(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 February 2024 (01.02.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/023527 A1

(51) International Patent Classification:
*G06F 21/56* (2013.01) *H04L 9/40* (2022.01)

(21) International Application Number:
PCT/GB2023/052001

(22) International Filing Date:
28 July 2023 (28.07.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2211125.6 29 July 2022 (29.07.2022) GB

(71) Applicant: **PREDATAR LTD** [GB/GB]; Bloxham Mill, Barford Road, Bloxham, Banbury Oxfordshire OX15 4FF (GB).

(72) Inventor: **NORGATE, Richard**; Bloxham Mill, Barford Road, Bloxham, Banbury Oxfordshire OX15 4FF (GB).

(74) Agent: **HUDSON, Daniel**; Potter Clarkson, Chapel Quarter, Chapel Bar, Nottingham Nottinghamshire NG1 6HQ (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Published:**
— *with international search report (Art. 21(3))*

(54) Title: DETECTION OF ANOMALOUS BACK-UP COPIES

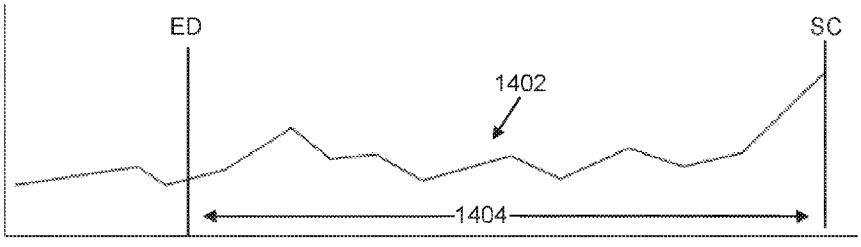


Figure 14

(57) **Abstract:** The disclosure relates to a method for detecting a suspected infection event, the method comprising: receiving data associated with back-up copies of a plurality of machines including at least a first machine and a second machine, in which the data is indicative of a size of the associated back-up copy; and determining whether to classify data associated with at least one back-up copy associated with at least a second machine as anomalous based on an anomalous pattern identified in data associated with a back-up copy associated with a first machine.

WO 2024/023527 A1

# DETECTION OF ANOMALOUS BACK-UP COPIES

## FIELD

The present disclosure relates to apparatus, systems and methods for restoring a computer system detecting a suspected infection event in back-up data, which may be used in recovering one or more of a plurality of machines of a computer system following a malware or ransomware attack.

## BACKGROUND

Public and private organisations of all shapes and sizes rely on the reliable and secure operation of their computer networks. It is not uncommon for a particular proprietor's network to contain thousands of machines, or virtual machines, spread across the globe. The interconnectivity of these machines allows collaboration but also increases the risk of infection by malware or ransomware, as the number of infection points increases. Ransomware can be especially problematic as it may encrypt the contents of a computer system's memory and demand that a ransom be paid by the proprietor in exchange for the decryption keys. Such infections can have devasting financial and reputational consequences for organisations and the scale of existing networks and their interconnectivity can create significant challenges in recovering from such events. In particular, the scale of loss can be reduced and resilience of the network can be improved by the effectiveness of the disaster recovery process. Disaster recovery is typically achieved by the restoration of back-up copies recorded at an earlier time. However, in existing solutions the process of restoring a network containing a number of machines may be time consuming and complex work, requiring significant operator skill to identify and prioritise nodes in need of recovery and implement an effective recovery strategy. One or more aspects of the present disclosure are directed to alleviating such difficulties.

## SUMMARY

According to a first aspect of the present disclosure there is provided a computer-implemented method for a computer-implemented method for detecting a suspected infection event, the method comprising:

receiving data associated with back-up copies of a plurality of machines including at least a first machine and a second machine, the data may be indicative of a size of the associated back-up copy; and

determining whether to classify data associated with back-up copies of at least a second machine as anomalous based on a pattern identified in data associated with back-up copies of the first machine.

The method may further comprise receiving data associated with each of a plurality of back-up copies associated with the first machine. The method may further comprise searching the plurality of back-up copies associated with the first machine to identify the earliest back-up copy that comprises a signature of the infection event. The pattern in data associated with back-up copies of the first machine may comprise a behaviour shape between the earliest back-up copy that comprises the signature of the infection event and the most recent back-up in which the infection event was detected. The behaviour shape may be a back-up data transfer profile as a function of time. The behaviour shape may be based on the back-up data transfer profile as a function of time. The behaviour shape may comprise a subset of the back-up data transfer profile as a function of time.

In response to classifying a back-up copy of the second machine as anomalous, the method may determine a score indicative of a likelihood of infection based on when the most recent antivirus scan was performed on the second machine.

The method may further comprise generating a graphical user interface (GUI). The GUI may provide an indication whether back-up copies of one or more of the plurality of machines have been classified as anomalous. The score may be indicative of a likelihood of infection associated with one or more of the one or more of the plurality of machines have been classified as anomalous.

A machine for comparison with the pattern may be selected based on user input.

The method may further comprise prioritizing the recovery of machines associated with a score indicative that infection is more likely over the recovery of machines with a score indicative that infection is less likely.

The score may be scaled based on the time the most recent antivirus scan was performed on the second machine between, for example, a time associated with the earliest back-up copy of the first machine that comprises the signature of the infection event; and a time associated with the most recent back-up of the first machine.

The method may further comprise receiving data associated with each of a plurality of back-up copies associated with the first machine, in which the data is indicative of a size of the associated back-up copy. The method may further comprise training a pattern matching algorithm to determine whether to classify data associated with back-up copies as an anomalous pattern using the data associated with each of a plurality of back-up copies of the first machine.

The method may further comprise using the trained pattern matching algorithm to determine whether to classify the data associated with back-up copies of the second machine as anomalous.

The method may further comprise scanning a back-up copy that is classified as anomalous using antivirus software.

The method may further comprise scanning metadata of a back-up copy that is classified as anomalous using antivirus software.

Determining whether to classify data associated with back-up copies of the second machine as anomalous may be based one or more patterns identified in data associated with back-up copies associated with a first plurality of machines.

A plurality of back-up copies may be associated with each of the plurality of machines. For each of the plurality of machines, each of the plurality of back-up copies may be searched to identify one or more clean-back-up copies that

do not comprise a signature of the infection event. Each of the plurality of machines may be a physical machine or a virtual machine.

The plurality of back-up copies may comprise one or more back-up copies stored on secondary storage and/or one or more snapshots stored on primary or secondary storage. The one or more back-up copies may be stored remotely from the primary data and systems to which they relate. The one or more snapshots may be stored adjacent or remotely to the primary data and system to which they relate. The one or more snapshots may relate to an exact image of a machine. The one or more snapshots may be obtained with a greater frequency than the back-up copies stored on secondary storage, or with the intention of a shorter term retention than the back-up copies such as those stored on secondary storage. For example, a snapshot may be taken from a machine on a periodic basis, such as every hour, whereas a back-up copy may be taken at less regular intervals, such as nightly, once a week or once month, for example. The snapshot copies may be discarded after a period of time has lapsed, such as a day or a week. Back-up copies such as those stored on secondary storage may be intended to be retained for a longer period than the snapshots, such as greater than a week, greater than a month or greater than a year, for example.

The method may comprise determining an infection-datum-time for the computer system by identifying a creation time of a clean-back-up copy created before an earliest back-up copy that comprises a signature of the infection event.

The first machine may be geographically situated within a first-geographical-area. The second machine may be geographically situated within a second-geographical-area. The first-geographical-area may be non-overlapping with respect to the second-geographical-area.

The first machine may use a first operating system. The second machine may use a second operating system. The first operating system may be distinct from the second operating system.

Restoring one or more of the plurality of machines may further comprise restoring the one or more machines in a computing environment that is isolated from potential sources of infection.

Restoring one or more of the plurality of machines may comprise installing anti-virus software dated subsequent to the infection-datum-time on at least one of the plurality of machines. The anti-virus may be installed on a virtual machine based on a back-up copy to be scanned. The anti-virus may be installed in a virtual quarantine area and scan the virtual machine within the quarantine area. A definition of the anti-virus software may be obtained automatically from an update server.

Anti-virus scanning may be applied to a plurality of the back-up copies. In some examples, anti-virus scanning may be applied to a restored Virtual Machine that has been restored from one or more back-up copies. Knowledge of one or more infection signatures or one or more virus code objects identified in one scan may be used in subsequent scans.

The signature of the infection event may be a user defined signature. Alternatively, a definition of the signature of the infection may be obtained automatically from an update server. The same update server may provide both the definition of the anti-virus software and the definition of the signature of the infection. In some examples, the update server may be a Global File Infection Search server.

The method may comprise providing a graphical user interface comprising elements associated with actions in the method for restoring the system. The method may comprise providing a graphical user interface comprising elements associated with the plurality of machines. The method may comprise receiving a user selection associating one or more of the plurality of machines with one of the actions.

The election may be achieved by dragging one or more elements associated with the plurality of machines to one of the actions. The elements associated

with the plurality of machines may include an indication of a status for each of the plurality of back-up copies associated with the respective machines.

The method may comprise providing a graphical user interface comprising elements associated with actions in the method for restoring the system. The method may comprise providing a graphical user interface comprising elements associated with the first-recovery-group. The method may comprise providing a graphical user interface comprising elements associated with the second-recovery-group. The method may comprise receiving a user selection associating the first- or second-recovery-group with one of the actions.

The method may comprise determining which back-up to use for a different machine based on the identified back-up copy for a first machine. The back-up copy for different back-up copy may be as old or earlier than the identified back-up.

According to a further aspect, there is provided a computer-implemented method for detecting a suspected infection event. The method comprises:

receiving data associated with each of a plurality of back-up copies associated with a machine, and in which the data is indicative of a size of the respective back-up copy;

training a pattern matching algorithm using the data associated with each of a plurality of back-up copies to identify a periodic variation in back-up size; and

using the trained pattern matching algorithm to determine whether to classify data associated with a further back-up copy associated with the machine as anomalous.

In this way, a pattern matching algorithm may be based on the output of a machine learning process using historical data for the network. Such an approach is advantageous in that it takes into account the typical periodicity in use of the network. For example, the data rate on days in which back-ups are scheduled may be substantially higher than that on other days. Similar considerations apply to different times of day. Machine learning methods may

be applied to large quantities of network traffic data taken over an extended period of time.

Each of the plurality of back-up copies may be a different version back-up. The further back-up copy may be obtained subsequently to the plurality of back-up copies.

A back-up copy that is classified as anomalous may be scanned using antivirus software. Alternatively, metadata associated with the back-up copy that is classified as anomalous may be scanned using antivirus software. That is, it may not be necessary to scan the whole file. The metadata may include a table of file names contained in the back-up copy.

According to a further aspect, there is provided a computer-implemented method for restoring a computer system following an infection event, the computer system having a plurality of machines, in which a plurality of back-up copies are associated with each of the plurality of machines, and in which each of the plurality of back-up copies associated with a particular machine is a different version back-up, the method comprising restoring one or more of the plurality of machines using a respective clean-back-up copy.

In one example, the method may comprise selecting a back-up copy for a particular machine, moving the selected back-up copy to a cleaning environment, cleaning the selected back-up copy in the cleaning environment and applying the cleaned, selected back-up copy to a respective machine in a live environment. Selecting the back-up copy may comprise receiving an indication of a back-up copy to be cleaned from a user. Cleaning may be achieved using antivirus software.

In another example, the method may comprise identifying the most recent back-up copy for a particular machine, moving the most recent back-up copy to a cleaning environment, cleaning the most recent back-up copy in the cleaning environment and applying the cleaned most recent back-up copy to a respective machine in a live environment. Cleaning may be achieved using antivirus software.

In a further example, the method may comprise searching the plurality of back-up copies to identify one or more clean-back-up copies that do not comprise a signature of the infection event. The method may comprise determining an infection-datum-time for the computer system by identifying a creation time of a clean-back-up copy created before an earliest back-up copy that comprises a signature of the infection event. The method may comprise identifying a back-up copy created after the infection-datum-time, moving the back-up copy to a cleaning environment, cleaning the back-up copy in the cleaning environment and applying the back-up copy to a respective machine in a live environment.

The method, system, computer-program may be configured to operate on a operate on computer that has a system bus with an even number of bits. The computer-implemented method may not make use of IBM antivirus or infection-event recovery software. The computer-implemented method may be performed using one or more compiled, executable files.

According to a further aspect, there is provided a graphical user interface for restoring a computer system following an infection event. The graphical user interface may comprise any of the elements configured to perform the functionality described herein.

The aspects described above may be provided in combination. Features described with reference to one aspect may be implemented in combination with any other aspect.

According to a further aspect, there is provided an apparatus comprising:

at least one processor; and

at least one memory including computer program code for one or more programs,

the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform any method disclosed herein or provide any of the graphical user interfaces described herein.

According to a further aspect, there is provided a computer program product including one or more sequences of one or more instructions which, when executed by one or more processors, cause an apparatus to at least perform any method disclosed herein or provide any of the graphical user interfaces described herein.

While the disclosure is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that other embodiments, beyond the particular embodiments described, are possible as well. All modifications, equivalents, and alternative embodiments falling within the spirit and scope of the appended claims are covered as well.

The above discussion is not intended to represent every example embodiment or every implementation within the scope of the current or future Claim sets. The Figures and Detailed Description that follow also exemplify various example embodiments. Various example embodiments may be more completely understood in consideration of the following Detailed Description in connection with the accompanying Drawings.

## BRIEF DESCRIPTION OF DRAWINGS

One or more embodiments will now be described by way of example only with reference to the accompanying drawings in which:

Figure 1 shows an architecture and data flow for a standard back-up system;

Figure 2 shows a system for backing up and subsequently restoring a computer system 202 using a back-up server;

Figure 3 shows an architecture and logical flow of a computer-implemented method for operating a Ransomware Recovery system;

Figure 4 illustrates a graphical user interface (GUI) layout comprising a number of portions;

Figure 5 illustrates a navigation portion and a activity portion of a GUI;

Figures 6a to 6d illustrate various style of boxes that may be used in combination with the GUI layout of figure 4 at various stages in the process;

Figures 7a to 7e illustrate various aspects of the actions portion;

Figure 8 illustrate a GUI profile providing a data rate transfer report;

Figure 9 illustrates an example of a protection map;

Figure 10 illustrates another example of a protection map;

Figure 11 illustrates an example computer program product;

Figure 12 illustrates a functional process flow for a procedure for detecting anomalies in back-up data;

Figure 13 illustrates another method for detecting anomalies in back-up data;

Figure 14 illustrates a pattern observed in back-up data;

Figure 15a and 15b illustrates the same pattern observed in different back-up data with shorter and longer timeframes; and

Figures 16a to 16c illustrate various threat scores based when a scan was most recently performed within a period in which a pattern indicating a suspected infection event is observed.

## DETAILED DESCRIPTION

The present disclosure provides an environment for improving the efficiency of infection recovery management, which may be achieved by iterative search of recoveries in a secure quarantined area, isolated from potential sources of infection, to ensure the latest uninfected recovery point can be recovered. This can be undertaken with the involvement of third-party software. For example, existing back-up systems may not natively provide the functionality to scan for malware within the back-up files at the point of recovery.

Previously unrecognized malware captured in back-up operations can be eliminated during restore operations, so that recovery from a malware attack does not reseed the recovery environment with the malware that caused the original attack.

It may be advantageous to prioritize restore processes. Initial efforts can focus on the most critical applications, so in some examples it is important to have defined a preferred restore order and the procedures in place. In the first instance this can be fulfilled by inspecting the candidates in pre-configured Recovery Groups of machines that form the computer system and removing immune assets from those groups where they are not integral to the function

of the group. Immunity can often be inferred by Operating System type and version, since some systems/versions may not be vulnerable to certain attacks. An option can apply conditions on the group that automatically remove immune assets in bulk from recovery groups.

It may not be possible to ensure a clean environment. Often the presence of ransomware can go undetected resulting in back-ups that contain the ransomware, which can be triggered again after recovering from the initial attack. Therefore, it is important to ensure that back-up software can scan for and remove ransomware during recovery from a ransomware attack. This requirement is satisfied by the Quarantine and Clean module of the present disclosure or by a Global File Infection Search (GFIS) where it is supported by the back-up application. This feature allows suspected infection signatures to be entered centrally in a GUI. GFIS then automatically runs a search for these filenames / extensions against multiple supported back-up applications to identify the youngest back-up held that does not contain the infection signature. This uninfected back-up may then marked as ready for recovery and antivirus (AV) scan in the PQC (Predatar Quarantine and Clean) production.

The recovery methodology disclosed herein is applicable to restoring machines based on long term back-up copies or snapshot images of machines, both of which may be considered to provide back-up copies for the machine. That is, the recovery methods may be applied to back-up copies obtained for remote back-up storage processes and also using primary storage snapshots generated by a machine on a periodic basis during use to allow rapid recovery with limited data loss between back-up cycles.

Figure 1 shows the architecture and data flow for a standard back-up system 100, which will be contrasted with the Ransomware Recovery Logical flow described below in relation to Figure 3, which describes aspects of the applicant's soon-to-be launched 'Predatar' Recovery product.

The standard back-up system 100 is divided between components that are part of a client organisation 102 that receive the back-up service and a datacentre 104 that provides the back-up service.

The standard back-up system 100 may be implemented using a known data protection and recovery system that is widely used, such as the IBM® Spectrum Protect™ (Tivoli® Storage Manager, TSM). Spectrum Protect permits an organisation to recover their data either onsite or at a disaster response site. Services provided by Spectrum Protect include tracking and managing the retention of data from organisations, providing centralised data protection, to assist with the retrieval of previously backed up and archived data and to allow for local site recovery and DR operations at second site. An overview of the service a TSM provides, how TSM works and the structure of a TSM system can be found at http://www.redbooks.ibm.com/redbooks/pdfs/sg248134.pdf, as viewed on 28 January 2021.

