セキュリティ強化した RPA のロボット開発

Robotic Process Automation 2.0

RPA とセキュリティ

RPA が各企業に浸透していく中、ガバナンスを担保する上で、セキュリティ面でのリスクや懸念点が多くの企業で浮き彫りになってきました。

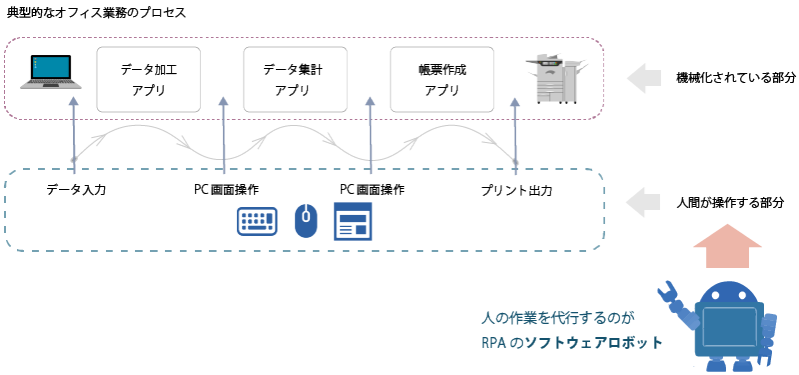
RPA は、導入するだけで何もセキュリティ対策を行なわないと、明らかにセキュリティリスクが高まり危険です。
ソフトウェアロボットは、ある部分では人と同じような仕事ができます。それゆえ、通常のソフトウェアより危険な面もあり、RPA ツールの特性を理解して、RPA を本格導入する前にリスク対応計画を準備しておかないと、ブラックボックス化してセキュリティホールになったり、ロボットが乗っ取られたり、暴走や突然の停止などのリスクが発生します。
RPA を導入することによって、財務報告に関わる部分では内部統制の不備を指摘されるかもしれないという懸念もあり、また、個人情報や、企業機密に関わるような業務も要注意です。

一方、自動車は交通事故原因の約 9 割がヒューマンエラーということで、自動運転が普及した方が事故率が減ります。同様に、RPA もしっかりしたセキュリティ対策を講じて導入すれば、人が行なっていた業務をロボットがミスなく遂行するため、運用の安全性が高まります。
つまり、人に頼った作業は、どうしても疲れやストレスから操作ミスや勘違いが発生しがちですが、ロボットにより自動化される業務は、指示通り正確に行われるため、その分品質が向上し、オペレーションミスがなくなります。各種のセキュリティ対策も、必要に応じてロボットの力を借りた方が、人の目検より圧倒的に正確で、速く、大量にチェックができます。よって、セキュリティ強化の仕組みもセキュリティソフトなどを組み合わせ、RPA 自身で実現することがたいへん有効です。

それでは、RPA のセキュリティを強化するロボット開発とは、どのような方法で行なうべきでしょうか？

RPA が行なうこと

RPA のセキュリティを考えるにあたり、あらためて RPA とはどのようなものか確認します。たとえば、典型的なオフィス業務では、データの加工や集計など機械化出来る部分は、コンピュータで処理させるよう自社のアプリや、特定のツールを使って処理します。人は、そのアプリを起動したり、データ入力したり、画面操作をして、プリンターへ印刷指示したりします。この部分における、人が操作して、機械化されていなかった作業を、ソフトウェアロボットに代行させて自働化するのが RPA です。



こうしてみると、ロボットは、人の代行として働いてもらうのだから、新人に引き継ぐ時と同じように、ロボットも人の採用と同じように管理すればいい、という考え方もガバナンス的には有効な方法です。但しセキュリティについては、加えて特別な新規のソフトウェアとして取り扱うという配慮も必要となります。
ロボットもソフトウェアですので、PC やサーバに導入されているアプリケーションやシステムと同等のセキュリティ対策は当然必要なのですが、ロボットだからこそのセキュリティ対策も別途必要となります。たとえば、人なら入退出管理をしますが、ロボットはコンピュータ上にいるため、ロボットの不穏な動きを管理するにはロボット監視用にログを取得して、行動履歴を取る仕組みがあらたに必要です。

