を検討しなくてはならない。これまでに提案されている対策は、クライアント側のチェックにより、チャレンジが ASCII 図形文字以外であれば、不正なチャレンジであると判定し、それ以降の処理を中断することである。これまでの Sasaki らの攻撃では、常に ASCII 図形文字になるようなチャレンジを生成することができないため、一時的な対策としては、機能する。しかし、今後の研究の発展を考慮すると、これも盤石な対策とはいいがたい。

次に考えられる対策は、レスポンスの生成時に、チャレンジとパスワードの順番を逆

にすることである。APOP の仕様では、レスポンスは、MD5(Challenge||pw)として成しているが、チェレンジとパスワードの順番を入れ替え、MD5(pw||Challenge) とすることで攻撃の回避が可能であるかもしれない。実際、Sasaki らの攻撃では、パスワードが後半に来ることが必須である。そのため、この対処法は、有効となりうる。しかしながら、Wang らは、MD5 ではなく、MD4 に対してではあるが、パスワードが前半部であっても、攻撃が成功することを示している[WOK08]。そのため、この対処法も、完全ではない。

以上のように、いくつかの容易に想像しうる対処法はあるが、いずれも今後の研究の進展を考慮に入れると、本質的に有効であるとは言い難い。そのため、さらに有効な攻撃の回避法を検討することが急務の研究課題である。

２.１.４. 参考文献

[BB93] Bert den Boer and Antoon Bosselaers, “Collisions for the Compression Function of MD5,” in Proc. of EUROCRYPT1993, LNCS 765, pp.293-304, 1994.

[DL05] Magnus Daum and Stefan Lucks, “Hash Collisions (The Poisoned Message Attack) The Story of Alice and her Boss,” Presented at the rump session of Eurocrypt2005.
http://th.informatik.uni-mannheim.de/people/lucks/HashCollisions/.

[Dobbertin96-1] Hans Dobbertin, “Cryptanalysis of MD5 compress,” Announcement at the Rump session of Eyrocrypt’96, 1996.

[Dobbertin96-2] Hans Dobbertin, “The Status of MD5 After a Recent Attack,” CryptoBytes The technical newsletter of RSA Laboratories, a division of RSA Data Security, Inc. VOLUME 2, NUMBER 2 - SUMMER 1996, 1996.

[GIS06] Max Gebhardt, Georg Illies and Werner Schindler, “A note on the practical value of single hash collisions for special file formats,” In Jana Dittmann, editor, Sicherheit, LNI 77, pp.333-344, GI, 2006.

[Handbook04] 電子情報通信学会編, ”情報セキュリティハンドブック,” オーム社, 2004.

[Klima05] Vlastimil Klima, “Finding MD5 Collisions on a notebook PC using multi-message modifications,” International Scientific Conference Security and Protection of Information, 2005.

“Tunnels in Hash Functions: MD5 Collisions Within a Minute,” Cryptology ePrint Archive, Report 2006/105.
http://eprint.iacr.org/2006/105.pdf.

[LW05] Arjen K. Lenstra and Benne de Weger, “On the possibility of constructing meaningful hash collisions for public keys,” Information Security and Privacy,

10th Australasian Conference - ACISP 2005, LNCS 3574, pp.267-279, Springer-Verlag, 2005.

[LWW05] Arjen Lenstra, Xiaoyun Wang, Benne de Wager, “Colliding X. 509 Certificated based on MD5-Collisions,” 2005年3月1日.
http://www.win.tue.nl/~bdeweger/CollidingCertificates/

[Leurent07] Gaetan Leurent, “Message Freedom in MD4 and MD5 Collisions: Application to APOP,” Fast Software Encryption - FSE 2007, LNCS 4593, pp.309-328, Springer-Verlag, 2007.

[Rivest91] Ron Rivest, “The MD4 message digest algorithm,” in Proc. of CRYPTO'90, LNCS537, pp. 303-311, 1991.

[Rivest92] Ronald L. Rivest, “The MD5 Message Digest Algorithm,” RFC 1321, April 1992.
ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt.

[PO96] Bart Preneel and Paul C. van Oorschot, “On the Security of Two MAC Algorithms,” in Proc. of EUROCRYPT'96, LNCS 1070, pp.19-32, 1996.

[Myers96] J. Myers, M. Rose, ““Post Office Protocol - Version 3,” RFC 1939, May 1996.
http://tools.ietf.org/html/rfc1939

[SNKO05] Yu Sasaki, Yusuke Naito, Noboru Kunihiro, and Kazuo Ohta, “Improved collision attack on MD5,” (IACR Cryptology ePrint Archive: Report 2005/400 http://eprint.iacr.org/2005/400), 2005.

[SNKO07] Yu Sasaki, Yusuke Naito, Noboru Kunihiro and Kazuo Ohta, “Improved Collision Attacks on MD4 and MD5,” IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E90-A No.1, pp.36-47, Jan 2007 (The initial result was announced as [SNKO05]).

[SYA07] Yu Sasaki, Go Yamamoto, and Kazumaro Aoki, “Practical Password Recovery on an MD5 Challenge and Response,” Cryptology ePrint Archive, Report 2007/101.

[SWOK07] Yu Sasaki, Lei Wang, Kazuo Ohta, Noboru Kunihiro, “Extended APOP Password Recovery Attack,” in the presentation of rump session of FSE2007.

[SWOK08-1] 佐々木悠, 王磊, 太田和夫, 國廣昇, “MD5チャレンジ・レスポンス方式の安全性について: APOPパスワード復元攻撃の拡張,” SCIS2008, 3A3-1, 2008.

[SWOK08-2] Yu Sasaki, Lei Wang, Kazuo Ohta and Noboru Kunihiro, “Security of MD5 Challenge and Response: Extension of APOP Password Recovery Attack,” to appear in CT-RSA2008.

[SLW07] Marc Stevens, Arjen Lenstra and Benne de Weger, “Chosen-prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities,” in Proc.

of EUROCRYPT2007, LNCS 4515, pp.1-12, 2007.

[WOK08] Lei Wang, Kazuo Ohta and Noboru Kunihiro, “Password Recovery Attack on Authentication Protocol MD4(Password||Challenge),” To appear in ASIACCS2008.

[WL04] Xiaoyun Wang, Xuejia Lai (& Dengguo Feng, Hongbo Yu), “Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD,” in the presentation of Rump Session of CRYPTO2004.

[WY05] Xiaoyun Wang and Hongbo Yu, “How to Break MD5 and Other Hash Functions,” in Proc. of EUROCRYPT 2005, LNCS 3494, pp.19-36, 2005.

２.２. 実証モデルの検討

２.２.１. 実証システムを構築する上での課題と対策

(1) なりすまされた通信の実現

なりすまされた通信を実現するために下記の機能を実現することが必要である。

(a) 攻撃者による偽装メールサーバの構築

攻撃者は、POP サーバと同様の動作をする攻撃プログラムを組み込んだ攻撃者のメールサーバ（攻撃サーバ）を用意する。いくつかの POP サーバはソースコードが公開されているので、それに攻撃プログラムを組み込むことが可能である。

(b) クライアントの攻撃サーバへの誘導

クライアントと正規のメールサーバ（正規サーバ）との間に、攻撃サーバと繋がったルータを挿入する。このルータのルーティングテーブルを制御することにより、クライアントからのパケットを正規のメールサーバと攻撃者のメールサーバに振り分け、本来の通信となりすまされた通信の振り分けを実現する。

(2) 実証データの取得

攻撃者はクライアントに対して正規のサーバになりすまし、クライアントと通信することによってパスワードを解読するが、クライアントやそれを利用するユーザに攻撃を受けていることを知られてはならない。従って、常になりすますことはできず、ある割合で偽装メールサーバ（攻撃サーバ）になりすます必要がある。攻撃を検知されないためには、この割合をできるだけ少なくすればよいが、本実証実験においては脆弱性の確認が主題であり、短期間での確認ができるように攻撃の割合を高くする。得られるデータから適切な割合で攻撃を行った場合の状況は推定が可能である。

以上を考慮して、具体的な実証データの取得について以下の機能により実現する。

(a) 攻撃周期の短縮化

攻撃者はクライアントからの認証要求を待たなければならない。実験時間の短縮のため、メールクライアントは最短の認証間隔を設定する。

(b) 複数のクライアントへの攻撃

脆弱性を確認するには、複数のパスワードに対してパスワードを解読する必要がある。実験時間の短縮のため、1種類のパスワードに対して1回の実験を行うのではなく、複数種のパスワードを設定した複数台のクライアントを使用して、同時に解読を行う。

また、限られた計算機環境で複数台のクライアントを実現するために仮想マシンを使用する。

２.２.２. 機能ブロック図と各ブロックの役割、必要なツール

実証実験を行う上で必要となる機能要素は、クライアント、メールサーバ、ルータ、攻撃者である。これらはそれぞれ表 ２.２-１のように被攻撃メールクライアント、正規のメールサーバ、なりすましを実現するルータ、攻撃者のメールサーバと位置づけられる。各々は図 ２.２-１のように接続する。

表 ２.２-１　機能ブロック

コンピュータ名	機能ブロック	主な機能
クライアント	被攻撃メールクライアント	APOP 対応メールクライアント
正規サーバ	正規のメールサーバ	POP サーバ
ルータ	なりすましを実現するルータ	ルーティング設定
攻撃サーバ	攻撃者のメールサーバ	攻撃プログラム

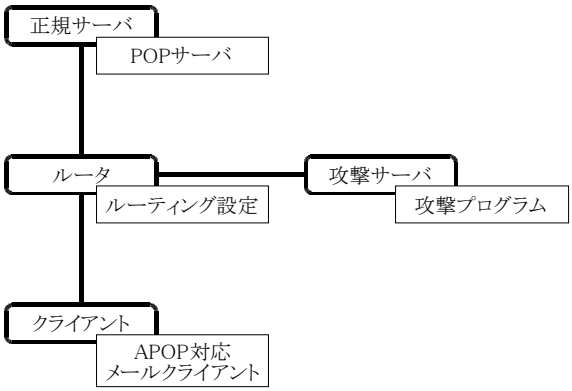

図 ２.２-１　各機能ブロックの接続図

２.２.３. 実験内容

目的の実証データを取得するために、3 種類の実験を行う。

(a) 実験１：なりすまされた通信を使用したパスワードの解読

実証実験システムにおいて、クライアントと正規サーバが正常に通信できる環境から、攻撃システムを作動させ、パスワード解読を行う。攻撃サーバとの認証回数や、接続からチャレンジが送られてくるまでの応答時間を確認する。

(b) 実験 2 : なりすましによって新着メールの到着が遅れる事象の確認

クライアントがなりすまされた通信をしている間に、正規サーバに新しくメールが到着した場合、そのメールはすぐに受信できない。本来新着メールがあるはずの認証において、攻撃サーバと通信していたために受信ができなかった、という事象がどの程度発生するかを確認する。クライアントへの新着メールは、正規サーバ内で疑似的に生成する。

(c) 実験 3: APOP 対応メールクライアントが攻撃サーバと通信したときの挙動の調査

APOP 対応メールクライアントが攻撃サーバと通信したときの挙動と、正規サーバと通信したときの挙動の差異を確認する。

２.２.４. 具体的なパラメータ

(1) 実験 1

実験 1 において、下記をパラメータとして変化させることで MD5 の脆弱性の限界を検証する。

(a) 解読するパスワードの文字数
(b) パスワードに使用する文字の種類
(c) 攻撃頻度（なりすましの割合）
(d) 辞書利用攻撃の有無

実証実験の具体的に選定したパラメータを表 ２.２-２に示す。表の項目は実験 1 の中での実験番号である。パスワードを先頭から 1 文字ずつ特定をするという解読方法から、パスワード 8 文字の解読に関する情報は 12 文字の解読で得られる情報から取り出せるため、実際に行う実験は 12 文字の解読のみとする。

辞書攻撃なしの場合、パスワードの文字の推定方法は、ASCII 図形文字を昇順に巡る。

表の中で「-」とした箇所は、ASCII コードパスワードの文字種に関する区別はないため、辞書攻撃なしの攻撃では英数字と記号で構成されるパスワードに対する解読のみ実行する。

