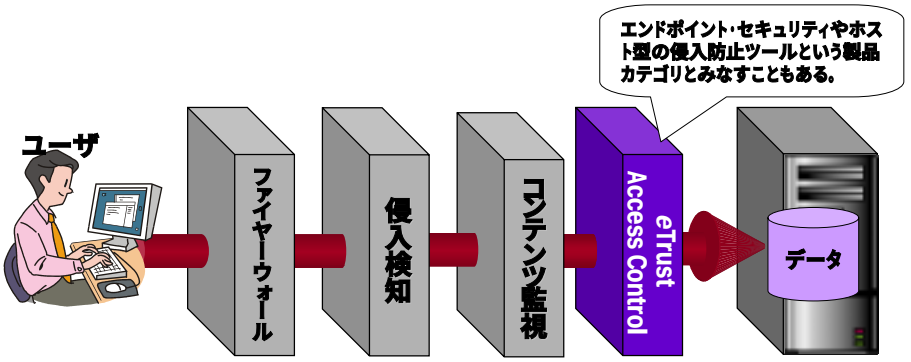
*e*Trust Access Control
デモンストレーション・ハンドブック

コンピュータ・アソシエイツ株式会社

*e*Trust Access Controlとは？

- UNIX(Linux, Windows)のシステムとデータを保護するための**セキュリティ管理インフラ**
  - **カーネルに変更を加えないので、現在稼働しているシステム環境にそのまま適用可能。**
  - **セキュリティポリシーの統合管理を実現**
- **主要機能**
  - **ログイン制御**
  - **パスワードポリシー設定**
  - **ネットワーク制御**
  - **システムリソースの保護**
  - **ファイル／プロセスの保護**
  - **管理権限の分割**
  - **システム監査**
  - **集中管理**

一般的な企業内のセキュリティ対策における *eTrust Access Control*の位置付け



ログイン制御

- ユーザ、グループごとに様々なログイン条件の指定が可能
  - ログイン可能な端末
  - ログイン可能な時間帯
  - ログインに使用できるプログラムを制限
- ユーザIDの失効
  - 例）10日間ログインがなかったユーザIDは自動的に失効させる、管理者ユーザ以外は、休日はログインさせない、など
- SUコマンド制限
  - 例）アプリケーション用ユーザID以外でのrootへのなりかわり制限、など

パスワードルール設定

- 異なるOSに対して、同じパスワードルールの設定が可能。OSよりも詳細なパスワードルールの設定が可能
  - 例）パスワードルールの設定例
  - 大文字を最低2文字以上含める
  - aaaなどの文字の反復使用の禁止
  - 3回ミスをするとパスワードをロックする
  - 旧パスワード3世代分の使用禁止
  - 文字種はアルファベットと数字の混在
  - 有効期限は1ヶ月
  - パスワード使用禁止文字の指定、など

ネットワーク制御

- ホスト型ファイアウォールの機能を実現
  - すべてのTCP/IPサービスをモニタリングし、制御することが可能
    - アクセスの通知と監査ログ
    - アクセス制御
    - ホスト、ネットワーク単位
    - サービス名、ポート番号、ポート範囲単位
    - 曜日と時間の指定
  - outgoingとincomingに対する両方の制限が可能
    - 例）DMZ上の公開サーバ間での通信をアプリケーション間で発生する通信に限定する、など

システムリソースの保護

- **重要なOSファイルに対する改ざんを監視**
  - ファイルの最終更新日時やサイズ情報などの改変を検知する
  - 特権(setuid/setgid)プログラムに対し、置き換えや改ざんが行われた場合、　その実行を禁止する
  - 例) “トロイの木馬”型プログラムの起動抑止
- **バッファオーバーフロー防止**
  - 独自プログラムによりメモリスタックを保護、OSやアプリケーションのセキュリティ・ホールへのアタックを防御する

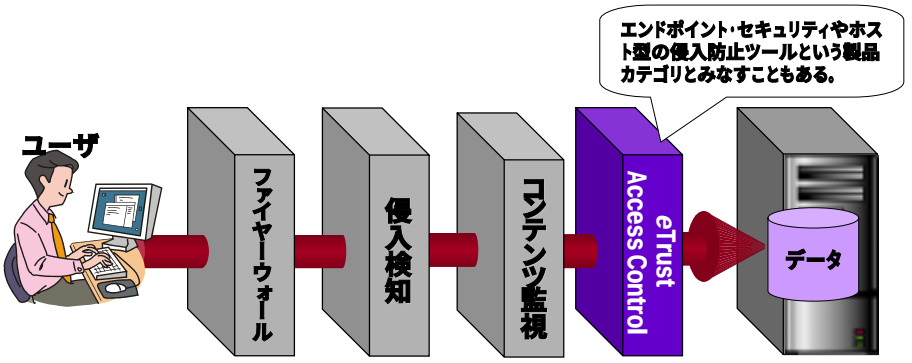
ファイル／プロセスの保護

- **ファイルアクセス制御**
  - ファイル／ディレクトリに対してきめ細かなアクセス権限の設定が可能
  - Read, Write, Execute, Delete, Update, Chown, Chmod, Chdir, ALL, Note…
  - UFS, HPFS, CDFS, FAT, NTFSなど全てのファイルシステムに対応
- **ユーザ／グループ単位でのファイルアクセス制御**
  - 例) アプリケーションのコマンド経由以外でのデータファイルへのアクセス　　不可、など
- **プロセスに対するKillアクセスを制限**
  - OSのネイティブなプロセス
  - 様々なアプリケーション・プロセス
  - eTrust Access Controlのプロセス

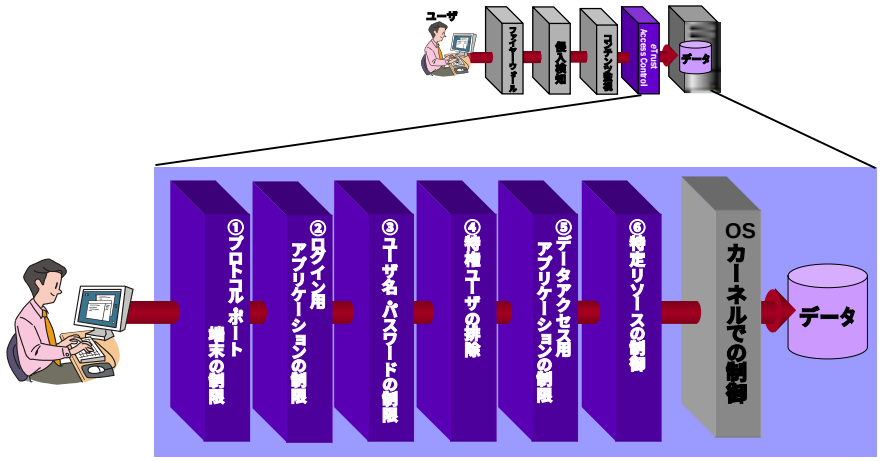
セキュリティポリシーの集中管理

- ポリシー・モデルDBによるセキュリティ・ポリシーの一元管理
  - ユーザ・アカウント、アクセス権の設定を一元管理
  - 複数のマシンの監査ログをセンタサーバにに集約して一元管理
  - マルチプラットフォーム対応