RPA は、ロボット自身と、ロボット開発したソースプログラムの両方について、セキュリティ対策を強化しておかなければいけません。
ソフトウェアロボットは、人と同じような操作ができるので、今までオペレータが必要だったエラー時の通報や、障害対応のオペレーションなども自動化でき、IT 部門の運用の現場でも活躍できます。ロボットの監視・管理の部分についても、人手に頼らず、ロボットに代行させるという考えは、RPA を利用する上で大変有効な手段です。

RPA で見落としがちなセキュリティ

RPA を構築する場面で見落としがちなのが、それぞれの企業のセキュリティポリシーに基づいたロボット開発をするということです。

RPA が、業務プロセスの自動化であるため、ロボットも人と同じように手順通りに成果物を作成できるかについては、もちろん各企業の業務統制に従って注意を払われていますが、ロボットを使用することが、自社のセキュリティポリシーに照らし合わせて問題ないかという監査については、多くの RPA 導入企業で見落されがちです。
業務プロセス通りの手順は、RPA ツールの操作記録方式などの機能を使用することによって、比較的簡単にロボットへ設定ができ、何も問題がなければ、そのロボットは意図した通り業務を自動実行できます。しかしながら、ちょっとした画面ボタンのずれや、例外データが表示された時など、人なら普段からあたり前に対処しているオペレーションであっても、ロボットでは臨機応変に対応出来ず止まってしまう場合があります。
ロボットにもセキュリティ教育が必要なのですが、指示通りにしか動かないロボットには、想定できるセキュリティリスクをすべて洗い出し、事前に対応できるように設定しておく必要があります。

現在注目されている自動運転車は、車が自ら周囲の動きを認知し、判断して行動します。それによって、人が運転しなくてよい車が開発されています。無人でも、安全で事故を起こさない運転ができるように、より多くの経験を自動車に学習させています。
RPA の最終形も、完全無人運転なので、将来は自らが学習する仕組みが必要になります。ただ現状では、まだまだ自身で学習できるロボットは分野が限られているので、少しでも安全に業務を実行させるためには、想定されるリスクは自動的に対応できるように、可能な限り事前にロボットへプログラムしておくべきです。
自動車であっても、危険を察知してブレーキをかけるとか、衝突時にはエアバッグが作動するとか、まずは適応できるところから自動化を実現してきました。

セキュリティポリシーは、PDCA サイクルで常にリスクへの対応を知識ベースに蓄え、より安全が高まるように見直されます。
それに伴いロボットも、新たなトラブルへの対処方法を繰り返し覚えさせ、進化させることで、RPA は安全な無人運転に近づきます。

ロボット特有の注意点

RPA のセキュリティの対策は、ロボットと、それを動かすプログラム（スクリプト）について、それぞれ誰がどの様に管理するかが重要です。RPA について企業がセキュリティポリシーに沿って適切な対応を取るためには、ロボット特有の注意すべき点も多くあります。

セキュリティに関して、ロボットだから気を付けるべきポイントの例

① 例外データ等に起因した誤作動がありえる
② エラーがわからない場合がある
③ 指示しないと対処できない
④ 五感がないため、自分の異常に気付かない
⑤ ロボットに指示するためは、専門技術が必要な場合がある
⑥ ログの種類や出力形式がツールに依存する
⑦ アラーム通知の設定が必要
⑧ 割り込みやエラー時のロボットの終了方法の手順が必要
⑨ ロボットのライセンス管理が必要
⑩ バージョンアップやバグ対応が必要
⑪ 実行環境変化への対応が必要
⑫ バックアップ／リカバリー手順と実施が必要
⑬ 権限と認証の管理が必要

今後進化していくことが期待されますが、現状はソフトウェアロボットは、神経を持たないので痛がることもなく、自分に異常があってもわからないことがほとんどです。よって、誰かに乗っ取られても、いつもと違う動きをしていても基本的には自分ではわかりません。また、その異常に誰かが気づいたとしても、止め方が分からなかったり、リカバリー手順が通常の業務とは違ってくる場合があります。

厄介なのは、そのRPAツールの専門知識がないと、対処できない場合が想定されることです。結局、セキュリティを保つためには、そのための対処をロボットに事前に教えておくか、常に厳重な監視体制を敷いて、いつでも対応できる状態にしておくしかありません。
とはいえ、直ぐにすべてのリスクを洗い出して対応できるはずもなく、いつまでも人が監視と対処を行なっていなければならないようでは、自動化のメリットが薄れてしまいます。