**Collecting data from Spectrum Protect server 108**

A client 110 runs SELECT or QUERY commands on a Spectrum Protect Administrative command line (dsmadmc) 112. The client 110 reads ANR/ANE (Application Not Responding/Ascending Numerical Order) messages that appear on the Spectrum Protect Administrative command line console (dsmadmc in console mode) 114. The client 110 parses the text output on the Administrative command line 114 and sends it to a server 116. The data is sent over a secure connection 118 to the server 116, which may use 1024-bit AES/RSA encryption methods. The server 116 receives the data and inserts it into a database 120. The server 116 notifies a portal 122 of the data insertion. The portal 122 filters the data to check if the data is breaching any defined thresholds. The Portal 122 then raises an alert or ticket in the database 120. The Portal 122 gets data from the database 120 to display it on the Portal 122.

**Sending commands from the Portal 122 to the Client 110 for Spectrum Protect server 108**

A User 124 may issue a command from the Portal 122 such as running a data collection or restarting the Client 110. The command is picked up by the Server 116. The command is sent to the Client 110 (the Client 110 can keep polling the Server 116 for any commands). The Client 110 reads the command and performs the relevant action such as running the command on the Spectrum

Protect Administrative command line (dsmadmc) 112 or restarting itself, for example.

**Running disaster recovery test for a VM from the Portal 122**

The User 124 can issue a restore command from the Portal 122. A Recovery command listener 126 picks up the command from the Portal 122. The Recovery command listener 126 can keep polling the Portal 122 for any command. The Recovery command listener 126 runs the restore command on a Back-up-Archive command line (dsmc) 128. A Virtual Machine (VM) is restored in a specified hypervisor 130. The Recovery command listener 126 reads the result of the restore from the Back-up-Archive command line 128. The Recovery command listener 126 sends the result of the restore to the Portal 122. The Recovery command listener 126 may delete the VM from the Hypervisor 130 at a user-defined point in time once the restore has finished.

**Collecting data from Spectrum Protect Plus server 134**

A second client 132 makes API calls to the Spectrum Protect Plus server 134 to collect information, such as a list of protected and non-protect VMs or vSnap utilization. The second client 132 and the client 110 may be provided as one. The second client 132 reads the results return by the API call. The second client 132 sends the results to the second server 136. The data is sent over a secure HTTPs connection 138 through Web API calls. The second server 136 receives the data and inserts it in the database 120. The Portal 122 gets data from the database 120 to display it on the Portal 122 as reports, for example.

**Sending commands from the Portal 122 to the Second client 132**

The User 124 issues a command from the Portal 122 such as running a data collection. The command is picked up by the second server 136. The second server 136 and the server 116 may be provided as one. The command is sent to the second client 132 (the second client 132 can keep polling the Server 136 for any commands). second client 132 reads the command and performs the action such as making an API call to the Spectrum Protect Plus server 134.

Whilst suites such as Spectrum Protect and their implementation in such systems may be extremely powerful, their use in an organisation of any

significant size quickly becomes very complex and requires active management. Experts are therefore required to configure and manage the data protection system and develop and test bespoke data protection policies and recovery procedures. Known data protection solutions and disaster recovery contracts with third party organisations can also be expensive for an organisation.

For ease of reference, the discussion of each of the figures will be followed by a reference table providing a list of the numerals used in that figure.

| FIG 1. ARCHITECTURE AND DATA FLOW FOR A STANDARD BACK-UP SYSTEM | |
|---|---|
| **BEHAVIOUR NODE** | **DESCRIPTION** |
| 100 | ARCHITECTURE AND DATA FLOW FOR A STANDARD BACK-UP SYSTEM |
| 108 | SP SERVER |
| 110 | CLIENT1 |
| 112 | DSMADMC COMMAND LISTENER |
| 114 | DSMADMC IN CONSOLE MODE |
| 116 | SERVER1 |
| 118 | SECURE CONNECTION 1024-BIT AES/RSA ENCRYPTION METHODS |
| 120 | DATABASE |
| 122 | PREDATAR PORTAL |
| 124 | USER |
| 126 | RECOVERY COMMAND LISTENER |
| 128 | BACK-UP-ARCHIVE COMMAND LINE |
| 130 | HYPERVISOR |
| 132 | CLIENT2 |
| 134 | SPP SERVER |
| 136 | SERVER2 |
| 138 | HTTPS CONNECTION |

Figure 2 shows a system 200 for backing up and subsequently restoring a computer system 202 using a back-up server 204. The computer system 202 is connected to the back-up server 204 by an appropriate two-way network connection 206. The computer system 202 has a plurality of distinct machines, including a first machine 210a, a second machine 210b, a third machine 210c and an Nth machine 210n. Each one of the plurality of machines 210a-n may be physically located in the same or different locations, and may run the same or different operating systems. A machine as described herein may in general be a physical machine or a virtual machine.

Back-up copies of any one or more of the plurality of machines 210a-n can be transmitted to the back-up server 204. The back-up server 204 can store a first back-up copy 220a, a second back-up copy 220b and further back-up copies including an Nth back-up copy 220n for each machine 210a-n. Each of these back-up copies 220a-n may be a different version with respect to one another, such as by having been created at different points in time. This can enable the computer system 202 to be restored to its condition as it was at different times in the past.

One purpose for which the back-up copies may be used is to restore the computer system 202 after it has been attacked by malware or ransomware. Such attacks can be classified as infection events. A problem with restorations following an infection event can arise because ransomware, in particular, may not initiate its attack against the computer system 202 as soon as it has infected one of the machines 210a-n. The ransomware may lie dormant for a period of time before launching an attack. Therefore, restoring one or more of the machines 210a-n using one or more of the back-up copies 220a-n may restore the computer system 202 to condition in which the ransomware is still present.

To prevent this problem, of restoring to an already-infected state, it is possible to search the back-up copies 220a-n for a signature of the infection event, such as a digital signature of the ransomware responsible for launching the attack. If each of the first to Nth back-up copies 220a-n contain the signature then none of these copies will be suitable for providing restoration of the computer system 202, at least not without additional processing to deal with the malware that infects the back-up copies 220a-n. However, further searching of the back-up server 204 may identify a first clean-back-up copy 222a and further clean-back-up copies, such as an Nth clean-back-up copy 222n, that do not contain the signature of the malware. Any one of the clean-back-up copies 222a-n can then be used to restore one or more of the machines 210a-n of the computer system 202 to an uninfected state that does not include the malware detected by the presence of the signature in the later back-up copies 220a-n.

In the event that all of the machines 210a-n of the computer system 202 are infected with malware it is possible to search the back-up server 204 to identify one or more clean back-up copies for each one of the machines 210a-n and then to restore each of the machines 210a-n with an appropriate clean-back-up copy. In this instance, each of the clean-back-up copies should be free from the presence of any signature of the infection event.

To identify suitable clean back up copies it can be advantageous to determine an infection datum time (which may also be called a ransom infection datum time) for the computer system 200. The infection datum time can be the same time as a creation time of a clean back up copy that was created directly before the earliest back-up copy that does contain a signature of the infection event. By identifying the earliest back-up copy that does contain the signature of the infection event it is then readily possible to identify suitable clean back-up copies for any machine within the computer system 202 that should also be free of infection, since the clean back-up copies were created at or before the infection datum time.

In some examples, different clean back-up copies may be identified for each of the plurality of machines 210a-n by identifying back-up copies that were created at or before the infection datum time that are relevant to each in turn of the machines 210a-n and can be used to restore each of the machines 210a-n. This may be advantageous, even where for a particular machine an apparently clean back-up copy may be identified that was created after the infection datum time. Having been created after the infection datum time it is possible that the back-up concerned is not truly clean since it may contain malware related to the infection event, but which does not contain the particular signature that has been identified to determine the infection datum time. Not all malware associated with a specific infection event will contain the same signature.

In some examples it may be possible, for a particular machine, to identify both a pre-event clean-back-up copy that was created before the infection-datum-time and a post-event back-up copy that was created after the infection-datum-time. The post-event back-up copy may contain valuable information

that is not present in the pre-event clean back-up copy, but may also contain undetected ransomware, or other malware. However, it can be possible to restore the particular machine using both the pre-event clean-back-up copy and the post-event back-up copy by combing the two different back-ups. This combination may be achieved by identifying areas of the post-event back-up copy that have a low risk of containing malware and using those areas for the restoration, while using data from the pre-event back-up copy, that corresponds to high-risk parts of the post-event back-up copy, to complete the restoration. For example, to make a merged protected machine, there may be provided a method comprising: i) recovering the pre-event clean back-up copy (restore a); ii) recovering the post-event back-up copy (restore b); iii) identifying the file systems or files of restore b that show infection patterns and that should not be restored; iv) using restore a, comparing files from that system with equivalent files from restore b; and v) if those files have a more recent date stamp and are flagged as clean, moving those files across from the image in restore b to the image in restore a.

It is possible to treat each machine 210a-n separately, to determine an infection-datum-time for each machine separately and select an appropriate clean back-up copy for each machine based on its particular infection-datum-time. It is also possible to determine an infection-datum-time for the entire computer system 202. However, in other examples, it may be advantageous to partition the machines 210a-n into distinct groups of machines, such as a first-recovery-group consisting of the first machine 210a and the second machine 210b, and a second-recovery-group consisting of the third machine 210c to the Nth machine 210n.

The first-recovery-group and the second-recovery-group can be analysed and restored separately from one another. The machines 210a-b of the first-recovery-group can be considered, and an earliest back-up of either machine 210a-b that does not contain a signature of an infection event may be identified and used to specify a first infection datum time. Clean back-up copies can then be identified for the first-recovery-group of machines 210a-b that were created before the first infection datum time and used to restore the machines 210a-b of the first recovery group. Similarly, a second infection datum time can be

defined by finding the earlies instance of an infection signature in back-up copies relevant to the machines 210c-n of the second-recovery-group. All machines 210c-n of the second recovery group can then by restored using clean back-up-copies created before the second infection datum time, which can be earlier than, or later than, the first infection datum time.

An advantage of partitioning the computer system 202 into a plurality of different recovery groups of machines can arise because different groups of machines may have different levels of exposure to malware threats and/or different levels of defences or immunity to malware threats. For example, different recovery groups may be formed based on the machines within each recovery group using different operating systems, since some threats may be specific to a particular operating system while being ineffective in infecting other different operating systems. Similarly, some recovery groups may be formed based on all machines within a particular recovery group being located within a common geographical area or within a predetermined distance threshold of each other. Where all machines in a recovery group are present in the same location, they may all share a vulnerability to a malware attack such as where that attack is launched by a person who has physical access to the location and thereby to the machines. Machines physically situated at other remote locations, that do not overlap with the vulnerable location, may not be subject to the same mode of attack and may therefore be immune to that attack.

If for any reason a particular machine that would otherwise be part of a given recovery group is determined to be immune from the attack (such as a machine that is present in a vulnerable location, but which uses a different operating system or other software compared to other machines in the same location) then the immune machine can be removed from the recovery group concerned. Conversely, immune machines may be grouped together into their own recovery group, such that a different infection datum time can be determined for the immune machines and recovery can be effected accordingly.

Irrespective of how the infection datum time is determined for any particular machine or recovery group of machines, any machine that is to be restored

can be restored in a computing environment that is isolated from the rest of the computer system 202. This isolation may ensure that if an infected back-up is used, the infection cannot spread to other machines within the computer system 202. Conversely, the isolation of the machine to be restored can also ensure that the machine will not be subject to any further attacks or attempted attacks during the restoration process.

Optionally, during the restoration of any machine, anti-virus software can be installed on the machine. Preferably, the anti-virus software may have been produced after the infection datum time such that it may be configured to neutralize the malware responsible for the infection event.

The signature of the infection event may be determined by any suitable means. It may be obtained from a database of signatures of malware or in some instances it may be determined or provided by a user of the computer system 202.

Definition for the anti-virus software or the signature(s) of the infection event may be manually or automatically updated at the computer system 202. In particular, updates may be obtained by the computer system 202 automatically interacting with an update server that is dedicated to maintaining such definitions. The computer system 202 may be configured to obtain the updates from the updated server on a periodic basis using a predefined protocol. The provision of such updates on a dedicated update server allows the efficient updating of a number of unrelated systems with relevant definitions for use in the disaster recovery process.

In summary, a process for facilitating disaster recovery may include –

1) detecting that an infection event has occurred (alternatively, the process could be entirely user-initiated from observations);

2) obtaining one or more signatures of infection from the user for the infection event;

3) for selected machines, scanning the back-up files for the one or more signatures of infection to determine a recovery point (or that no back-up is available);

4) restoring the selected one or more machines from the back-up files at a recovery point in a quarantined area;

5) optionally checking restored machines;

6) patching the restored machines with up-to-date antivirus in the quarantine area; and

7) moving clean machines from the quarantine area to a normal environment.

A specific implementation of such a method is described below with reference to Figure 3. Implementation of such a method may be provided by a graphical user interface, which is described further below with reference to Figure 4 onwards.

Such software products may be used to implement disaster recovery. "War Room" users are users drawn from a Service Provider and Organisation staff that orchestrate the recovery process. War Room Users may not all be regular Predatar users, but they may have skills that will be key to a concerted recovery, such as, for example:

Network / Security Specialist - prepared to provide a list of local admin system accounts and password preferably in advance or immediately on request in the event of a War Room scenario. These accounts and passwords may be necessary to populate the PHP script in the PQC environment and in order to run the anti-virus against suspect VM's before the Production.

Provisioning Team - Individuals familiar with the organisations processes for commissioning both virtual and physical machines across the environment,

Senior Management - Nominated executives from both the Service Provider and the Organisation.

FIG 2. SYSTEM FOR BACKING UP AND SUBSEQUENTLY RESTORING A COMPUTER SYSTEM USING A BACK-UP SERVER

| BEHAVIOUR NODE | DESCRIPTION |
|---|---|

| | |
|---|---|
| 200 | SYSTEM 200 FOR BACKING UP AND SUBSEQUENTLY RESTORING A COMPUTER SYSTEM 202 USING A BACK-UP SERVER 204. |
| 202 | COMPUTER SYSTEM |
| 204 | BACK-UP SERVER |
| 206 | TWO-WAY NETWORK CONNECTION |
| 210A | MACHINE 1 |
| 210B | MACHINE 2 |
| 210C | MACHINE 3 |
| 210N | NTH MACHINE |
| 220A | BACK-UP COPY |
| 220B | 2ND BACK-UP COPY |
| 220N | NTH BACK-UP COPY |
| 222A | FIRST CLEAN BACK-UP COPY |
| 222N | NTH CLEAN BACK-UP COPY |

Figure 3 shows an architecture and logical flow 300 of a computer-implemented method for operating a Ransomware Recovery system from the point of execution of a Global File Infection Search up to a Predatar Quarantine and Clean (PQC) production that constitutes a successful restoration of a computer system. The method may be performed on a system comprising a number of machines, each having a plurality of back-up files, such as that described previously with reference to Figure 2.

At a first step 302, a single machine, multiple machines or a recovery group of machines of a computer system are selected, which may be achieved by dragged and dropping icons by a user onto an action labelled "Live Enter Infection Signature, Search for uninfected back-ups" in a graphical user interface. At a second step 304, a command is sent to the Client. The command contains a list of file extensions specified on the action of step one 302 that provide signatures relevant to infection events. The command also includes the list of machines that have been dragged and dropped onto the action.

A third step 306 constitutes a decision step that determines whether the machine or machines are back-up up by IBM® Spectrum Protect (SP) or IBM® Spectrum Protect Plus (SPP). If the machine(s) are back-up up by SP then the method moves to a forth step 308 in which a command is run on the SP back-up and archive command line. This command will tell if it has found any files with an infection signature in the back-up application. Alternatively, if the

machine(s) are backed-up up by SPP then the method moves to a fifth step 310 in which an API (application programming interface) is called which will return a list of files with an infection signature that were backed up under a relevant Virtual Machine (VM).

In either instance, SP or SPP, the method proceeds to a sixth step 312, which is a decision step. The method determines whether any files have been identified that have an infection signature within a back-up. If such files have been found then the method moves to a seventh step 314 in which the back-up is marked as dirty (i.e. as infected with malware), which may be marked by providing a flashing red light on a grid of back-up entries. The method then proceeds to an eighth step 316, which is a decision step. The method checks to determine whether there exists any previous (i.e. earlier) back-up. If such a back-up does exist (which may, by virtue of having been created earlier, be infection free) then the method returns to the third step 306 to iterate through the step from the third step 306 onwards. However, if no such back-up is found at the eighth step 316 then the method proceeds to a tenth step 320 in which the machine is marked as critical (possibly with a flashing red indication, for example) on the grid of back-ups indicating that there is no immediately recoverable back-up and an message may be issued to alert a user. The method then proceeds to an eleventh step 322 as discussed further below.

If the method, at the sixth step 312, determines that no infected files have been found then the method proceeds to a ninth step 318 in which the back-up is marked as clean with a flashing green indicator on the protection map grid shown on the recovery engine page. Such back-ups may be moved directly to PQC, as discussed below.

At the eleventh step 322, a user can drag and drop the machines shown as dirty onto the 'Recovery GFIS Clean to PQC' action. The method then proceeds to the twelfth step 324 in which a command is sent to the Client containing a list of machines dragged and dropped on to the 'Recover GFIS Clean to PCQ' action. The method then proceeds to a thirteenth step 326, which is a decision step. For a machine backed up by SP, the method proceeds to a fourteenth step 328, in which the SP VMs Client will send the command to SP for the VM

to recover the relevant machines to a quarantine network that is isolated from the rest of the computer system. Alternatively, where a machine is backed up by SPP, the method proceeds to a fifteenth step 330 in which the VMs Client calls an SPP API to recover the machines to a quarantine network. In either case, SP or SPP, the method then proceeds to a sixteenth step 332.

