2C－7

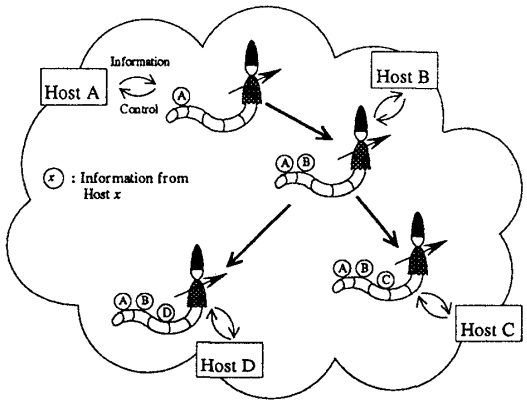
ネットワークワームを利用したネットワーク管理システムのためのセキュリティ向上手法[1]

清水亮博 (東京工業大学大学院理工学研究科情報科学専攻)[2]
大野浩之 (東京工業大学大学院情報理工学研究科)[3]

1　はじめに

これまで筆者らは、ネットワークワームを用いたネットワーク管理システム (NMW System) を開発してきた [3, 2, 4]。

NMW System は、ネットワークワーム (以下ワーム) によるネットワーク管理システムが実現可能であることを明らかにし、さまざまな特性 (例：図 1) を実証したが、セキュリティのための十分な機能がなかった。しかし、今後広域ネットワーク等において実証試験を行なうためには、セキュリティを強化する必要がある。そこで筆者らは公開鍵暗号を用いて、ワームのユーザ認証、改竄および盗聴の防止を行なった。



計算機間での移動に伴う調査、分析、制御

図 1: ワームによるネットワーク管理システムの特徴の例

2　セキュリティ向上に必要な条件

本システムはネットワーク管理用のシステムであるため、ワームの発信者の認証等、セキュリティには十分注意する必要がある。そのため、以下のような点を考慮する必要がある。

- ワームや実行結果等を改竄されない。
- 他の発信者が発信したワームに見せかけることができない。
- 作業内容や、その結果を盗聴されない。
- あるホストのセキュリティが破られても、他のホストへの被害が最小限に押えられる。

上の条件を満たすために、暗号化を用いる。また発信者の認証を行なうために、公開鍵暗号を用いた電子署名も採用する。通常の電子署名では発信者の公開鍵を一般に公開するが、ここでは公開鍵をその各ホストに配布するものの、各ホストの一般ユーザには公開しない。

ここで公開鍵方式暗号を用いるのは、発信者の秘密鍵でワーム全体を暗号化しておけば、ワーム自身はその発信者にしか暗号化できないため、ワーム本体が改竄されることはなくなるためである。もし秘密鍵暗号方式で暗号化すると、訪れたホストのセキュリティが破られた場合にワームを改竄することが可能であるが、公開鍵暗号を用いればこのような場合にもワーム本体が改竄されることはない。また、ワームがデータをワームの発信者に送り返す際には、発信者の公開鍵で暗号化すれば、秘密鍵を持っている発信者以外には実行結果を見られることはない。

ところでワームが各ホストで得たデータを発信者の公開鍵で暗号化すると、他のホストにおいてワームはそのデータを利用できなくなる。これでは、ワームがネットワーク上を移動しながら作業を行なう際に、訪れているホストのデータのみしか利用できなくなるので、複数のホスト間でデータを照合する場合等に障害となる。この解決法としては、秘密鍵暗号方式の鍵をワームに持たせ、途中でのデータをその鍵で暗号化する方法 (方法 1 とする) と、各ホスト毎に公開鍵と秘密鍵の組を用意し、その秘密鍵で暗号化する方法 (方法 2 とする) がある。

以上の議論を踏まえた暗号化手法を図 2に示す。この手法を用いれば、上記の条件をほぼ満たすことができる。

3　セキュリティ向上についての考察

方法 1 は、ワームが訪れたホストで得た情報を暗号化する鍵をワーム自身が持っているため、別途鍵を配送する必要がないという特徴がある。しかし、システム中

[1]Security Enhanced Network Management Worm System
[2]Akihiro Shimizu, Tokyo Institute of Technology
[3]Hiroyuki Ohno, Tokyo Institute of Technology

1. 発信者 A がワーム W を実行する場合、方法 1 ではまず K_W を作成し、W, a と共に K_a^S で暗号化する。方法 2 では K_W は不要。送信する際には、平文の a を付けておく。

$$\text{方法 1} \quad a, K_W, W \xrightarrow{K_a^S} a, [a, K_W, W]K_a^S$$
$$\text{方法 2} \quad a, W \xrightarrow{K_a^S} a, [a, W]K_a^S$$

2. ホスト x はワームを受けとると、平文の発信者の識別子 a より公開鍵 K_a^P を用いてワームを復号化する。復号化された発信者の識別子と平文の識別子が一致すれば、ワームは A が発信したものと考えて良い。なお、復号化する際には元の暗号化されたワーム本体も保存しておく。

$$\text{方法 1} \quad a, [a, K_W, W]K_a^S \xrightarrow{K_a^P} a = a, K_W, W$$
$$\text{方法 2} \quad a, [a, W]K_a^S \xrightarrow{K_a^P} a = a, W$$