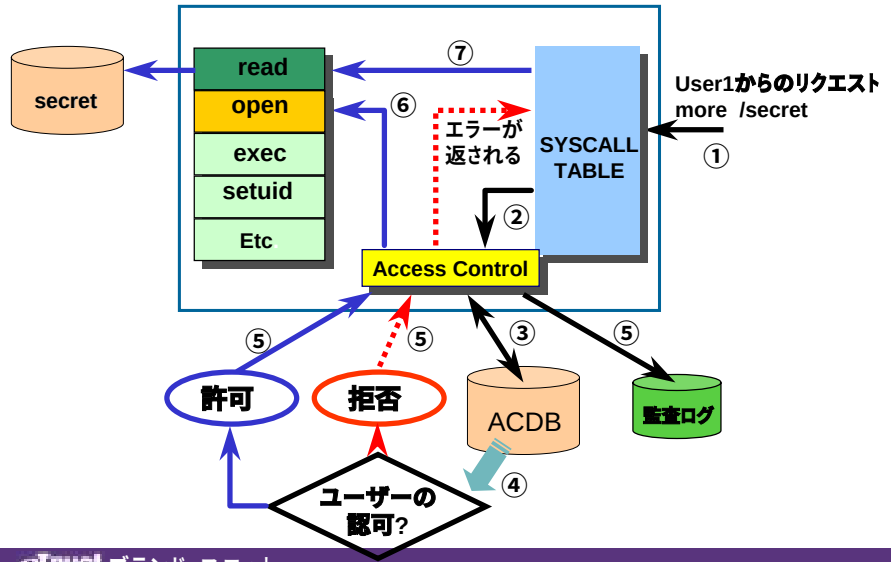
一般的な企業内のセキュリティ対策における *eTrust Access Control*の位置付け



eTrust Access Controlの中身



eTrust Access Controlの仕組み



*e*Trust Access Controlで、なにができるのか、何を防ぐことができるのか？

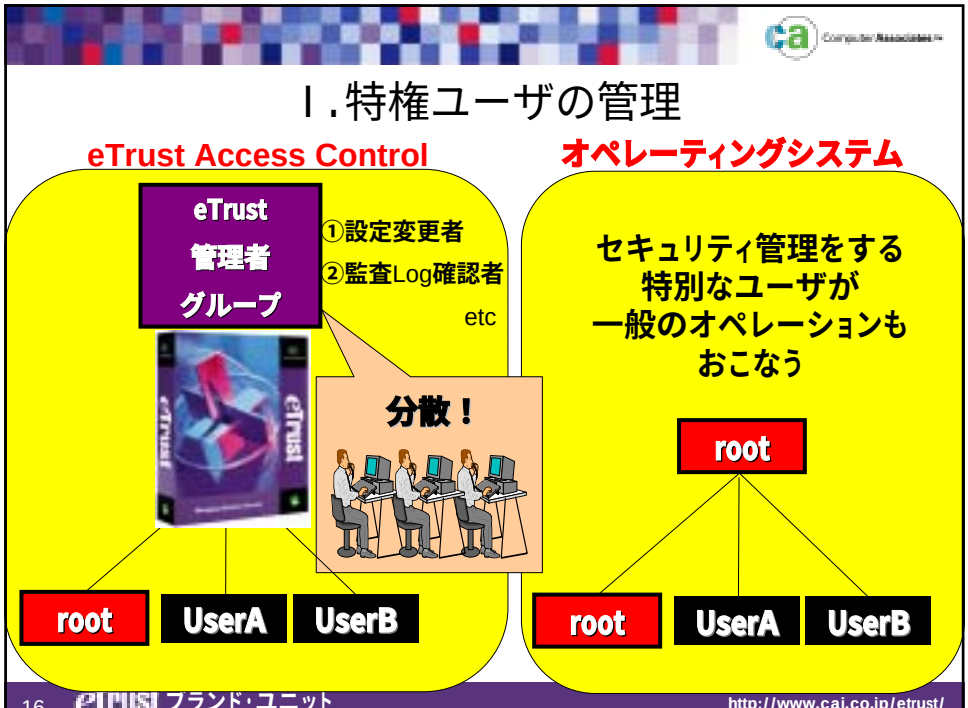
eTrust Access Controlではなにができるのか？

- root（administrator）等の特権ユーザ権限を分割し、必要なユーザに必要な権限だけを与える。
  - 外部からの不正侵入でrootでのアクセスされても、被害を最小限に食い止めることができる
  - アクセスログを確実に残すことができる。（rootであっても自分のアクセスログを消すことはできない）
- ユーザ（特に特権ユーザ）の誤操作によるトラブルを防止することができる。
  - 重要なファイルを誤って「削除」や「上書き」
  - 誤ったコマンドを発行
  - 間違ったサーバーをShut Down
  - サービスやプロセスを停止
- UNIXのOSでは履歴を残すことのできない、ファイルの移動、リネーム、削除、更新の制御とログを残すことができる。
- 複数のマシンのポリシーをリアルタイムに管理運用する