At the sixteenth step 332, the Client sends a command back to the portal indicating if the recovery has been successful or has failed. The method then proceeds to a seventeenth step 334, which is a decision step. If the recovery has not been successful then the method proceeds to an eighteenth step 336 in which an error message on the portal is displayed to show that the recovery was unsuccessful. Alternatively, if the recovery has been successful, then the method proceeds to a nineteenth step 338 in which the Client receives the command and cuts off the connection to the main network and connects instead to the quarantine network.

The method then proceeds to a twentieth step 340, which is a decision step, in which the method determines whether to run a PHP (Predatar Hunt Pack) script. The PHP is a custom script that can be dropped onto the recovered machine in quarantine that will search specifically for infection signatures. In a simple case, just a change in filesystem extension, for example to ".RYK", may be used to identify malware, rather then performing a full virus scan, which would be more time consuming. The infection signature may be a filename, partial file name or a file extension. The PHP script may be a standalone script or may be embedded in the Client itself.

If it is determined to run the PHP script, then the method proceeds to a twenty first step 342 in which the Client runs the PHP script on the recovered VM to determine if there are files present with certain file extensions on the recovered VMs that are infection signatures. The method proceeds to a twenty second step, which is a decision step, in which the method determines whether any infected files have been found by the PHP script. If infected files have been found then the method proceeds to a twenty third step 346 in which the infected machine is shut down. The method then returns to the seventh step 314 to iterate through the subsequent steps. Alternatively, where the twenty

second step does not find any infected files then the method proceeds to a twenty fourth step 348. If the twentieth step 340 determines that it is not appropriate to run the PHP script then the method also proceeds to the twenty fourth step 348.

At the twenty fourth step 348, the Client installs anti-virus software and runs a scan on the recovered VM. The method then proceeds to a twenty fifth step 350, which is a decision step. If the twenty fifth step 350 determines that a virus has been found then the method proceeds to a twenty sixth step 352 in which the infected machine is shut down. The method then returns to the seventh step 314 to iterate through the subsequent steps. Alternatively, if no virus is found at the twenty fifth step 350, then the method proceeds to a twenty seventh step 354 in which the machine is shown as clean (possibly with a flashing green indicator) on the protection grid of back-up entries in the GUI.

The method proceeds to a twenty eighth step 356 at which the user can drag and drop the machines that are shown as green on to the 'PQC Move Clean to production' action. Then, at a twenty ninth step 358 a command is sent to the Client containing the list of the machines that have been recovered. The method moves on to a thirtieth step 360 in which the Client receives the command and will the recover the VM to the production environment, i.e. the recovered machine will be returned to active duty within the computer system. Finally, at a thirty first step 362 the result of the restoration is sent to the portal.

Various steps of the method may be actioned manually by the user using a graphical user interface. Alternatively, various steps of the method may be further automated or semi-automated. For example, the twentieth step 340 to the twenty fifth step 350 may be automated by making use of a virus scanner.

A virus scanner can be used prior to or instead of installation of antivirus software in the twenty fourth step 348. A virus scanner acts on the virtual machine (VM) in the quarantined area, which may also be referred to as a clean room, to perform a virus scan on the virtual machine. The virus scanner may

be installed in the clean room and is able to act on the virtual machine from outside of the virtual machine without requiring knowledge of the credentials of the virtual machine.

VMware's NSX Guest Introspection software (the latest released version as of 27 August 2021 for example) is an example of software that allows virus scanning, and is a virus scanner known in the art that allows external scanning without requiring knowledge of the credentials of the virtual machine. In the present disclosure, the inventors have identified that virus scanners, which have not previously been deployed in the context of quarantine environments in automated disaster recovery, beneficially allows for the automated scanning of a plurality of VM versions as part of a batch recovery process. Further, once one VM has been scanned by the anti-virus, subsequent VMs can be scanned with knowledge of infection signatures and objects identifies in the previous anti-virus scan.

Some virus scanners may not be able to clean virtual machines without user credentials. Once a batch of VMs have been scanned and the outcome of the scans is provided to a user, the user can login to a selected VM to enable antivirus software to clean that VM. The batch of VMs may relate to different machines or to the same machine.

FIG 3. ARCHITECTURE AND LOGICAL FLOW OF A COMPUTER-IMPLEMENTED METHOD FOR OPERATING A RANSOMWARE RECOVERY SYSTEM FROM THE POINT OF EXECUTION OF A GLOBAL FILE INFECTION SEARCH UP TO A PREDATAR QUARANTINE AND CLEAN (PQC) PRODUCTION

| BEHAVIOUR NODE | DESCRIPTION |
|---|---|
| 300 | COMPUTER-IMPLEMENTED METHOD FOR OPERATING A RANSOMWARE RECOVERY |
| 302 | STEP 1 – SELECTED MACHINES (SINGLE, MULTIPLE,GROUP) |
| 304 | STEP 2 – COMMAND SENT TO CLIENT CONTAINING LIST OF FILE EXTENSIONS |
| 306 | STEP 3 DECISION STEP - DETERMINES WHETHER THE MACHINE OR MACHINES ARE BACK-UP UP BY IBM® SPECTRUM PROTECT (SP) OR IBM® SPECTRUM PROTECT PLUS (SPP). |
| 308 | STEP 4 – A COMMAND IS RUN ON THE SP BACK-UP AND ARCHIVE COMMAND LINE |
| 310 | STEP 5 - API (APPLICATION PROGRAMMING INTERFACE) IS CALLED WHICH WILL RETURN A LIST OF FILES WITH AN INFECTION SIGNATURE THAT WERE BACKED UP UNDER A RELEVANT VIRTUAL MACHINE (VM). |
| 312 | STEP 6 – DECISION STEP - DETERMINES WHETHER ANY FILES HAVE BEEN IDENTIFIED THAT HAVE AN INFECTION SIGNATURE WITHIN A BACK-UP. |
| 314 | STEP 7 - BACK-UP IS MARKED AS DIRTY (I.E. AS INFECTED WITH MALWARE) |
| 316 | STEP 8 – DECISION STEP - CHECKS TO DETERMINE WHETHER THERE EXISTS ANY PREVIOUS (I.E. EARLIER) BACK-UP. |
| 318 | STEP 9 - BACK-UP IS MARKED AS CLEAN |
| 320 | STEP 10 - MACHINE IS MARKED AS CRITICAL |
| 322 | STEP 11 - USER CAN DRAG AND DROP THE MACHINES SHOWN AS DIRTY ONTO THE 'RECOVERY GFIS CLEAN TO PQC' ACTION. |
| 324 | STEP 12 - A COMMAND IS SENT TO THE CLIENT CONTAINING A LIST OF MACHINES DRAGGED AND DROPPED ON TO THE 'RECOVER GFIS CLEAN TO PCQ' ACTION. |
| 326 | STEP 13 – DECISION STEP – IS THIS AN SP BACK UP? |
| 328 | STEP 14 - SP VMS CLIENT WILL SEND THE COMMAND TO SP FOR THE VM TO RECOVER THE RELEVANT MACHINES TO A QUARANTINE NETWORK THAT IS ISOLATED FROM THE REST OF THE COMPUTER SYSTEM. |
| 330 | STEP 15 - VMS CLIENT CALLS AN SPP API TO RECOVER THE MACHINES TO A QUARANTINE NETWORK |
| 332 | STEP 16 - THE CLIENT SENDS A COMMAND BACK TO THE PORTAL INDICATING IF THE RECOVERY HAS BEEN SUCCESSFUL OR HAS FAILED |
| 334 | STEP 17 – DECISION STEP – IS THE RECOVERY SUCCESSFUL? |
| 336 | STEP 18 - AN ERROR MESSAGE ON THE PORTAL IS DISPLAYED TO SHOW THAT THE RECOVERY WAS UNSUCCESSFUL |
| 338 | STEP 19 - THE CLIENT RECEIVES THE COMMAND AND CUTS OFF THE CONNECTION TO THE MAIN NETWORK AND CONNECTS INSTEAD TO THE QUARANTINE NETWORK. |
| 340 | STEP 20 – DECISION STEP - DETERMINE WHETHER TO RUN A PHP (PREDATAR HUNT PACK) SCRIPT |
| 342 | STEP 21 - THE CLIENT RUNS THE PHP SCRIPT ON THE RECOVERED VM TO DETERMINE IF THERE ARE FILES PRESENT WITH CERTAIN FILE EXTENSIONS ON THE RECOVERED VMS THAT ARE INFECTION SIGNATURES. |
| 344 | STEP 22 – DECISION STEP - DETERMINE WHETHER ANY INFECTED FILES HAVE BEEN FOUND BY THE PHP SCRIPT. |
| 346 | STEP 23 - THE INFECTED MACHINE IS SHUT DOWN |
| 348 | STEP 24 - THE CLIENT INSTALLS ANTI-VIRUS SOFTWARE AND RUNS A SCAN ON THE RECOVERED VM |
| 350 | STEP 25 – DECISION STEP - DETERMINES THAT A VIRUS HAS BEEN FOUND? |
| 352 | STEP 26 - INFECTED MACHINE IS SHUT DOWN |

| | |
|---|---|
| 354 | STEP 27 - THE MACHINE IS SHOWN AS CLEAN IN THE GUI. |
| 356 | STEP 28 - THE USER CAN DRAG AND DROP THE MACHINES THAT ARE SHOWN AS GREEN ON TO THE 'PQC MOVE CLEAN TO PRODUCTION' ACTION |
| 358 | STEP 29 - A COMMAND IS SENT TO THE CLIENT CONTAINING THE LIST OF THE MACHINES THAT HAVE BEEN RECOVERED |
| 360 | STEP 30 - THE CLIENT RECEIVES THE COMMAND AND WILL THE RECOVER THE VM TO THE PRODUCTION ENVIRONMENT |
| 362 | STEP 31 - THE RESULT OF THE RESTORATION IS SENT TO THE PORTAL. |

Figures 4 to 14 illustrate aspects of a graphical user interface (GUI) that may be used for example by a user, to action or implement steps of the methods described previously. The following disclosure provides a description of the elements of the GUI, and aspects of the GUI that implement features of the method described previously.

Figure 4 illustrates a GUI layout 400 comprising a number of portions, including a navigation portion 402, an activity portion 404, an actions portion 406, a protection map (or grid) portion 408 and a groups portion 410. It will be appreciated that the layout of the various portions on the display may be varied as required, and that not all portions 402 – 410 may be displayed in a single GUI layout. Various features of the respective portions are described further with reference to the figures below.

FIG 4. A GRAPHICAL USER INTERFACE (GUI) LAYOUT COMPRISING A NUMBER OF PORTIONS.

| PORTION | DESCRIPTION |
|---|---|
| 400 | GUI LAYOUT. |
| 402 | NAVIGATION PORTION. |
| 404 | ACTIVITY PORTION. |
| 406 | ACTIONS PORTION. |
| 408 | PROTECTION MAP (OR GRID) PORTION. |
| 410 | STEP 5 - API (APPLICATION PROGRAMMING INTERFACE) IS CALLED WHICH WILL RETURN A LIST OF FILES WITH AN INFECTION SIGNATURE THAT WERE BACKED UP UNDER A RELEVANT VIRTUAL MACHINE (VM). |

Figure 5 illustrates the navigation portion 402 and the activity portion 404. The navigation portion 402 comprises a number of buttons, or tabs, that link to various features of the software package. In the present case, the recovery tab 503 is selected, resulting in the display of the GUI layout shown in figure 4. The activity portion 404 comprises a panic button 505 to initiate the

recovery engine. The activity portion 404 also comprises a list 407 of recent events. The number of recent events displayed may be controlled and the events may be colour-coded. The events may comprise a list of the actions taken by the software, as discussed below.

When a user is informed of or suspects that a threat has occurred, they may initiate the recovery engine using the panic button 505. Activating the panic button 505 causes a dialogue box to be rendered for the user to enter a description of the threat, which may be provided to one or more users (which may be predefined in a War Room scenario as described previously). The panic button 505 initiates the recovery process and causes a dialogue box to be generated as described below with reference to figure 6a.

FIG 5. A NAVIGATION PORTION AND AN ACTIVITY PORTION OF A GUI.

| PORTION | DESCRIPTION |
|---|---|
| 402 | NAVIGATION PORTION. |
| 404 | ACTIVITY PORTION. |
| 503 | RECOVERY TAB. |
| 505 | PANIC BUTTON UNDER THE ACTIVITY PORTION. |
| 507 | LIST OF EVENTS UNDER THE ACTIVITY PORTION. |

Figures 6a to 6d illustrate various style of boxes that may be used in combination with the GUI layout of figure 4 at various stages in the process.

Figure 6a illustrates an example of the dialogue box or receiving user input to describe a threat situation. The elements of the dialogue box are described in the table below.

| # | ELEMENT | TITLE | DESCRIPTION |
|---|---|---|---|
| 1 | BUTTON | DONE | INPUT COMPLETE, AUTHENTICATE, AND INFORM WAR ROOM USERS OF THREAT EMAIL / TICKET |
| 2 | FIELD | REASON / THREAT | FREE TEXT FIELD TO DESCRIBE THE THREAT |
| 3 | RID TIME | FIELD | DATE FIELD FOR SUSPECTED INFECTION OR SYMPTOMS. |
| 4 | PW AUTHENTICATION | PASSWORD | PASSWORD FOR WAR ROOM USER REQUIRED TO INVOKE RECOVERY RESPONSE. |

On entering the information into the dialogue box, a ticket may be raised, or an email distributed, to send a message to the predefined War Room users. The message may contain the information provided by the user and additional pre-agreed information, such as muster details, for example, attend zoom meeting/come into office, etc.

An example of elements of the message to the War Room users is provided in the table below.

| # | ELEMENT | TITLE | DESCRIPTION |
|---|---|---|---|
| 1 | FIELD | REASON / THREAT | FREE TEXT FIELD TO DESCRIBE THE THREAT |
| 2 | FIELD | MUSTER DETAILS | FREE TEXT WITH ZOOM DETAILS ETC... |
| 3 | RID TIME | FIELD | DATE FIELD FOR SUSPECTED INFECTION OR SYMPTOMS. |

The global RID times may be auto-populated for all recovery groups of machines or virtual machines. The RID recovery time may also be changed at a number of levels, for example at machine level, at group level or globally. Such functionality may be useful because the RID time may vary by geography, operating system or machine domain.

Figure 6b illustrates a dialogue box for setting RID time information, and contains the elements described in the table below.

| # | ELEMENT | TITLE | DESCRIPTION |
|---|---|---|---|
| 3 | RID TIME | FIELD | DATE FIELD FOR SUSPECTED INFECTION OR SYMPTOMS. |

Turning to Figures 9 and 10, which show examples of the protection map 908, 1008, or grid, introduced previously with reference to figure 4, the protection map graphically illustrates the status of the respective back-ups for each machine of interest as a function of time/date. Colour coding may be used to denote a back-up that is known to be infected (for example red or flashing red for a confirmed infected back-up, bright green for a confirmed clean back-up, pale green for an assumed clean back-up).

In figure 9, the whole first row in the protection map is marked to signify that the node is critical because no uninfected back-up is available. In such case, the War Room users may be alerted by Email / Ticket.

The RID time for each machine may also be displayed. In this example, each machine is provided on a different row, each column represents a different back up time slot, and a vertical line 909 represents an RID time associated with a particular machine. A header in the protection map 908 includes headings for the server, machine type, entity and the date/time of each back-up slot.

FIG 9. AN EXAMPLE OF A PROTECTION MAP.

| FEATURE | DESCRIPTION |
|---|---|
| 908 | PROTECTION MAP GRAPHICALLY ILLUSTRATING THE STATUS OF THE RESPECTIVE BACK-UPS FOR EACH MACHINE OF INTEREST AS A FUNCTION OF TIME/DATE. |
| 909 | VERTICAL LINE REPRESENTING AN RID TIME ASSOCIATED WITH A PARTICULAR MACHINE. |

An example of a description of the interface elements for the status portion is provided in the table below.

| # | Element | Title | Description |
|---|---|---|---|
| 1 | RID Time | Toggle | Switch to remove or add RID Time icon for suspected infection or symptoms. |
| 2 | RID Time Marker | Icon | Blue line through cells displaying latest RID update for the cell. |
| | In the Protection map may provide an option for viewing RID time markers, which when turned on will show the RID marker lines for each machine.<br>If no RID time has been specified for a machine, then the RID time specified at a recovery group level that machine is included in may be used. | | |

Figures 7a to 7e illustrate various aspects of the actions portion 406.

| Figure | Description |
|---|---|
| 7a | An illustration of the actions portion. |
| 7b | A blown-up illustration of the actions portion, showing example secondary selections for a user. |
| 7c | A blown-up illustration of the actions portion, showing an example sub menu for a user. |
| 7d | A blown-up portion of the actions portion, showing a graphical representation provided at the cursor when multiple nodes have been selected and dragged from the protection map to the actions portion. |
| 7e | A blown-up portion of the actions portion, showing a graphical representation that is provided at the cursor when an entire group is selected from the groups portion and dragged to an action in the actions portion. |

The actions portion may provide a menu of configurable or preconfigured actions. In one mode of use, a user may drag and drop groups of machines from the group portion 410 onto specific actions in the actions portion 406. In this way, the user may control the technical task of performing data recovery for a group of machines in an improved manner compared to existing interfaces, which may involve entering many lines of command line instructions in order to achieve the same effect. In order to that implement such functionality, it may be convenient for the groups portion to be located adjacent to the actions portion 406. In addition to the functionality of group selection provided using the abstraction of the predefined groups in the groups portion 410, the machines associated with the individual rows in the protection map

408 may be selectable by clicking and dragging that row, for example, onto an action in the actions portion 406 in order to perform a selected task for a particular machine. For this reason, it may also be preferable for the protection map 408 to be provided adjacent to the actions portion 406.