攻撃頻度は、1/2 と表記されているものは、例えば認証間隔を 1 分間とした場合、4 分間のうち 2 分間、1/4 のものは 8 分間のうち 2 分間攻撃システムを作動させ、攻撃を行うという意味である。

なお、各項目の結果のばらつきを吸収するため、各々の実験において 10 種類のパスワードに対して解読を行う。文字種に関しては同一のパスワードを使用する。

表 2.2-2　実験 1 のパラメータと実験番号

攻撃者側パラメータ	辞書攻撃		なし		あり	
	攻撃頻度		1/2	1/4	1/2	1/4
ユーザ側パラメータ	文字の種類	英数字	-	-	1-5	1-6
		英数字と記号	1-1	1-2	1-3	1-4

解読対象のパスワードを表 2.2-3 に示す。パスワード 1 からパスワード 5 は人手により作成したもの、パスワード 6 からパスワード 10 はランダムに生成したものである。パスワード 1 は安易なものを、パスワード 2 は単語の間に文字を挿入したものを、パスワード 3 は日常使われている単語との組み合わせで作成した。

表 2.2-3　解読対象のパスワード

パスワード番号	英数字と記号	英数字のみ
パスワード 1	1!2"3#4$5%6&	password0123
パスワード 2	C-l_i~e=n|t?	q1w2e3r4t5y6
パスワード 3	sato@ice.uec	Blue1997Deep
パスワード 4	^5+er|5ked**	BObutarou119
パスワード 5	!Psyvariar2R	AABCgnOO5ABG
パスワード 6	Xbo#G_QN8VFK	OMwPVtpJfHye
パスワード 7	dLa~HfhwRTBr	uoyU6c3ZDSLg
パスワード 8	Q6|/VgTz9;[=	EcgBsvf384JQ
パスワード 9	n&<$9}fMm-,4	4FlfjSzPHDkN
パスワード 10	(OPeQS'wko@>	OAe3Kd9mblnE

(2) 実験 2

実験 2 において、クライアントへの新着メールは、最大 30 分のランダムな間隔で生成される。また、新着メールは Windows を搭載したクライアントのアカウントへ送信する。実験 1 で関係するパラメータは攻撃頻度である。

(3) 実験 3

実験 3 において、APOP 対応クライアントは Windows を搭載したコンピュータ上で実行する。また、ルータは常にクライアントを攻撃者と通信させるように動作させる。

２.２.５. 取得データ

各々の実験において取得するデータをそれぞれ表 ２.２-４、表 ２.２-５、表 ２.２-６に示す。

表 ２.２-４　実験１において取得するデータ

取得するコンピュータ	取得内容	確認事項
ルータ	すべての通信内容	通信内容の変化
攻撃サーバ	攻撃プログラムのログ	攻撃プログラムの挙動

表 ２.２-５　実験２において取得するデータ

取得するコンピュータ	取得内容	確認事項
ルータ	接続先の切り替えログ	クライアントの通信相手
正規サーバ	新着メール生成ログ	新着メールが送信された時刻
	POP サーバログ	クライアントの受信状況

表 ２.２-６　実験３において取得するデータ

取得するコンピュータ	取得内容	確認事項
ルータ	すべての通信内容	通信内容
クライアント	エラー画面	エラーなどが起きた時の情報

２.２.６. 実験手順

(a) 開始手順

1. ルータ：クライアントと正規サーバが通信できるようにルーティングテーブルを書き換え
2. 正規サーバ：POP サーバ開始
3. クライアント：メールクライアント起動（自動で送受信）
4. 正規サーバ：クラアイントが定期的に受信しているかを確認
5. クライアント：パケットキャプチャプログラム起動、キャプチャ開始
6. ルータ：パケットキャプチャプログラム起動、キャプチャ開始
7. 攻撃サーバ：攻撃プログラム開始
8. ルータ：ルーティング切り替え開始

(b) 動作監視中

1. 攻撃サーバのログを監視、解読完了したクライアントを停止

2. 停止するクライアントが Windows マシンなら、メールクライアントとパケットキャプチャプログラム、正規サーバのメール配信をすべて停止
3. 停止するクライアントが CentOS 上の仮想マシンなら、メールクライアントと仮想マシンを停止

（c）解読完了後

1. ルータのパケットキャプチャプログラム停止
2. 攻撃サーバの攻撃プログラム停止
3. （正規サーバはそのまま）

２.３. 実証結果の分析

後述の第３章の実験結果に示しているように、設定した全ての実験パターンにおいて、ごく一般的な PC を用いてパスワードを解読できることがわかった。

また、その解読に要する時間も、一部処理上（アルゴリズム）の問題、実験システム上（ルータの切り替え手順）の問題等が発生しているため若干異なる部分もあるが、ほぼ想定時間以内に収まることが実証された。

以下、実験結果についての分析を示す（分析の元となる詳細なデータは第３章を参照）。

２.３.１. 危険性のレベル

今回の実証実験にあたっては、現実的な時間以内に実行するため加速環境で実験を実施した。

具体的には、

・クライアントのメール到着確認が 1 分間おきに定期的に実行されている

・クライアントのメール到着確認を 2 回に 1 回ないし 4 回に 1 回攻撃者がハッキングする

との条件下で実施した。

現実的には、

・定期的に実行されるメール到着確認は 30 分に 1 回程度である[1]

・攻撃者がハッキングする周期については、ハッキングが行われていても現実の主なクライアントのメールソフトウェアの挙動では特に異常を検出できないこともあり、上記の条件下でも現実的と想定される

であることを考慮すると、攻撃者がパスワードを解読できる認証回数は今回の実験の約 30 倍程度となるものと推定される。

実験結果（図 ２.３-１、図 ２.３-２：図 ３.３-１、図 ３.３-２の再掲）が示すパスワード解読を完了するために必要とした認証回数が、パスワード長が 12 文字の場合でも 1,000 回前後であることから、現実の使用環境においては 60,000 分ないし 120,000 分、即ち約 1,000 時間～2,000 時間（41 日～83 日）程度であることを示している。

このことは、かなり頻繁にパスワードを定期的に変更しない限り、脆弱性の攻撃により、パスワードをクラッキングされる危険性を示している。

---

[1] 主要なメールクライアントソフトウェアを調査した結果、メール到着確認周期のデフォルト設定では 30 分を設定しているものが多い。尚、Sylpheed のように 10 分と短い値に設定されているものもある。

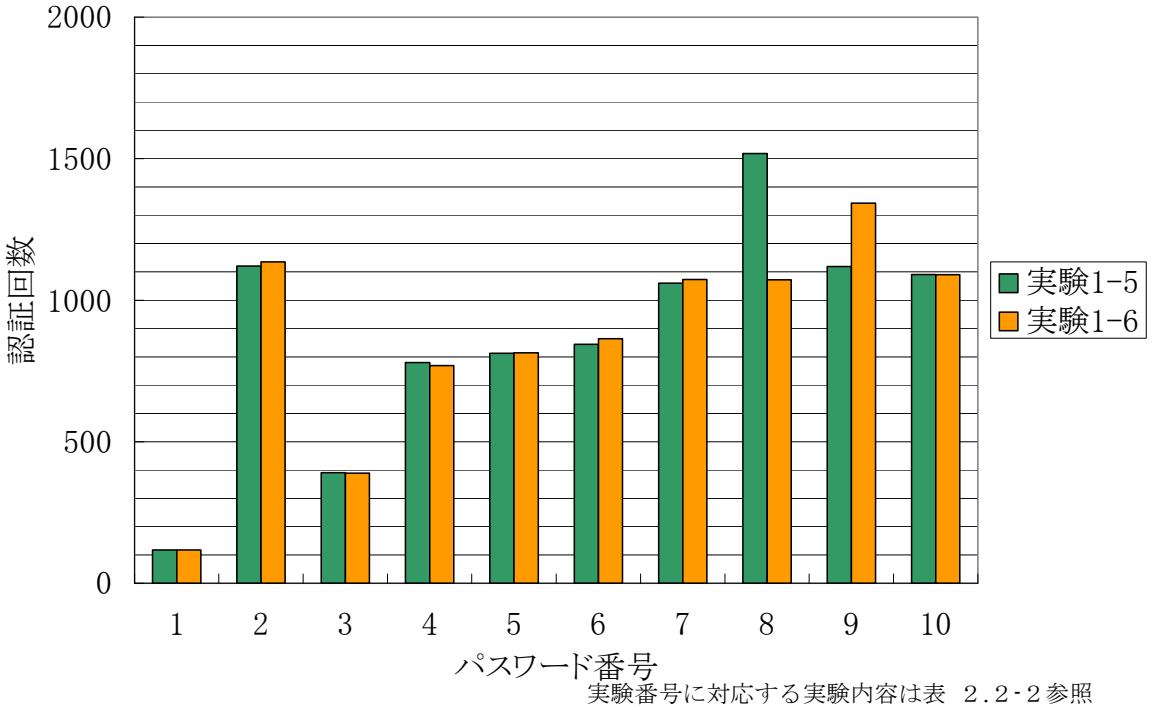

図　2.3-1　攻撃サーバとの認証回数（英数字のみ）

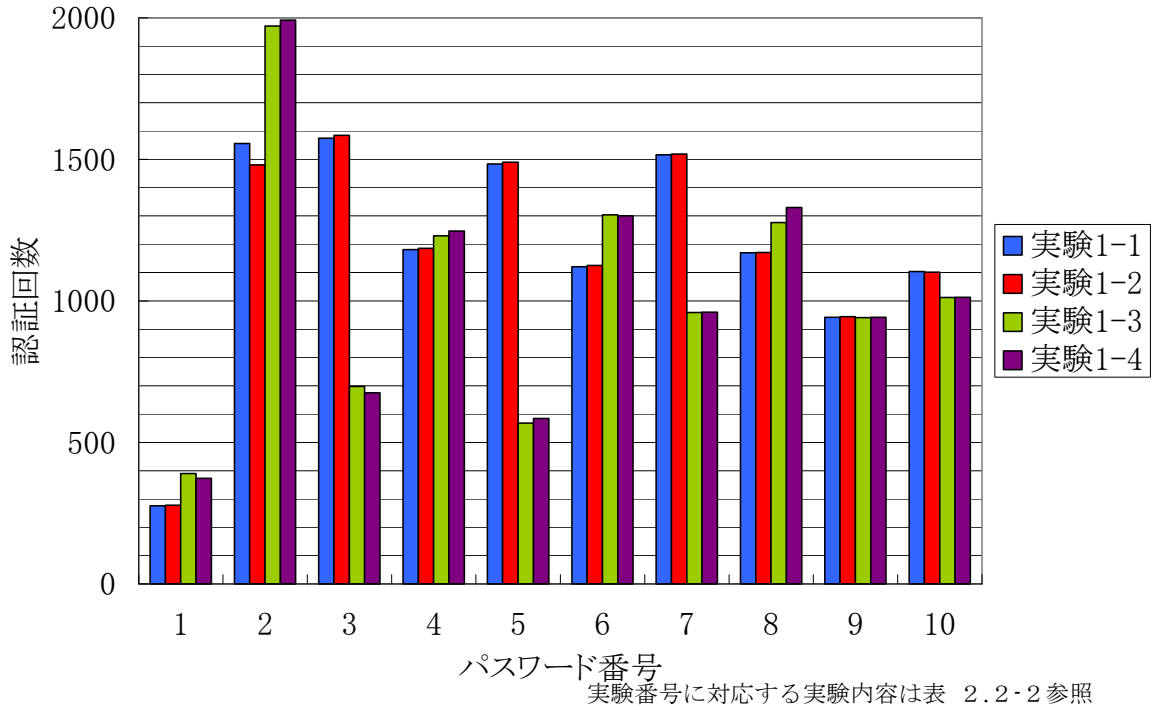

図　2.3-2　攻撃サーバとの認証回数（英数字と記号）

２.３.２. パスワードのあり方

今回の実験においては、攻撃者におけるパスワード推定ロジックとして、インターネットで流通しているパスワード辞書（パスワードクラッキングや運用管理者が管理対象でのパスワードの脆弱性の有無の確認に通常使用するもの）を利用した場合の効果についても確認している（図 ２.３-３、図 ２.３-４）。