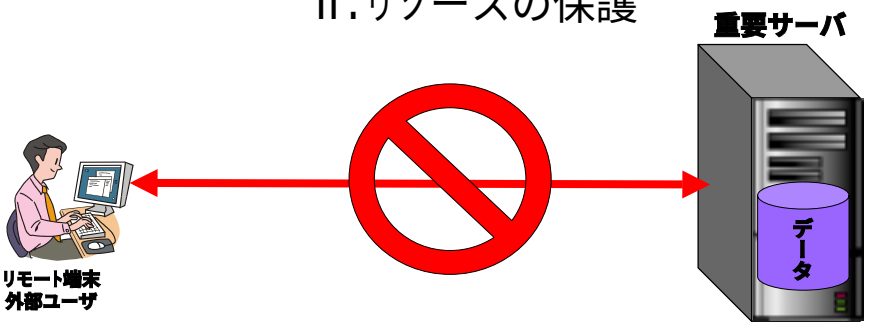
デモで紹介する機能について

I. 特権ユーザの管理
II. リソースの保護
III. 監査ログ
IV. ポリシー･監査の一元管理

Ⅰ.特権ユーザの管理



II.リソースの保護

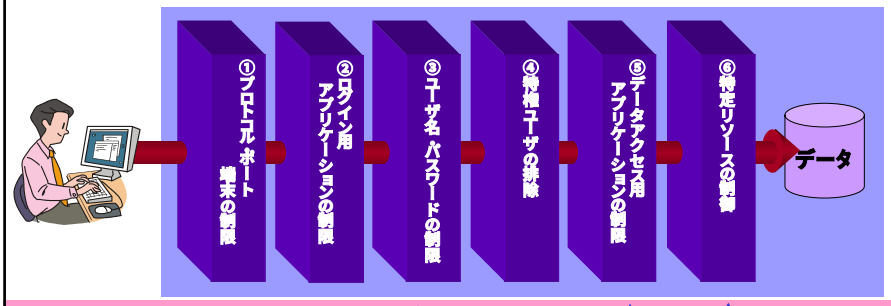


リソース保護の基本“All Deny”（すべての穴を閉じる）

必要なモノだけを許可。

- ポート
- 端末
- 使用するアプリケーションプロトコル
- 利用アプリケーション

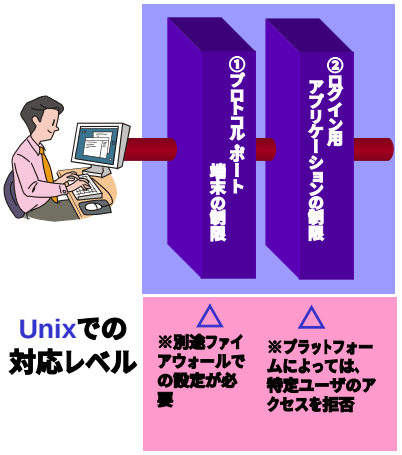
II.リソースの保護：データへのアクセス制御プロセス



Unixでの対応レベル	△	△	△	X	△	△
	※別途ファイアウォールでの設定が必要	※プラットフォームによっては、特定ユーザのアクセスを拒否	※単純に指定したパスワードでの制御ができる		※特定のシェル、アプリケーションの使用制限が出来ない。	※大まかな制限しか出来ない

II.リソースの保護
データへのアクセス制御プロセス：ログインの絞り込み

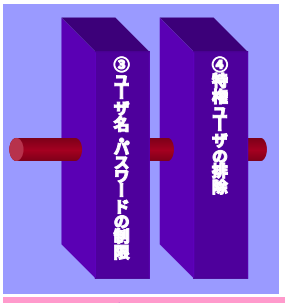
- **①プロトコル･ポート･端末の制限**
  ※全ての穴を閉じた後で…
  - HTTPで
  - Port８０で
  - 端末 aaaaから
- **②ログインアプリケーションの制限**
  ※全てを使用不可にした後で…
  - ローカルLoginに限定可能
  - Telnet,rlogin等のLogin
    を不可とする事が可能



Unixでの対応レベル	△ ※別途ファイアウォールでの設定が必要	△ ※プラットフォームによっては、特定ユーザのアクセスを拒否

II.リソースの保護　データへのアクセス制御プロセス
：ユーザの権限とパスワードの制御

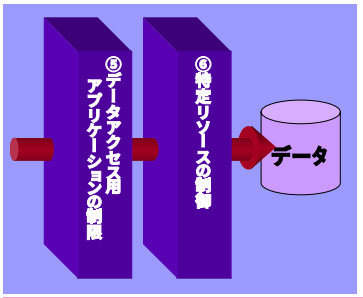
- **③ユーザパスワードの管理**
  - ユーザ名と一致させない
  - 特定文字列での条件付け
    - 英数字、特殊文字を強制的に使用させる
  - 最長有効期間･最短有効期間の設定
- **④OSに依存した特別な権限の排除**
  - rootに特別な権限をもたせない
  - 適切なユーザに権限を与える
  - suでのユーザスイッチの制限



Unixでの対応レベル	△ ※単純に指定したパスワードでの制御ができる	X

II.リソースの保護　データへのアクセス制御プロセス
:アプリケーション･リソースへの制限

- ⑤データアクセス用アプリケーションの制限
  - 特定コマンドのみアクセス
- ⑥特定リソースの制限
  - 資産ファイルファイル
  - 設定ファイル
  - プロセス
  - デーモン

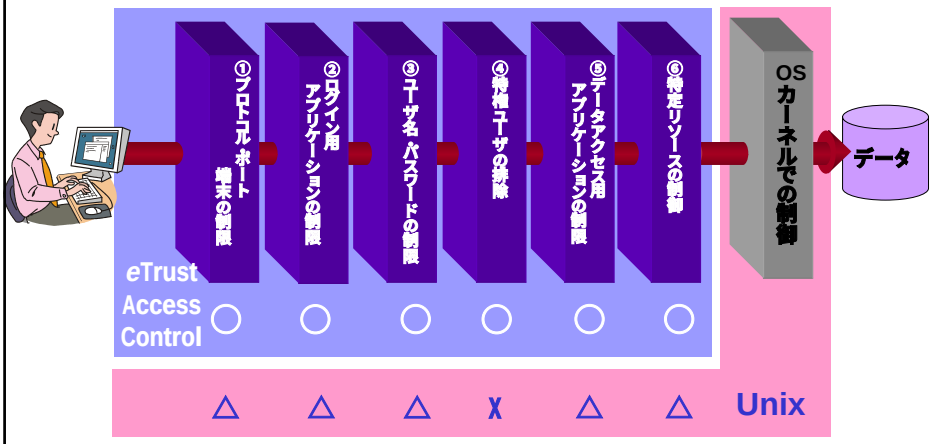


Unixでの対応レベル

△ ※特定のシェル、アプリケーションの使用制限が出来ない。

△ ※大まかな制限しか出来ない

II.リソースの保護　データへのアクセス制御プロセス
eTrust Access ControlとUnixの比較



III．監査ログ

- ***e*Trust Access Control**でなにがとれる
  - 4W1H
    - What　**システムリソースに対して**
    - Who　**誰が**
    - When　**いつ**
    - Where　**どのマシンから**
    - How　**どのプログラムによって**

III．監査ログ
*e*Trust Access Controlの詳細な追跡

	Information Title	内容
	Access	アクセスのタイプ
	Administrator	選択したレコードの中でコマンドを実行した管理者名
	Class	コマンドを実行して編集したクラス
	Command	アクセサが入力したコマンド
	Command Type	選択したレコードの中で使われたコマンドのタイプ
	Daemon	開始・終了したコマンド名
いつ	Date	コマンドを実行した日付（日・月・年）
	Details	実行したコマンドの詳細
	Effective user ID	プロセスの実効UID
	Event type	発生イベントのタイプ
	File	アクセスしたファイル名
	Host name	リモートからの接続が実行(試み)が行われた時のリモートホスト名
だれが	Login user ID	プロセスのUNIX UID
何に	Object	アクセスしたリソース
なにを介して	Program	イベントを実行するのに使われたプログラム
	Real user ID	プロセスの実UID（eACが管理している）
	Resource	アクセスもしくは更新したリソース名
	Service	リモートホストからの要求があったサービス名
	Status	ユーザアクセスした（試みた）時に起こった状態　例）Logout
どこから	Terminal	イベントが実行されたターミナル
	Time	コマンドを実行した時間
	Trace Information	トレースに関する情報
	User name	コマンドを実行したアクセサ名

