ファイル改ざんの検出および自動修復が可能な被害回復支援システム

千葉大学自然科学研究科
張　亮

発表の流れ

- 背景
- 目的
- システムの構成
- システムの実現
- 動作確認実験
- 考察
- まとめ
- 今後の課題

背景

- インターネットを利用した犯罪は激増
  - WEBページの改ざん
  - バックドアの設置
  - ウイルスによる被害の拡大
- 多くの研究や開発が進められている
  - ファイアウォール
  - ウイルス対策ソフト
  - IDS、IDP
- 不正アクセスの件数は増加する傾向にある

背景

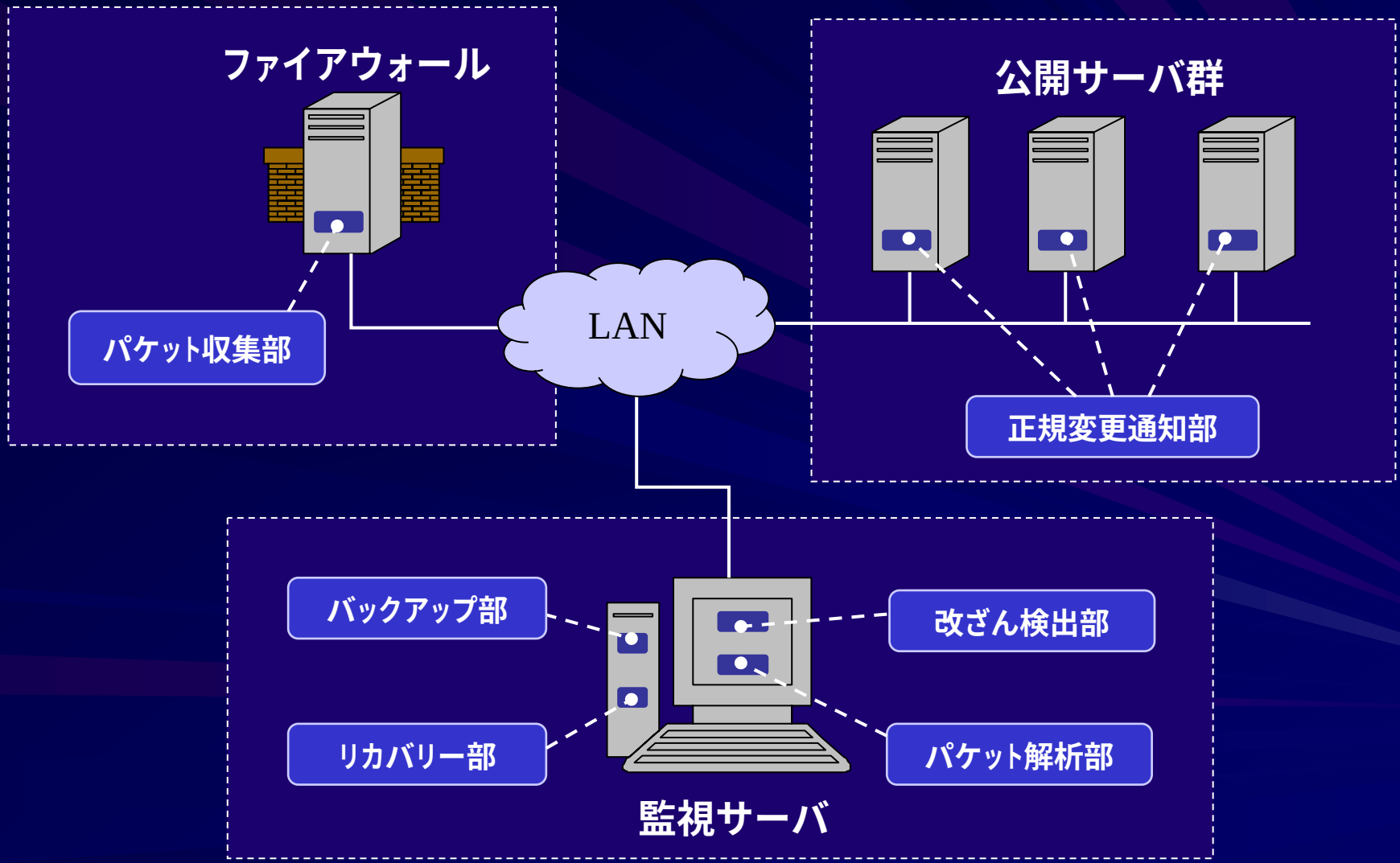
- 不正アクセス手法の多様化
  - すべての攻撃手法へ対応できない
- 不正アクセス手法の進化
  - その進化のスピードに追いつかない