その結果、実験に用いた 10 種類のパスワード（表 ２.３-１：表 ２.２-３の再掲）のうち、パスワード 2 のように 1 文字おきに記号が挿入され、かつそれ以外の英数部分も含めて意味のある単語の痕跡がなくランダムに組み合わせられている場合には比較強度が高いことを示している（図 ２.３-４）。逆に、英数字と記号を組み合わせてはいるものの、意味のある単語の痕跡が残るパスワード 3、パスワード 5 では強度の低い顕著な例となっている（図 ２.３-３）。

また、攻撃者が辞書攻撃を用いると、安易なパスワード 1 の例では、24 回のみの攻撃者との APOP 認証によりパスワードが解読されてしまうこともわかった。

今回の実験結果からどのようなパスワードが適当（攻撃に強い）かは一概には言えないし、攻撃者側でのパスワード推測ロジックにかなり依存する部分もある。従来から英数字だけでなく記号をパスワードに含めることが推奨されていたが、単に含めるだけでは効果がなく、記号以外の英数字部分に意味のある単語の痕跡が残る状況では、解読に要する認証回数が大きく減少しており、強度的に強くないことが示された。

一方、比較的強度が強いことが示されたパスワードは、英数字と記号がかなりランダムな意味のない状況で交互に組み合わされたものであるが、現実的にこの種のパスワードを利用者が記憶し利用するのは困難（機械的に記憶している場合を除く）かもしれない。

表 ２.３-１　解読対象のパスワード

パスワード番号	英数字と記号	英数字
パスワード 1	1!2"3#4$5%6&	password0123
パスワード 2	C-l_i~e=n|t?	q1w2e3r4t5y6
パスワード 3	sato@ice.uec	Blue1997Deep
パスワード 4	^5+er|5ked**	BObutarou119
パスワード 5	!Psyvariar2R	AABCgnOO5ABG
パスワード 6	Xbo#G_QN8VFK	OMwPVtpJfHye
パスワード 7	dLa~HfhwRTBr	uoyU6c3ZDSLg
パスワード 8	Q6|/VgTz9;[=	EcgBsvf384JQ
パスワード 9	n&<$9}fMm-,4	4F|fjSzPHDkN
パスワード 10	(OPeQS'wko@>	OAe3Kd9mb|nE

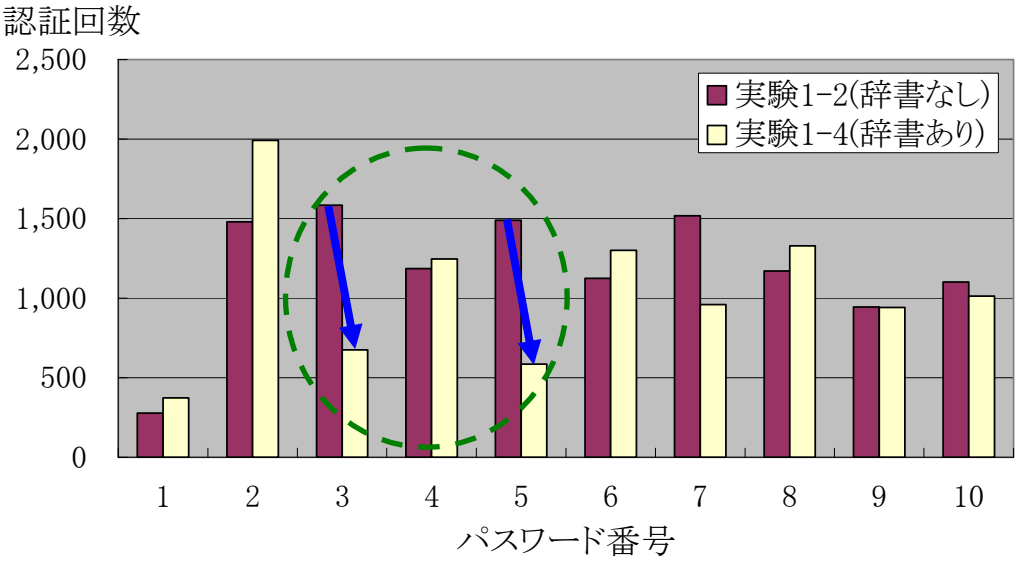

図　2.3-3　パスワード辞書攻撃の評価

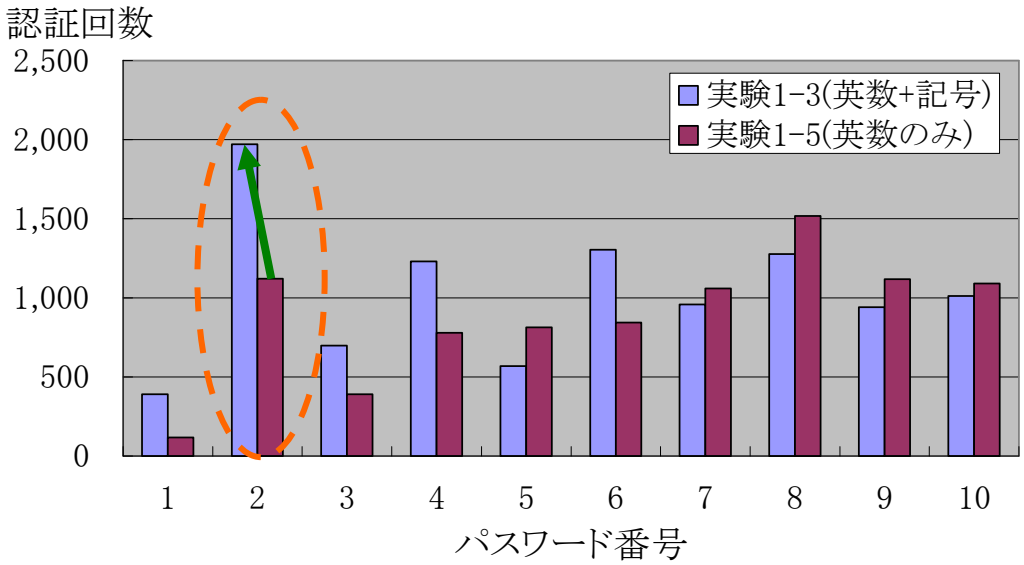

図　2.3-4　パスワードにおける記号の評価

2.3.3. クライアントでの認知状況

攻撃を受けている状況がクライアントにどのように見えるかについてメールサーバとの通信におけるサーバの応答時間、新着メールの到達時刻、メールクライアントソフトウェアでの異常検出の観点から確認した。

(a) サーバの応答時間

通常の PC を用いた実験環境においても、攻撃を受けている場合と攻撃を受けていない場合とで、サーバの応答時間が極端に延びることはなく、増加分は NW における遅延時間と同程度のレベルであり、タイムアウトを起こす時間でもないため、認知することは困難と考えられる。

(b) 新着メール到達時刻

攻撃者によって偽装メールサーバに接続された場合には、本来その時刻までに受信しておりクライアントに送付されるはずのメールが送付されず、次回の正規メールサーバに接続できた際に送付されることになる。即ち、メールサーバへのメール到達時刻を確認すると、クライアントでメール到着確認時刻との関係で矛盾があることがわかる。この状況は今回の実験においても明らかになっている。

このような矛盾が頻発（特に定期的に）する場合、攻撃を受けている可能性が高いと認知できる。この場合、サーバとクライアントの絶対時刻が一致していなければ判定が難しいため、厳密な認知は困難かもしれない。

(c) メールクライアントソフトウェアでの異常検出

現在、攻撃者が生成できるチャレンジ文字列は、RFC で定められている書式を厳密に満たすことは困難である。従って、この文字列を厳密にチェックすることで攻撃を受けている可能性をチェックすることが可能である。

今回の実験においては主要なメールクライアントソフトウェアを用いて確認を行った。その結果、フリーソフトウェアとして流通している 2 種類のソフトウェアに関して、エラーメッセージが表示されることを確認した。

このような対策を講じているメールクライアントソフトウェアを利用することで、さしあたりの攻撃を回避することが可能である。

なお、ここで用いられているチェック手法は“msg-id”[2]に準拠したチャレンジ文字列を生成することができないという点を利用している。将来“msg-id”に準拠したチャレンジ文字列を用いて攻撃が可能となった場合にはこの攻撃回避方法は無効になるため、他の対策について今後検討する必要がある。

---

[2] POP3 を規定している RFC1939 には、“A POP3 server which implements the APOP command will include a timestamp in its banner greeting. The syntax of the timestamp corresponds to the ‘msg-id’ in [RFC822], and MUST be different each time the POP3 server issues a banner greeting.　For example, on a UNIX implementation in which a separate UNIX process is used for each instance of a POP3 server, the syntax of the timestamp might be: <process-ID.clock@hostname>”とあり、RFC822（最新版は RFC2822）に準拠した簡単な例が示されている。

２.３.４. 再検証の必要性

今回行った実験では下記の点を重視した。

・実際の環境に近い環境を攻撃対象にする

・攻撃システムは攻撃を検出されないように正規サーバになりすます

・攻撃システムは市販のコンピュータで実現する

実際に悪意のある者が攻撃を行う方法はこれに極めて近いと考えられるため、実験実証システムの構築方法に関しては十分である。

実験内容に関しては実験時間の都合上、パスワード 12 文字となった。今回実装した攻撃手法では 31 文字まで解読できることから、より長いパスワードに対しても確認を行う必要がある。

２.３.５. 残された課題

攻撃サーバとの認証回数が理論値を超える結果となっている。これは、クライアントが各サーバと行う POP セッションとは無関係にルータが接続先を切り替える方式での実装を行ったためである。

POP 認証のセッション実行中、非実行中を認識して切り替える等、攻撃者のルータがより高度にパケットを操作することができれば、このようなロスは少なくなると考えられる。

２.４. 影響範囲の調査

本調査においては、電子メールクライアントソフトウェアとして APOP 方式を採用しているシステムないしソフトウェア、並びに電子メールサービスに APOP 方式をサポートしているプロバイダについて調査し、APOP に係る脆弱性による脅威を受ける可能性の影響範囲を明らかにした。

２.４.１. APOP 方式を採用している電子メールクライアントソフトウェアの調査

(1) 調査手法

市場に流通している主要な電子メールクライアントソフトウェア並びにオープンソースとして広く利用されている主要な電子メールクライアントソフトウェアについて、その仕様をインターネット検索等により確認し、APOP のサポート状況及び APOP の脆弱性に関する警告あるいは対応の状況等について調査した。

(2) 調査結果

(a) 調査対象とした電子メールクライアントソフトウェア

パッケージソフトウェアとして販売、提供（シェアウェア等も含む）されているもの、フリーソフトウェアとして公開されているもの、OS 等にバンドルされているものの内、Windows の OS で広く利用されている下記の 16 の電子メールクライアントソフトウェアを調査対象として選択した。

① Outlook Express
② Outlook
③ Windows Mail
④ Windows Live Mail
⑤ Netscape
⑥ Becky! InternetMail
⑦ AL-Mail32
⑧ Thunderbird
⑨ Sylpheed
⑩ Shuriken
⑪ Winbiff with EditX
⑫ WeMail32
⑬ Eudora

⑭ 秀丸メール
⑮ 電信八号
⑯ Yosaku

(b) 調査の項目

調査対象とした電子メールクライアントソフトウェアについて、下記の項目について詳細に調査した。

① バージョン
② 最新更新日
③ APOP 対応の有無
④ APOP 脆弱性への警告等の有無
⑤ 脆弱性対策

(3) 調査結果

調査した結果を表 ２.４-１に示す。

表 2.4-1　主要な電子メールクライアントソフトウェアの APOP に関する対応状況

APOP 対応	メールソフト	バージョン	最新更新日	脆弱性への警告等	脆弱性対策	備考	参考（リンク情報等）	（参考）実証実験結果
	Outlook Express	6		―				―
	Outlook	2007		―				―
	Windows Mail	―		―			http://www.microsoft.com/japan/office/2007/Outlook/mailappscomp.mspx	―
∨	Windows Live Mail	β		―		Microsoft 初の APOP 対応	http://www.microsoft.com/japan/office/2007/Outlook/mailappscomp.mspx	―
	Netscape	7.1		―			http://home.jp.netscape.com/ja/	―
∨	Becky! InternetMail	2.42	2007.10.31	×			http://www.rimarts.co.jp/becky-j.htm	アタック成功
∨	AL-Mail32	1.13	2006.1.30	×			http://www.almail.com/	アタック成功
∨	Thunderbird	2.0.0.9	2007.10.18	○	http://www.mozilla-japan.org/security/announce/2007/mfsa2007-15.html	Ver.2.0.0.4(2007.5.30)で修正済	http://www.mozilla-japan.org/products/thunderbird/	認証失敗の警告
∨	Sylpheed	2.4.7	2007.10.4	○	http://sylpheed.sraoss.jp/ja/news.html	Ver.2.4.0(2007.4.20)で修正済	http://sylpheed.sraoss.jp	認証失敗の警告