4W1H

ログインした時のオリジナル　ユーザID

III．監査ログ　監査情報の違い

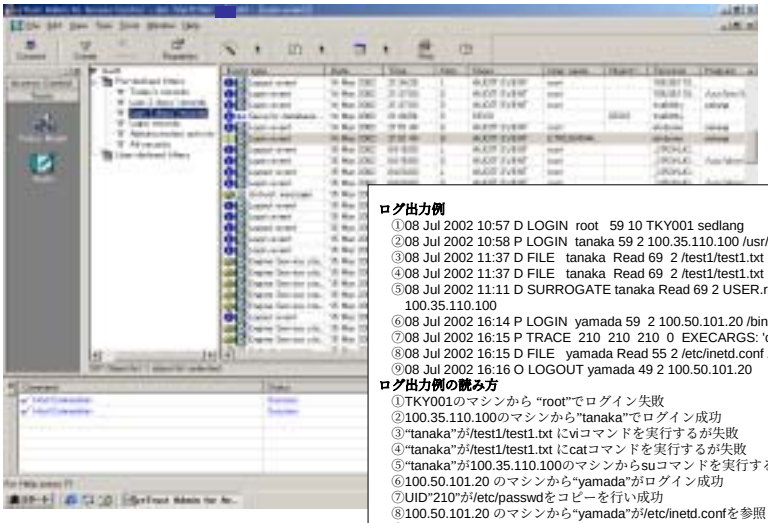
例) ユーザが次のオペレーションを行った場合…

1. ユーザが“hanako”でログイン失敗
2. ユーザが“taro”でログイン成功
3. “su”で“root”に切り替わる
4. 特定ファイル</JINJI>にアクセス成功
5. 特定ファイル</KEIRI>にアクセス失敗

- UNIX上での履歴
  1. “hanako”ユーザのログイン失敗履歴は残さない　〔※HP-UXのみ履歴を残す〕
  2. “taro”ユーザのログイン成功履歴を残す
  3. ”su”で切り替わった後は、rootで履歴を残す
  4. ファイルアクセス成功の履歴は残さない
  5. ファイルアクセス失敗履歴は残さない

- eTrust Access Control上での履歴
  1. “hanako”ユーザのログイン失敗履歴を残す
  2. tarooがログインしたことを履歴に残す
  3. ”su”で切り替わった後も“taro”で履歴を残す
  4. ”taro”で</JINJI>ファイルアクセス成功履歴を残す
  5. ”taro”で</KEIRI>ファイルアクセス失敗履歴を残す

III．監査ログ　サンプル



ログ出力例

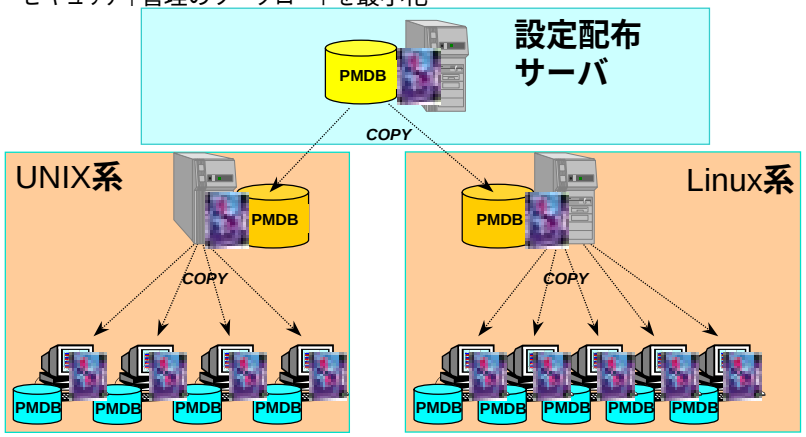
①08 Jul 2002 10:57 D LOGIN root 59 10 TKY001 sedlang
②08 Jul 2002 10:58 P LOGIN tanaka 59 2 100.35.110.100 /usr/bin/login
③08 Jul 2002 11:37 D FILE tanaka Read 69 2 /test1/test1.txt /bin/ vi
④08 Jul 2002 11:37 D FILE tanaka Read 69 2 /test1/test1.txt /bin/cat
⑤08 Jul 2002 11:11 D SURROGATE tanaka Read 69 2 USER.root /bin/su
100.35.110.100
⑥08 Jul 2002 16:14 P LOGIN yamada 59 2 100.50.101.20 /bin/login
⑦08 Jul 2002 16:15 P TRACE 210 210 210 0 EXECARGS: 'cp /etc/passwd *'
⑧08 Jul 2002 16:15 D FILE yamada Read 55 2 /etc/inetd.conf /bin/cat 100.50.101.20
⑨08 Jul 2002 16:16 O LOGOUT yamada 49 2 100.50.101.20

ログ出力例の読み方

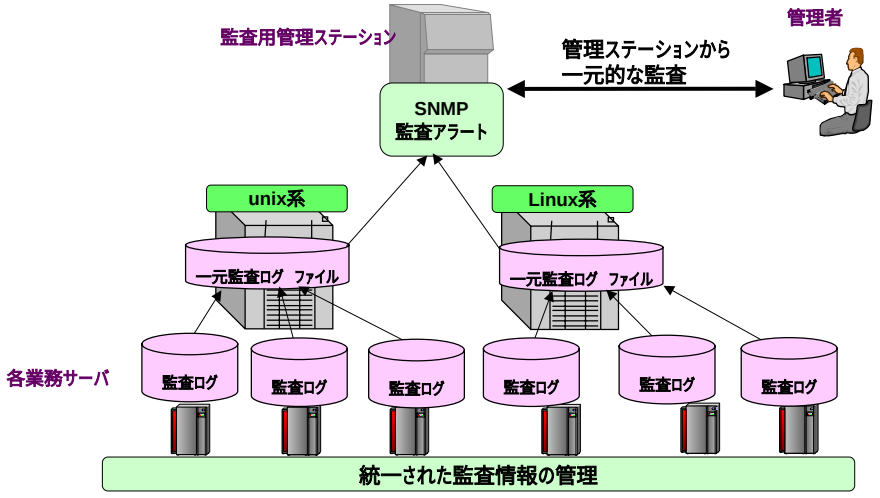
①TKY001のマシンから “root”でログイン失敗
②100.35.110.100のマシンから”tanaka”でログイン成功
③“tanaka”が/test1/test1.txt にviコマンドを実行するが失敗
④“tanaka”が/test1/test1.txt にcatコマンドを実行するが失敗
⑤“tanaka”が100.35.110.100のマシンからsuコマンドを実行するが失敗
⑥100.50.101.20 のマシンから“yamada”がログイン成功
⑦UID”210”が/etc/passwdをコピーを行い成功
⑧100.50.101.20 のマシンから“yamada”が/etc/inetd.confを参照しようと試みたが失敗
⑨100.50.101.20 のマシンから“yamada”がログアウト