The actions portion may contain the actions to: enter infection signature, search for uninfected back-ups, recover GFIS clean to PCQ, Install and Run PHP in PQC, Install and run AV, PQC Move Clean to production, Flag as condemned hardware for OS reinstall and notify when ready for BA Client/TDP Restore (reinstallation of the Operating System and then flag when that is done so that the Back-up Client Software can be reinstalled manually), Declare OS platforms/versions clear here and prompt for removal from recovery groups, Search for latest Pre RID Time back-ups and recover to safety, and Immediately back-up all clearly scanned boxes.

With reference to these actions in the recovery process, a user can proceed through the recovery exercise by dragging and dropping individual or multiple recovery point cells from the protection map onto the actions, or by dragging whole groups onto the actions, as described previously. The cell status in the protection map will change as the process proceeds.

Figure 7b illustrates a blown up portion of the action portion 406 illustrated in figure 7a. In this example, if a user performs a secondary selection of a particular action (for example right click) the user can choose from a sub menu to edit, delete or copy the action.

If the user is to perform a primary selection (for example, left click mouse) on one of the actions, a dialogue box may be generated. For example, if a user were to primary select the live enter infection signature action illustrated in figure 7b, the dialogue box illustrated in figure 6c may be generated to receive user input.

Turning to the production map illustrated in figure 10, it can be seen that all of the nodes are flagged such that they are ready to be moved to production. A user may perform this action by dragging the required nodes to the "*flag as*

*confirmed hardware for OS reinstall and notify when ready for BA client/TP restore"* action, which generates the sub menu shown in figure 7c, including the options: move to production, view status, clean up and view history. Alternatively, as shown in figure 9, one of the nodes has failed its antivirus process for the most recent back-up, resulting in the RID time 909 for that node being pushed back to the next most recent back-up.

Figure 7d illustrates a graphical representation that is provided at the cursor when multiple nodes have been selected and dragged from the protection map to the actions portion.

Figure 7e illustrates a graphical representation that is provided at the cursor when an entire group is selected from the groups portion and dragged to an action in the actions portion.

In both figures 7d and 7e, the graphical representation of the node or group of nodes is translucent over the remainder of the GUI layout.

Figure 6d illustrates an example of a dialogue box that may be generated when a plurality of nodes are selected and dragged to a particular action. The dialogue box seeks confirmation that the action is required for each of the selected nodes. A description of the action is provided in the table below.

| # | ELEMENT | TITLE | DESCRIPTION |
|---|---|---|---|
| 1 | ACTION BUTTON | BUTTON CUSTOMISATION | PORT AND ENHANCE ACTION EDIT FUNCTION FROM CLIENT EVENTS TO ALLOW RECOVERY POINT OR POINTS TO BE DRAGGED AND DROPPED FROM THE RECOVERY GROUP SIMILAR TO CLIENT EVENTS FUNCTION. |

Figure 8 illustrates a GUI profile providing a data rate transfer report, which may be provided in addition to those included in Figure 4. A ransomware attack may result in a large increase in back-up size due to the encryption of data on the node. Similarly, an attack may result in an increase in data traffic from a node in cases in which data is removed from the node by malware. Unusual

changes in data rate or back-up size may therefore be used to determine the RID time of an infection event.

In Figure 8, a GUI provides a deviation report which may be togglable with the protection map in Figure 4. The deviation report shows a data transfer size corresponding to each of the back-ups for each machine. The grid structure corresponds to that described previously with reference to the protection map. The same machine order may be maintained between both views. In this way, the user may find a suspect deviation then toggle to the same node in the protection map to understand if there is a correlation with the RID Time.

FIG 10. A FURTHER EXAMPLE OF A PROTECTION MAP.

| FEATURE | DESCRIPTION |
|---|---|
| 1008 | PROTECTION MAP GRAPHICALLY ILLUSTRATING THE STATUS OF THE RESPECTIVE BACK-UPS FOR EACH MACHINE OF INTEREST AS A FUNCTION OF TIME/DATE. |

Figure 11 shows an example computer program product 1100 (equivalently, a computer readable memory medium) that contains instructions that, when executed, cause an system, as described in relation to figure 2, to at least perform steps of any of the methods described herein.

FIG 11. AN EXAMPLE COMPUTER PROGRAM PRODUCT.

| FEATURE | DESCRIPTION |
|---|---|
| 1100 | COMPUTER PROGRAM PRODUCT (EQUIVALENTLY, A COMPUTER READABLE MEMORY MEDIUM). |

Figure 12 illustrates a functional process flow for a procedure for detecting anomalies in back-up data. When ransomware infects a machine, it is typical for the size of back-up files taken from the machine to substantially increase in size because, for example, the ransomware may be encrypting particular files on the machine and so creating substantial differences in the files seen by back-up software compared with earlier versions. Ransomware also may increase general data traffic from a particular machine by transmitting files, either encrypted or unencrypted to a malicious third party. Such increases in back-up file size due to ransomware may be considered to result in anomalies in the backed-up data.

After the process initiates 1202, a back-up process, such as a scheduled back-up routine, may be executed 1204 on one or more machines. The back-up process may be managed by the software application described previously with reference to figures 1-11. In this example, the software is further configured to collect analytics regarding the back-up data. The types of anomaly in the back-up data that may be detected are twofold, as discussed below.

Firstly, every time a back-up is taken, the amount of data that is associated with that particular back-up may be recorded. The size of each back-up may be recorded over a prolonged period. For example, the software may retain the size of respective back-up copies associated with a particular machine for a period of 30, 60 or 90 days, for example. The software may be configured to determine whether the size of a particular back-up is greater than the size of corresponding historical back-ups by a predetermined amount, which may be set by a user. Alternatively, the software may be configured to execute pattern matching algorithms or similar in order to learn back-up behaviour associated with a particular machine. The method is not specific to a particular implementation of pattern matching algorithm and may be implemented using standard methods used in machine learning. For example, if a back-up routine is configured to perform a daily back-up in which only deltas (difference files) are transmitted on a network, and transmits a full back-up once a week, for example on Saturdays, then the day that is the subject of a full back-up will have a substantially greater data transfer rate than the other days in which deltas are transmitted. However, although the increase in data flow for the back-up process on the full back-up day may be, for example, 20 times greater than that for the other days, nothing untoward is necessarily taking place. By observing network data traffic over a prolonged period, the software may learn such back-up behaviour associated with a particular machine without a priori knowledge of the settings of the back-up software on that machine. In this way, the software may be configured to ignore expected increases in back-up size and not flag them as anomalous back-ups.

Secondly, in addition to monitoring data transfer sizes of back-up copies in order to detect anomalies as discussed above, the software may be further

configured to search back-up files for references to known viruses. Such scanning may be achieved using conventional antivirus software. The list of virus definitions may be updated periodically and automatically as described previously with reference to figures 1-11, for example. This differs from a conventional approach in which, for example, an absolute deviation in data rate may be identified as a possible indicator of a ransomware attack. That is, the system may look for periods in which the data rate is significantly higher than a mean value, for example. Such prior art approaches may result in false negatives that are avoided by the present machine learning approach.

It has been found that particularly advantageous method for searching back-up copies for known infection objects may be achieved by scanning only metadata associated with the back-up copies to search for signatures of infection (for example, particular terms in file lists that indicate the activity of a known piece of ransomware). Such scanning, focused on the metadata, is computationally more efficient than scanning the whole body of the back-up in question. In addition, scanning metadata associated with a back-up copy may be achieved without the need to have a decrypted the bulk of the data for the particular back-up. In this way, scanning the metadata associated with a back-up may be achieved for some back-up systems that incorporate end-to-end encryption without the need for decrypting data either on route or at the back-up server.

The process in effect automatically detects 1208 whether an infection event is likely to have occurred. If no anomaly is detected 1208 by the software using the above processes, the process ends 1212. If the software does detect 1208 an anomaly, the back-up that is subject to the anomaly may be flagged 1210 for further action. For example, potentially infected back-ups may be moved to a quarantine area for full virus scanning and cleansing, if necessary. A process for orchestrating the recovery of potentially infected back-ups was described previously as reference to figures 1-11, and the system of raising tickets or flags and alerting the configured end user described previously may be implemented to allow potentially infected back-up copies to be treated. Once the back-up copy has been found to be safe or cleansed, the process ends 1212.

FIG 12. A FUNCTIONAL PROCESS FLOW FOR A PROCEDURE FOR DETECTING ANOMALIES IN BACK-UP DATA.

| STEP | DESCRIPTION |
|---|---|
| 1202 | THE PROCESS IS INITIATED. |
| 1204 | A BACK-UP PROCESS, SUCH AS A SCHEDULED BACK-UP ROUTINE, MAY BE EXECUTED ON ONE OR MORE MACHINES. |
| 1208 | AUTOMATIC DETECTION OF WHETHER AN INFECTION EVENT IS LIKELY TO HAVE OCCURRED. |
| 1210 | IF THE SOFTWARE DOES DETECT AN ANOMALY, THE BACK-UP THAT IS SUBJECT TO THE ANOMALY MAY BE FLAGGED FOR FURTHER ACTION. |
| 1212 | IF NO ANOMALY IS DETECTED BY THE SOFTWARE USING THE ABOVE PROCESSES, THE PROCESS ENDS. |

Figure 13 illustrates a further method 1300 for detecting a suspected infection event. The method may be implemented within a system for backing-up and subsequently restoring a computer system using a back-up server such as that described previously with reference to figure 2, for example. As described previously, such a system has a plurality of distinct machines to be backed-up, including a first machine and a second machine, and each machine may be a physical or virtual machine.

As a starting point, the method 1300 receives 1302 an indication that a machine in the system, for example the first machine, is infected with software such as ransomware or malware. The determination that the first machine is infected may be made previously by one of the other methods described herein, for example using the machine learning algorithm described previously with reference to figure 12, based on an antivirus scan or infection signature search in the processes described previously with reference to figure 3, or even based on manual user input using a graphical user interface such as those described previously with reference to Figures 4 to 10.

The presence of ransomware on a machine typically results in changes to back-up data traffic associated with that machine. In a further step of the method 1300, a pattern in the back-up data traffic associated with the first machine is identified 1304. The pattern may comprise data points that are each indicative of a size of an associated back-up copy from the first machine taken at

different, respective points in time. The size of a particular back-up copy may be based on that of the copy itself or its metadata.

The pattern in the size of data associated with back-up copies from the first machine that has been identified 1302 is subsequently used to determine 1306 whether to classify data associated with at least one back-up copy from one or more other machines in the system, such as the second machine. Whether the pattern is compared with data from any other particular machine or group of machines in the system may be directed by user input via a graphical user interface element such as those described previously with reference to Figures 4 to 10. In this way, the automated method 1300 is able to determine the suspected presence of an infection on one machine based on similarities between the back-up data traffic over time from that machine and corresponding back-up data traffic for another machine.

The first machine and second machine may be operated by entirely unconnected users. For example, the first machine and second machine may be devices belonging to different businesses and used for entirely different purposes. In employing such methods, there is no need to share private information between the first machine and the second machine. In one example, the method for detecting anomalies in the back-up data may be performed by a server that is remote from both the first machine and the second machine, in which case there is no need to share any information about the first machine with the second machine, and vice versa. Alternatively, if the first machine is determined to be infected, the pattern identified in the data associated with back-ups of the first machine may be shared directly with the second machine. However, the pattern that is shared does not need to provide any information identifying the first machine, and may merely represent anonymised data transfer information for an unknown machine.

In some examples, the pattern that is used in the process of determining whether a second machine is infected may comprise raw back-up data transfer against time information. The pattern may have a component attributable to the infection event and another component attributable to ordinary usage. Alternatively, the pattern may be derived from the raw data using conventional

statistical methods or machine learning tools in order to extract from the raw data components that are attributable to the infection event. For example, the pattern may be compared with an earlier corresponding usage period, or multiple periods, to attempt to identify infection events as described previously with reference to Figure 12.

Figure 14 illustrates a profile of the back-up transfer size against time for a particular machine. In this example, an infection has been found on the machine. The profile of Figure 14 illustrates a behaviour shape 1402 over a time period 1404 between the earliest detection (ED) and the time the scan completed (SC). The behaviour shape 1402 may be considered to provide the pattern for comparison with the data from other machines.

The behaviour shape 1402 for the infected machine may be stored by a server, for example in a global repository in the cloud. The behaviour shape 1402 may then be used as a template for comparison with other nodes that have their back-up process managed by the server. In some examples, corresponding data usage for all customers with back-ups managed by the server may be compared with the same behaviour shape found from the infected server.

It has been found that the same behaviour shape for a particular type of infection is maintained but the timeframe over which the infection spreads may differ depending on the machine and its environment. For this reason, the behaviour shape from the infected machine may be compared with traffic data from other machines by pattern matching the shape profile irrespective of the timeframe. That is, if the same pattern is seen it may be an indication of an infection irrespective of whether the matching data corresponds to a timeframe of hours, days, weeks or months.

Figure 15a illustrates a behaviour shape 1502a that corresponds to the behaviour shape identified in the example of Figure 14 over a shorter time period, 2 weeks in this example.

Figure 15b illustrates a behaviour shape 1502b that corresponds to the behaviour shape identified in the example of Figure 14 over a longer time period, 2 months in this example.

If it is determined that a back-up data transfer profile of a second machine is similar to the behaviour shape found in an infected first machine, a threat score can be determined to provide an indication of the likelihood that the second machine is in fact infected. The use of a threat score can assist in reducing the number of false positives and so provide a better means for determining whether a particular machine is infected.

The concept of the threat score is discussed further with reference to figures 16a to 16c. In this example, the threat score is a rating between 0 and 10, where 0 is the lowest possible threat and 10 is the highest possible threat.

Figure 16a illustrates a back-up data transfer profile in which the threat score is 0. In this example, the threat score has been assigned the value 0 because the back-up data transfer profile of the node matching the behaviour shape had been scanned in the same time period as the most recent scan (scan complete) the first node in which the infection was detected and found to have been clean. That is, the second node may have been scanned on the same day, for example, as the scan complete scan. As such, it is unlikely that the second node is in fact infected even though the back-up data transfer profile of the second node has been found to substantially match the behaviour shape of the first node.

Figure 16b illustrates a back-up data transfer profile for a node that in which the threat score is 10. If a node in which the back-up data transfer profile has been found to match the behaviour pattern has never been scanned or was last scanned before the earliest data in which the infection signature was detected in the first node then the second node will be marked with a threat score of 10 (the highest possible mark) because there is a higher likelihood that the match is indeed indicative of an infection event.

Figure 16c illustrates a back-up data transfer profile for a node that in which the threat score is 3, 5 and 7. A scaling factor of threat score between 0 and 10 (the lowest and highest possible ratings) is assigned if the last scan of the node is between the earliest date of infection of the first node and the scan completion date of the first node. In the illustrated example, the threat score scales lineally between 0 at the scan completion date and 10 at the earliest detected date and examples are illustrated of markers for the most recent scan of the second node corresponding to threat score of 3, 5 and 7.

## CLAIMS

1. A computer-implemented method for detecting a suspected infection event, the method comprising:

receiving data associated with back-up copies of a plurality of machines including at least a first machine and a second machine, in which the data is indicative of a size of the associated back-up copy; and

determining whether to classify data associated with back-up copies of at least a second machine as anomalous based on a pattern identified in data associated with back-up copies of the first machine.

2. The method of claim 1, further comprising:

receiving data associated with each of a plurality of back-up copies associated with the first machine in which the data is indicative of a size of the associated back-up copy;

searching the plurality of back-up copies associated with the first machine to identify the earliest back-up copy that comprises a signature of the infection event; and

wherein the pattern in data associated with back-up copies of the first machine comprises a behaviour shape between the earliest back-up copy that comprises the signature of the infection event and the most recent back-up in which the infection event was detected.

3. The method of claim 2, wherein the behaviour shape is a back-up data transfer profile as a function of time.

4. The method of claim 2 or claim 3, further comprising:

in response to classifying a back-up copy of the second machine as anomalous, determining a score indicative of a likelihood of infection based on when the most recent antivirus scan was performed on the second machine.

5. The method of claim 4 comprising generating a graphical user interface providing:

an indication whether back-up copies of one or more of the plurality of machines have been classified as anomalous; and

the score indicative of a likelihood of infection associated with one or more of the one or more of the plurality of machines have been classified as anomalous.

6. The method of claim 4 or claim 5 comprising prioritizing the recovery of machines associated with a score indicative that infection is more likely over the recovery of machines with a score indicative that infection is less likely.

7. The method of any of claims 4 to 6, in which the score is scaled based on the time the most recent antivirus scan was performed on the second machine between:

a time associated with the earliest back-up copy of the first machine that comprises the signature of the infection event; and

a time associated with the most recent back-up of the first machine.

8. The method of claim 1, further comprising:

receiving data associated with each of a plurality of back-up copies associated with the first machine, in which the data is indicative of a size of the associated back-up copy; and

training a pattern matching algorithm to determine whether to classify data associated with back-up copies as an anomalous pattern using the data associated with each of a plurality of back-up copies of the first machine.

9. The method of claim 8, further comprising using the trained pattern matching algorithm to determine whether to classify the data associated with back-up copies of the second machine as anomalous.

10. The method of any preceding claim, further comprising scanning a back-up copy that is classified as anomalous using antivirus software.

11. The method of any of claims 1 to 9, further comprising scanning metadata of a back-up copy that is classified as anomalous using antivirus software.