RDA タイプのセキュリティリスク

RPA の流行は、プログラミング経験がない現場の業務担当者でも、業務の PC 操作をロボット化できるという RPA ベンダーからのアピールが大きく影響しています。RDA（Robotic Desktop Automation）と呼ばれる、クライアント型の RPA ツールは、業務部門の担当者でも、簡単な単純作業であれば比較的容易に自動化することができ、大流行しました。

RPA の構築が簡単な RDA タイプのツールは、導入も手軽で、それだけにエクセルなどのオフィスソフトと同等に取り扱われる可能性があります。しかしながら､RPA ツールはロボットの性格上、「サービス停止」や、「ロボットの暴走」､「悪意によるロボット操作」などのリスクに対しても、新たに対策をとる必要があります。
例えば、重要なデータは取り扱わず、PC の単純操作のみの業務でしか RPA を使用しないという場合は、セキュリティ対策は特にいらないはずだと思われがちです。しかしながら、ソフトウェアロボットが PC に導入されただけで、特に何もさせていないという状況であったとしても、そのロボットが誤作動を起こしたり、誰か悪意のある第三者に乗っ取られる危険性もあります。

業務部門の人達は IT の専門家ではないので、どのようなセキュリティ対策の仕組みをロボット開発時に組み込むべきなのかについては、あまり知識と経験がありません。さらに、ロボット向けにエラー対処のロジックを組むためには、条件式や変数などのプログラミング知識が必要です。しかも、PC 操作の記録型で開発が簡単な RPA ツールでは、あまり複雑なロジックの追加はいろいろと制限されています。

野良ロボットが不正な動きをしないよう監視したりするためには、業務担当者が作成したロボットをセキュリティポリシーに違反がないように監視・制御するロボットを、運用や開発に詳しい IT 部門の人が別途作成すべきです。この場合は、別のロボットになるため、必ずしも同じ RPA ツールで作成しなければならいないという制限はありません。

つまり、業務担当者は、自身の業務自動化の部分のみ RDA タイプのツール等を使って開発をして、担当者にとっては苦手かもしれないロボットのセキュリティ対策は、IT 部門の人がガバナンス等を考慮して別の仕組みとして開発してあげることが現実的です。

- 完全性チェック
- 妥当性検査
- ＩＤ識別
- アプリ認証
- アクセス承認
- フォレンジック
- 入出力データ保証

サーバタイプのセキュリティ対策

RPA ツールには、サーバを使用してロボットの集中管理を行なうものがあります。
中には実行されるロボットも、物理 PC でなく、仮想サーバ上で動くものさえあります。

サーバタイプの RPA ツールは、スケジュール管理や、実行結果のログ管理などが標準で備わっており、一般的なセキュリティリスクが可能な限り最小になるようにシステム化されているため複数のロボットでも管理がし易くなっています。会社組織として、一括で管理できるので、ガバナンスやセキュリティについては優れています。ただ、その管理手法が企業のセキュリティポリシーに準じているかどうかの確認は必要です。

このタイプの RPA ツールにも、セキュリティに関して懸念点があります。

① ツールシステム内がブラックボックスとなる

サーバタイプの RPA ツールは、複数のロボットを効率的に管理するために、基本は中央のサーバからのコントロールで、実行スケジュールはもちろん、ロボットの停止や、エラー検知、ログ解析などを独自の手法で行なっています。そのため、現場担当からは、コントロールできない不明な部分が多くなってしまい、そこにリスクがあります。

② ネットワーク環境の整備が必要

サーバからロボットをコントロールするために、接続しているネットワーク通信について、不正や異常な動きがないかどうかセキュリティソフト等によるネットワーク監視が必要です。

③ 専門技術を持った人しかロボットを使用できない

比較的高価なサーバタイプの RPA ツールは、機能が豊富なためそのツール専用の技術の取得が必要です。従来のシステム管理手法がそのまま使用できるわけではないため、ツールに熟知した運用管理者の育成が必要となります。