Ⅳ. ポリシー・監査の一元管理
セキュリティポリシーの一元管理

- 設定の一元配布。親マシンで設定したルールは、子マシンにもコピー生成される。「異種OS」環境でも可能。
- セキュリティ管理のワークロードを最小化



Ⅳ. ポリシー・監査の一元管理
ユーザアクセスログの一元管理

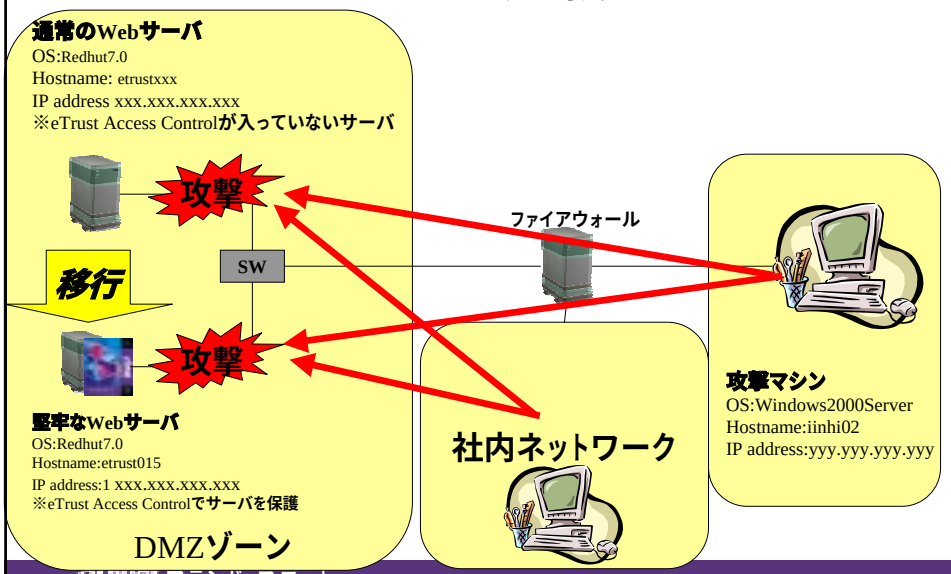


eTrust Access Controlの4つの機能

- ① 特権ユーザの管理
- ② リソースの保護
- ③ 監査ログ
- ④ ポリシー・監査の一元管理

デモを，ご覧下さい。

デモの想定環境



デモ実施の流れ：アクセスパターン

例1）外部ユーザの不正アクセスパターン

ポートスキャン → パスワード解析 → 管理者権限の取得 → システム環境改竄

入手した管理者パスワードによりシステムへ侵入

- 設定ファイルの改竄
- デーモンの停止
- Webコンテンツの改竄
- 痕跡の消去

一般ユーザでログイン → 管理者権限でログイン → システム環境操作 → 不正アクセス 情報漏洩 オペレーションミス

- Root（管理者）でのデーモン停止
- Root（管理者）でのファイル削除
- Root（管理者）でのオペレーションミス

例2）オペレータの不正アクセスパターン

通常のWebサーバ　デモの動き

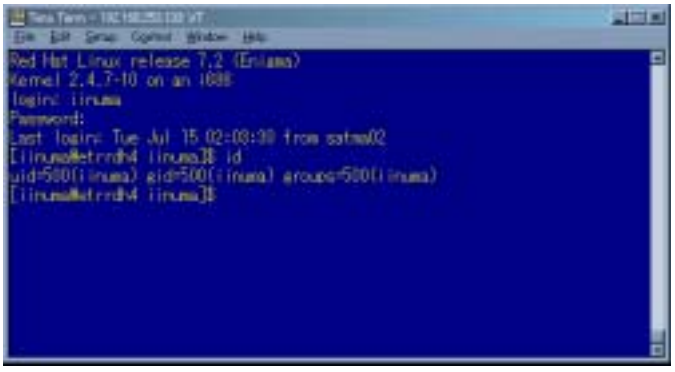
①最初の状態

- **Webサーバが正常に稼動している**
- **サービスが起動している**
- **Webページが正しい情報である**

通常のWebサーバ　デモの動き

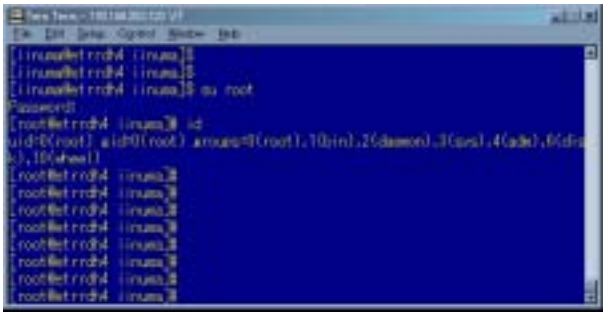
※一般ユーザでアクセスがすでにできることを前提にします

②一般ユーザ　iinuma
でログイン



通常のWebサーバ　デモの動き

③一般ユーザからOSの
スーパーユーザ“root”
になりかわる



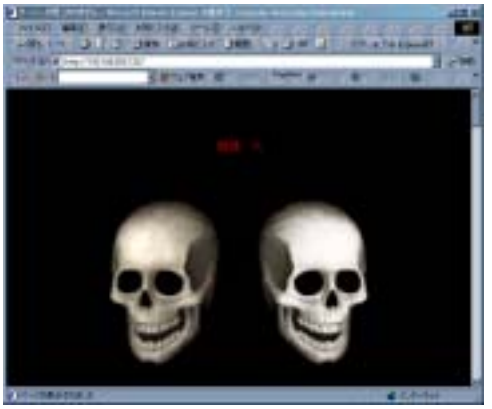
④OSのスーパーユーザ“root”で

正しいファイルに不正アクセスをする
正しいファイルを改竄する
正しいファイルを変更する

通常のWebサーバ　デモの動き

⑤その結果

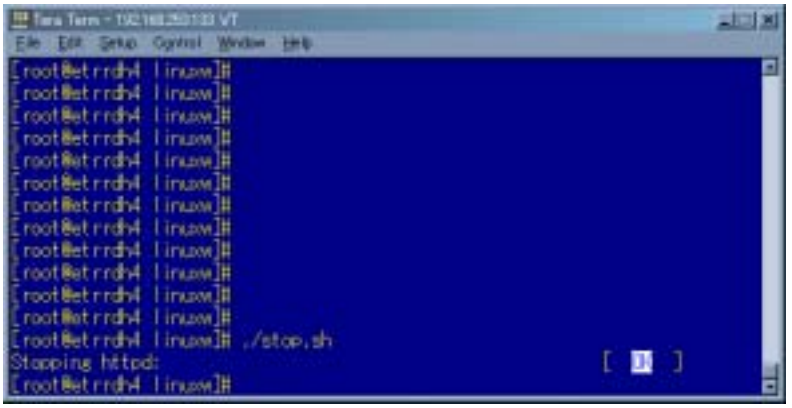
ファイルが変更されている
ファイルが改善させてしまう
誤った情報を外部に提供してしまう