- 完璧なセキュリティシステムの実現は困難
- 外部から不正に侵入される可能性がある

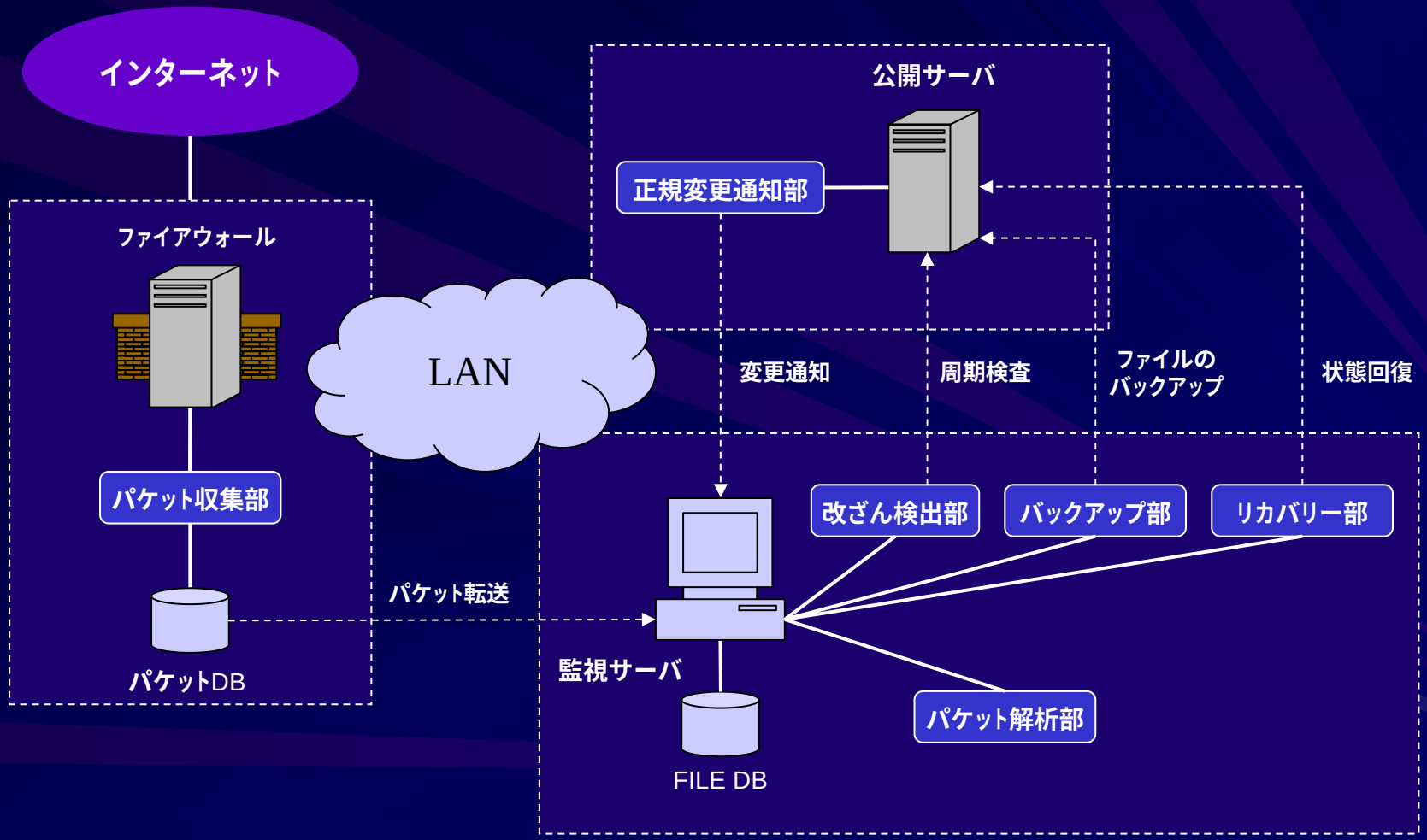
研究目的

- 被害の早期発見、自動回復、原因究明を支援する技術が重要となっている
- ファイルの改ざん問題を対象とした被害回復支援システムの構築
  - 改ざんされたファイルの検出･修復の自動化
  - 独自のパケット解析機構を用いて、ファイルの改ざんに関連するパケットの解析