12. The method of any preceding claim, wherein determining whether to classify data associated with back-up copies of the second machine as anomalous is based one or more patterns identified in data associated with back-up copies associated with a first plurality of machines.

13. An apparatus comprising:

at least one processor; and

at least one memory including computer program code for one or more programs,

the at least one memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform the method of any of any preceding claim.

14. A computer program product including one or more sequences of one or more instructions which, when executed by one or more processors, cause an apparatus to at least perform the method of any of claims 1 to 12.

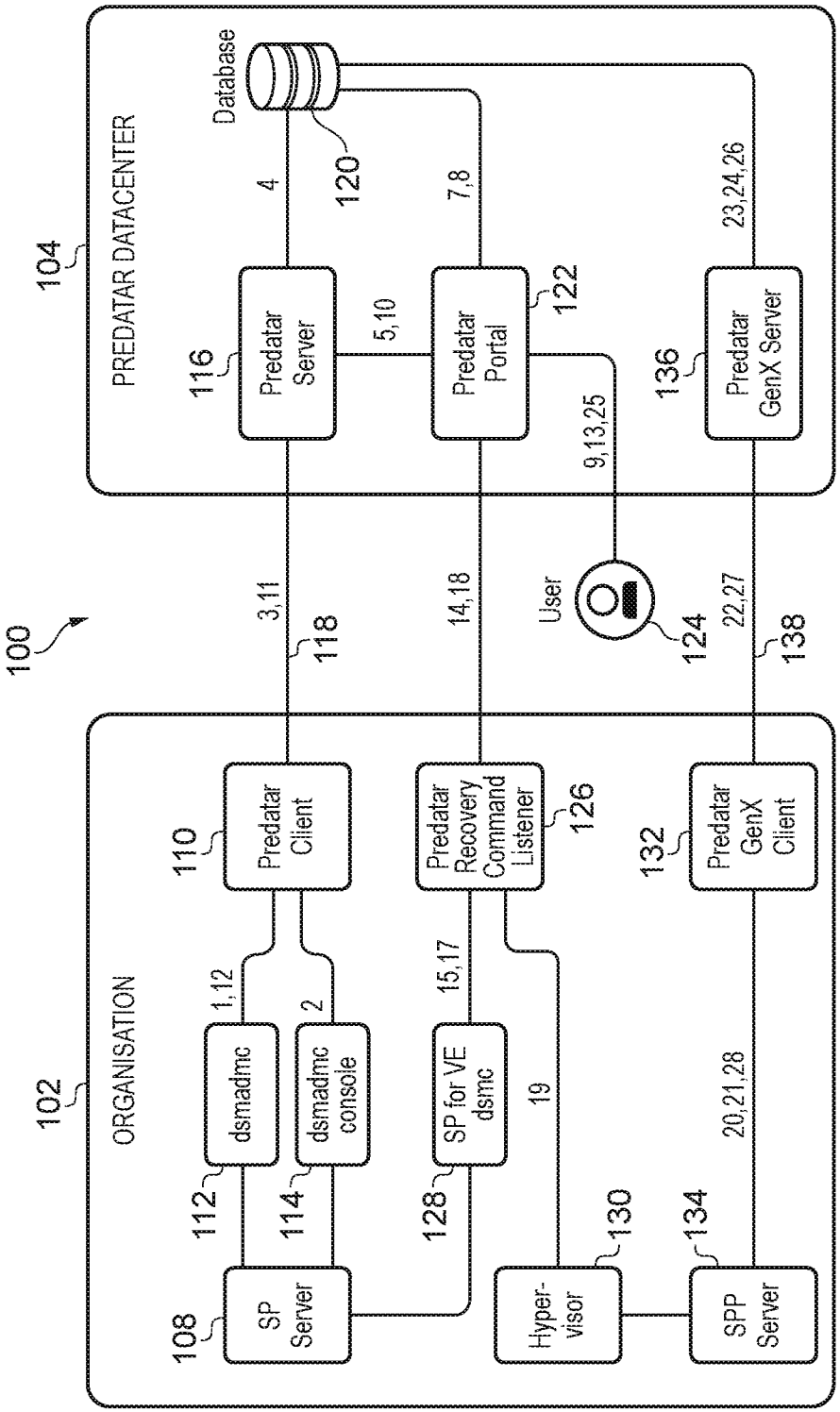
100
102
ORGANISATION
104
PREDATAR DATACENTER
Database
120
Predatar Server
116
Predatar Portal
122
Predatar GenX Server
136
Predatar Client
110
Predatar Recovery Command Listener
126
Predatar GenX Client
132
dsmadmc
112
dsmadmc console
114
SP for VE dsmc
128
SP Server
108
Hyper-visor
130
SPP Server
134
User
124
4
5,10
7,8
9,13,25
23,24,26
3,11
118
14,18
22,27
138
1,12
2
15,17
19
20,21,28

FIG. 1

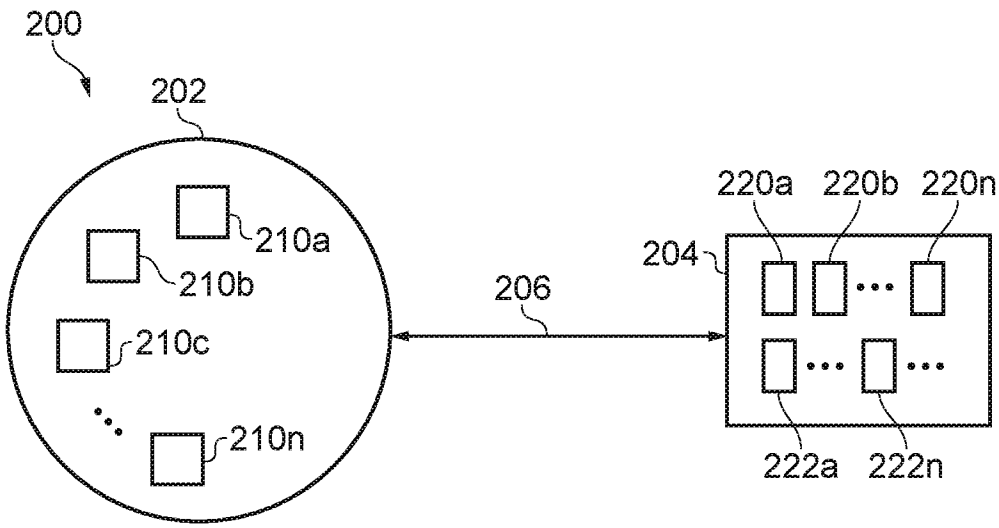


FIG. 2

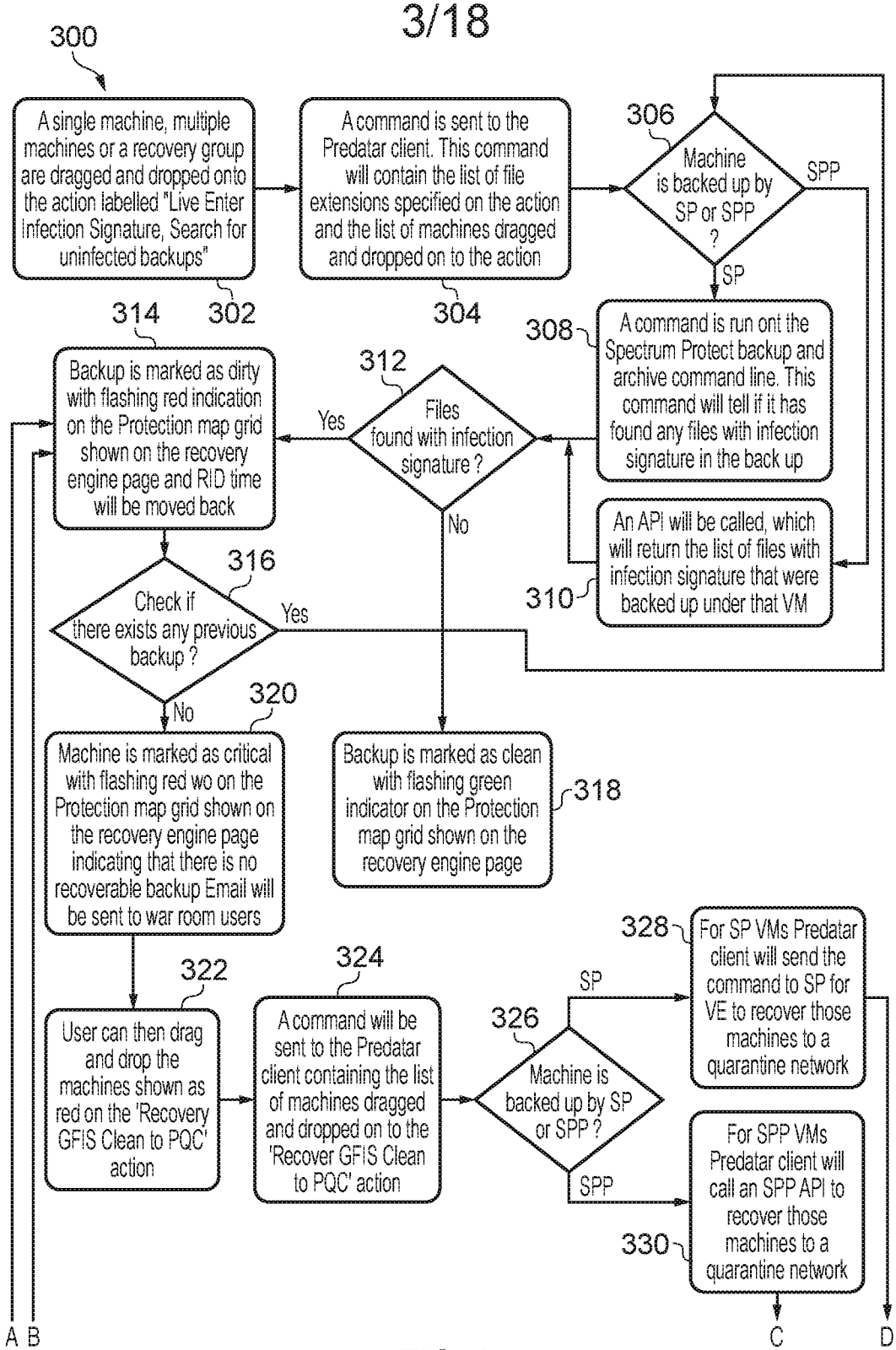
300
A single machine, multiple machines or a recovery group are dragged and dropped onto the action labelled "Live Enter Infection Signature, Search for uninfected backups"
302
A command is sent to the Predatar client. This command will contain the list of file extensions specified on the action and the list of machines dragged and dropped on to the action
304
306
Machine is backed up by SP or SPP ?
SPP
SP
308
A command is run ont the Spectrum Protect backup and archive command line. This command will tell if it has found any files with infection signature in the back up
310
An API will be called, which will return the list of files with infection signature that were backed up under that VM
312
Files found with infection signature ?
Yes
No
314
Backup is marked as dirty with flashing red indication on the Protection map grid shown on the recovery engine page and RID time will be moved back
316
Check if there exists any previous backup ?
Yes
No
318
Backup is marked as clean with flashing green indicator on the Protection map grid shown on the recovery engine page
320
Machine is marked as critical with flashing red wo on the Protection map grid shown on the recovery engine page indicating that there is no recoverable backup Email will be sent to war room users
322
User can then drag and drop the machines shown as red on the 'Recovery GFIS Clean to PQC' action
324
A command will be sent to the Predatar client containing the list of machines dragged and dropped on to the 'Recover GFIS Clean to PQC' action
326
Machine is backed up by SP or SPP ?
SP
SPP
328
For SP VMs Predatar client will send the command to SP for VE to recover those machines to a quarantine network
330
For SPP VMs Predatar client will call an SPP API to recover those machines to a quarantine network
A
B
C
D

FIG. 3

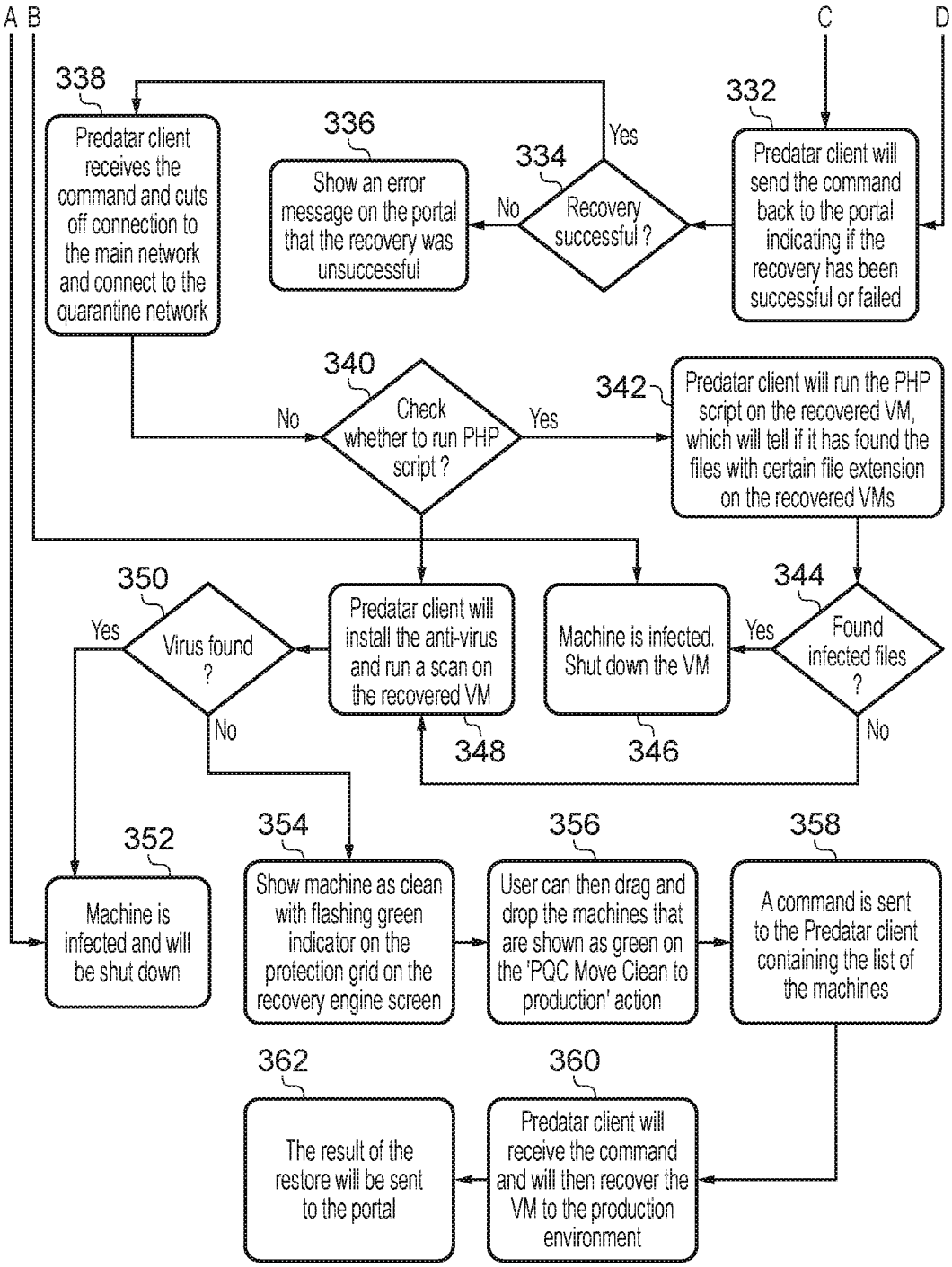


FIG. 3 (Continued)