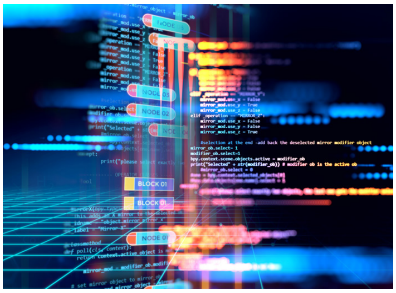
他にもセキュリティ上注意が必要なポイントはありますが、概ねサーバタイプは、RDA タイプと比べ、大規模に RPA が展開されるケースが多いため、その分セキュリティのリスクも大きくなります。

開発タイプのセキュリティ対策

RPA ツールの中には、汎用的プログラミング言語で API を通してロボットに指示ができる開発型もあります。
ROBOWARE を例にとれば、Ruby、Java、C#、PHP のいづれかの汎用プログラム言語を使用して、従来のプログラム開発と同様にプログラムをコーデイングし、そのプログラムの中から ROBOWARE 専用の API を使ってソフトウェアロボットへ動作の命令ができるため、高度な技術知識がなくても人が行なっているマウスやキーボードなどの PC 操作等が、簡単に自動化できます。

この場合、ROBOWARE 用に開発したソースプログラムは、従来の Ruby や Java などのシステム開発時に作成されるソースコードと同じ管理方法に合わせることができます。そのため、特別な管理手法を用いなくても、ソースプログラムのバージョン管理によって、野良ロボットを出さないように統制することはもちろん、ソースの改ざん防止方法など、その企業のセキュリティポリシーに従い管理できます。

開発型でも、インストールしたソフトウェアロボット本体の管理が必要ですが、こちらについてもその企業のポリシーに従って、既存の業務アプリケーションやデータベースなどに対する場合と同じ仕組みで、セキュリティソフト等から監視対象にすることで対応でができます。
ROBOWARE は、Manager の管理用端末から起動することもできますが、サーバタイプの RPA ツールのような集中管理方式ではないため、それぞれの PC で独立して稼働も可能です。よって他のアプリケーションと同様、稼働する PC 上で取得するシステムログ、操作ログ等と連動して実行管理をすべきです。
もちろん、ROBOWARE 本体のモジュールに対しては、適宜バージョンアップをして最新にしておくことが望まれます。



開発型のタイプは、他の RPA ツールに比べると、前提としてプログラミング知識が必要になりますが、開発時に、複雑な処理や、様々なシステムとの連携を組み込むことができ、従来の運用が活かせるため、IT 部門の運用系の監視業務等の RPA にも実績が多く、セキュリティの強化をしやすい RPA といえます。

フェーズ毎のセキュリティ対策

業務部門で導入された RPA ツールは、各部門の予算で購入される場合が多く、業務担当の方が直接管理責任を負う場合もあります。
よって、全社で一括管理しない場合でも導入、開発、運用と分けたフェーズ毎に注意が必要です。

① 導入フェーズ

- ロボットの使用ができる人を限定する
- 暗号化されたハードディスクなどのセキュアな場所へのインストールする
- ロボットへ与える権限を業務ごとに制限する
- ロボットを稼働できる PC 等の使用環境の範囲を限定する

② 開発フェーズ

- ID、パスワードをソースプログラムに記述しない
- 業務プロセス通りの稼働をテスト検証する
- 業務秘密のテストデータはマスキング処理する
- 中間ワークデータは終了後に消去するロジックにする
- 緊急時に割り込みができる仕様にする
- エラー時にアラームが通知されるようにする
- ポイント毎にアプリケーションバックアップを取得する
- ソースプログラムを、セキュアな場所で管理する

③ 運用フェーズ

- ロボットの実行監視、エラー監視をして行動を制御する
- PC の操作ログ、アクセスログを取得して統合管理する
- ロボットのバージョンを最新に更新する
- セキュリティポリシー合わせ運用する
- 定期的なリスク分析をして、セキュリティポリシーを見直す
- BCP に基づき、緊急時の連絡体制と対応フローを作成する
- 管理対象外のロボットの定期的な存在チェックをする