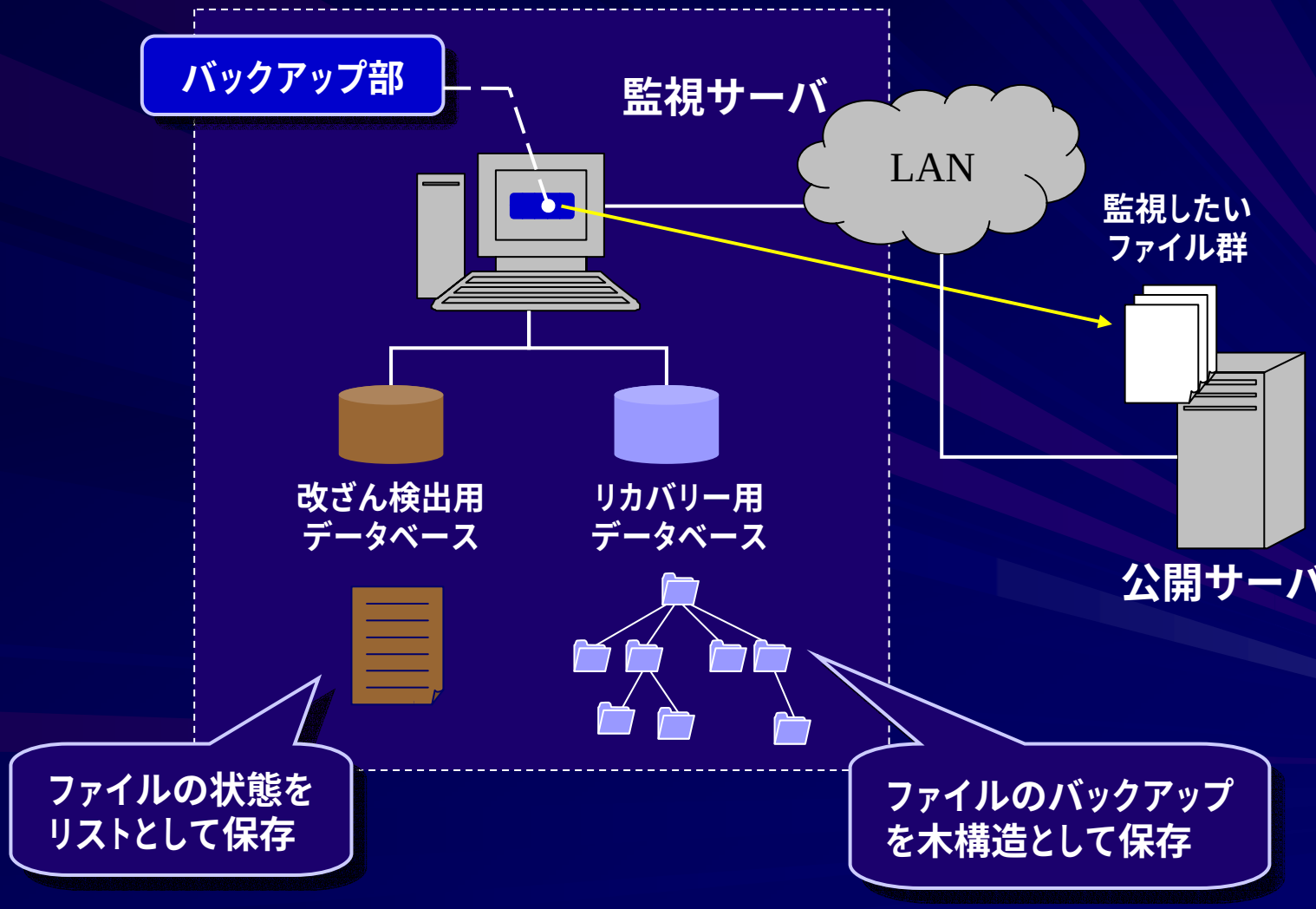
システムの構成


ファイアウォール
パケット収集部
LAN
公開サーバ群
正規変更通知部
バックアップ部
改ざん検出部
リカバリー部
パケット解析部
監視サーバ

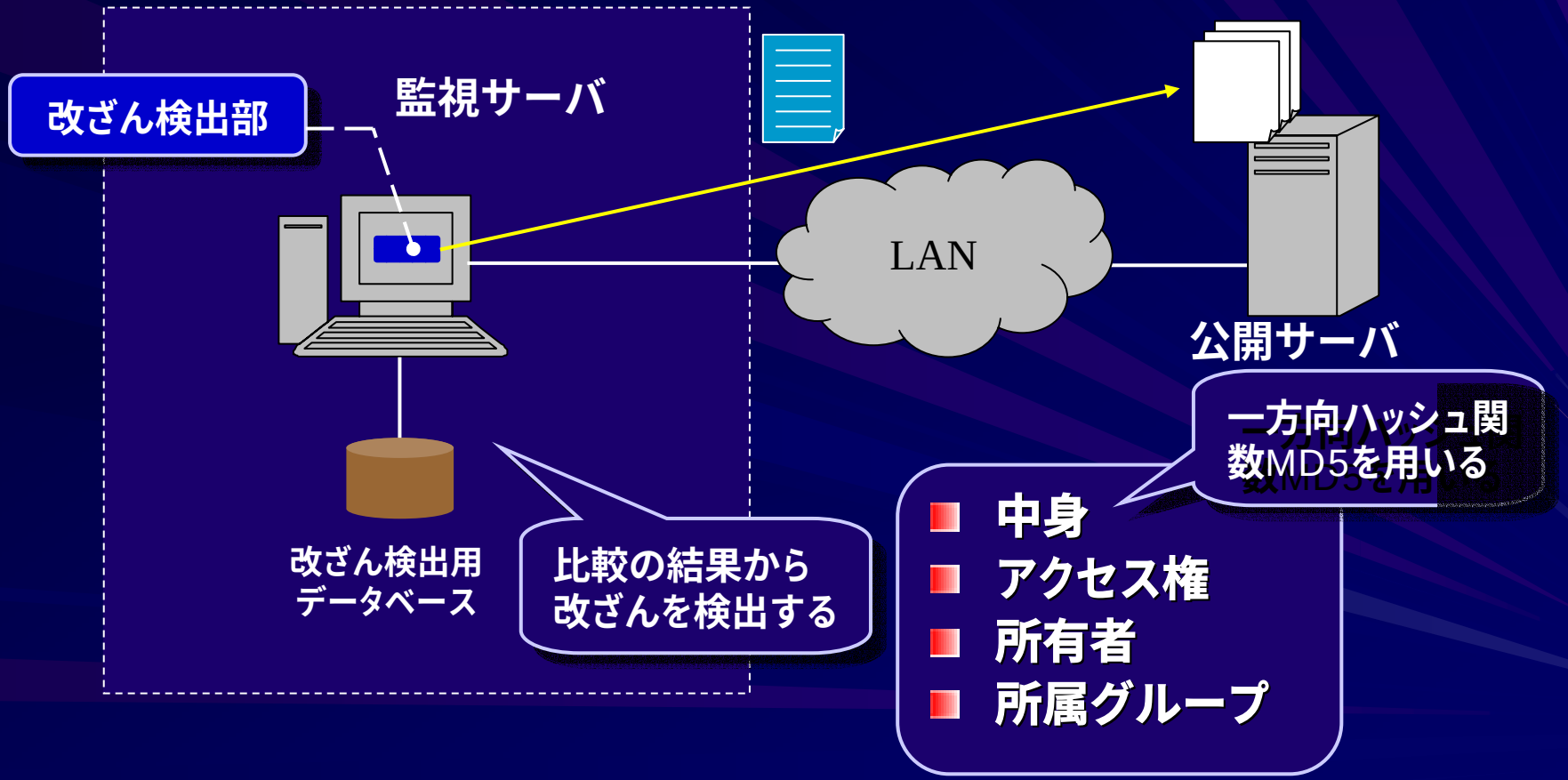
動作の原理


インターネット
公開サーバ
正規変更通知部
ファイアウォール
LAN
変更通知
周期検査
ファイルの
バックアップ
状態回復
パケット収集部
改ざん検出部
バックアップ部
リカバリー部
パケット転送
監視サーバ
パケット解析部
パケットDB
FILE DB