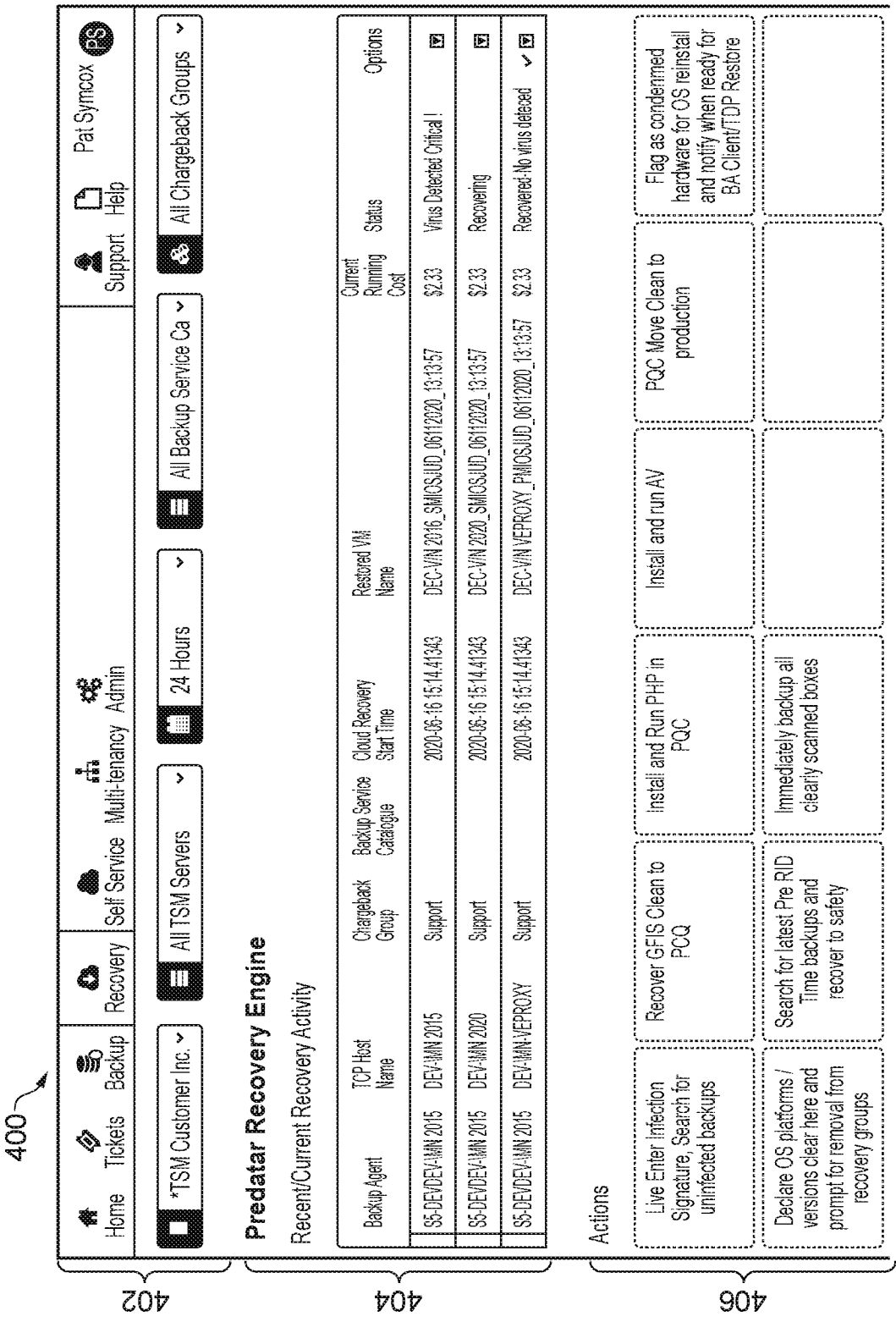
400
Home
Tickets
Backup
Recovery
Self Service
Multi-tenancy
Admin
Support
Help
Pat Symcox
PS
*TSM Customer Inc.
All TSM Servers
24 Hours
All Backup Service Ca
All Chargeback Groups
402
Predatar Recovery Engine
Recent/Current Recovery Activity
Backup Agent
TCP Host Name
Chargeback Group
Backup Service Catalogue
Cloud Recovery Start Time
Restored VM Name
Current Running Cost
Status
Options
S5-DEV\DEV-\MN 2015
DEV-\MN 2015
Support
2020-06-16 15:14.41343
DEV-VM 2016_SMIOSJUD_06112020_13:13:57
$2.33
Virus Detected Critical !
S5-DEV\DEV-\MN 2015
DEV-\MN 2020
Support
2020-06-16 15:14.41343
DEV-VM 2020_SMIOSJUD_06112020_13:13:57
$2.33
Recovering
S5-DEV\DEV-\MN 2015
DEV-\MN-VEPROXY
Support
2020-06-16 15:14.41343
DEV-VM VEPROXY_PMIOSJUD_06112020_13:13:57
$2.33
Recovered-No virus detected
404
Actions
Live Enter Infection Signature, Search for uninfected backups
Recover GFIS Clean to PCQ
Install and Run PHP in PQC
Install and run AV
PQC Move Clean to production
Flag as condemned hardware for OS reinstall and notify when ready for BA Client/TDP Restore
Declare OS platforms / versions clear here and prompt for removal from recovery groups
Search for latest Pre RID Time backups and recover to safety
Immediately backup all clearly scanned boxes
406

FIG. 4

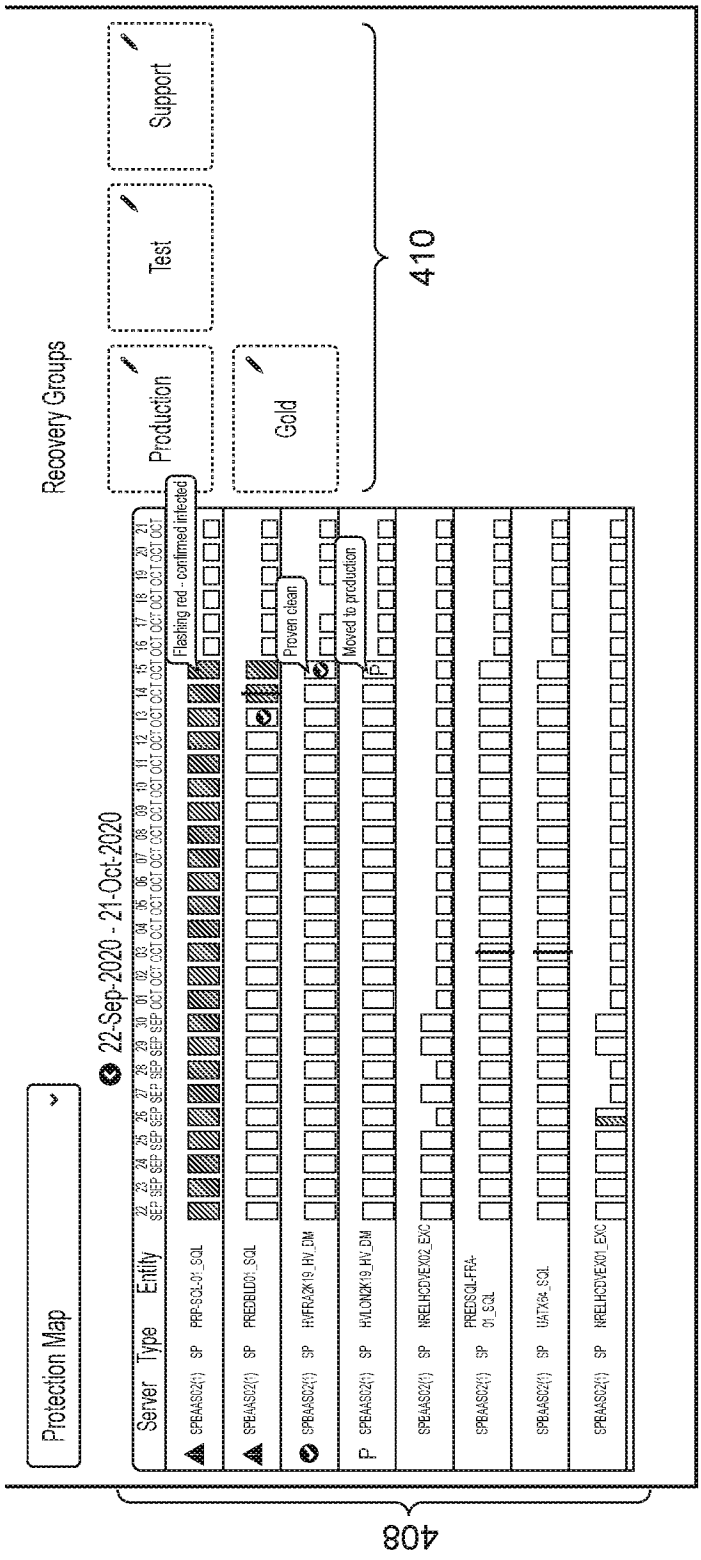


FIG. 4 (Continued)

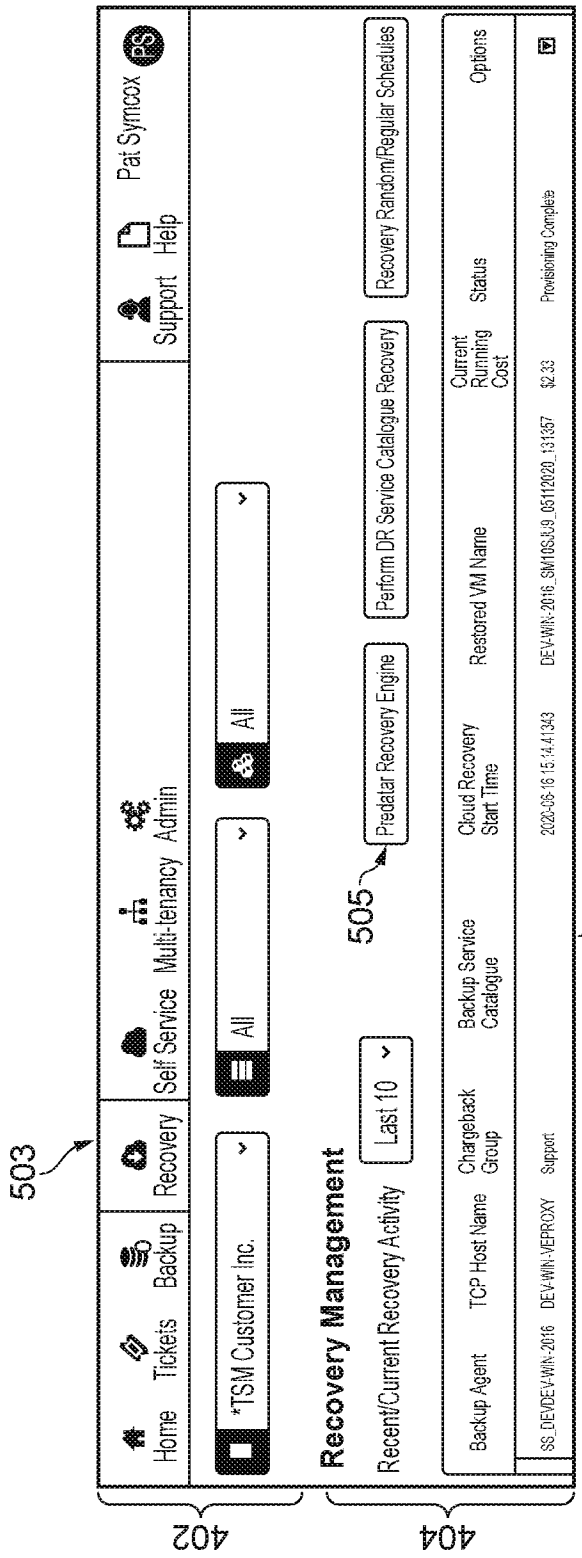
503
Home
Tickets
Backup
Recovery
Self Service
Multi-tenancy
Admin
Support
Help
Pat Symcox
*TSM Customer Inc.
All
All
Recovery Management
Recent/Current Recovery Activity
Last 10
505
Predator Recovery Engine
Perform DR Service Catalogue Recovery
Recovery Random/Regular Schedules
Backup Agent
TCP Host Name
Chargeback Group
Backup Service Catalogue
Cloud Recovery Start Time
Restored VM Name
Current Running Cost
Status
Options
SS_DEVDEV-WIN-2016
DEV-WIN-VEPROXY
Support
2022-06-16 15:14:41343
DEV-WIN-2016_SM10SJ0J9_05112020_131357
$2.33
Provisioning Complete
507
402
404

FIG. 5

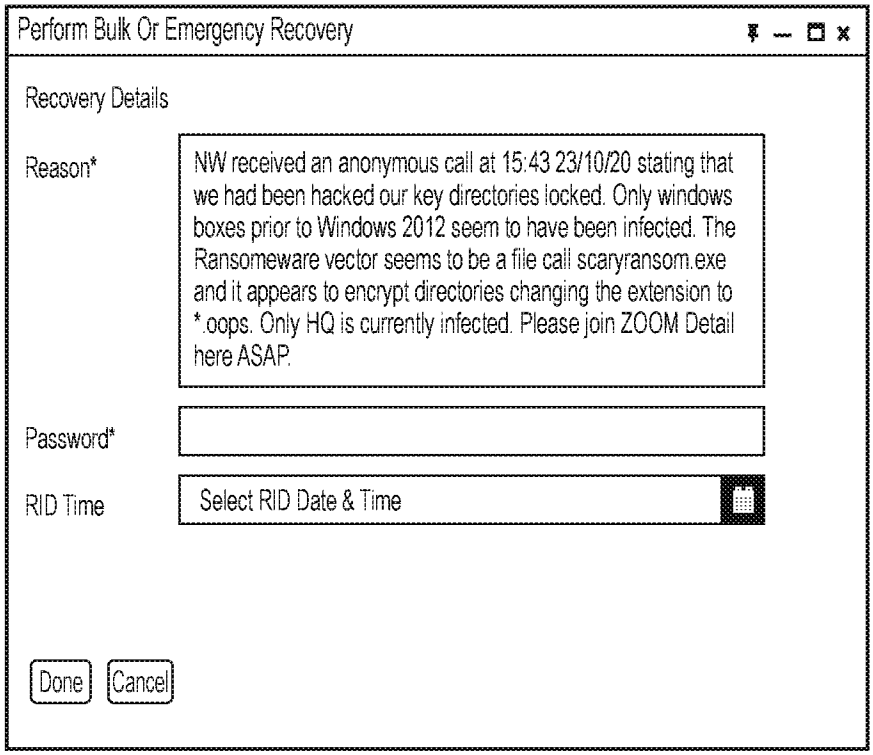
Perform Bulk Or Emergency Recovery
Recovery Details
Reason*
NW received an anonymous call at 15:43 23/10/20 stating that we had been hacked our key directories locked. Only windows boxes prior to Windows 2012 seem to have been infected. The Ransomeware vector seems to be a file call scaryransom.exe and it appears to encrypt directories changing the extension to *.oops. Only HQ is currently infected. Please join ZOOM Detail here ASAP.
Password*
RID Time
Select RID Date & Time
Done
Cancel

FIG. 6A

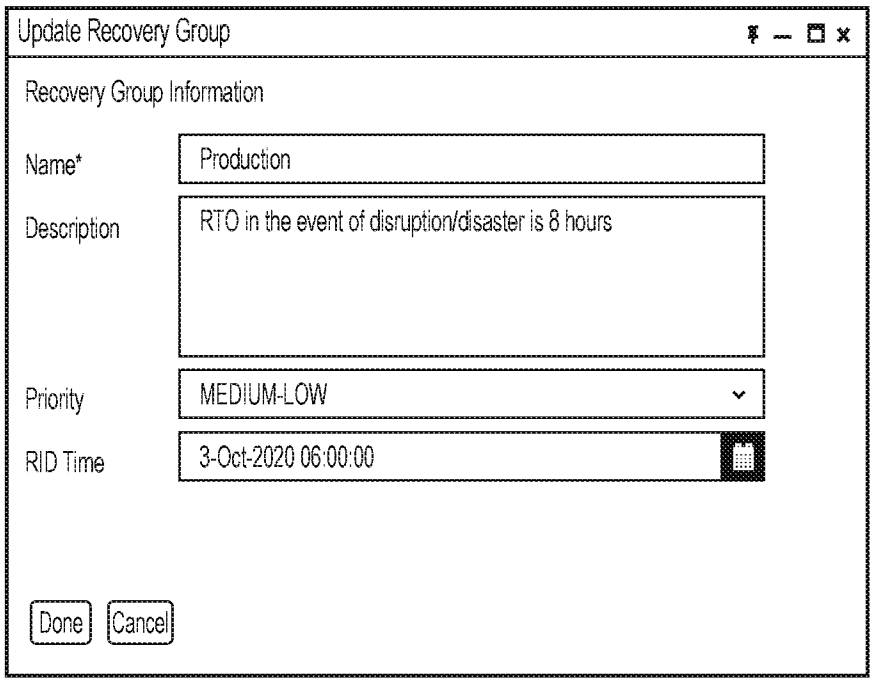
Update Recovery Group
Recovery Group Information
Name*
Production
Description
RTO in the event of disruption/disaster is 8 hours
Priority
MEDIUM-LOW
RID Time
3-Oct-2020 06:00:00
Done
Cancel

FIG. 6B

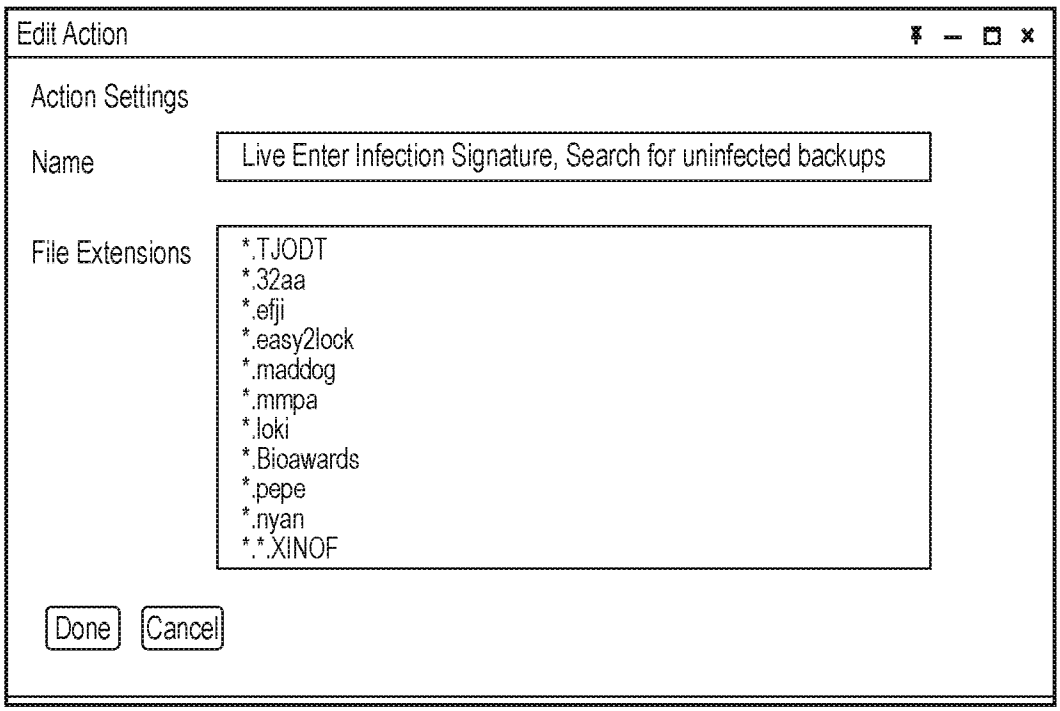
Edit Action
Action Settings
Name
Live Enter Infection Signature, Search for uninfected backups
File Extensions
*.TJODT
*.32aa
*.efji
*.easy2lock
*.maddog
*.mmpa
*.loki
*.Bioawards
*.pepe
*.nyan
*.*.XINOF
Done
Cancel