通常のWebサーバ　デモの動き

⑥OSのスーパーユーザ “root”で

Webサービスを停止する



通常のWebサーバ　デモの動き

⑦その結果

外部へサービスが提供できなくなる

⑧結論
適切なユーザと権限をもった場合でも

不正アクセスは発生する
サービスは停止する
オペレーションミスは発生する

eTrust Access Control導入サーバ　デモの動き

①最初の状態

・Webサーバが正常に稼動している
・サービスが起動している
・Webページが正しい情報である

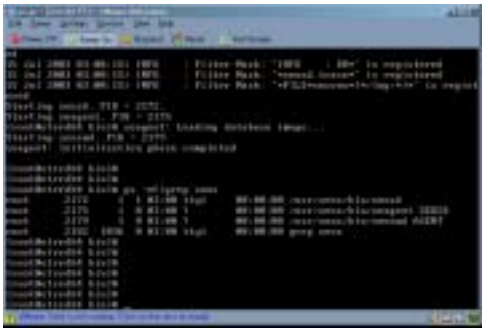
eTrust Access Control導入サーバ　デモの動き

②eTrust Access Control を起動

・HTMLファイルのコンテンツを保護

・サービスの制御の保護

・アプリケーションログの保護

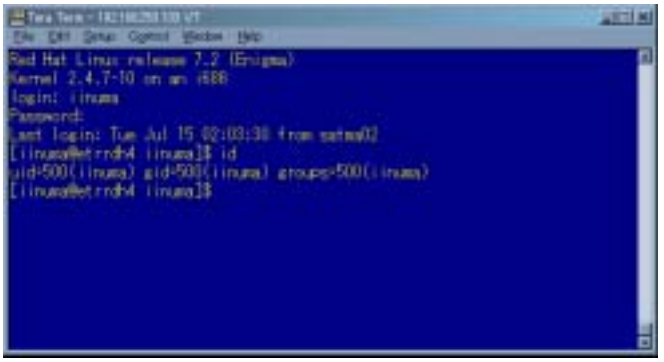


・OSのスーパーユーザ“root”ではアクセスさせていない
・Webコンテンツ管理者にのみファイルの書き込みを許可
・Webアプリケーション管理者にのみサービスの起動・停止を許可

eTrust Access Control導入サーバ　デモの動き

※一般ユーザでアクセスがすでにできることを前提にします

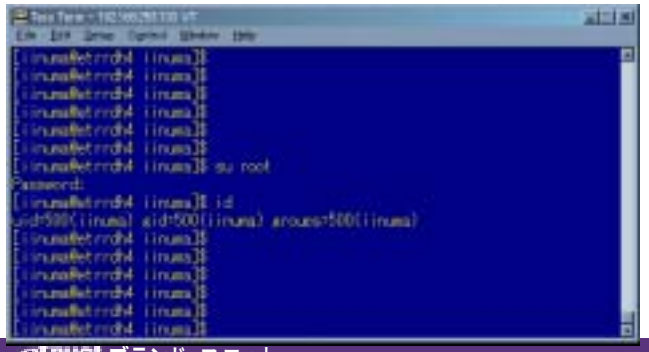
③一般ユーザ　iinuma
でログイン



eTrust Access Control導入サーバ　デモの動き

④一般ユーザから
OSのスーパーユーザ“root”に切り替えようとする

⑤eTrust Access Controlで
特定ユーザからのみスーパーユーザへの切り替えを許可
一般ユーザ“iinuma”からスーパーユーザへの切り替えは拒否

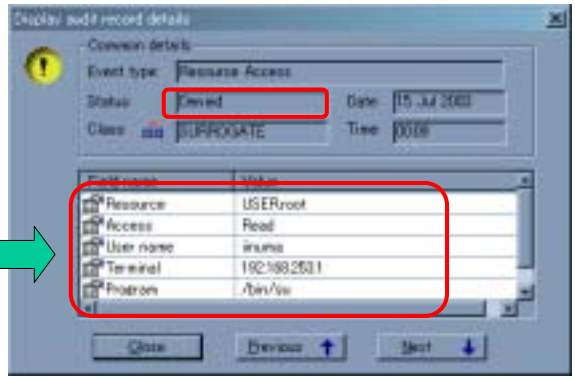


eTrust Access Control導入サーバ　デモの動き

⑥eTrust Access Controlで以下のオペレーションに関する監査ログを確認

特定ユーザからのみスーパーユーザへの切り替えを許可
一般ユーザ“iinuma”からスーパーユーザへの切り替えは拒否

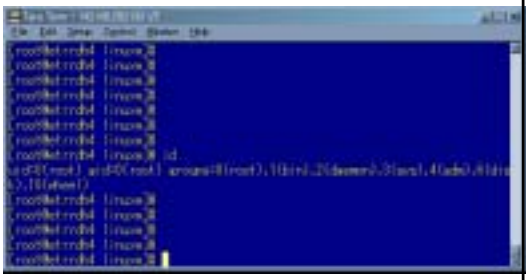
User :iinuma
が
/bin/su
を使用し
Root
になることを防御



eTrust Access Control導入サーバ　デモの動き

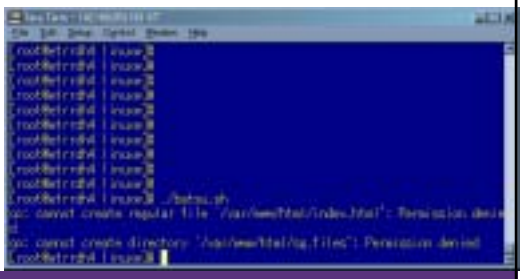
⑦eTrust Access Control
で許可されているユーザが

スーパーユーザ“root”に
切り替わる



⑧OSのスーパーユーザ“root”
で以下のオペレーションを試みる

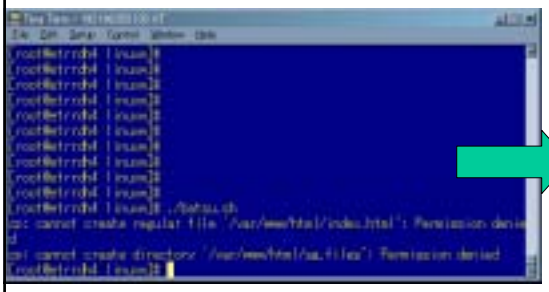
正しいファイルに不正アクセスをする
正しいファイルを改竄する
正しいファイルを変更する