バックアップ部


バックアップ部
監視サーバ
LAN
監視したい
ファイル群
改ざん検出用
データベース
リカバリー用
データベース
公開サーバ
ファイルの状態を
リストとして保存
ファイルのバックアップ
を木構造として保存

改ざん検出部

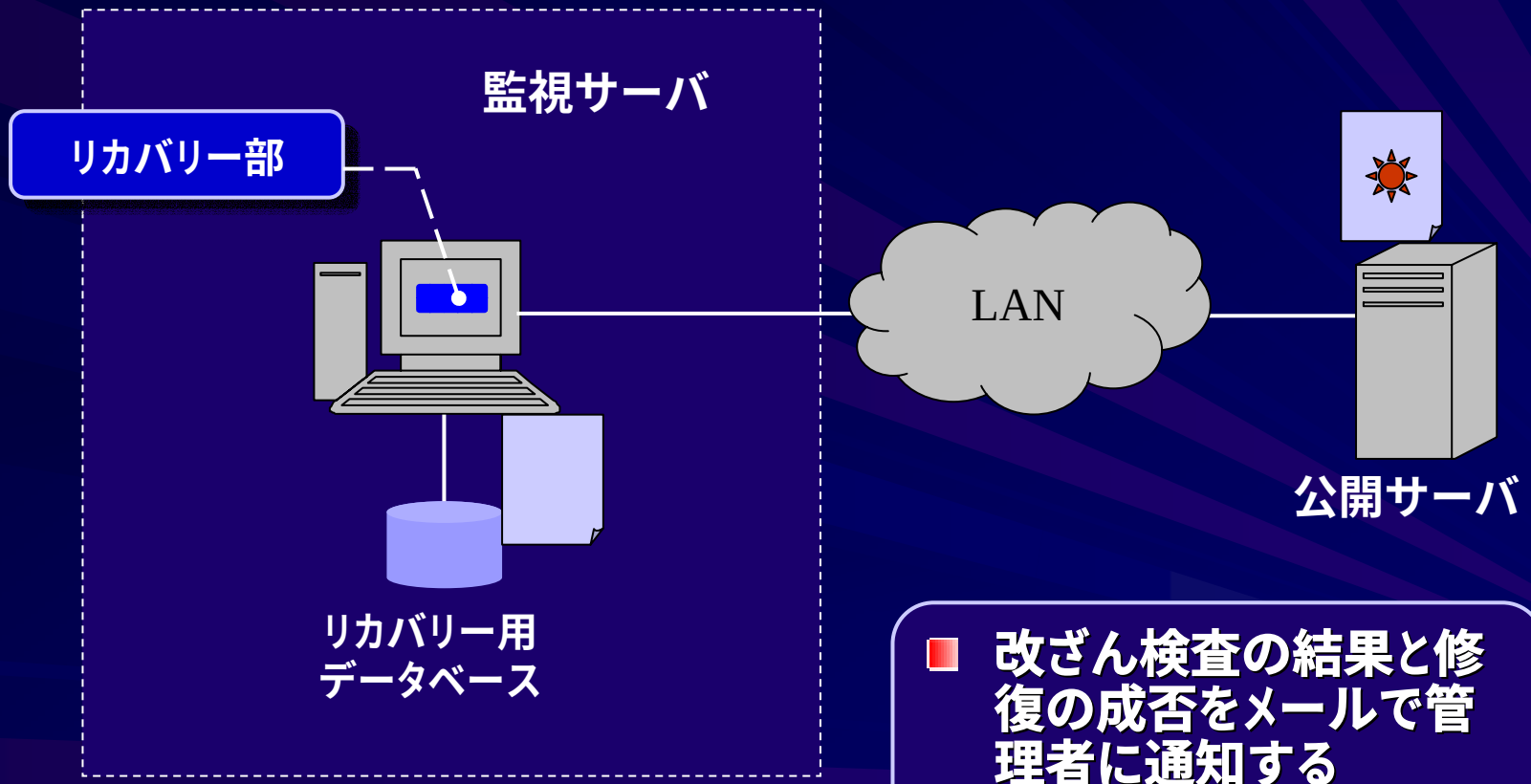

改ざん検出部
監視サーバ
LAN
公開サーバ
改ざん検出用
データベース
比較の結果から
改ざんを検出する
一方向ハッシュ関数MD5を用いる
中身
アクセス権
所有者
所属グループ