∨	Shuriken	2007	2007.2.6	×			https://www.justmyshop.com/app/servlet/c6	アタック成功
∨	Winbiff with EditX	2.51PL1	2007.11.9	×		Mobile用pasomailやxGateもAPOP対応	http://www.orangesoft.co.jp/	アタック成功
∨	WeMail32	2.52	2007.6.12	×			http://www.ntes.co.jp/WeMail/	アタック成功
∨	Eudora	7J	2006.7.14	×		2007.10.1商用版の販売を終了		アタック成功
∨	秀丸メール	4.83	2007.11.5	×			http://hide.maruo.co.jp/software/tk.html	アタック成功
∨	電信八号	32.1.6.1	2007.6.6	×			http://denshin8.esprix.net/	アタック成功
∨	Yosaku	1.32	2006.9.10	×			http://donko.homeip.net/	強制終了

平成19年11月15日現在

(4) 分析

OS 等とバンドルされ提供されている一般的な電子メールクライアントソフトウェア（Outlook 系等）等については APOP 対応を行っていないが、その他の主要なソフトウェアは APOP に対応している。

しかしながら、APOP の脆弱性に対しての明確なアナウンス（警告）を行っているソフトウェアはほとんど見当たらず、フリーのソフトウェアとして提供されている Thunderbird 並びに Sylpheed が脆弱性に対する警告、並びに、ある程度の対策を施したバージョンを既に提供済みである。

このことは、インターネットコミュニティー内において、メジャーなフリーソフトウェアあるいはオープンソースソフトウェアの柔軟性、関係者のコントリビューションの大きさを象徴するものと考えられる。

尚、当該ソフトウェアにおける具体的な対策内容としては、チャレンジワード内に不適切なコードが含まれているようなケースをチェックアウトする等によりアタックに対して防御を行っている模様である。

(5) まとめ

主要な電子メールクライアントソフトウェアについて、APOP 方式をサポートしているか否か､APOP の脆弱性に対する警告ないし対応をしているか等の調査を行った。

全体として、利用者数が最も多いと思われる Outlook 系については APOP サポートではないため本問題に関する影響の数値的な度合いは比較的大きくない。

しかしながら、APOP を利用するユーザに的を絞った場合、脆弱性に対する対処を考慮しているソフトウェアが少ないことから、問題はかなり深刻であると判断すべきである。本件調査結果を早期に公表し、再度の警告を行うと共に、可能な対処策を早急に講じることが必要である。

２.４.２. APOP 方式をサポートしているプロバイダに関する調査

(1) 調査手法

主要なプロバイダが提供している電子メールサービスについて、そのサービス内容をインターネット検索等により確認し、APOP のサポート状況及び APOP の脆弱性に関する警告あるいは対応の状況等について調査した。

(2) 調査内容

(a) 調査対象としたプロバイダ

電子メールサービスを提供している主要なプロバイダの中から、下記の 19 のプロバイダを調査対象として選択した。

① au one net
② So-net
③ GyaO 光
④ BIGLOBE
⑤ OCN
⑥ AOL
⑦ @nifty
⑧ ODN
⑨ hi-ho
⑩ ASAHI ネット
⑪ DTI
⑫ ぷらら
⑬ Yahoo!BB
⑭ WAKWAK
⑮ Highway Internet
⑯ IIJ インターネットサービス
⑰ Info Sphere
⑱ @T COM
⑲ eo 光

(b) 調査の項目

調査対象としたプロバイダについて、下記の項目について詳細に調査した。

① 運営組織
② APOP 対応
③ APOP 脆弱性への警告等
④ 広報日
⑤ 脆弱性対策

(3) 調査結果

調査した結果を表 ２.４-２に示す。

表 2.4-2　主要なプロバイダの APOP に関する対応状況

APOP 対応	プロバイダ	運営組織	脆弱性警告等	広報日	脆弱性対策	備　　考	参考（リンク情報等）
V	au one net	KDDI	×			註：メールにて確認	http://cs119.kddi.com/auone-net/faq.jsp?faqno=DSE01014
V	So-net	ソネットエンタテイメント	×				http://www.so-net.ne.jp/support/qa/ans/s1000/s1004.html
	GyaO 光	USEN	×			註：電話にて確認	http://hikari.gyao.ne.jp/service/flets_faq_service.html#09
V	BIGLOBE	NEC ビッグローブ	×				http://support.biglobe.ne.jp/settei/mailer/apop.html
V	OCN	NTT コミュニケーションズ	○	2007.4.19	http://www.security.ocn.ne.jp/information/news/nf20070419_01.html	パスワードの定期的変更	http://www.ocn.ne.jp/mailon/menu/p-2b.html
	AOL	eAccess	×			POP 対応なし（IMAP のみ）	http://support.aol.co.jp/manual/mail/win_oe.html
V	@nifty	ニフティー	△	2007.4.19	http://clip.nifty.com/entry/954644de59590d575a4cbbc9aadf1b311ebe3487/73361	会員からのクリップ情報	http://support.nifty.com/support/faq/mail_qa/mail/mail_qa_ans24.htm?qt=APOP
	ODN	SoftBank Telecom	×			註：メールにて確認	http://www.odn.ne.jp/service/

	hi-ho	ハイホー	×			註：メールにて確認	http://home.hi-ho.ne.jp/home/support/
V	ASAHI ネット	朝日ネット	×				http://asahi-net.jp/support/guide/0555.html
V	DTI	ドリーム・トレイン・インターネット	×				http://dream.jp/mail/code.html
V	ぷらら	ぷららネットワークス	×				http://www.plala.or.jp/member/option_service/secuplus/secure/service.html
	Yahoo!BB	Yahoo!JAPAN	×			YAHOO!ウェブホスティングでは APOP 可能	https://ybb.softbank.jp/support/connect/index.php
V	WAKWAK	NTT ME	○	2007.5.11	http://www.wakwak.com/news/2007/hd/0511.html	パスワードの定期的変更	http://www.wakwak.com/option/index.html
	Highway Internet	USEN	×				http://www.highway.ne.jp/service/service-mail.html
V	IIJインターネットサービス	インターネットイニシアティブ	×				http://www.iij.ad.jp/service/system/IIJ-Mailbox-spec_857.html
V	Info Sphere	ＮＴＴＰＣ	×				http://customer.sphere.ne.jp/faq/mail/index.html#faq04
	@T COM	ビック東海	×				http://www.t-com.ne.jp/

V	eo 光	ケイオプティコム	○		http://eonet.jp/mail/apop/	「SSL による暗号化通信」を利用する「メール盗聴防止サービス」	http://cs.eonet.ne.jp/Contents/1287/W_1287.html

平成 19 年 11 月 30 日現在

また、調査対象としたプロバイダが利用者規模的にどの程度であるのか、別の意味で言うなら、APOP の脆弱性の問題の影響を受ける可能性のある利用者がどの程度であるのかを見極めるデータとして、直接の利用者数のデータが入手困難であったため、あるページ「便利ページ」への時々刻々のアクセスランキングを発表しているデータがあったので、それを参考として推定したものを表 2.4-3に示す。このデータは直接のプロバイダ規模を示すのもではないが、およその相対値的な規模データとしては利用できるものと考えている。

表 2.4-3　プロバイダ規模の目安

順位	プロバイダ	アクセス数	%	調査対象	APOP 対応
1	OCN	31,494	18.6%	V	○
2	Yahoo!BB	16,224	9.6%	V	
3	Plala	12,483	7.4%	V	○
4	DION (au one net)	10,747	6.3%	V	○
5	@nifty	10,307	6.1%	V	○
6	http://www.ucom.ne.jp/	9,822	5.8%		
7	ASAHI ネット	8,119	4.8%	V	○
8	So-net	6,666	3.9%	V	○
9	BIGLOBE	6,334	3.7%	V	○
10	@NetHome	5,885	3.5%		
11	eo	4,571	2.7%	V	○
12	vectant	3,899	2.3%		
13	WAKWAK	3,047	1.8%	V	○
14	InfoSphere	2,717	1.6%	V	○
15	DTI	2,334	1.4%	V	○
16	ODN	2,042	1.2%	V	
17	ＺＡＱ	1,552	0.9%		
18	Powered Internet	1,366	0.8%		
19	IIJ4U	1,261	0.7%	V	○
20	T-com ADSL	1,163	0.7%	V	○
21	http://www.pwd.ne.jp/	1,126	0.7%		
22	TNC	1,101	0.7%		
23	freebit	1,082	0.6%		
24	Hi-HO	774	0.5%	V	
25	豊橋ケーブルネットワーク	629	0.4%		

26	CatNet	556	0.3%		
27	bit-drive	553	0.3%		
28	eAccess	508	0.3%		
29	http://www.megaegg.ne.jp/	493	0.3%		
30	SANNET	492	0.3%		
31	KCN-Net	462	0.3%		
32	Fiberbit	450	0.3%		
33	TOKAI ネットワーククラブ	422	0.2%		
34	アイタイネット	421	0.2%		
35	Fenics	418	0.2%		
36	TIKITIKI	417	0.2%		
37	http://www.ccnw.ne.jp/	410	0.2%		
38	http://www.em-net.ne.jp/	377	0.2%		
39	http://www.ayu.ne.jp/	364	0.2%		
40	http://www.mecha.ne.jp/	364	0.2%		
41	KATCH CATV Internet	338	0.2%		
42	avis インターネット	260	0.2%		
43	トーカイハイウェイネット	259	0.2%		
44	豊島ケーブルネットワーク	249	0.1%		
45	NSK	245	0.1%		
46	http://www.mediatti.net/	245	0.1%		
47	http://www.cyberhome.ne.jp/	245	0.1%		
48	ユーキャット	244	0.1%		
49	北ケーブルネット	238	0.1%		
50	http://www.pikara.ne.jp/	236	0.1%		
総数（51 位以降も含む）		169,355			
全体の%		―	71.0%	59.1%	

V	：表１の調査対象
○	：上記対象中 APOP サポート

註：「便利ページ」へのアクセスランキングより作成　（平成 19 年 11 月 25 日～12 月 1 日）

http://city.cyberoz.net/~nagatuma/rnk/prv.html

(4) 分析

全体の 70%程度の利用者数を占めると考えられる主要な 19 のプロバイダを調査した結果、その 63%にあたる 12 のプロバイダで APOP をサポートしていることが判明した。このプロバイダは利用者規模的に見ておよそ 60%に相当すると考えられる。

即ち、全（日本国内）インターネット利用者の半数以上が APOP を利用できる環境下にあることが判明した。

その 12 のプロバイダの中で、APOP の脆弱性を明確にして、利用者に注意喚起を行っているプロバイダは 3 プロバイダ（約 25%）のみである。利用者規模的には約 23.1%に相当すると考えられる。

