Using I-TRIZ for Computer Security Innovation

Ron Fulbright
University of South Carolina Upstate
800 University Way
Spartanburg, SC 29303
(864) 503-5683
rfulbright@uscupstate.edu

ABSTRACT
The last fifteen years has seen a rise in interest in a body of work called TRIZ (pronounced “trees”) comprising a set of tools, methodologies, knowledge bases, and cognitive approaches designed to turn creative and innovative thinking into an algorithmic exercise. Since TRIZ is new to the computer security domain, this paper gives a brief summary of classical TRIZ dating back to 1946 and continues to present day by introducing one of the contemporary derivatives of TRIZ called I-TRIZ. Principal I-TRIZ components are discussed and we show how to use I-TRIZ in two problems involving issues in computer security.

1. INTRODUCTION
TRIZ (pronounced “trees”) is a Russian acronym for “Teoriya Resheniya Izobretatelskikh Zadatch”, the Theory of Inventive Problem Solving; an effort begun by Genrich Altshuller [9] in 1946 and continued through to today by numerous colleagues [1][3]. TRIZ is a broad title representing methodologies, tool sets, knowledge bases, and model-based technologies for generating innovative ideas and solutions. TRIZ aims to create an algorithmic approach to the invention of new systems and the refinement of existing systems and is based on the study of the patterns of evolution of various technological systems, scientific theories, organizations, and works of art. The motivation behind all TRIZ efforts is the belief that universal methods can be developed, based on the discovered patterns of evolution, and when applied to future problems in a systematic way can stimulate new ideas thereby leading to innovative solutions to difficult problems. As [10] states:

The most important result...is that Altshuller set out to develop a method that would help technical individuals handle difficult technological problems. In fact, he accomplished much more than this, revealing the basic patterns and principles of evolution and creativity that are applicable to any field of human activity requiring creative solutions. He also succeeded in systematizing these patterns and principles, making them available for wider use.

There have been three distinct phases in the 60-year history of TRIZ [10]:
- **Classical TRIZ** – development led by Altshuller from the mid-1940s to the mid 1980s.
- **Contemporary TRIZ Phase I** – development during perestroika in the former Soviet Union from the mid-1980s to the early 1990s.
- **Contemporary TRIZ Phase II** – penetration into the Western world from the early 1990s to the present.

Following the fall of the Berlin wall and the collapse of the Soviet Union, in 1989, many TRIZ researchers and practitioners moved to the West, established their own companies, and have either served as consultants and educators. Many have continued to extend TRIZ. Today, TRIZ is a vibrant culture employed by large corporations and individual practitioners alike. A large selection of TRIZ-centric Web sites, workshops, books, tutorials, journals, and seminars are available. Every year, the number of TRIZ devotees grows.

One of the companies formed in the early 1990s is Ideation International, Inc. in Southfield, Michigan. Composed of a number of the original Altshuller team members, Ideation has developed extensions to classical TRIZ and entirely new TRIZ-based methods and tools collectively called I-TRIZ. I-TRIZ has been applied to a large number of application domains including, but not limited to: scientific work [12], quality management [5], and other non-technical areas [4][13], but never to the computer security domain. This paper introduces I-TRIZ to the computer security domain and shows how it can be used to stimulate innovative solutions to computer security problems. Since TRIZ and I-TRIZ are new to the computer security domain, we begin with a brief outline of TRIZ history and a description of a portion of classical TRIZ. We then move into the contemporary era by discussing the major components of I-TRIZ. Finally, we show how to employ I-TRIZ to stimulate creative solutions to two problems in the computer security domain.

2. CLASSICAL TRIZ
The classical TRIZ era started in 1946 with Altshuller’s seminal research on creativity and ranged to the mid-1980s when Altshuller reasoned that development of TRIZ was complete [10]. Altshuller and his colleagues analyzed over 40,000 patents looking for recurring patterns and documenting human innovation. Altshuller realized that technical systems develop according to certain patterns and that these “patterns of evolution” for different systems have much in common. The main accomplishments during the Classical TRIZ era were:
- The Principle of Ideality—that we desire to engineer systems to a perfect state of having only beneficial impact and no undesirable effects—is the goal of any system’s evolution.
• The realization that resolving those conditions keeping systems from being ideal (called contradictions) is vital to the evolution of a system.
• The development of a systematic approach (a methodological approach) to the problem-solving and inventive process.
• Formal problem models and standard solutions.
• An innovation knowledge base featuring case studies illustrating the application of fundamental innovative concepts.
• Methods for reducing psychological inertia.
• 40 Innovation Principles and the Contradiction Table.
• Separation Principles.
• Patterns of Evolution.
• Substance-Field Analysis.