MD5 (Message Digest 5)

- MITのRonald Rivest氏によって開発されたメッセージダイジェストを作成するアルゴリズム

- MD5の特徴
  1. 任意の長さの入力メッセージに対して128ビットのハッシュ値を生成する
  2. 出力値から入力メッセージは導けない
  3. 異なる入力メッセージが同じ出力値を生成する確率は殆んどゼロ
  4. 計算は速い

リカバリー部



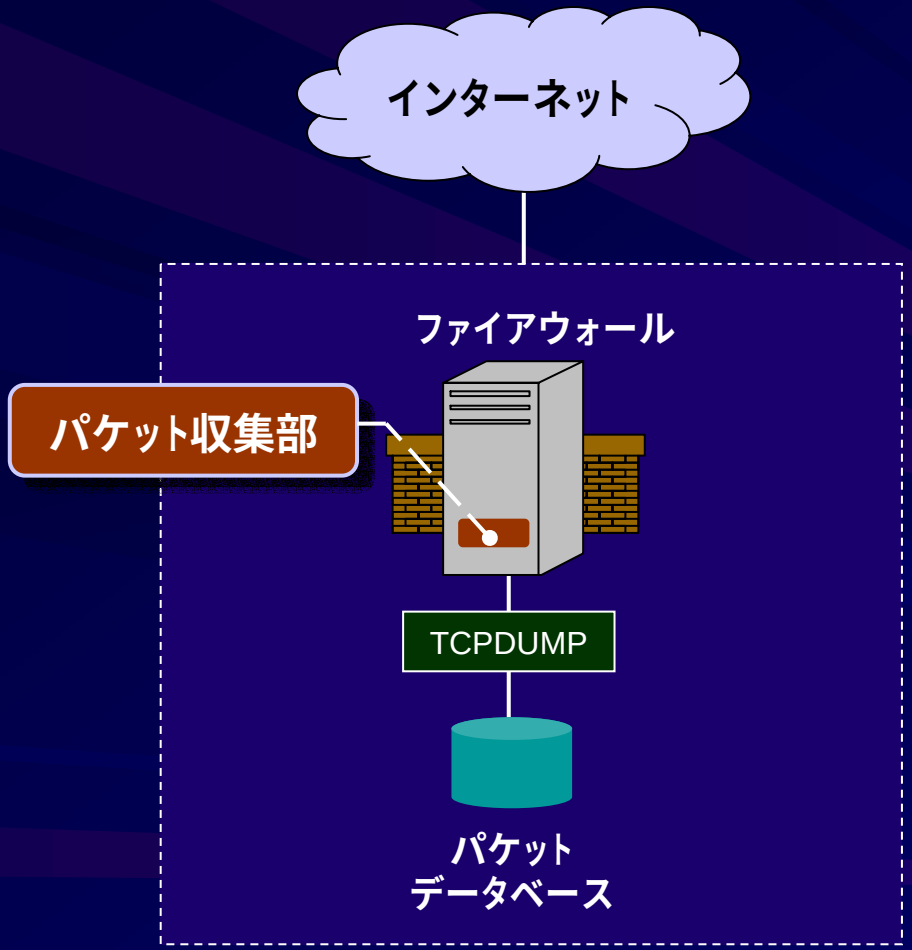
- 改ざん検査の結果と修復の成否をメールで管理者に通知する

パケット解析機構

- 改ざんに関連するパケットの解析
- 事件の早期解決とセキュリティ向上に努める

- パケット解析機構の構成
  - パケット収集部
    - 過去のパケットを保存する
  - パケット解析部
    - 関連あるパケットを解析する

パケット収集部

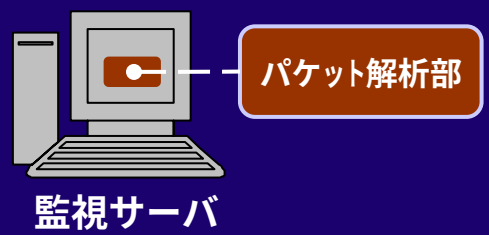


- パケットを保存するファイルの名前に保存された時刻がわかるようにつける
  - 20040117132205.p
- 一定時間ごとに保存するファイル名を変更してパケットを収集
  - 今回は60秒
- 保存されている無用なパケットを定期的に削除する
  - 今回は1日