以上の関係を図　2.4-1 に示す。

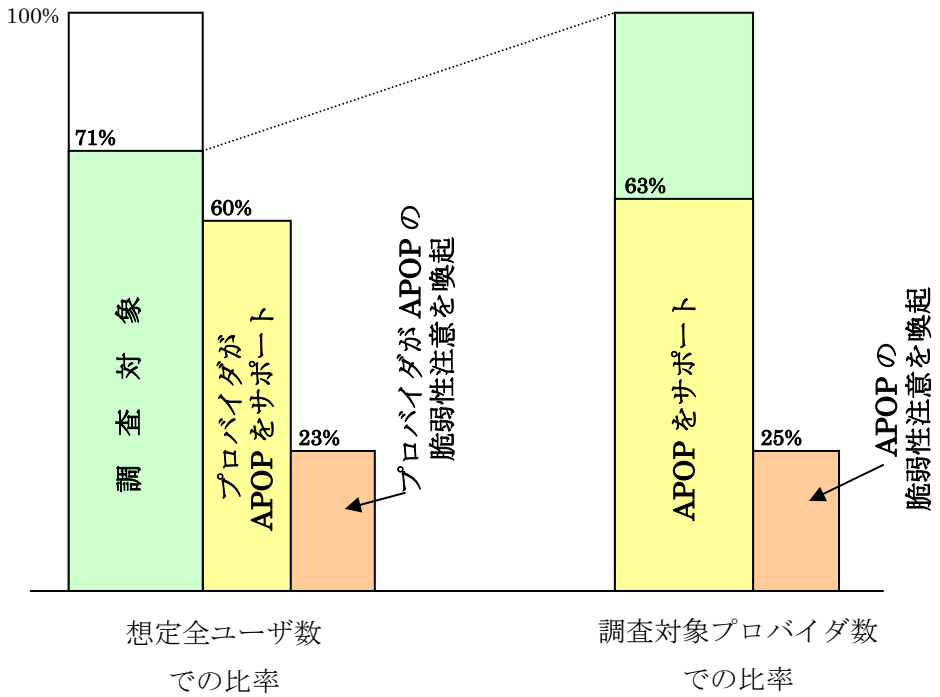

図　2.4-1　調査結果の概要

尚、APOP をサポートしているプロバイダに加入している利用者のうち、実際に APOP のサービスを利用している利用者数に関しては公表にされていないため、具体的な脅威の規模は不明である。

以上の結果、かなりの割合、規模的に APOP の脆弱性の影響下にあることから、認証方式の変更、アタック状況の検出等早急な対応策の実施が必要である。

(5) まとめ

主要なプロバイダが提供している電子メールサービスについて、APOP のサポート状況及び APOP の脆弱性に関する警告あるいは対応の状況等について調査を行った。

プロバイダ数的、利用者規模的共に半数以上が APOP の脆弱性の影響を受ける状況にあることが判明し、認証方式の変更、攻撃の有無の検出等の対応策の早急な実施が必要である。

２.５. 当面の対応策の検討

APOP における脆弱性の攻撃に対する対策としては、プロトコルの改善、メールサーバの改善、メールクライアントソフトウェアの改善等の側面がある。

２.５.１. プロトコル改善方法

APOP における脆弱性をプロトコルの側面から改善する手法を検討した。

今回の脆弱性は、レスポンス文字列を計算するハッシュ関数として MD5 を使用していることが問題であり、このハッシュ関数を SHA-2 などの他のものに変えれば攻撃を防ぐことができる。

しかし、将来に渡って同様の攻撃が他のハッシュ関数に対しても発見される可能性は否定できない。さらに、現在稼働しているすべてのメールサーバ及びメールクライアントが新しい認証方式に対応するまでにかかる歳月を考えると、プロトコル自体を修正する方法は短期的対策としては現実的ではない。

２.５.２. メールサーバ改善方法

APOP における脆弱性をメールサーバで改善する手法を検討した。

APOP の脆弱性を攻撃するには、攻撃者がクライアントに対して正規のメールサーバになりすます必要がある。これを防ぐには、サーバが SSL を用いた POP3S に対応すればよい。これによって、通信相手が正規のメールサーバであるかどうかを判定でき、なりすましを防止できる。

これは根本的な解決方法だが、ユーザはサービスプロバイダの対応を待たなければならない。その間、ユーザは脆弱性の脅威にさらされることになる。

２.５.３. メールクライアントの改善方法

(1) チャレンジ文字列の書式

POP3 の APOP 認証で用いられるチャレンジ文字列は RFC822 の“msg-id”という名前で書式が規定されている[3]。書式にはいくつか種類があるが、最も多く使用されている

[3] POP3 を規定している RFC1939 には、“A POP3 server which implements the APOP command will include a timestamp in its banner greeting. The syntax of the timestamp corresponds to the ‘msg-id’ in [RFC822], and MUST be different each time the POP3 server issues a banner greeting.　For example, on a UNIX implementation in which a separate UNIX process is used for each instance of a POP3 server, the syntax of the timestamp might be: <process-ID.clock@hostname>”とあり、RFC822（最新版は RFC2822）に準拠した簡単な例が示されている。

例を図 ２.５-１に示す。

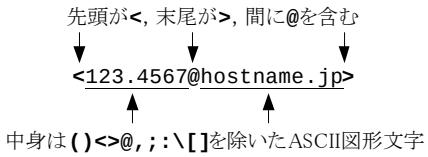

図 ２.５-１　チャレンジ文字列の書式の例

現在報告されている攻撃は、図 ２.５-１を含む書式を満たすようにチャレンジ文字列を生成することができない。

(2) チャレンジ文字列のチェックアルゴリズムの例

図 2.5-1 に示したように、チャレンジ文字列はすべて ASCII 図形文字で構成されているべきである。従って、チャレンジ文字列が書式を満たしているか厳密にチェックすれば、通信相手が攻撃者であるかを判断できる[4]。

チェックアルゴリズムの例を図 ２.５-２に示す。工程 2. がチェック行程である。

1. (サーバから送信されたメッセージからチャレンジ文字列を抽出する)
2. チャレンジ文字列の全ての文字について、以下を繰り返す
   2-1. 文字が ASCII 図形文字である場合
      1-2-1.　次の文字にすすむ
   2-2. 文字が ASCII 図形文字ではない場合
      1-2-2.　エラー画面を出し、サーバとの接続を終了する
3. レスポンス文字列を計算し、サーバに送信する
4. (以下、メール受信操作などをする)

図 ２.５-２　チャレンジ文字列チェックアルゴリズム

２.５.４. 既存のクライアントの調査

(1) 既存の主要な APOP 対応メールクライアントの挙動調査

既存の主要な APOP 対応メールクライアントに対して攻撃が成功するか否かを表 ２.５-１に示す。

[4] 3.1.1 項の説明で引用した Leurent の論文[Leurent07]においても、現実的な対策として ASCII コードのチェックを述べている。

表 2.5-1　APOP 対応メールクライアントに対する攻撃の可否

ソフト名	バージョン	攻撃の可否
AL-Mail32	1.13a	可
Becky! Internet Mail	2.42.00	可
Eudora	7J	可
Shuriken	2007	可
Sylpheed	2.4.7	**不可**
Mozilla Thunderbird	2.0.0.9	**不可**
WeMail32	2.52	可
Winbiff	2.51PL1	可
Windows Live Mail	12.0.1606	可
Yosaku	1.32	**否*1**
秀丸メール	4.83	可
電信八号	32.1.6.1	可
AL-Mail32	1.13a	可

*1:何らかの異常を誘発しているようであり、プログラムが強制終了してしまうため、攻撃は出来ても、その時点で利用者側もメールがそれ以降利用できなくなってしまう。

調査の結果、APOP 脆弱性に対する対策をしていると謳っている Thunderbord 2.0.0.9 と Sylpheed 2.4.7 は、攻撃サーバからチャレンジ文字列を受信した途端にエラーが表示され、レスポンス文字列を送る前に認証が中断した。その際に出たエラー画面を図 2.5-3、図 2.5-4 に示す。

他のソフトに関してはエラー画面も出ず、「新着メールはありません」等の正規のサーバと通信したときと変わらない挙動を示した。

以上の結果、ほとんどのメールクライアントは相手が攻撃サーバであるということを検知できないということがわかった。

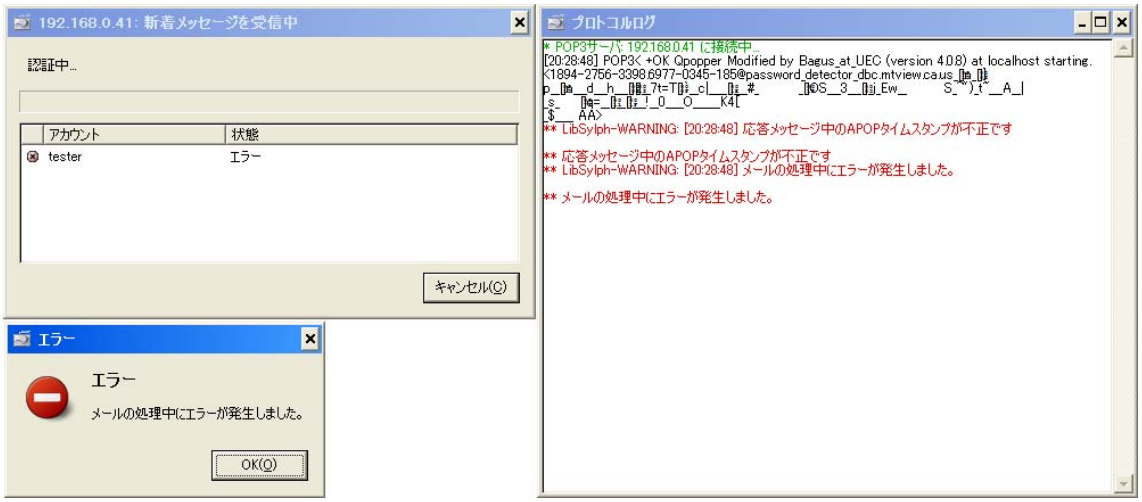

図　2.5-3　Sylpheed 2.4.7 のエラー画面

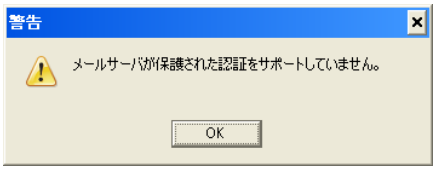

図　2.5-4　Mozilla Thunderbird 2.0.0.9 のエラー画面

(2) 攻撃不可メールクライアントソフトウェアの詳細

Sylpheed2.4.7 および Thunderbird2.0.0.9 では、チャレンジ文字列の書式を厳密にチェックすることで、攻撃を回避している。

一例として、Sylpheed2.4.7 のソースコードにおいてチャレンジ文字列をチェックしている部分を図　2.5-5 に示す。

ソースコード	処理内容

gboolean is_ascii_str(const gchar *str)
{
const guchar *p = (const guchar *)str;
 	
while (*p != '¥0') {
if (*p != '¥t' && *p != ' ' &&
*p != '¥r' && *p != '¥n' &&
(*p < 32 || *p >= 127))
return FALSE;
p++;
}
 	
return TRUE;
}
	入力はチャレンジ文字列
	
	文字列の先頭から末尾までの文字が
	タブ文字(¥t)・スペース・
	CR(¥r)・LF(¥n)ではなく、
	ASCII コード 32 未満 127 以上
	のとき「チャレンジは不正」を返す
	
	チェックを通過した場合「チャレンジは正当」を返す

図 2.5-5 Sylpheed2.4.7 のチャレンジ文字列チェック関数

尚、ここで用いられているチェック手法は“msg-id”に準拠したチャレンジ文字列を生成することができないという点を利用している。将来“msg-id”に準拠したチャレンジ文字列を用いて攻撃が可能となった場合にはこの攻撃回避方法は無効になるため、他の対策について今後検討する必要がある。