ロボットの制御

ロボットの危険についてよく話題にされるのが、ロボットによる暴走です。
意図した作業以外についても、動きを止めず、余計な仕事までして問題を起こすパターンです。

これは、もちろん通常のプログラムでもありえますが、RPA の場合は、操作したいアプリケーション画面のレイアウトが少し変わっただけでも、予期しない動きをソフトウェアロボットがする可能性があり、比較的簡単に誤作動が発生してしまう危険があります。
もちろん、これもリスク対応計画に入れるべきですが、ここで注意が必要なのは、暴走してしまった場合のそのジョブへの割り込みの仕方が、RPA ツールによっては通常の処理と違う場合があるということです。
汎用的なプログラムでロボットへ指示しているときには、通常のジョブキャンセルと同様なので、他の人でも対処し易いのですが、一般的な RPA ツールは、それぞれ独自のシステムに従っているため、暴走を発見したからといって、簡単にジョブを止められません。
特にサーバ管理型の RPA ツールなどでは、その稼働を強制終了させると、他のロボットの仕事にも影響します。
ここで問題なのが、その RPA ツールを取り扱える技能を持っていない人が運用している場合、そのロボットに介入することが難しいという点です。
RPA ベンダー任せで開発してしまったロボットの場合は、最悪そのベンダーが対応してくれるまで待たないといけないケースさえ考えられます。

これを解決するためにも、やはりそのロボットの内部仕様を知らない人でも、適切に人が介入できるように、あらかじめロボットにいつでも割り込みができる仕組みを開発しておく必要があります。たとえば、あるキーを押したら、安全な方法でロボットを停止して、その時点の作業の状況と処理中のデータをアウトプットし、分析できるように設定しておくなどです。

もちろん、影響度や優先度によって、どこまでロジックを組み入れるかは業務によって変わりますが、最低限、暴走が発見されたら、人によって安全に停止でき、データを壊さず終了できる仕組みが必要です。

ロボット乗っ取り対策

ロボット特有のリスクとしてあげられる事象の一つに、ロボットの乗っ取りがあります。
乗っ取られてしまえば、ロボットの暴走の原因にもなりえます。
但し、わかりやすい暴走よりも、最近の標的型攻撃のように、知らない間にロボットが乗っ取られて、機密情報や個人情報を漏洩してしまうケースなどの方がむしろ危険です。そうなると発見が遅れ、被害が拡大するリスクもあり注意が必要です。
ここで取り上げている乗っ取りには、悪意を持った第三者だけではなく、内部犯行によって業務以外の作業をロボットにさせるケースも含みます。ロボットは夜間でも苦も無く仕事をしてくれるため、従来の勤怠管理では通用しません。加えて通常では、アクセスできないように制御されているデータでも、業務遂行のためにロボットには読み書きを許してしまう場合があるため、他のシステムよりリスクが高くなります。

まずは、本番業務作業として動かしているロボットのジョブについて、決められた動き以外をしていないかどうかを常にモニタリングする必要があります。
ロボットが乗っ取られている場合、そのロボットが自分で出力しているログは、それさえも改ざんされている恐れがあるため、あまり信用できません。よって、そのソフトウェアロボットが稼働する PC またはサーバ上に、操作ログ、システムログ等を自動収集して、ロボットの本来の稼働スケジュール以外の動きや、業務目的以外のリソースのアクセスがあった場合に、自動的に通知できる仕組みが必要です。この時、RPA ツール側で出力するログも合わせて事実確認の突合せをする必要があるので、統合ログ管理のシステムが必要です。
さらに、ログを解析して、アラームを通知するだけでなく、その動きを止める仕組みまで構築できることが望ましいので、ログ管理もロボット化することがお勧めです。
ロボットの稼働状況をログ管理することにより、野良ロボットが不正に使用されていないかどうかもチェックできます。

ロボットをリモートでも動かせる RPA ツールが多いため、セキュリティリスクとしては、ネットワークも併せて監視することが重要なポイントとなります。

そして、ソースプログラムもロボット本体も改ざんされていないかも監視すべきなため、変更管理や改ざん検知用のツールなどを採用することも有効です。
また、セキュリティホールにならないよう、ロボット自身のバージョンを最新にしておくことも重要です。