パケット解析部

- 動作原理
  - 改ざんを検出した際、パケット解析部に通知する
  - 解析に必要なパケットをファイウォールから取得し、パケット解析を行う

- パケットの抽出
  - 改ざんされたファイルの最終更新時刻の前後に流れるパケットのみ
  - 最終更新時刻は改ざんの時刻と一致する
  - この時間帯のパケットには証拠が入っている可能性が高い



パケット解析部の実装

- Snortを用いて実装する
- Snortの概要
  - シグネチャマッチング型のIDS
    - 攻撃のパターンを記述したルールの集合体をシグネチャという
    - シグネチャにマッチしたパケットは不正なパケット
  - Tcpdump形式のパケットを直接解析できる
  - 解析の結果をログファイルに出力される

動的シグネチャ

- 動的シグネチャの作成
  - パケットの中に改ざんされたファイル名が含まれていれば、警告を出す
- ファイルを操作するときに、最低限ファイル名を指定する必要がある
- 既存のシグネチャで検出できない攻撃にもある程度の情報を提供できる

実用化に向ける問題点

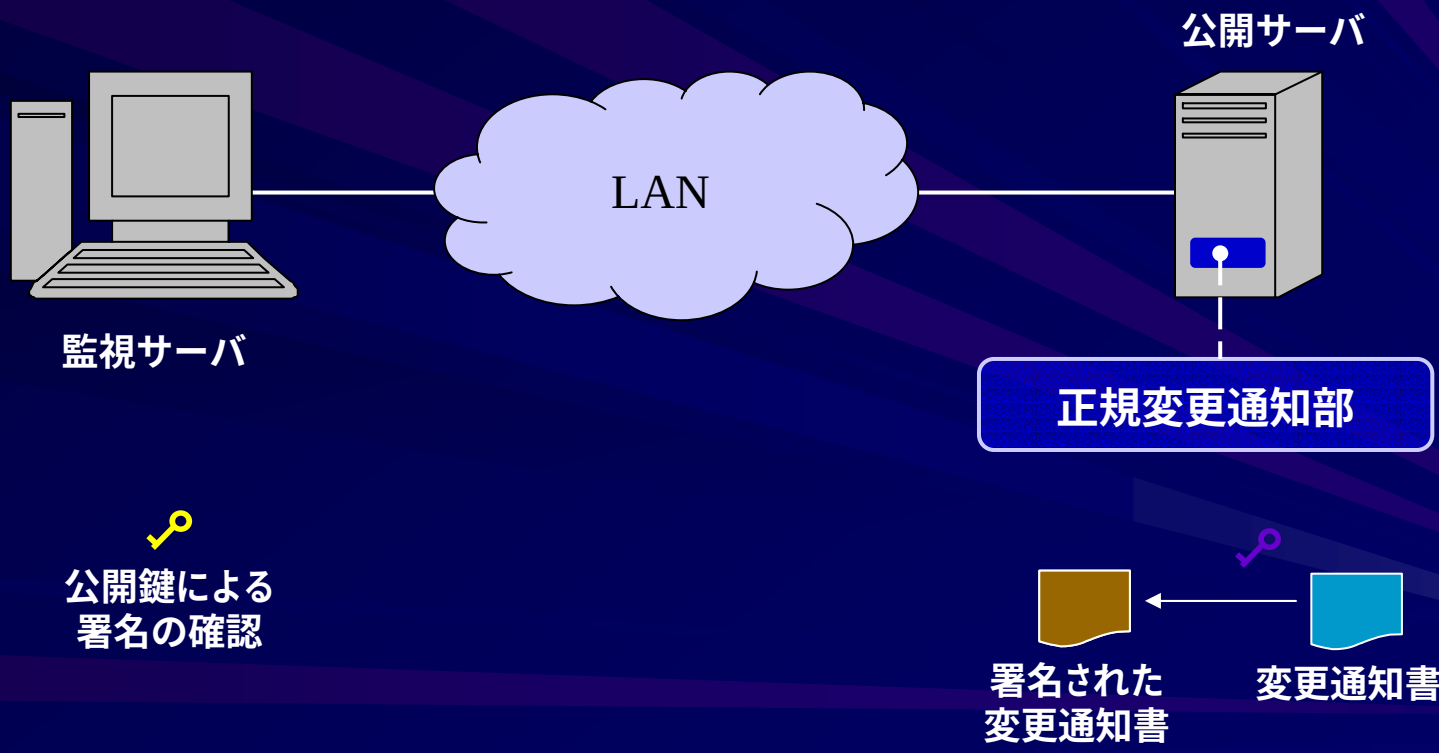
- ファイルの正規変更
  - データベースを更新する必要がある
  - 手動で行うのは不便で、実用化には困難
- 正規変更通知部
  - 正規ユーザによるファイルの変更を監視サーバに伝え、データベースを自動的に更新する
  - ユーザの認証が必要
    - PGPを用いたユーザの認証