３．MD5 解読手法の実ネットワーク環境での実証調査

３.１．実証実験システムの構成

実証実験システムの構成を図　３.１-１に示す。

また、各機能ブロックにおける詳細構成を表　３.１-１に示す。

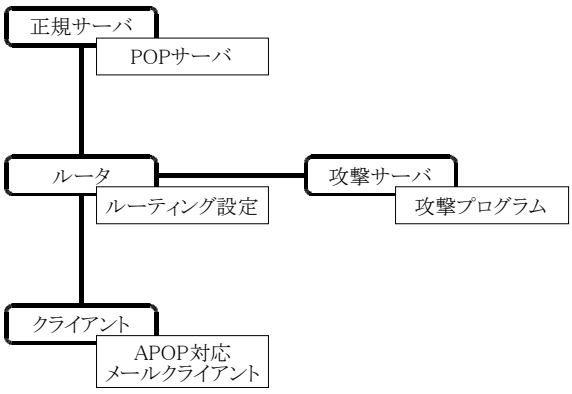

図　３.１-１　各機能ブロックの接続図

表 3.1-1　各機能ブロックの詳細構成

機能ブロック	機種名	機　能	機能を実現するソフト名
クライアント	MCJ Lm-i440S	OS	CentOS 5
		APOP 対応 メールクライアント	Mozilla Thunderbird 1.5.10
		仮想マシンモニタ	xen 3.0.3
	Panasonic Let's Note R1	OS	Windows XP
		APOP 対応 メールクライアント	Mozilla Thunderbird 2.0.0.0
	IBM ThinkPad X30	OS	Windows XP
		APOP 対応 メールクライアント	Mozilla Thunderbird 2.0.0.0
	IBM ThinkPad X23	OS	Windows XP
		APOP 対応 メールクライアント	Mozilla Thunderbird 2.0.0.0
正規サーバ	DELL PRECISION 380	OS	CentOS 5
		POP サーバ	Dovecot 1.0.rc15
		擬似メール生成	sendmail 8.13.8
攻撃サーバ	Lenovo ThinkPad X60s	OS	Vine Linux 4.1
		攻撃プログラム	Qpopper 4.0.8 を基に改造
ルータ	Acer Aspire M1100	OS	CentOS 5
		ルーティング設定	Linux, iptabels の機能を利用
		パケットキャプチャ	Wireshark 0.99.5

構築した実証実験システムの全景を図 3.1-2に示す。机上のノート PC は奥からクライアント(Let's Note R1)、クライアント(ThinkPad X30)、クライアント(ThinkPad X23)、攻撃サーバ、床のデスクトップ PC は奥から正規サーバ、クライアント (MCJ Lm-i440S)、ルータである。

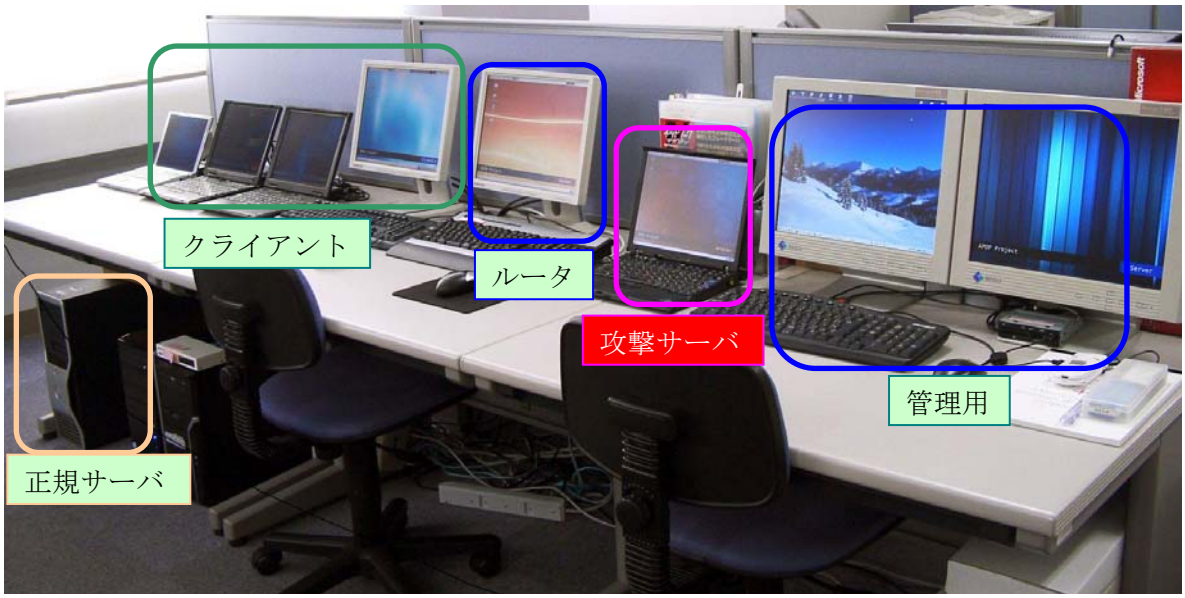

図　3.1-2　実証実験システム全景

3.2. 各実験項目の実施予定

3.2.1. 実験の予想完了時間の見積もり

本実験において、メールクライアントがメールサーバに接続してメール到着の有無をチェックする周期（認証周期）は 1 分間隔とした。

パスワードに使うことのできる ASCII 図形文字は 95 種類ある。攻撃手法より、1 つの文字がパスワードの文字であるか否かを判定するのに 2 回の認証が必要なことから、

1 文字を特定するのに必要な最大認証回数：95 文字×2 回/文字=190 回

となる。

又、メールクライアントが 1 分間隔で認証を行うため、

1 項目の実験にかかる最大時間：　190 回/文字×12 文字×1 分/回×1/攻撃頻度

となる。

この計算式を元にして計算した各実験番号の実施予想完了時間を表　3.2-1 に示す。

表 3.2-1　実験１の予想完了時間

実験番号	文字種	辞書攻撃	攻撃頻度	予想完了時間（最大）
1-1	英数字+記号	なし	1/2	760 時間（32 日）
1-2			1/4	1,520 時間（64 日）
1-3	英数字+記号	あり	1/2	760 時間（32 日）
1-4			1/4	1,520 時間（64 日）
1-5	英数字	あり	1/2	760 時間（32 日）
1-6			1/4	1,520 時間（64 日）

上記の実行時間はパスワード 10 種類を、1 種類ずつ実験をした場合の時間である。

3.2.2. 各実験項目の実施計画

上記予測時間によると、1 種類ずつ実験を実施した場合、総時間は最大で 6,840 時間(285 日)となると考えられる。現実的な時間内で実験を完了するためには、特にクライアントの数を増加させることが必要となる。

実験時間を短縮する現実的な方法として、1 台のコンピュータ上で複数のメールクライアントを実行する方法が考えられるが、攻撃サーバにおいて攻撃の対象を明確にするためにはクライアントの IP アドレスが固定化されている必要がある。そのため、別の方法として、1 つの PC 上に仮想的なコンピュータ（仮想マシン)を構成する。本実験では 6 台の仮想マシンを構成して実験を行うことにより、10 種類のパスワードを同時に解読することにした。

以上の結果、3.2.1 項の表 3.2-1 の予想完了時間は 10 分の 1 に短縮できる。

実験 2、実験 3 を含めた全体の実施予定を表 3.2-2 に示す。

表 3.2-2　予定実験期間

実験番号	実験完了に要する予想時間（最大）	予定実験期間
1-1	76 時間　（4 日）	10 月 22 日 - 10 月 25 日
1-2	152 時間（6 日）	10 月 29 日 - 11 月 3 日
1-3	76 時間　（4 日）	11 月 5 日 - 11 月 8 日
1-4	152 時間（6 日）	11 月 10 日 - 11 月 13 日
1-5	76 時間　（4 日）	11 月 14 日 - 11 月 17 日
1-6	152 時間（6 日）	11 月 18 日 - 11 月 23 日
2	-	実験 1 と同時に実施
3	-	11 月 25 日

３.３. 実験結果

３.３.１. 実験の実施状況

実際に実験を行った期間を表 ３.３-１に示す。

前述の３.２.２項で見積もった実験完了予想時間とずれがあるのは、実験中に攻撃プログラムがチャレンジ文字列の生成に失敗することがあったためである。実験番号 1-1 では該当するクライアントのみ始めから解読をやり直した。実験番号 1-2 以降では解読データを操作することで、解読を失敗したところから再開できる手法を検討し、効率的な実験を行えるようにした。

表 ３.３-１　実験期間

実験番号	実行完了所要時間	実験期間
1-1	135 時間	10 月 22 日 - 10 月 28 日
1-2	146 時間	10 月 29 日 - 11 月 3 日
1-3	91.5 時間	11 月 5 日 - 11 月 9 日
1-4	142 時間	11 月 10 日 - 11 月 16 日
1-5	51 時間	11 月 16 日 - 11 月 19 日
1-6	88 時間	11 月 19 日 - 11 月 22 日
2	実験 1 と同時に実施	実験 1 と同時に実施
3	11 月 25 日	12 月 3 日

３.３.２. 実験結果

(1) 攻撃サーバとの認証回数

攻撃サーバはそれぞれのクライアントに設定したパスワードをすべて正しく解読した。クライアントと攻撃サーバが解読完了までに行った認証回数を表 ３.３-２に示す。また、グラフを図 ３.３-１及び図 ３.３-２に示す。

実際にはルータによる接続先の切り替えの頻度に対応して、クライアントは表の値の 2 倍ないし 4 倍の回数の認証を行っている。

実験 1-1 と 1-2、1-3 と 1-4、1-5 と 1-6 はそれぞれ同一のパスワード・攻撃方法だが、同一のパスワードにおいて認証回数が異なっている。これは、クライアントと攻撃サーバが認証中にルータが接続先を切り替えてしまい、認証が途中でタイムアウトし中断をした回数も含まれることが原因である。

表 ３.３-２ (1)　攻撃サーバとの認証回数（実験 1-1、1-2）

実験番号	実験 1-1		実験 1-2		ASCII 昇順で巡った場合の理論値	
パスワード番号	8 文字	12 文字	8 文字	12 文字	8 文字	12 文字
パスワード 1	168	276	168	278	168	276
パスワード 2	986	1,556	907	1,480	904	1,474
パスワード 3	1,105	1,575	1,114	1,585	1,104	1,574
パスワード 4	867	1,181	872	1,186	866	1,180
パスワード 5	1,054	1,484	1,060	1,490	1,054	1,484
パスワード 6	803	1,121	805	1,125	802	1,120
パスワード 7	1,080	1,516	1,081	1,519	1,080	1,516
パスワード 8	890	1,170	890	1,171	890	1,170
パスワード 9	698	942	699	944	698	942
パスワード 10	672	1,104	670	1,102	670	1,102

表 ３．３－２ (2)　攻撃サーバとの認証回数（実験 1-3、1-4）

実験番号	実験 1-3		実験 1-4	
パスワード番号	8 文字	12 文字	8 文字	12 文字
パスワード 1	252	390	235	373
パスワード 2	1,129	1,971	1,135	1,992
パスワード 3	378	698	363	675
パスワード 4	1,131	1,230	1,139	1,247
パスワード 5	379	568	388	585
パスワード 6	817	1,304	812	1300
パスワード 7	756	959	757	960
パスワード 8	836	1,277	837	1,330
パスワード 9	727	941	727	942
パスワード 10	826	1,012	827	1,013

表 ３．３－２ (3)　攻撃サーバとの認証回数（実験 1-5、1-6）

実験番号	実験 1-5		実験 1-6	
パスワード番号	8 文字	12 文字	8 文字	12 文字
パスワード 1	24	118	24	118
パスワード 2	353	1,121	357	1,135
パスワード 3	279	391	268	389
パスワード 4	602	780	591	769
パスワード 5	469	813	470	814
パスワード 6	696	844	716	864
パスワード 7	810	1,060	812	1,073
パスワード 8	964	1,518	674	1,072
パスワード 9	677	1,119	901	1,343
パスワード 10	751	1,091	750	1,090

