不正アクセスから自分を守る方法

渇さん（青宮泉若太）

目次

- 受動的対策
  - バックアップ
    - 全体バックアップ
    - 差分バックアップ
    - さまざまなバックアップの方法
- 能動的対策
  - セキュリティ対策
    - ワンタイムパスワード
    - コールバックシステム
    - 暗号化

受動的対策

受動的対策とは何か？

メイプルに限らず、インターネットに接続している環境では誰もが不正アクセスなどを受ける可能性がある。

それらを防ぐ方法はあるのだろうか？

⇒1つの方法として不正アクセスを受けにくいようにするという考え

⇒能動的対策（後述）

⇒もう1つの方法としては不正アクセスをもし万が一受けてしまった場合の対策を立てるという考え

⇒**受動的対策**

受動的対策

不正アクセスを受けてからでは遅いのではないか？

受動的対策に意味があるのだろうか？

⇒あらかじめ不正アクセスを受けるものと考えて準備をしておく

⇒**バックアップ**

バックアップの考え

自分のコピーをあらかじめ用意しておけば、もしやられても大丈夫ではないだろうか？

（参考）メイプルの墓システム的な考え方

メイプルでは墓を落とすと町まで戻されるが、経験値が減る。

ここで、町まで戻される＝バックアップ復元、経験値が減る＝減った分の経験値に相当する更新は反映されないとすれば、バックアップの考え方のトポロジになる。

仮にもし墓してメイプルのキャラが削除されたと考えれば、バックアップはいかに重要なことかが分かるだろう。

主なコピーの方法は2通り

- 全体バックアップ
- 差分バックアップ

全体バックアップ

今現在のデータ全てのコピーを作成する。

メイプルで言えば再インストール。

全ての分のコピーなので大きな容量のデータを保持する装置が必要。

⇒DAT（テープ）

⇒しかし、現在はDVD－RWや容量の大きいUSBメモリなどが登場しているため、それらをバックアップ装置に使っても可。

全体バックアップの弱点

- 全体をバックアップするのでバックアップ1回当たりの容量が大きくなる。
- よって、バックアップを定期的に行った場合、1回の作業に時間がかかり、非効率的。
- バックアップに失敗した場合、また1からやり直しになる。

⇒差分バックアップも利用する

差分バックアップ

前回にバックアップした分のデータと現在のデータを比較して、更新されたものだけバックアップする。

　メイプルで言えばパッチ。

更新されたものだけなのでバックアップに必要な容量はあまりいらない。

⇒しかし弱点もある

差分バックアップの弱点

- バックアップ復元時に差分バックアップのデータだけでは復元できないので、別のバックアップデータが必要になる。
- ゆえにバックアップ装置をたくさん用意しなければいけないことになり、装置がかさばる。

⇒そこで全体バックアップと併用することでこの問題を解決できる

バックアップ

- 最初の１回、および3ヶ月に1回の程度の頻度で全体バックアップを行う

  ⇒これにより全体バックアップの弱点であったバックアップにかかる時間が短縮できる。

- ２週間に１度ぐらいの程度の頻度で差分バックアップを行う

  ⇒3ヶ月に1回で全体バックアップを行うので、それまでの差分バックアップデータは破棄できるので、その装置を再利用できる。

- この頻度で行えば、バックアップ装置は全体バックアップ用に1つと、差分バックアップ用に5つの計６つで済む。

さまざまなバックアップの方法

- これまでにあったような全体バックアップ、差分バックアップでも、さらにバックアップの方法が色々ある。
- 代表的なものを挙げておく
  - デュアル式
  - データ分散式

デュアル式バックアップ

- 全く同じバックアップデータをもう1つ用意する。
  - ⇒これにより、もしバックアップに失敗した、あるいはバックアップ装置の物理的な破損によりバックアップデータが誤って消去された場合でも困らない
  - ⇒しかし、この方法はバックアップ時間が2倍になってしまうので、全体バックアップのような時間のかかるバックアップでこれをやると時間が多くかかってしまう
  - ⇒どちらかと言うと差分バックアップ向けだが、全体バックアップデータがもし破損してしまうと、差分バックアップデータも台無しになってしまう。