PGPとは

- PGP (Pretty Good Privacy)
- メッセージの暗号化を行なうセキュリティ規格
  - 電子メールの暗号化やデジタル署名

- 特徴
  - 暗号化と復号化には公開鍵暗号を使用する
  - メッセージの作成者を保証する
  - メッセージが送信する途中で改ざんされたかどうかを確認できる

PGPを用いた認証



実装

- 提案したシステムはLinux上に実装した
- 実装環境

サーバ	CPU	メモリ	HD	OS
ファイアウォール	Intel Pentium III 600Mhz	128MB	30G	Redhat Linux 9.0
公開サーバ	Intel Celeron 1.4Ghz	256MB	80G	Redhat Linux 9.0
監視サーバ	Intel Pentium 4 1.8Ghz	512MB	40G	Redhat Linux 9.0

動作確認実験

- 実験での設定

監視ファイル数	総容量	検査の実行間隔	改ざんしたファイル数	改ざんしたファイルの総容量
12498	150MB	1時間	10個	1.7MB

- 実験の結果
  - 1回の改ざん検出にはおよそ70秒が掛かった
  - ファイルの修復には9秒掛かった
  - メールによって周期的な検査の動作を確認できた

実験結果への考察

- 実験結果は運用する環境によって異なる
- 改ざん検出にかかる時間を左右する要因
  - コンピュータのスペック
  - 監視するファイル数とその容量
- リカバリーにかかる時間を左右する要因
  - ネットワークの環境
  - 改ざんされたファイル数とその容量

Tripwireとの比較

- Tripwireはファイル改ざんの検出はできるが、修復はできない

- データベースの保管
  - Tripwireはデータベースを公開サーバ上に保存
  - 本システムはデータベースを監視サーバ上に保存する
    - セキュリティの面ではtripwireより優れている

パケット解析への考察

- パケット解析に対する評価は行えていない
  - 攻撃手法は様々である
  - すべてを試すことが不可能
- 改ざんの原因をすべてパケットの解析によって解明できるとは言えない
  - 暗号化されたパケット
  - 潜伏期間のあるプログラム

パケット解析への考察

- 短時間で原因を解明する可能性が高い
  - 改ざんを行った時刻前後のパケットのみを解析
  - それらの中には改ざんに関する証拠が入っている可能性が非常に高い
- 改ざんされたファイルの名前を元に作成されたシグネチャによる解析
  - ファイルに対してどんな不正な操作を行ったかを調べることが可能

時刻の同期

- 時刻同期の必要性
  - パケットの抽出にはファイルの最終更新時刻を用いる
  - 時間にずれがあると、検出率の低下と検査時間の延長につながる
- NTPを用いた時刻の同期を実現

まとめ

- 改ざんの検出、リカバリーおよびパケット解析を自動的に行う被害回復支援システムを構築した
- 実験を通して、システムの動作を確認した
- 諸考察を行い、有効性と実用性を確認した

今後の課題

- パケット解析精度や検出率の向上
  - 複数のIDSやパケット解析ツールにより解析
- 不正なプログラムの検出
  - 正常なサービスをあらかじめ登録し、定期的に比較する