Use of criticality principle in information protection systems at machine-building enterprises

K Yu Zhigalov¹, R R Turluyev² and O V Dudareva³

¹V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences
²Chechen State University, 17a, Dudaev Boulevard, Groznyy, 364015, Russia
³Irkutsk National Research Technical University, 83, Lermontov str., Irkutsk, 664074, Russian Federation

E-mail: kshakalov@mail.ru

Abstract. At present, the security of information systems plays a key role not only for large enterprises, but also for individuals. The information presented in the paper considers an innovative approach to the organization of multi-stage protection of information networks based on access matrices created and specified by data on the degree of criticality of protected files against illegal actions of third parties. The authors proposed to use several levels of criticality with three degrees: critical for program running; important for programs; not critical to programs. Besides, an algorithm for determining the criticality level by file extension as universal is proposed as the basic one. Timely response to changes in critical files not only of the operating system, but also of all programs significantly increases the security of enterprise systems. Moreover, the proposed method can be used to minimize the loss of important data.

1. Introduction

The main feature of modern enterprise information systems is that the system components are distributed in space and physical communication between them is carried out both through network connections and software [1, 2].

When users work remotely (as it happens now), remote attacks on the enterprise information infrastructure take the leading position in the number of attempts and the success of their application. In this regard, the control over the infrastructure software at the file level is becoming ever more relevant [3].

The research has shown that companies around the world lose about 6% of their revenues due to incidents related to various ways of information fraud and theft. The number of crimes in this area by country is as follows:
- USA 80%;
- Great Britain 85%;
- Germany 75%;
- France 80%;
- Russian Federation 90%.

The major disadvantages of the current information protection systems include the following:
- visibility of security programs to end users;
- large amounts of stored backup information.

It is known that in this particular case, forewarned is forearmed.
Only backup techniques can protect against accidental or deliberate destruction or modification of files by users with full access to them [4].

However, there are several disadvantages to existing backup methods, including:

- need for large amounts of physical storage to accommodate backup files. As practice shows, the amount of backup storage often exceeds the amount of data used by the company by 2-3 times even in the archived form;
- significant time required to recover data from backups. This is caused by the search for the necessary archive, unpacking the files and forwarding them over the network to the server or workstation on which they were used before their loss;
- no automated file integrity monitoring system. This aspect significantly complicates the search for lost information, since the emergency case can be detected by the user only after some time;
- significant network loading when performing backup procedures from several computers (therefore, this type of work is carried out mainly at night);
- lack of technical capability to verify the correctness of backup procedures.

2. Access matrices

To simultaneously minimize all factors, it is advisable to use the protection systems of a fundamentally different level based on access matrices created on the basis of criticality levels of certain files.

The Access Matrix is a table that displays access control rules [5].

File criticality (in this paper) is the ability of the software, to which the file belongs, to work if it is lost.

For several years Microsoft in its operating system (hereinafter referred to as the OS) based on Windows has been using the criticality criteria of some files in backup methods, which serve to:

- restore the previous OS;
- restore the OS operability.

In the first case, the operating system logs all configuration file changes.

In the second case, Microsoft proposes to rewrite important system files from the original storage (operating system installation).

However, these methods do not affect additional software installed by the user independently and the software not related to the operating system.

According to the studies, not all files installed with programs are critical to their work [6].

Thus, criticality criteria can be generically divided into the following levels:

- critical to the work of the program as a whole. If these files are lost, the program does not start or perform its main functions;
- critical for program operation (usually files of different program modules). If this group of files is lost, the program starts, but does not perform some secondary functions;
- not critical for operation (text files – instructions, backup files for recovery of software performance, files of previous updates, files intended for different configurations, log files). If this group of files is lost, the program can start and perform its main functions.

All software manufacturers create it individually, and therefore there is no single list of file extensions related to the above groups, either directly or indirectly. This fact forces to test each program and each of its files manually and put the results in a table (Table 1). To simplify the process, it is useful to initially group files by extension and scan all files that have this extension at the same time.

3. Criticality check algorithm

The verification algorithm is as follows:

1. Copy the directory containing the program into the archive;
2. Find files with one common extension and delete them;
3. Try to run the program, and, if possible, test the program. If the program does not start, the files fall into the category (1). If the program is running and performs only basic functions, the files fall into
the category (2). If the program is running and performing its functions, the files fall into the category (3):

4. Tabulate the result;
5. Return the deleted files from the archive;
6. Repeat with a different file type.

### Table 1. File extension criticality

| Operating system | Software | Software version | File extension | Criticality |
|------------------|----------|------------------|----------------|------------|
| W                | Avast    |                  | exe            | 2          |
| W                | Office   | 2                |                | 2          |

The program is tested using the black box method. The method is based on the fact that the tests are performed in accordance with software specifications or other documents describing the requirements for the system.