FIG. 6C

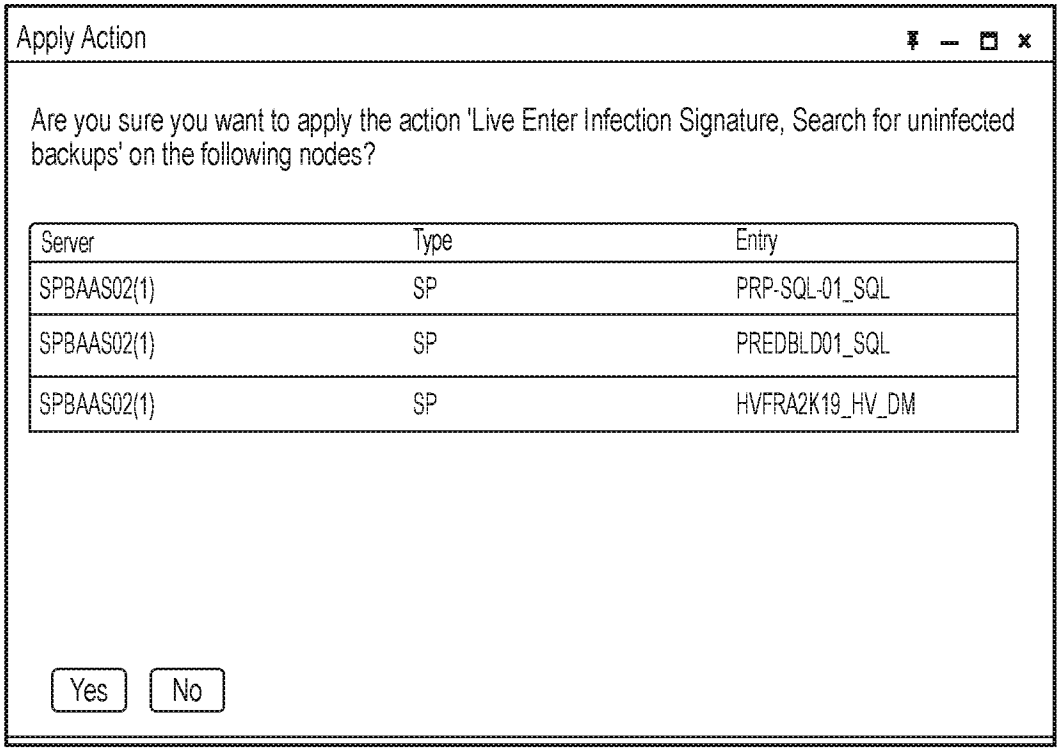
Apply Action
Are you sure you want to apply the action 'Live Enter Infection Signature, Search for uninfected backups' on the following nodes?
Server
Type
Entry
SPBAAS02(1)
SP
PRP-SQL-01_SQL
SPBAAS02(1)
SP
PREDBLD01_SQL
SPBAAS02(1)
SP
HVFRA2K19_HV_DM
Yes
No

FIG. 6D

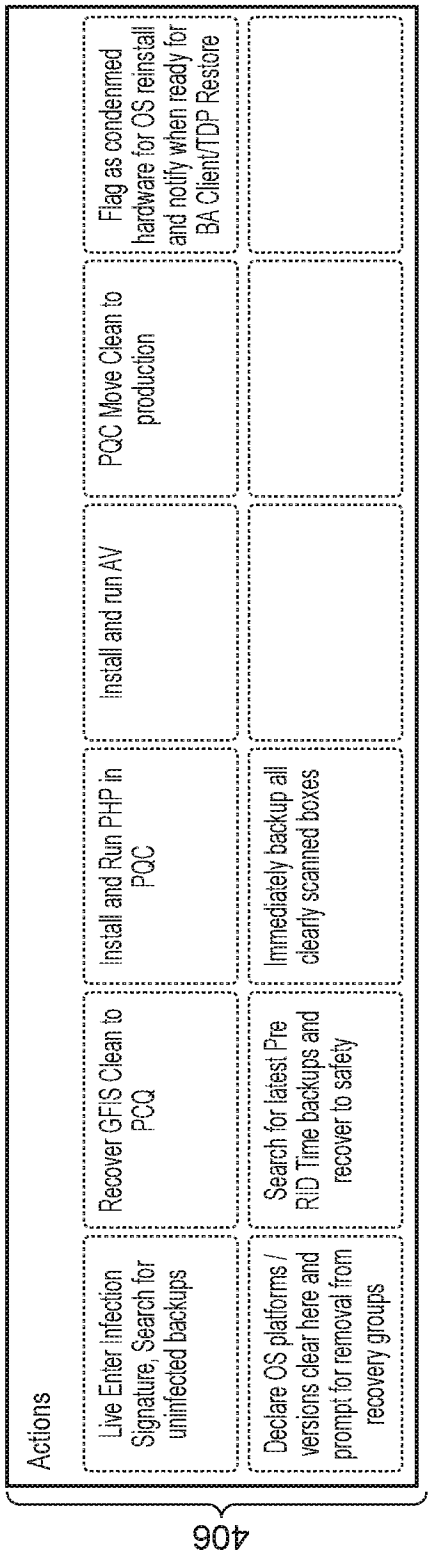
Actions
Live Enter infection Signature, Search for uninfected backups
Recover GFIS Clean to PCQ
Install and Run PHP in PQC
Install and run AV
PQC Move Clean to production
Flag as condemned hardware for OS reinstall and notify when ready for BA Client/TDP Restore
Declare OS platforms / versions clear here and prompt for removal from recovery groups
Search for latest Pre RID Time backups and recover to safety
Immediately backup all clearly scanned boxes
406

FIG. 7A

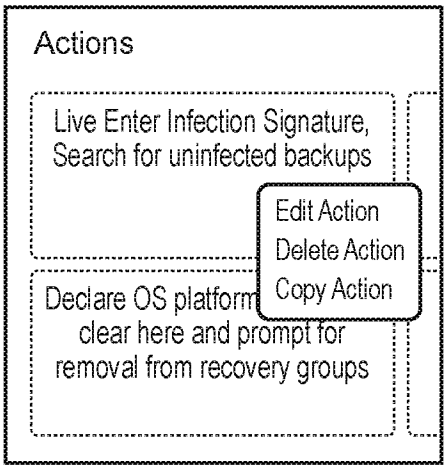


FIG. 7B

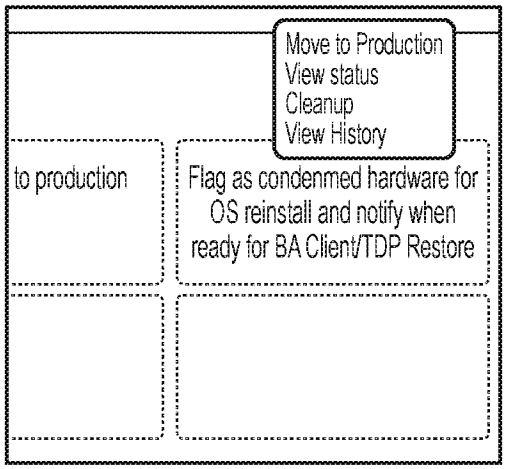


FIG. 7C

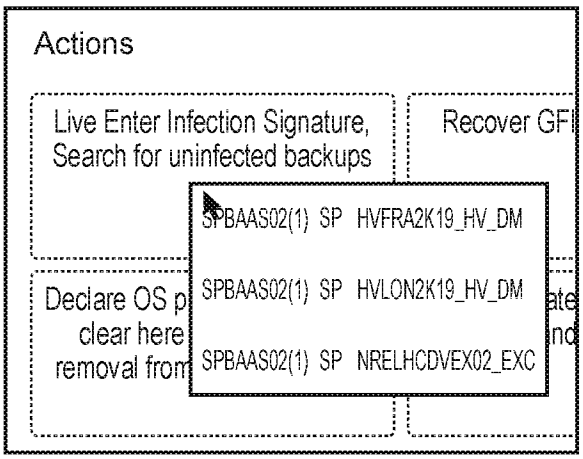


FIG. 7D

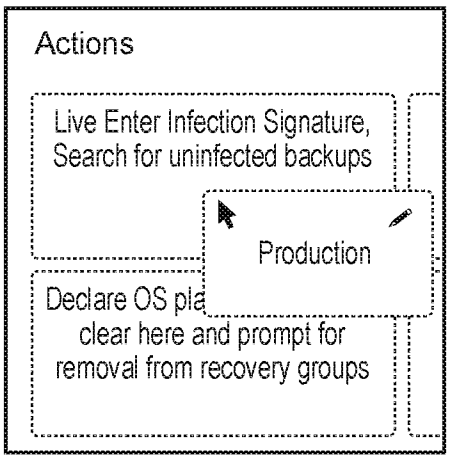


FIG. 7E

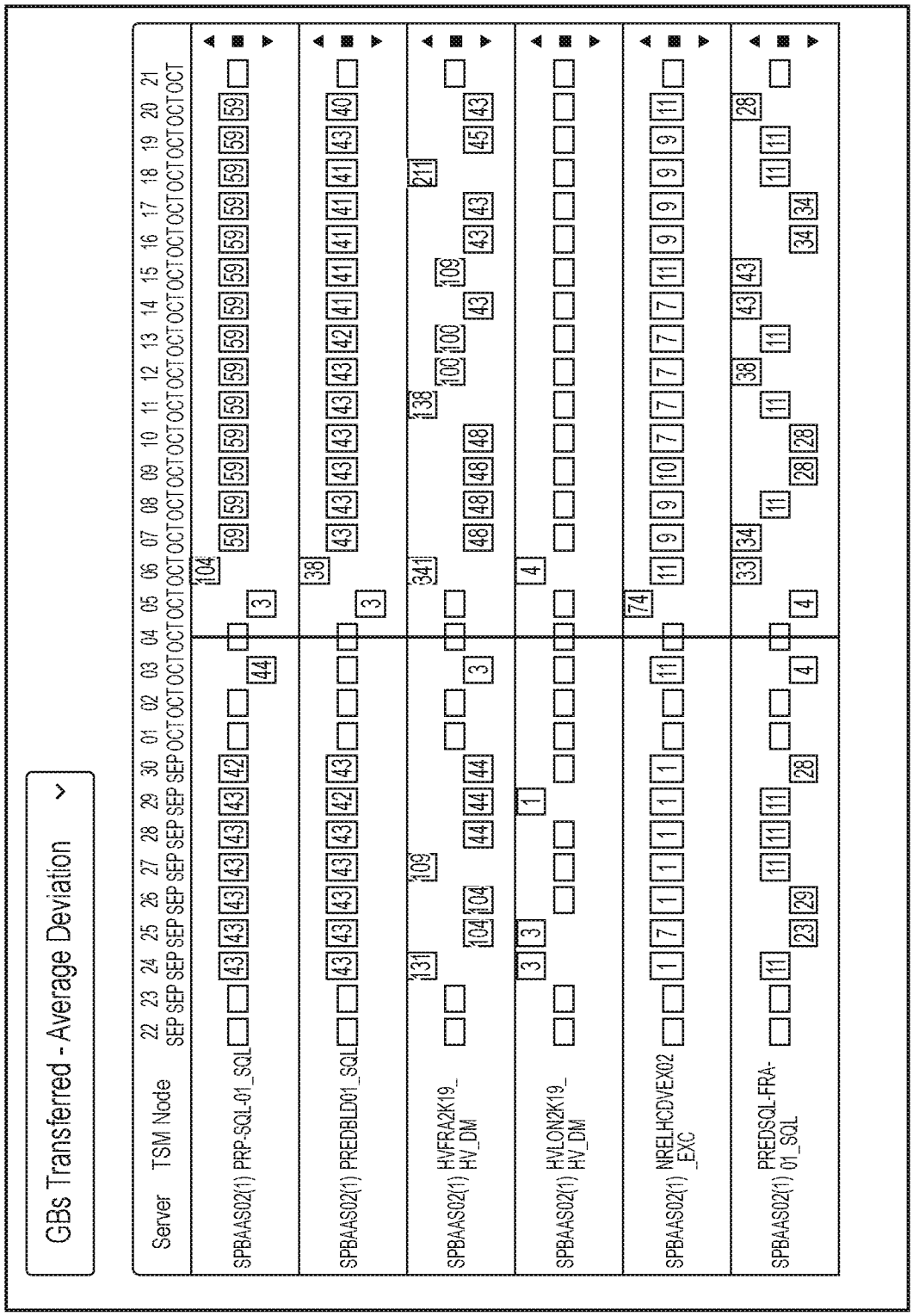
GBs Transferred - Average Deviation
Server
TSM Node
22 23 24 25 26 27 28 29 30 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
SEP SEP SEP SEP SEP SEP SEP SEP SEP OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT
SPBAAS02(1) PRP-SQL-01_SQL
SPBAAS02(1) PREDBLD01_SQL
SPBAAS02(1) HVFRA2K19_HV_DM
SPBAAS02(1) HVLON2K19_HV_DM
SPBAAS02(1) NRELHCDVEX02_EXC
SPBAAS02(1) PREDSQL-FRA-01_SQL

FIG. 8

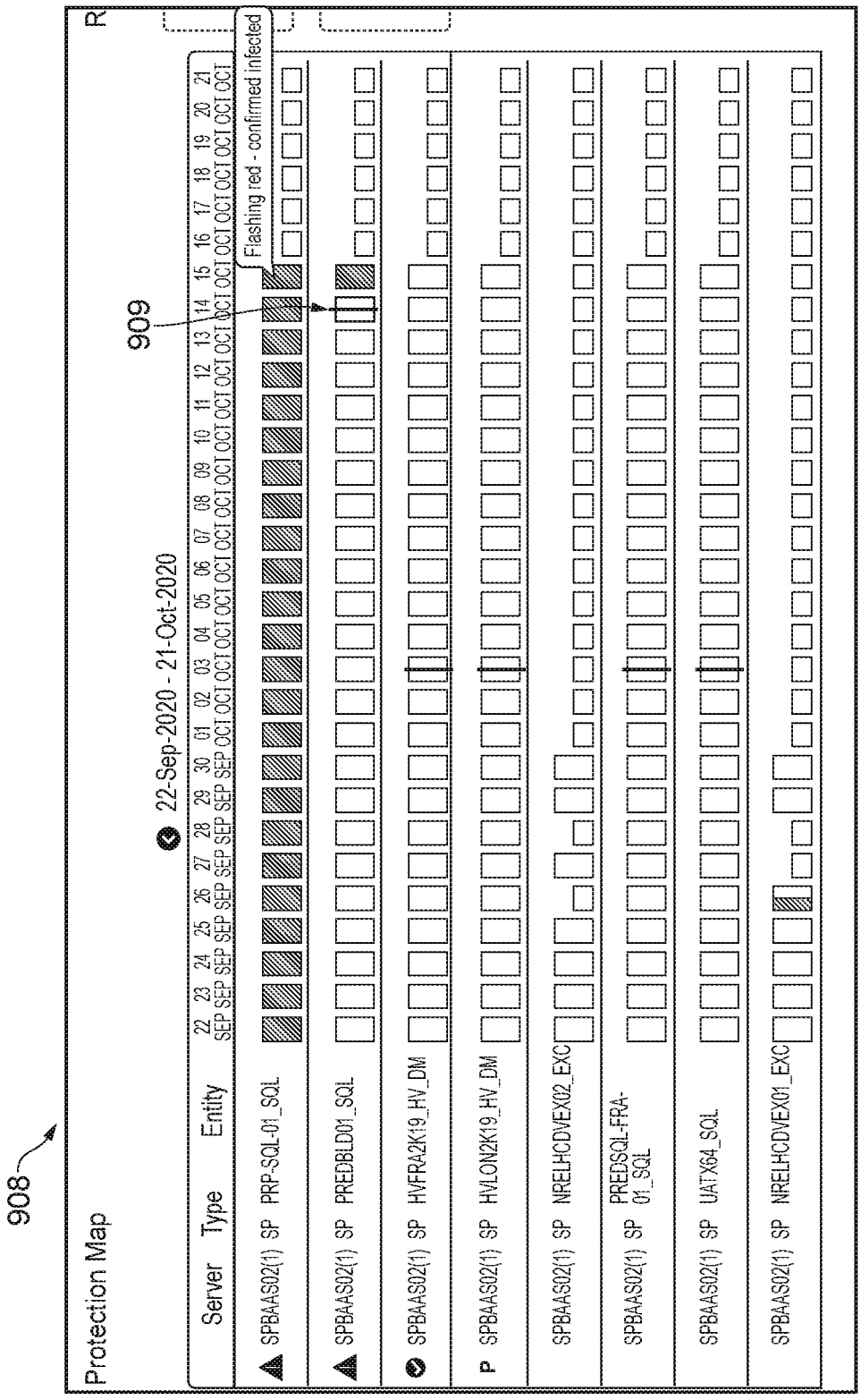
908
Protection Map
R
909
22-Sep-2020 - 21-Oct-2020
Flashing red - confirmed infected
Server Type Entity
22 23 24 25 26 27 28 29 30 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
SEP SEP SEP SEP SEP SEP SEP SEP SEP OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT OCT
SPBAAS02(1) SP PRP-SQL-01_SQL
SPBAAS02(1) SP PREDBLD01_SQL
SPBAAS02(1) SP HVFRA2K19_HV_DM
P SPBAAS02(1) SP HVLON2K19_HV_DM
SPBAAS02(1) SP NRELHCDVEX02_EXC
SPBAAS02(1) SP PREDSQL-FRA-01_SQL
SPBAAS02(1) SP UATX64_SQL
SPBAAS02(1) SP NRELHCDVEX01_EXC