3. ホスト x での実行が終った後、このホストで得た情報 I_x を付け加える。この時方法 1 では I_x を K_W によって暗号化する。方法 2 では I_x を x と共にホスト x の秘密鍵 K_x^S で暗号化する。これを暗号化されたままのワーム本体に加えて他のホストへ転送する。

$$\text{方法 1} \quad a, [a, K_W, W]K_a^S, I_x \xrightarrow{K_W} a, [a, K_W, W]K_a^S, [I_x]K_W$$
$$\text{方法 2} \quad a, [a, W]K_a^S, x, I_x \xrightarrow{K_x^S} a, [a, W]K_a^S, x, [x, I_x]K_x^S$$

実行結果 R_x は、公開鍵 K_a^P で暗号化して発信者 A へ送れば、秘密鍵 K_a^S を持っている発信者 A のみが R_x を復号化できる。

$$R_x \xrightarrow{K_a^P} [R_x]K_a^P, \qquad [R_x]K_a^P \xrightarrow{K_a^S} R_x$$

4. 次にホスト y でこのワームを受けとった時、2. と同様に復号化する。方法 1 では、K_W は秘密鍵暗号方式による鍵なので、$[I_x]K_W$ は他のホストでも復号化することができる。方法 2 では、あらかじめ K_x^P が配布されているので、復号化できる。

$$\text{方法 1} \quad a, [a, K_W, W]K_a^S, [I_x]K_W \xrightarrow{K_a^P} a = a, K_W, W, [I_x]K_W$$
$$[I_x]K_W \xrightarrow{K_W} I_x$$
$$\text{方法 2} \quad a, [a, W]K_a^S, x, [x, I_x]K_x^S \xrightarrow{K_a^P} a = a, W, x, [x, I_x]K_x^S$$
$$x, [x, I_x]K_x^S \xrightarrow{K_x^P} x = x, I_x$$

a	発信者 A の識別子	x	ホスト x の識別子
K_a^P	発信者 A の公開鍵	K_x^P	ホスト x の公開鍵
K_a^S	発信者 A の秘密鍵	K_x^S	ホスト x の秘密鍵
W	ワーム本体	R_x	ホスト x での実行結果
K_W	ワーム W の鍵 (秘密鍵暗号方式による)		

図 2: NMW System の暗号化手法

のあるホストのセキュリティが破られた場合、方法 1 ではその破られたホストでそれまでに得た情報を改竄される恐れがある。したがって、ネットワークの状態を調べるだけでなく、構成等を変更するような重要な作業を行なうワームには方法 1 は使用できない。

これに対し、方法 2 は暗号化のアルゴリズムと鍵が十分強ければ、訪れるホストのセキュリティが破られていても、その他のホストで得た情報は安全である。

なおこれらの手法を用いても、暗号化された通信を記録しておき、そのまま繰り返す再送攻撃を防ぐことはできない。この問題を解決するには、次の 2 つの方法がある。

有効期限を用いる方法 ワーム本体に、そのワームの有効期限日時を設定しておき、各ホストで有効期限が切れているものを排除すれば、有効期限よりあとでは再送攻撃を防ぐことができる。しかし動作時間が長いワームでは有効期限を長くせざるを得ないため、十分に長いワームでは防ぐことができない。

TTL(Time To Live) count を用いる方法 ワームに付加する情報として、IP datagram の TTL フィールドと同様の変数を導入する。各ホストに訪れる度に TTL は 1 ずつ減らされ、0 になるとそのワームは捨てられる。

この TTL の値とワーム自身の ID を各ホストで記録しておく。同じ ID のワームが到着した時には、前回の TTL と比べて減少していなければ不正なものとして捨てる。TTL やワームの ID は暗号化されているため、盗聴、改竄できないためこれで再送攻撃を防ぐことができる。

謝辞

WIDE プロジェクトの研究者からさまざまな助言を得た. ここに記して感謝する.

参考文献

[1] John F. Schoch and Jon A. Hupp. The 'Worm' Programs — Early Experience with a Distributed Computation. *Communications of the ACM*, Vol. 25, No. 3, pp. 172–180, March 1982.

[2] 清水亮博, 大野浩之. ネットワークワームを利用したネットワーク管理手法. 第 47 回 (平成 5 年度後期) 全国大会 講演論文集 (1), pp. 1–299. 情報処理学会, October 1993.

[3] 清水亮博. 寄生プログラム (ネットワークワーム) を用いたコンピュータネットワークの管理方式. 平成 4 年度卒業論文, 東京工業大学理学部情報科学科, February 1993.

[4] 清水亮博, 大野浩之. ネットワークワームを用いた コンピュータネットワーク管理機構. 分散システム運用技術研究グループ資料. 情報処理学会, July 1994. DSM-9407012.