ロボットの特権管理

ソフトウェアロボットは、様々な仕事ができてしまうため、１つのロボットでいくつもの業務を掛け持ちで担当している場合があります。
それは、RPA ツールに関わる費用を抑えるためでもあるのでしょうが、セキュリティ面を考えれば、業務担当者が持っていた権限以外の仕事をさせることで、別のリソースのアクセス権限まで持たせてしまうことはとても危険です。
たとえば、見積り処理をしていたロボットに、支払い、請求業務までやらせてしまうような場合です。通常、営業の事務担当者は、経理の支払いデータにはアクセスできなくされているはずです。にもかかわらず、見積り業務のロボットを作成した営業部門の担当者は、ロボットを通じて、経理部門のアクセス権限までも手に入れてしまうことになります。
それゆえ、ロボットは、その業務権限が及ぶ範囲毎に別々に設置すべきで、それぞれのロボットの権限は、その業務担当者と同じ権限にすべきです。そうでなければ、部門ごとのガバナンスが、遵守できない危険性が生じます。

やむを得ず購入ライセンス数の制限などで、ロボットに部門担当者より上の権限を与えなければならない場合は、新たな特権 ID を管理する場合と同じ様に、ロボット用に新規 ID を登録して、特別な管理が必要になります。
ロボットのプログラミングが出来る人は、通常社内に複数人います。特権 ID 管理と同様な管理とは、システム管理者用の共通 ID を誰がいつ、どのように使用したかなど特権を持った人の管理をする場合と同じように、ロボットについても、誰が、いつ、何の目的で使用したり、改変したりしたかを厳格に管理する必要があります。

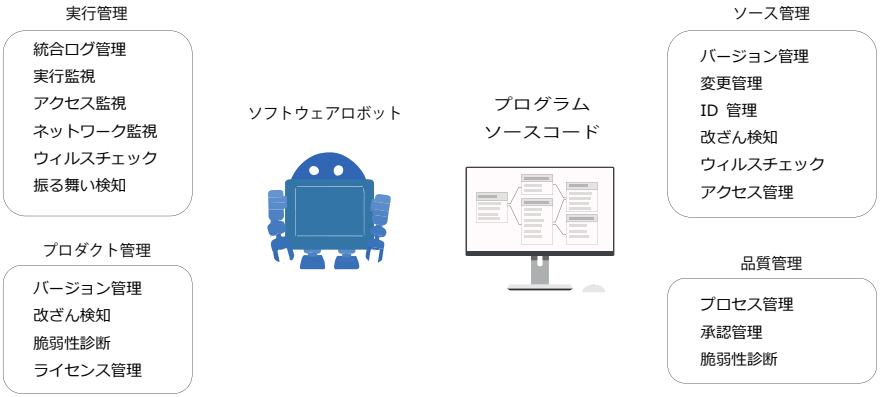
そのためには、利用の申請、承認のフローを厳密に管理し、ロボット使用者のユーザ認証や、操作ログについても業務の重要度に応じて、画面録画までして記録するなどの考慮が必要です。ここでも、市販のセキュリティソフト等を利用することは有効な手段です。
部門毎にロボットを割り当て、担当者と同じ権限で作業のみを行なうのであれば、通常のセキュリティポリシーに従い、ロボットを人と同じように管理でき、比較的運用し易いです。

ロボット利用申請・承認の記録
ロボットＩＤ、パスワードの管理
ロボットＩＤの貸し出し・権限貸与
アクセス制御
ログの記録・分析

ロボットの管理ポイント

RPA を構築する上でのセキュリティ上管理すべき対象は、大きく分けると、ソフトウェアロボット自身と作成したそれを制御するプログラムのソースコードです。RPA ツールによっては、ロボット本体に加え、それに付随するベンダー提供の管理ツールも含まれます。

下記の図が、セキュリティ管理すべき代表例です。



ソフトウェアロボットは、他の業務アプリやデータベースのシステムと同様、重要なシステムとして認識して徹底した運用管理が必要です。加えて不正利用がされないよう、ロボット毎のライセンス管理が厳格に行われるべきです。他部門で購入されたロボットが、安易に別の PC で稼働しているようなことは絶対にできないようにしなければいけません。