データ分散式バックアップ

- バックアップデータをさらに細かく分けて、装置に分散させてバックアップさせる。

  ⇒これにより、バックアップデータを時系列ではなく、データパックとして扱える。バックアップ装置個々の容量も少なくなる

  ⇒しかし、バックアップの復元に必要な装置の数が常に最大になる上に、ウイルスなどを誤ってバックアップしてしまった場合に復元が大変になる。

- デュアル式バックアップと組み合わせれば、信頼性は向上する。

バックアップ復元時の注意事項

- もしバックアップを復元しなければならないような状況になった場合、必ず最初にインターネットおよびLANなどのネットワークから切り離してから行うこと
- もしこれをやらないと、不正アクセスされたままになり、バックアップを復元できないようにされる可能性がある。
- あらかじめバックアップ用に起動ディスクを用意しておく
- 起動ディスクで起動した後に全体バックアップ→差分バックアップの順で復元する。

  ⇒全体バックアップと差分バックアップを組み合わせておいたありがたみが改めて分かる

能動的対策

では、不正アクセス自体を受けにくくするにはどうしたら良いか？

主な方法として以下の方法がある。

- ワンタイムパスワード
- コールバックシステム
- データの暗号化
  - 公開鍵と秘密鍵

ワンタイムパスワード

- １度だけ有効なパスワードを作成する
- １度使用すると、もう2度と同じパスワードは使えなくなる

  ⇒ログインするたびにパスワードを変更する
- これにより、万が一パスワードが漏洩しても大丈夫

ワンタイムパスワード

- メイプルで実装するにはどうしたら良いか？
  - ログインする前に必ず公式サイトにアクセスしてパスワードを変更する
    - パスワードの文字列は長めの方が良い
    - かつ、辞書にあるような単語ではない方が良い
    - さらに、生年月日など個人を特定できるようなパスワードにはしない方が良い

      ⇒例として・・・2つ以上の単語の組み合わせ
    - パスワードの文字列にAからZまでのアルファベットと数字が用いられているとすると、n文字のパスワードでは36^n通りになる
    - 仮に1通りを10^{-6}秒で計算したとすると、10文字のパスワードでは最悪約36億秒＝約100万時間＝約114年かかることになる。

ワンタイムパスワードの弱点

- しかし、毎回公式サイトでパスワードを変えるのは面倒くさい
- 何らかの事情で公式サイトにアクセスできなくてパスワードを変えられなくなると、メイプルにINできなくなる

コールバックシステム

- 一度ログインしたら、本人であることを確認するためにもう一度かけなおすシステム
- 不正アクセス防止にも本人確認は必要
- メイプルでの実装は難しい

暗号化

- そこでメイプルで本人確認をするために関係者しか知らない「合言葉」を用意する
- ログインするたびに「合言葉」を言う
- 「合言葉」を言えなかった時点で本人ではないと判断できる
- 「合言葉」を用意する方法として主なものを挙げる
  - 秘密鍵暗号方式
  - 公開鍵暗号方式

秘密鍵暗号方式

- 「合言葉」をリアル世界で伝達しておく
  - 例えば口頭で「合言葉は〇〇だよ」と言う
  - 例えば封書にして親展扱いで本人に郵送する
- そしてこの「合言葉」が文字通り合えば、本人確認ができたことになる。
- 「合言葉」の伝送が大変なのが弱点

秘密鍵暗号方式の応用

- 秘密鍵暗号方式をメイプルに応用させてみる
- リアル世界で伝達できるのが一番良いのだが、そうなるとリア友にしか伝達できないので不便
- リアル世界を拡大解釈してメイプル以外の世界とみなす
  - すると、伝達方式にメールで送るなどという選択肢ができる
  - ただし不特定多数が見るような掲示板などで伝達してはいけない。あくまで本人確認なので、その個人だけが見られるような形式にしないとまずい

公開鍵暗号方式

- 「質問」と「答え」を用意しておく
- 「質問」をした時に正しく「答え」が返ってくれば本人確認ができたことになる。
- 「質問」の「答え」は本人が回答しやすいものを選んでおく必要がある

  ⇒各人の親密度、コミュニケーション力、性格などが大きく影響する
- 用意する「質問」の選定が大変なのが弱点

  ⇒しかし「質問」さえきちんと答えられるものであれば、容易に本人確認ができる。

公開鍵暗号方式

- 「質問」は誰にでも言う(＝公開する)ものなので、公開鍵と呼ぶ
  - 不特定多数の人が見るような場所で公開しても**問題なし**。
- 「答え」および秘密鍵暗号方式の「合言葉」は関係者しか知らないものなので秘密鍵と呼ぶ
  - 不特定多数の人が見るような場所で公開しては**いけない**。
- なお、「質問」の文脈に「答え」が必ずマッチしている必要はない
  - 例:「質問」に「好きなスポーツは？」として、「答え」に「信濃川」としても良い