Not all software files with the same extension are equally critical to the program. If the number of files is small, there is a need to check all these files and assign the criticality criterion individually to each file. If the number of files is quite large, there is a need to limit to a quite large sample to determine the percentage of files critical for the program among the “not critical”:

$$C = \frac{\text{Number of critical files}}{\text{Number of checked files}} \times 100\%$$  \hspace{1cm} (1)

where C – coefficient of critical files among non-critical files. If a very small number of critical files were found in the sample, it makes sense to find these files to minimize the archive.

To unify file criteria matching, it is possible to use cloud technologies to increase the criticality table at the expense of future system users.

Upon completion of file criticality tables, a file system access matrix is created based on them (Table 2).

The criticality matrix adds system action options for each group of files and users.

According to the studies, the following possible program operation modes were integrated into each group:

- read lock (in fact, this is blocking the user’s access to these files);
- record lock (the user can read but not write files);
- monitoring the user actions (in this case, the Program can log user actions with these files (deleted/read/edited/copied/started);
- system-safe monitoring (the program logs the user’s actions and creates the following folder on the backup storage server: `\date\user login\`. Thus preserving the file structure from the root, we copy files deleted or modified by the user);
- safe optional monitoring (the program logs the user’s actions and creates the following folder: `\date\user login\`. It saves only deleted or modified files based on the file criticality table).
### Table 2. Access matrix (example)

| General data | Group data | File extension | File route | Software operation mode | Individual keys according to the user’s name or group |
|--------------|------------|----------------|------------|-------------------------|-----------------------------------------------------|
|              |            |                |            | Read lock               | Record lock                                        |
|              |            |                |            | Action monitoring       | System-safe monitoring                              |
|              |            |                |            | safe optional monitoring| Semenov                                             |
|              |            |                |            |                         | Sale Support                                        |
| Software files | OS files   | dll            | 1          | yy                      | xx                                                  |
|              |            | sys            | 1          | yy                      | yy                                                  |
| Software setup files | exe | 1 | yy | 1 | |
|              |            | bat            | 1          | yy                      | yy                                                  |
| Temporary files | imp | - | zz | 1 | 1 |
| System and software login | log | 3 | zz | 1 | 1 |
| Data files | Text documents | doc | z | 3 | 2 |
|              |               | docx           |            |                         |                                                     |
|              |               | odt            |            |                         |                                                     |
|              |               | txt            |            |                         |                                                     |
| DB | lcd | 1 | Uu | 4 | 2 |
|              |               | lgp            | 1          |                         |                                                     |
|              |               | sql            |            |                         |                                                     |
|              |               | mdf            |            |                         |                                                     |
|              |               | rdf            |            |                         |                                                     |
|              |               | jpeg           |            |                         |                                                     |
|              |               | raw            |            |                         |                                                     |
|              |               | bmp            |            |                         |                                                     |
|              |               | gif            |            |                         |                                                     |

| s | 2 | xx | yy | 1 |
4. Conclusion
The positive aspects of the access matrix include the following:

1. It can be used in various kinds of data protection software by differentiating access to them; protects critical data with backup.
2. The matrix makes it possible to choose the necessary level of protection and thereby save resources of the company’s internal network.
3. The file criticality matrix significantly reduces the time to configure the information protection system.

The negative aspects of the access matrix include the following:

1. The need to test all programs manually during the initial phase of technology implementation.
2. A matrix in the hands of an attacker will allow quickly copying important data.

The research has shown that an average from 25% (Windows) to 45% (Linux/Unix) of non-critical files can be contained on a computer.

The proposed approach to information security has a certain potential. This method allows responding to all changes in critical files in a timely manner (deleting a file), which significantly improves security and essentially eliminates the possibility of losing important data due to extremely long period since their change/deletion.

References
[1] Klyuev R V, Bosikov I I, Mayer A V, Gavrina O A 2020 Sustainable Development of Mountain Territories V.12 2(44) 283-290
[2] Bosikov I I, Klyuev R V, Tavasiev V Kh, Gobeev M A 2020 IOP Conference Series: Materials Science and Engineering 918 012223
[3] Klyuev R, Turluev R, Khadzhiev A 2020 International Ural Conference on Electrical Power Engineering (UralCon) INSPEC Accession Number: 20034001 324-329
[4] Bovkun A, Korodyuk I 2019 Analysis of the development of small innovative enterprises in the construction industry. IOP Conf. Ser.: Mater. Sci. Eng. 667
[5] Skorobogatova Y A, Bovkun A S, Ivanov M Yu, Shilova O S 2020 The role and place of small innovative enterprises in the construction industry in the modern economy of single industry towns. IOP Conf. Ser.: Mater. Sci. Eng. 880
[6] Repinskiy O D, Konyukhov V Yu, Bovkun A S, Schupletsov A F 2021 Improving the competitiveness of Russian industry in the production of measuring and analytical equipment. Journal of Physics: Conference Series 1728