eTrust Access Control導入サーバ　デモの動き

⑧OSのスーパーユーザ“root”で以下のオペレーションを試みる

正しいファイルに不正アクセスをする
正しいファイルを改竄する
正しいファイルを変更する



eTrust Access Controlにより

- **OSのスーパーユーザのオペレーションを制御**
- **Webコンテンツ管理者にのみファイルの書き込みを許可**

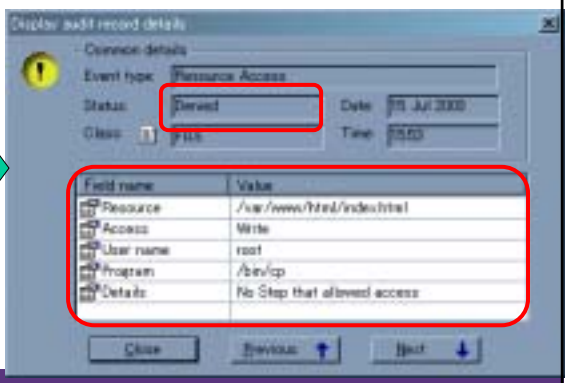
eTrust Access Control導入サーバ　デモの動き

⑨eTrust Access Controlで以下のオペレーションに関する監査ログを確認

Webコンテンツ管理者からのファイルへのアクセスを許可

スーパーユーザ“root”からのファイルへのアクセスを拒否

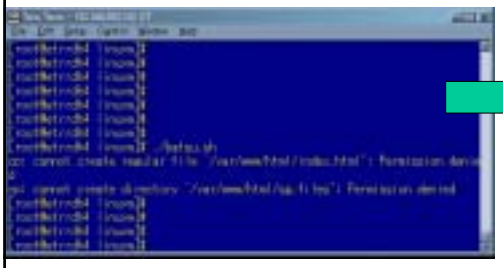
User :root
が
HTMLのコンテンツ
を
書き込み
することを
拒否



eTrust Access Control導入サーバ　デモの動き

⑩OSのスーパーユーザ “root”で以下のオペレーションを試みる

Webサービスを停止する



eTrust Access Controlにより

・OSのスーパーユーザ
　でのファイルアクセス制御
・Webアプリケーション管理者
　にのみサービスの起動・停止を許可

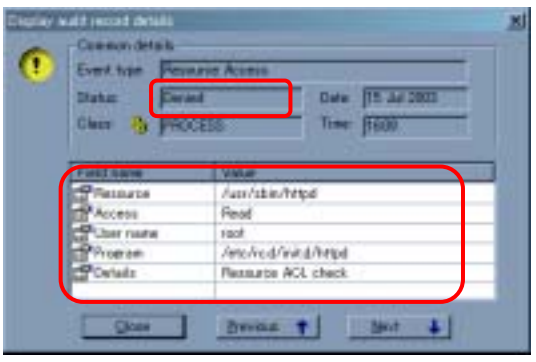
eTrust Access Control導入サーバ　デモの動き

⑪eTrust Access Controlで以下のオペレーションに関する監査ログを確認

Webアプリケーション管理者でのWebサービスの起動・停止許可

スーパーユーザ“root” でのWebサービスの起動・停止を拒否

User :root
が
Webサービス
の
停止すること
を
拒否



eTrust Access Control導入サーバ　デモの動き

⑫結論

Webアプリケーションのサービスは、正常に稼動する

Webコンテンツは、正しい情報を保持する

つまり

- **本来の適切なユーザにのみアクセス権を与える**
- **不正アクセスを防御する**
- **情報漏洩を防御する**

デモ：不正ログインの保護および監査

例1) 外部ユーザの不正アクセスパターン

ポートスキャン → パスワード解析 → 管理者権限の取得 → システム環境改竄

特定Host、ユーザ、アプリケーションプロトコルのログイン制御

suコマンドの制御
どのユーザからどのユーザへのスイッチがOKか？

入手した管理者パスワードによりシステムへ侵入

- 設定ファイルの改竄
- デーモンの停止
- Webコンテンツの改竄
- 痕跡の消去

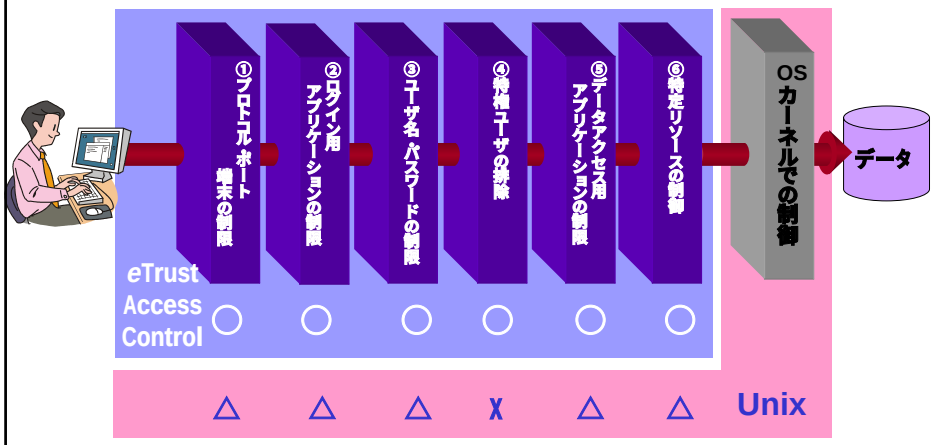
一般ユーザでログイン → 管理者権限でログイン → システム環境操作 → 不正アクセス 情報漏洩 オペレーションミス

この矢印ステップを制御する

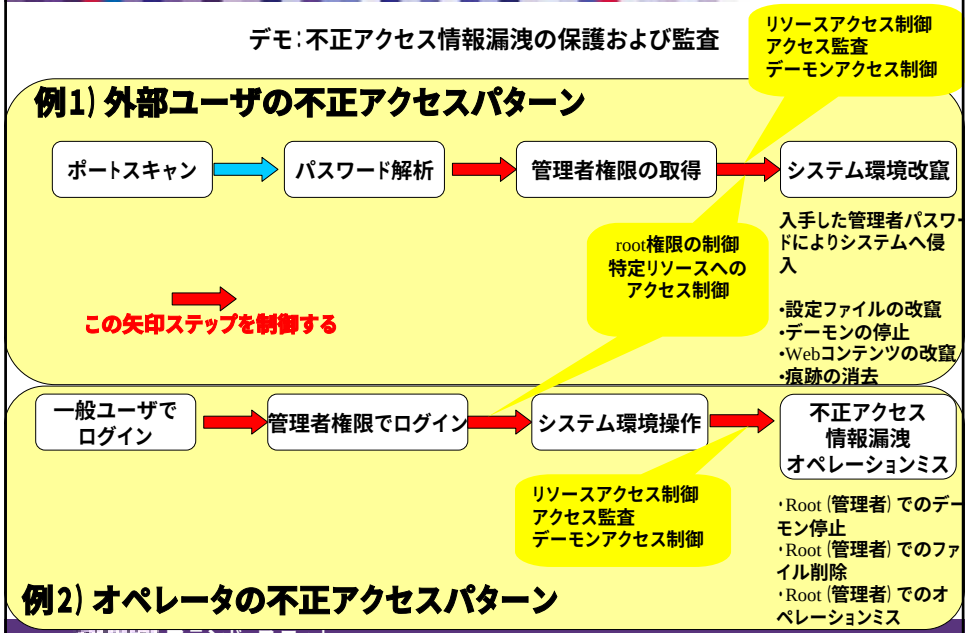
- Root（管理者）でのデーモン停止
- Root（管理者）でのファイル削除
- Root（管理者）でのオペレーションミス