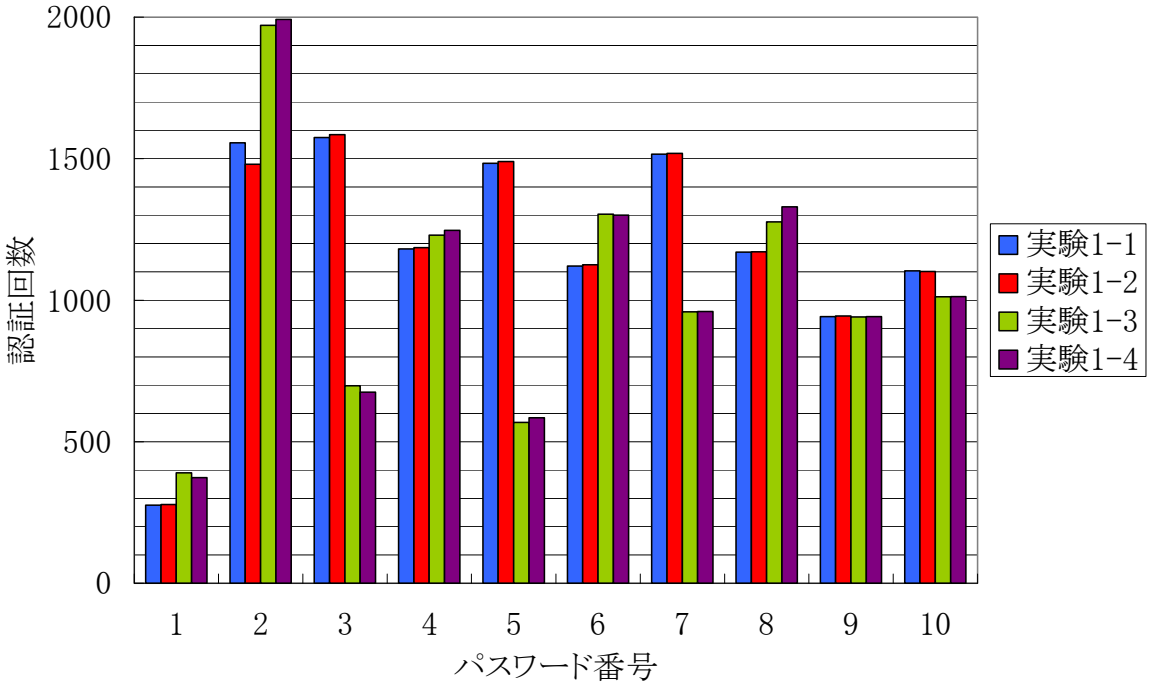

図 ３.３-１　攻撃サーバとの認証回数（英数字と記号）

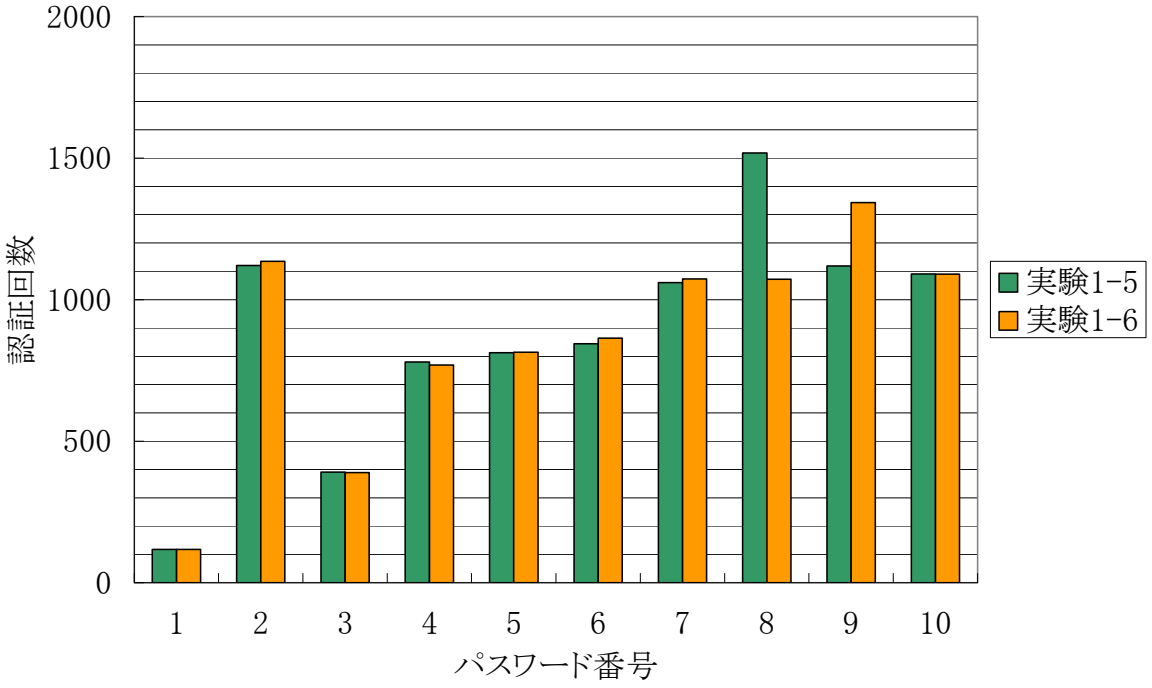

図 3.3-2 攻撃サーバとの認証回数（英数字のみ）

(2) 攻撃サーバの応答時間

攻撃プログラムはクライアントが接続してきてからチャレンジ文字列を生成する。

従って、チャレンジ文字列を生成する時間だけ、攻撃者の応答時間は正規のサーバよりも長いはずである。応答時間の中にはチャレンジ文字列生成以外の動作も含まれるが、10 ミリ秒オーダーと十分小さく無視できるため、図 3.3-3 のように TCP 接続確立後に攻撃サーバからチャレンジ文字列が送られてくるまでの時間を攻撃者の応答時間と定義する。

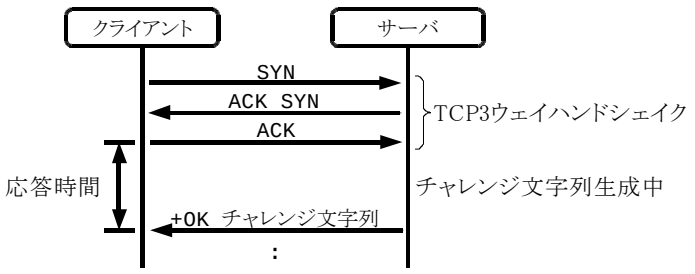

図 3.3-3 応答時間の定義

縦軸を応答時間の平均、横軸を推定する文字数目としたグラフを図 3.3-4 に示す。尚、チャレンジ文字列は推定する文字の 1 回目の認証で生成するため、2 回目の応答時間

は含んでいない。2 回目の応答時間は正規サーバと同様、10 ミリ秒オーダーのものがほとんどであった。

1 文字目から 11 文字目まではほぼ横ばいである。辞書攻撃なしの実験 1-1、1-2 では、12 文字目が他に比べて 2 倍程度になっている。これは攻撃方法のチャレンジ生成アルゴリズムによるものである。

1 文字目から 11 文字目までの応答時間が上下している理由としては、複数のクライアントが同時に接続してきた場合、他のクライアントに対するチャレンジ文字列を生成するプロセスが優先され、該当するプロセスが後回しになった結果、応答まで大幅に時間がかかったものが含まれることが原因であると考えられる。

辞書攻撃ありの実験 1-3、1-4 と実験 1-5、1-6 では、すべての文字数目に関してそれぞれ実験 1-1、1-2 よりも 1.25 秒から 2.5 秒ほど時間がかかっているが、文字数目に関しては実験 1-1、1-2 と同様の変化をしている。

これらの差は、辞書から次の推定する文字を探索している時間が原因と考えられる。

今回の実験の攻撃サーバは Intel Core Duo LV 1.5GHz を搭載した一般的なノート PC で行った。それぞれの認証は独立したプロセスで実行されるため、より速いプロセッサを多く並列化すれば攻撃サーバの応答時間はさらに短くなると考えられる。

いずれにしても、攻撃者の応答時間はメールクライアントに設定されたタイムアウト時間よりも十分に短く、ネットワーク的に遠いメールサーバの応答時間と多少長いか変わらない為、攻撃サーバを見分けることは困難である。

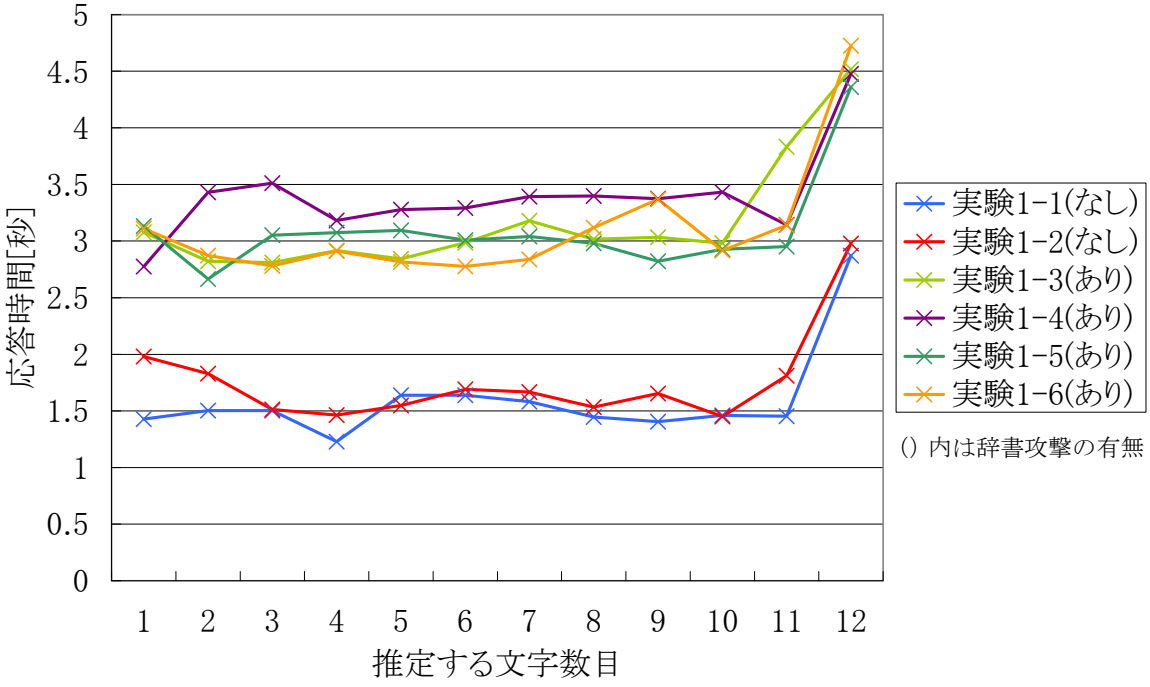

図 3.3-4 攻撃者の応答時間

(3) 新着メールの到着が遅れる割合

メールクライアントの設定で、新着メールを確認する間隔は 1 分に設定されている。従って、図 3.3-5 の例 A のようにメールサーバに新着メールが届いてから、クライアントソフトが受信するまでの間隔は 1 分以内になるはずである。例 B のように間隔が 61 秒以上の場合、新着メール無しであった前回の認証は攻撃サーバとの間で行われたと考えてよい。

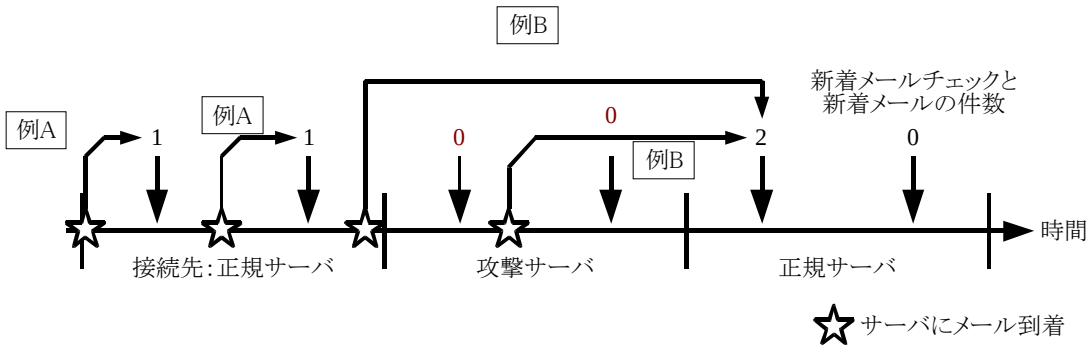

図 3.3-5 メール到着から受信までの時間の例

実験番号ごとの新着メール受信回数を表 3.3-3 に示す。表右の、例 B の到着回数を総受信回数で割った値は各実験で設定した攻撃頻度とほぼ一致する。

表 3.3-3 新着メール受信回数

実験番号	新着メール総受信回数	クライアントが受信するまでの間隔		総受信回数と61 秒以上の比
		0-60 秒 (例 A)	61 秒以上 (例 B)	
1-1	756	368	388	0.513
1-2	901	671	230	0.255
1-3	560	284	276	0.493
1-4	977	729	248	0.254
1-5	576	272	304	0.528
1-6	1,117	773	344	0.308