FIG. 9

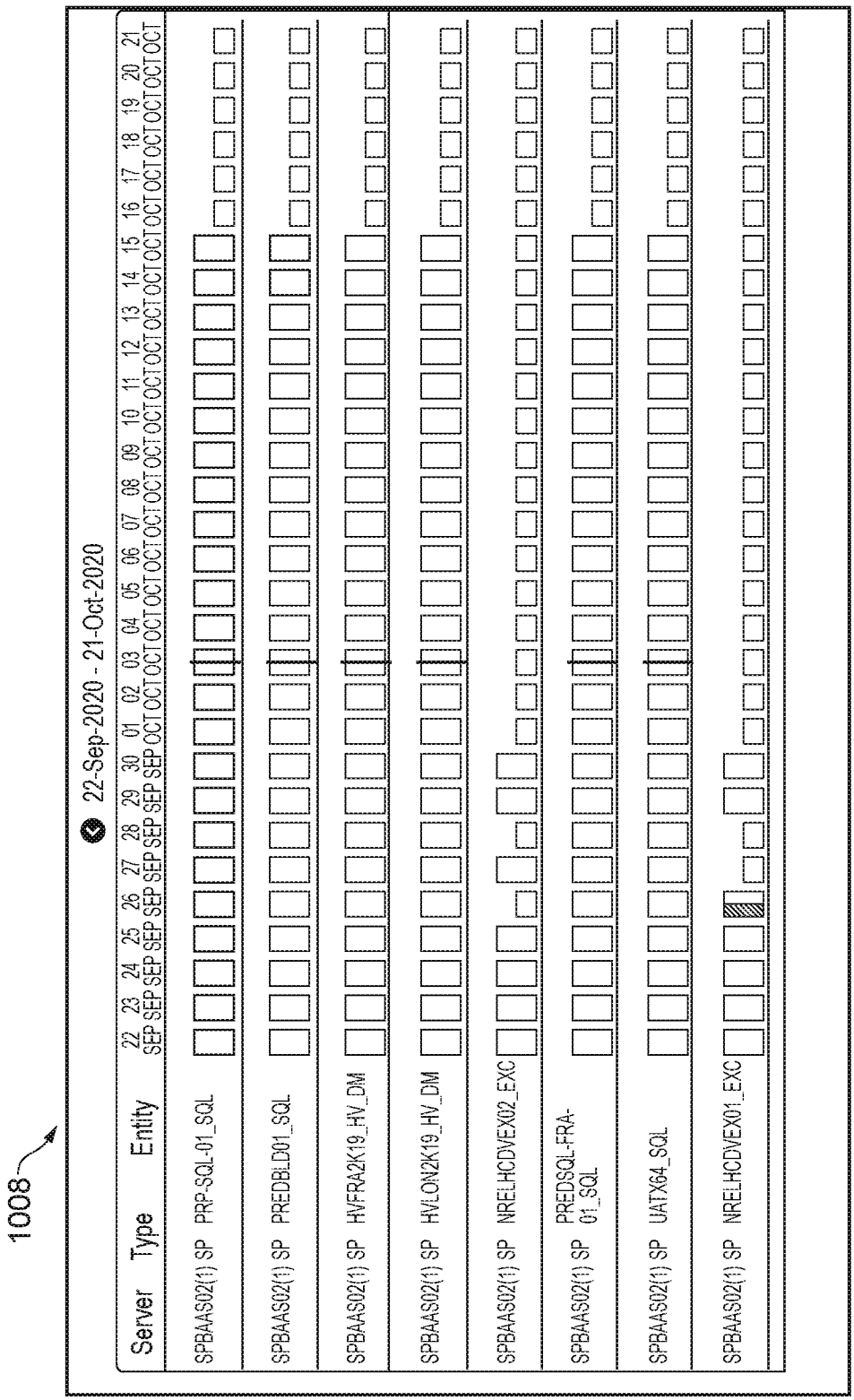
1008
22-Sep-2020 - 21-Oct-2020
Server
Type
Entity
22 SEP
23 SEP
24 SEP
25 SEP
26 SEP
27 SEP
28 SEP
29 SEP
30 SEP
01 OCT
02 OCT
03 OCT
04 OCT
05 OCT
06 OCT
07 OCT
08 OCT
09 OCT
10 OCT
11 OCT
12 OCT
13 OCT
14 OCT
15 OCT
16 OCT
17 OCT
18 OCT
19 OCT
20 OCT
21 OCT
SPBAAS02(1) SP
PRP-SQL-01_SQL
SPBAAS02(1) SP
PREDBLD01_SQL
SPBAAS02(1) SP
HVFRA2K19_HV_DM
SPBAAS02(1) SP
HVLON2K19_HV_DM
SPBAAS02(1) SP
NRELHCDVEX02_EXC
SPBAAS02(1) SP
PREDSQL-FRA-01_SQL
SPBAAS02(1) SP
UATX64_SQL
SPBAAS02(1) SP
NRELHCDVEX01_EXC

FIG. 10

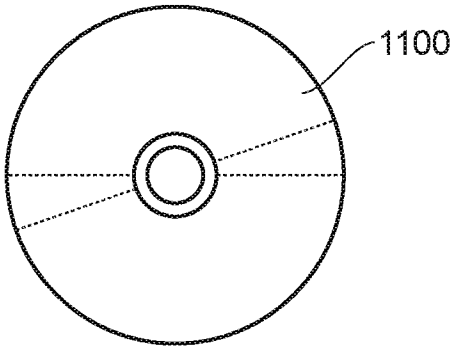


FIG. 11

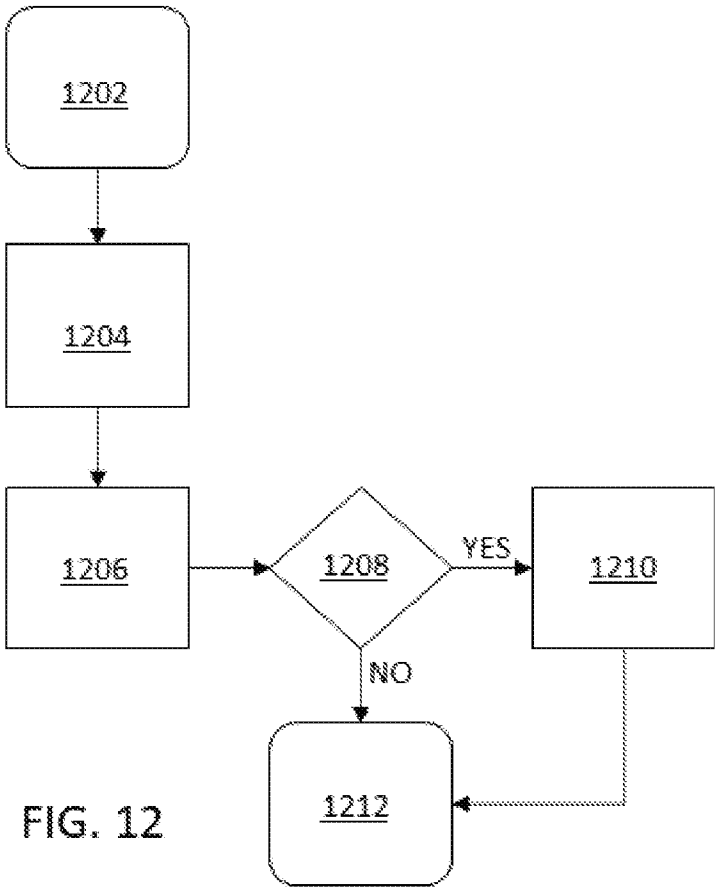


FIG. 12

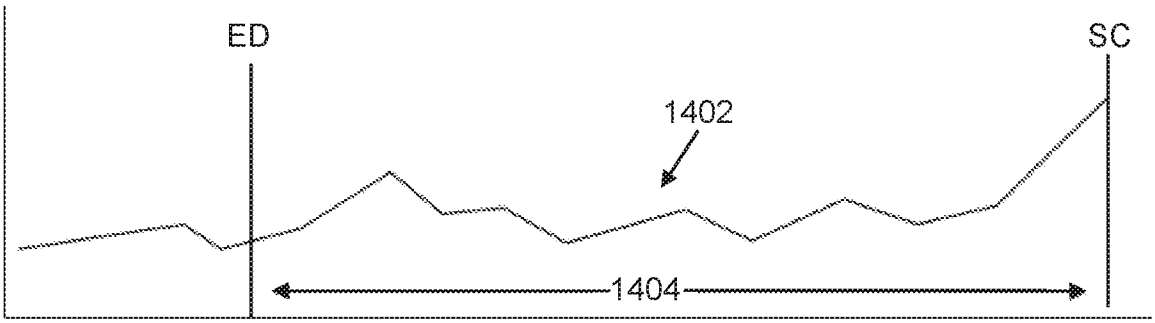


Figure 14

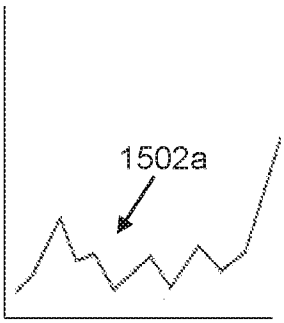


Figure 15a

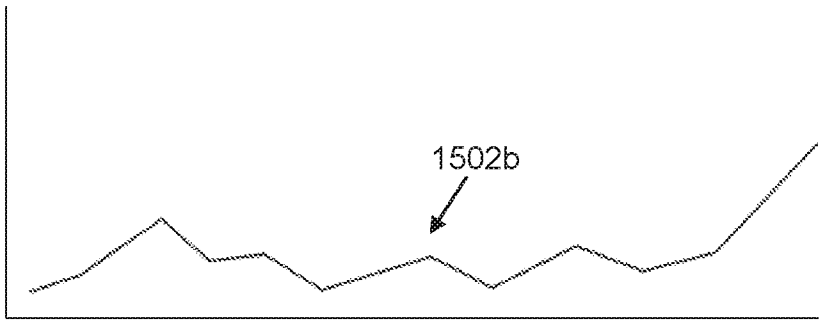


Figure 15b

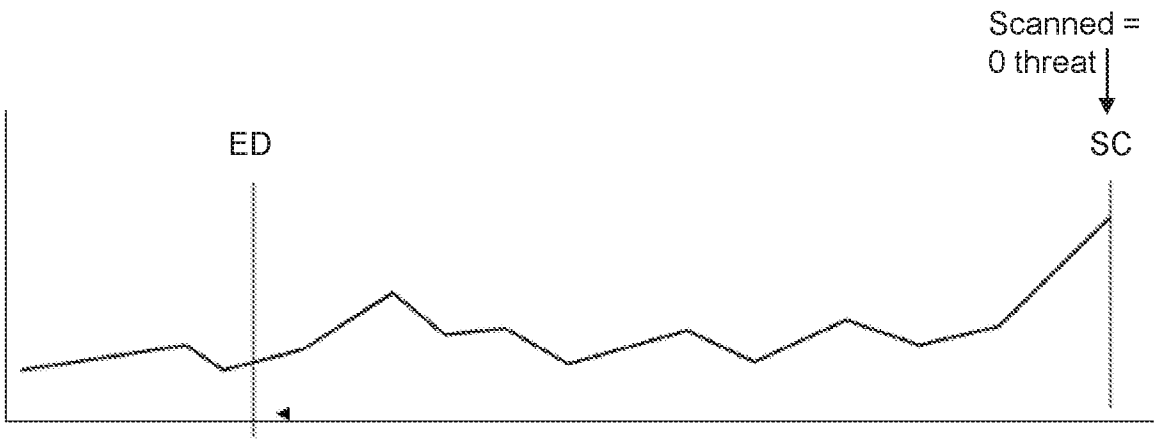


Figure 16a

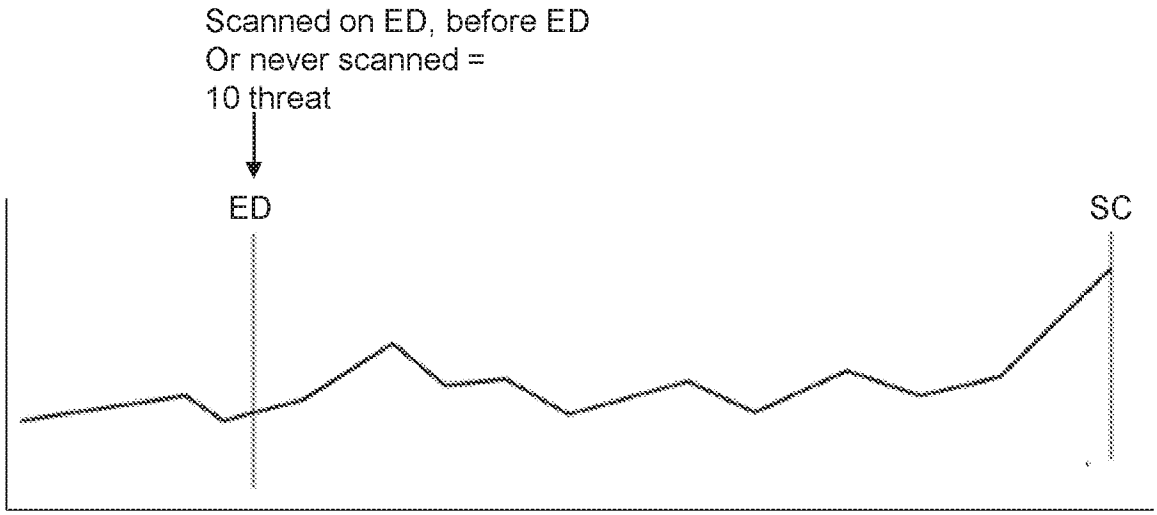


Figure 16b

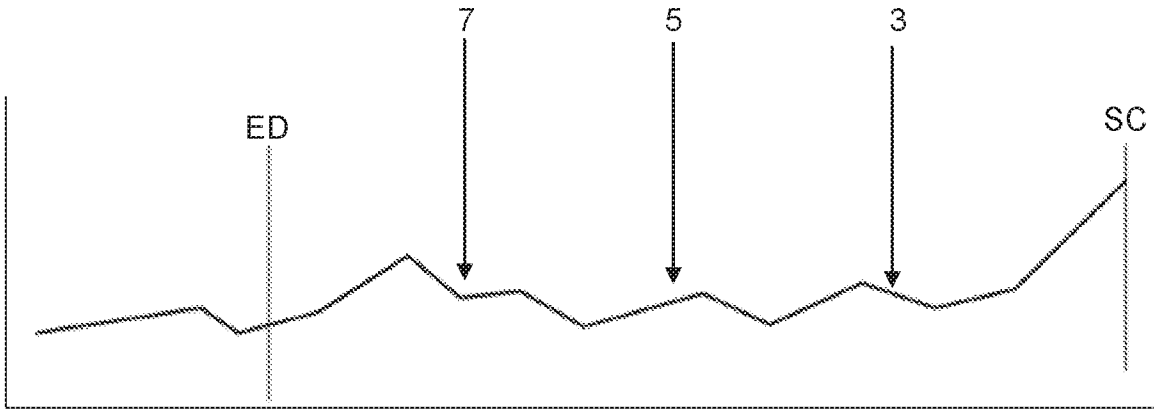


Figure 16c

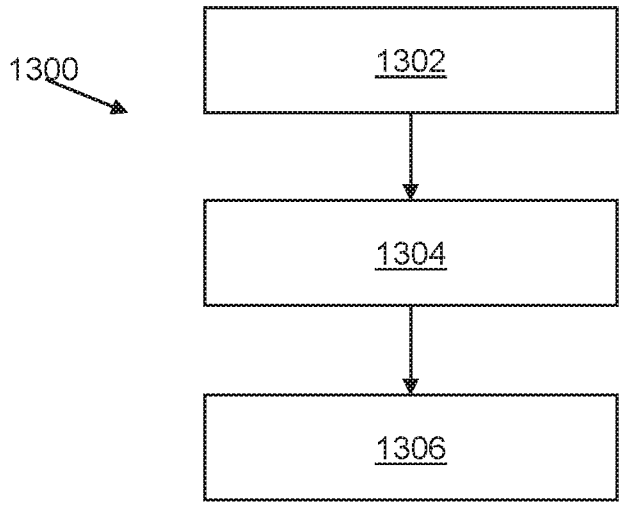


Figure 13

# INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2023/052001

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/56 H04L9/40
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 2019/236274 A1 (BRENNER ADAM [US])<br>1 August 2019 (2019-08-01)<br>paragraphs [0001], [0005], [0019] - [0032], [0039] - [0047]<br>figures 1, 2, 3 | 1-14 |
| X | US 2019/235973 A1 (BREWER KARL EDWARD [US] ET AL) 1 August 2019 (2019-08-01)<br>paragraphs [0004], [0066] - [0077], [0081], [0084] - [0089]<br>figures 6, 7 | 1-14 |
| A,P | GB 2 603 245 A (PREDATAR LTD [GB])<br>3 August 2022 (2022-08-03)<br>page 1, line 4 - page 1, line 7<br>page 5, line 13 - page 6, line 9<br>page 30, line 25 - page 33, line 22<br>claim 23<br>figure 12 | 1-14 |

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 4 October 2023 | 13/10/2023 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Volpato, Gian Luca |



Form PCT/ISA/210 (second sheet) (April 2005)

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2023/052001

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2019236274 A1 | 01-08-2019 | NONE | |
| US 2019235973 A1 | 01-08-2019 | NONE | |
| GB 2603245 A | 03-08-2022 | GB 2603245 A | 03-08-2022 |
| | | US 2022245250 A1 | 04-08-2022 |