例2) オペレータの不正アクセスパターン

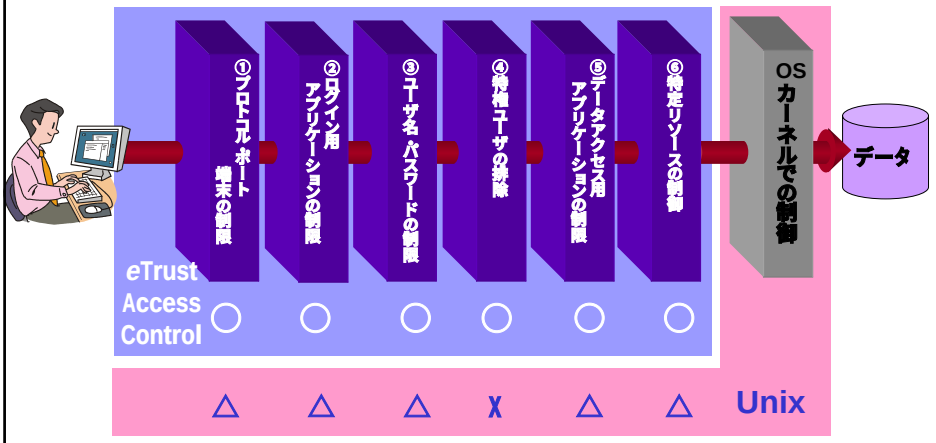
デモで使用するeTrust Access Control項目①～④



デモ：不正アクセス情報漏洩の保護および監査



デモで使用するeTrust Access Control項目④～⑥



制御ポイント →
この矢印ステップを制御する

リソースアクセス制御
アクセス監査
デーモンアクセス制御

例1) 外部ユーザの不正アクセスパターン

ポートスキャン → パスワード解析 → 管理者権限の取得 → システム環境改竄

特定Host、ユーザ、アプリケーションプロトコルのログイン制御

suコマンドの制御
どのユーザからどのユーザへのスイッチがOKか？

root権限の制御
特定リソースへのアクセス制御

入手した管理者パスワードによりシステムへ侵入

- 設定ファイルの改竄
- デーモンの停止
- Webコンテンツの改竄
- 痕跡の消去

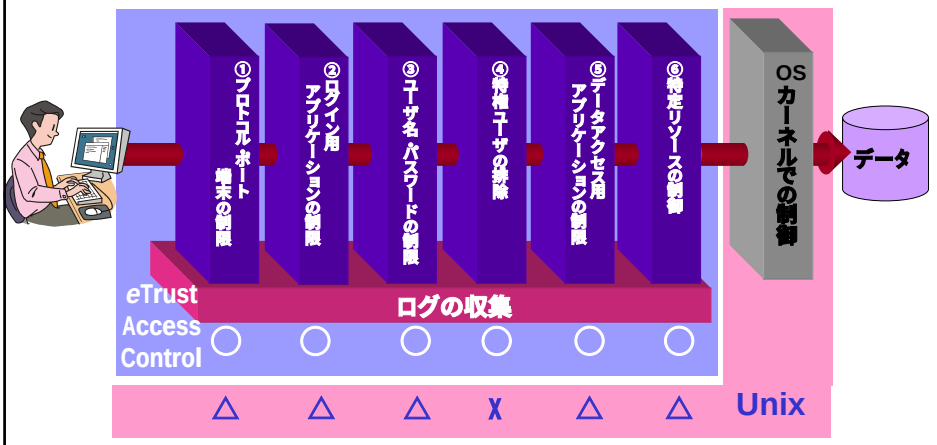
一般ユーザでログイン → 管理者権限でログイン → システム環境操作 → 不正アクセス 情報漏洩 オペレーションミス

リソースアクセス制御
アクセス監査
デーモンアクセス制御

- Root（管理者）でのデーモン停止
- Root（管理者）でのファイル削除
- Root（管理者）でのオペレーションミス

例2) オペレータの不正アクセスパターン

すべてのプロセスのログの収集

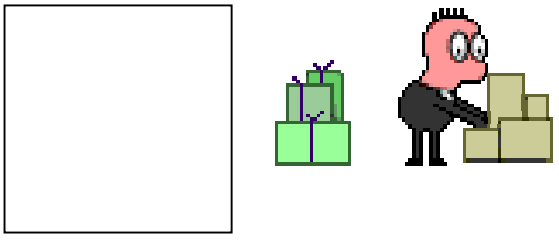


デモ終了

*e*Trust Access Control　○×クイズ

- ***e*Trust Access Control**は、金融機関や官公庁といった特定業種、特定業務向けのツールである。

○か×か？

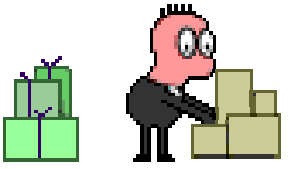


- ***eTrust Access Control*** は、root、administratorといった管理者ユーザにアクセス制限をかけることができる。

○か×か？

- ***eTrust Access Control*** は、OracleやDB2の個々のテーブルやカラムに　アクセス制御をかけることができる。

○か×か？

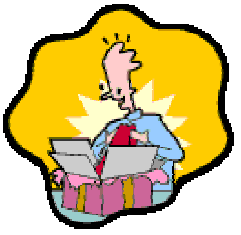


- ***e*Trust Access Control**では“su”コマンドでrootに成り代わったとしてもそのアクセスログはもとのログインユーザIDで記録することができる。

○か×か？

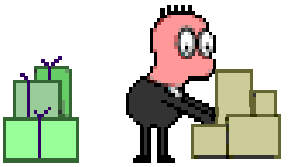
- IDS（侵入検知）をすでに導入していれば、***e*Trust Access Control**は必要ない。

○か×か？



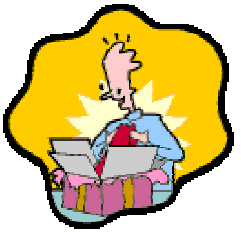
- あるファイルに対して、***eTrust Access Control***ではアクセス許可、OS上ではアクセス拒否の権限設定がされている場合は、ユーザはこのファイルにアクセスすることができない

○か×か？



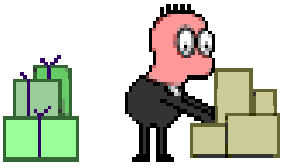
- ***eTrust Access Control***がサポートしているのは、UNIX、LINUXだけである。

○か×か？



- **eTrust Access Control**は、複数のOSのセキュリティポリシーとアクセスログを集中管理することができる。

○か×か？



- **eTrust Access Control**を導入するとセキュリティの管理の作業が増える

○か×か？