ロボット開発で作成されるソースコードの管理は、バージョン管理や、改ざん防止はもちろん、プログラム作成に関し業務処理で生成されるアウトプットまでを含めた品質管理が必要です。求められているアウトプットが、予定している品質を満たしているかの確認はもちろん、堅牢なプログラムが作成されているかどうかテストを重ね、監査が通って承認されていることが重要になります。
RPA ツールによっては、標準的に出てくるフローチャートに似たソースコードとは別に、複雑な処理を組み込むために別途作成するスクリプトがある場合は、その保管方法なども配慮が必要です。
また、事業継続計画（BCP) に従い、バックアップや復旧手順についても定め、災害時であってもビジネスインパクトを与えないように、対処できるようにしておくことも重要です。

セキュリティ強化したロボット開発

RPA の構築で、セキュリティ強化されたロボット開発を行なうためには、大きく３つのことが推奨されます。

① リスク回避処理をプログラムに組み込む

パッケージ化されたシステムを導入するのとは違い、新たにシステム開発する場合と同様に、ロボット用のプログラムを作成する段階から、エラーハンドリングなどセキュリティ面を考慮してデザインし、コーディングすることが必要です。そのプログラムが、セキュリティポリシーに合っているか、稼働してリスクはないのかをテストを重ね検証しながら、プログラムへの追加や修正が繰り返され品質保証された後、本番適用されるべきです。

② 統合ログ管理の仕組みを作る

ロボットの異常検知、乗っ取りや、野良ロボの監視など、ロボットを管理するためには、操作ログなど様々な種類のログを取得し、横断して分析や対処が出来る仕組みが必要です。

③ ロボットを管理するためのロボットを作る

セキュリティを強固にするために、ロボットやソースプログラムを管理することが重要になりますが、せっかく自動化したのに、管理のために人手を使っていてはメリットが薄れます。将来ロボットが増えても大丈夫なように、スケーラブルなデザインでセキュリティ管理するロボットを作成することがポイントです。

セキュリティの強化には、既存のセキュリティ対策ソフトや、市販されているパッケージ製品などをうまく組み合わせ、可能な限り自動で管理できる仕組みを構築することが、結果的に効率がよく安全です。

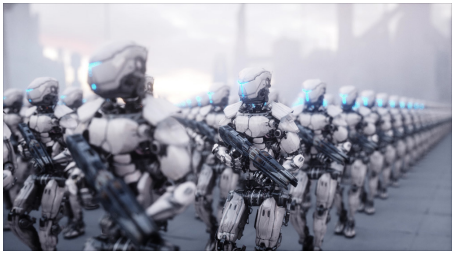
ロボット時代のセキュリティ

RPA2.0 によって完全無人化が進めば、操作ミスなどのヒューマンエラーも削減でき、セキュリティを含めた業務の運用リスクが格段に減ることが期待できます。
単純に RPA を導入するだけでは、セキュリティ脅威のリスクが増してしまいますが、RPA の特性を理解して適切な対処ができれば、人を管理するよりもロボットの方がセキュリティ面で向上させることができます。工夫次第で、ロボットがロボットを監視し、運用することが可能です。企業がどのような強固なセキュリティ対策をして運用していくかは、ベンダー任せにせず、自社で責任を持って管理・運用していくことが大切です。

今後、オフィス内で活躍するソフトウェアロボットは、間違いなく増えていきます。
最初から、完璧な運用ができなくても、まだ導入数が少ない今からセキュリティを考えた運用を実施しながら、都度セキュリティポリシーと共に見直し、知識ベースを蓄えて改善を継続して、他社より先んじて安心で効率的な運用を継続してくことが大切です。ロボットが増えても、むしろセキュリティが強固になる仕組みを作ることが急務なのです。

ロボットは日々進化していきます。将来は、中央集権型の管理に頼らずとも、ブロックチェーンなどの最新技術を用いて、ロボット一台一台の動きを改ざんされないように分散管理ができ、さらに堅牢なセキュリティが保てる柔軟性のある運用が可能になるかもしれません。
今この機会に全社一丸となってセキュリティが強化されたロボット環境を構築して、安心安全なロボット時代を迎えましょう。

（編集：春日井）



CCS 株式会社シーイーシーカスタマサービス
■お問合せ　ESECinfo@cec-ltd.co.jp
〒150-0022 東京都渋谷区恵比寿南 1-5-5 JR 恵比寿ビル 8F
TEL：03（5789）2443　FAX：03（5789）2575

（2019年2月22日公開の情報です）