不定期に受信操作をした場合、図 3.3-6のようになる。

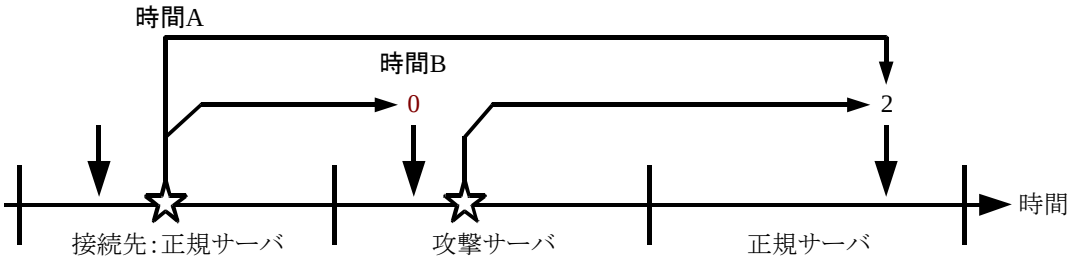

図 3.3-6 不定期に受信操作をした場合

このような新着メールの到着遅延が発生し得ることを利用して攻撃の有無を判断するためには、前回に新着メールをチェックした時間（時間 B）を記憶しておき、届いた新着メールが自分のサービスプロバイダのメールサーバに届いた時間（時間 A）を比較することである。前回のチェックで新着メールなしだったにもかかわらず、それより前の時間にメールが届いていたことを検出することができる。しかし、クライアントとサーバの時計の時刻を合わせておくことが前提であり、必ずそのような環境で運用しなければならない。

(4) APOP 対応メールクライアントが攻撃サーバと通信したときの挙動の調査

調査に使用したメールクライアントを表 3.3-4に示す。12 月 3 日時点で入手可能な最新版である。

表 3.3-4　APOP 対応メールクライアント

ソフト名	バージョン	攻撃の可否
AL-Mail32	1.13a	可
Becky! Internet Mail	2.42.00	可
Eudora	7J	可
Shuriken	2007	可
Sylpheed	2.4.7	否
Mozilla Thunderbird	2.0.0.9	否
WeMail32	2.52	可
Winbiff	2.51PL1	可
Windows Live Mail	12.0.1606	可
Yosaku	1.32	否*1
秀丸メール	4.83	可
電信八号	32.1.6.1	可
AL-Mail32	1.13a	可

*1:何らかの異常を誘発しているようであり、プログラムが強制終了してしまうため、攻撃は出来ても、その時点で利用者側もメールがそれ以降利用できなくなってしまう。

調査の結果、APOP 脆弱性に対する対策をしていると謳っている Thunderbird 2.0.0.9 と Sylpheed 2.4.7 は、攻撃サーバからチャレンジ文字列を受信した途端にエラーが表示され、レスポンス文字列を送る前に認証が中断した。その際に出たエラー画面を図 3.3-7、図 3.3-8 に示す。

Yosaku 1.32 はチャレンジ文字列を受信した段階で強制終了した。

Windows Live Mail では稀に認証エラーが発生したが、パスワードは正しく解読された。

他のソフトに関してはエラー画面も出ず、「新着メールはありません」等の正規のサーバと通信したときと変わらない挙動を示した。

以上の結果、ほとんどのメールクライアントは相手が攻撃サーバであるということを検知できないということがわかった。

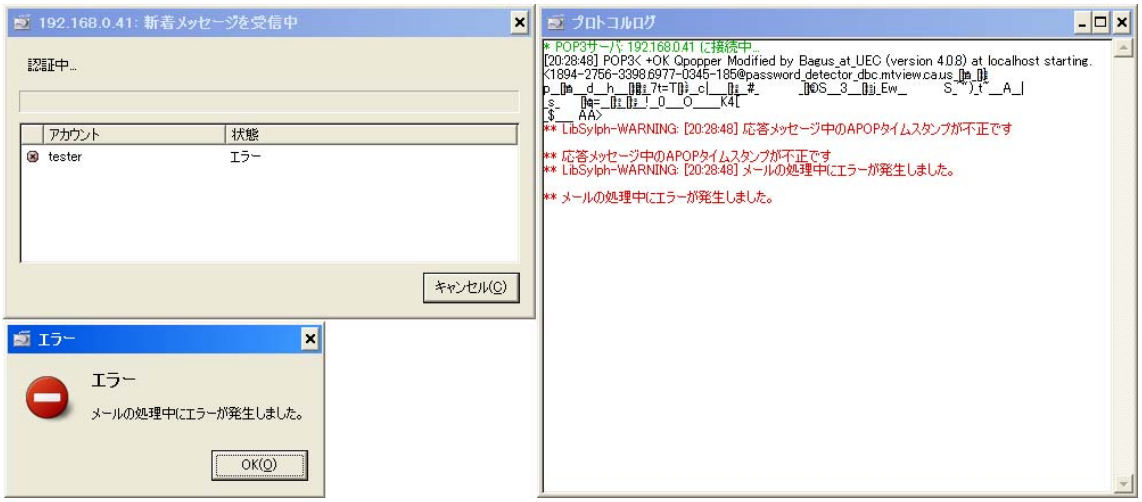

図　3.3-7　　Sylpheed 2.4.7 のエラー画面

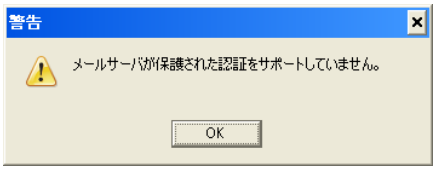

図　3.3-8　　Thunderbird 2.0.0.9 のエラー画面

現在、攻撃者が生成できるチャレンジ文字列は RFC で定められている書式を満たすことができない。認証が中断した 2 つのソフトは、サーバから送られてくるチャレンジ文字列を厳密にチェックし、書式を満たさない場合は相手が攻撃者であると判断してレスポンス文字列を送らずに認証を強制終了している。これによって、攻撃者はレスポンス文字列を得ることができず、攻撃に失敗する。

４．まとめ

今回の調査、実証実験により、ごく一般的な PC を用いても、メールのパスワードをすべて正しく解読することができることを確認した。

対症療法としては、従来からいわれている「パスワードを定期的に変更すること」に尽きるが、この変更周期についても、今回の実験からはかなり短い周期である必要性が浮かび上がってきた（実験環境の条件下での推定では**約 40 日程度**[5]）。

これらの状況に対して、APOP に係わる脆弱性を明確に注意喚起しているメールプロバイダは多くない。

更に、メールクライアントソフトウェアについても、ほとんどの主要ソフトウェアにおいて対策が講じられていないのが現状である。

また、メールのリアルタイム性を上げるためには頻繁にメールサーバに接続する必要があるが、そのたびに認証が行われるため攻撃の機会をより頻繁に与えることにもなりかねない。従って、必要最低限のメール到着確認周期にすることも、攻撃から身を守る術となろう。

今回の調査、実証結果を踏まえて、改めて APOP 方式の脆弱性について警告を発すると共に、当面の具体策に関する啓蒙を行う必要性がある。また、メールクライアントソフトウェア等における攻撃の回避対策を検討する必要がある。

４．１．MD5 の脆弱性に関する調査結果より

MD5 の脆弱性に関する各種論文、報告等を調査した結果、脆弱性に対する実環境での実証検証、具体的な対応策に関しての言及はほとんどないのが現状である。

攻撃者は、ネットワークの構築さえ出来てしまえばユーザに気づかれることなく、APOP のパスワードを入手することが実時間で可能である。そのため、現実の脅威として認識する必要がある。パスワードはいろいろな仕組みで共有している場合（特に、オンラインバンクなど）は、単に、メールの内容を読まれてしまうだけでなく、経済的な被害も生じる。

そのため、早急にこの攻撃に対する対策法を検討しなくてはならない。これまでに提案されている対策は、クライアント側のチェックにより、チャレンジが ASCII 図形文字以外であれば、不正なチャレンジであると判定し、それ以降の処理を中断することである。これまでの Sasaki らの攻撃では、常に ASCII 図形文字になるようなチャレンジを生成する

[5] 今回の実証実験では、メール到着確認の周期を 1 分間隔に設定したが、通常のメール利用者が設定しているメール到達確認周期は、もしデフォルト設定のままであると仮定すると多くのソフトウェアが 30 分と設定されている（Sylpheed では 10 分）との調査結果を適用して推定。

ことができないため、一時的な対策としては、機能する。しかし、今後の研究の発展を考慮すると、これも盤石な対策とはいいがたい。

次に考えられる対策は、レスポンスの生成時に、チャレンジとパスワードの順番を逆にすることである。APOP の仕様では、レスポンスは、MD5(Challenge||pw)として成しているが、チェレンジとパスワードの順番を入れ替え、MD5(pw||Challenge) とすることで攻撃の回避が可能であるかもしれない。実際、Sasaki らの攻撃では、パスワードが後半に来ることが必須である。そのため、この対処法は、有効となりうる。しかしながら、Wang らは、MD5 ではなく、MD4 に対してではあるが、パスワードが前半部であっても、攻撃が成功することを示している。そのため、この対処法も、完全ではない。

以上のように、いくつかの容易に想像しうる対処法はあるが、いずれも今後の研究の進展を考慮に入れると、本質的に有効であるとは言い難い。そのため、さらに有効な攻撃の回避法を検討することが急務の研究課題である。

４.２. MD5 解読手法の実証調査結果より

チャレンジ生成アルゴリズムを POP サーバに組み込み、攻撃者のメールサーバに実装した。攻撃を現実に近い環境で行うために、クライアントと正規のサーバの間にルータを挿入する方法でなりすましを行いながら攻撃実験を行った。

その結果、ごく一般的な PC を用いても、攻撃サーバはそれぞれのクライアントに設定したパスワードのすべてを正しく解読した。攻撃者は平均的には約 1,000 回のなりすましを行うことで解読が可能である。特に、攻撃者が辞書攻撃を用いると、安易なパスワードの 1 つでは、24 回の攻撃者との APOP 認証によりパスワードが解読されてしまうこともわかった。その解読に要する時間も、色々な設定条件により差異はあるものの、通常の使用環境においてはほぼ現実的な時間以内に収まってしまうことが判かった。

新着メールの遅れという点では、サーバに新着メールが届いているはずの時間に新着メール無しとなった場合、攻撃を受けている可能性がある。メールクライアントが前回の受信時間を記憶しておき、新着メールのヘッダの書いてある時間との比較を行うなどの工夫をすることで攻撃を検出できる。しかし、通信障害などの遅延は起こり得る現象であり、この方法に頼るのは万策ではない。

少数のメールクライアントソフトウェアにおいて、攻撃に対する現在可能な攻撃回避方法を実装しているが、一方で対策が施されていないものも多く存在する。対策が施されていない多くのメールクライアントに関しては早急な対策が必要である。ただし、この方法は“msg-id”に準拠したチャレンジ文字列を生成することができないという点を利用している。将来“msg-id”に準拠したチャレンジ文字列を用いて攻撃が可能となった場合にはこの攻撃回避方法は無効になるため、他の対策についても今後検討する必要がある。

また、POP の仕様の変更は短期的対策としては有効ではないが、長期的対策としては有

効であるため、今後プロトコル仕様の修正を行うことは十分に価値がある。

サーバが SSL に対応することは、もっとも根本的な対策であるが、ユーザはサービスプロバイダの対応を待たなければならない。

４.３. 今後の課題

APOP の脆弱性に対する攻撃を回避する手段として、メールクライアントソフトウェアにおいてチャレンジ文字列がプロトコル仕様に合致しているかを厳密にチェックする手段が当面有効である。

しかしながら、ウイルスとウイルス検出ソフトウェアとの場合と同様に、攻撃者側も進化していくことが考えられ、更なる対策が必要である。

従って、今後も引き続き攻撃回避の対策の研究、検討が重要である。

また、同時にプロトコルの見直し等も早急に進める必要がある。