Size constraints prevent this paper from discussing each of these in detail, but books, journals, workshops, and seminars are widely available. [6] and [1] are works well-suited to the beginner. Most newcomers to TRIZ first learn to use one or more of the 40 Innovation Principles [2]. As an example, consider potential solutions the following problem:

Painting a complex part with a paint sprayer does not work because the paint fails to cover 100% of the part.

The challenge is to think of alternative ways to paint the part. A simplistic methodology is to systematically apply each of the 40 principles of classical TRIZ in turn. One principle is segmentation:

<table>
<thead>
<tr>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Divide an object into independent parts.</td>
</tr>
<tr>
<td>• Make an object easy to disassemble.</td>
</tr>
<tr>
<td>• Increase the degree of fragmentation or segmentation.</td>
</tr>
</tbody>
</table>

When one applies segmentation to the painting problem, one immediately thinks of ways to disassemble the object into simpler parts and paint them individually. Combining this idea with another principle, preliminary action:

<table>
<thead>
<tr>
<th>Preliminary Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Perform a change before an object is needed.</td>
</tr>
<tr>
<td>• Pre-arrange objects to come into action in a convenient place without affecting delivery.</td>
</tr>
</tbody>
</table>

leads one to consider painting the constituent parts before they are assembled into the hard-to-paint assembly. Considering another principle called the other way round, which captures the concept of inversion, leads to thinking along the lines of dipping the complex part into the paint rather than spraying the paint on the part. Systematically stepping through each of the 40 principles is likely to trigger several dozen ideas for most problems. The idea behind TRIZ is to stimulate alternative thinking. Often, TRIZ leads to solutions that would never have been conceived by conventional means. As such, TRIZ is also a way to overcome psychological inertia—thinking that artificially constricts our thinking so we do not entertain potential creative solutions.

3. I-TRIZ

The freedom of perestroika in the 1980s permitted, for the first time, the use of classical TRIZ in a commercial way and it was quickly realized that TRIZ was too cumbersome for the mass market. Proficiency required hundreds of hours of training and years of experience. It was also realized that TRIZ did not cover the entire innovative process. In response, TRIZ schools and researchers began modifying TRIZ to enhance its usability and expand its applicability to the entire innovative process. This effort marks the beginning of contemporary phase of TRIZ, begun in the mid 1980s, and which continues to the present [8][11].

Ideation International was formed in the United States in the early 1990s and consists of a group of original TRIZ colleagues. The result of the last fifteen years of work is collectively called I-TRIZ (for Ideation TRIZ) and consists of four modules [7][8][10]:

- **IPS** Inventive Problem Solving
- **AFD** Anticipatory Failure Determination
- **DE** Directed Evolution
- **IP** Control of Intellectual Property

This paper focuses only on using the IPS module. Directed Evolution is used to identify future versions of a product and help manufacturers of the product select one of the future incarnations as the goal of future production efforts. The IP module allows one to protect future inventions from encroachment by competitors. AFD not only helps identify the causes of problems, but also helps to predict critical failure points in a system [7]. Each of these modules is applicable to the computer security domain. However, for brevity, we choose to focus only on the IPS module in this paper.

3.1 Operators

Underlying all of I-TRIZ is a collection of suggestions for incremental changes called operators. By definition, an operator

leads I-TRIZ operators are similar to the classical TRIZ principles. However, where 40 principles are identified in classical TRIZ, over 400 operators have been defined in I-TRIZ. I-TRIZ operators are more detailed in nature and tend to focus thinking toward a specific change. For example, the Segmentation principle in classical TRIZ is replaced by a group of five operators in I-TRIZ.
including four different ways to achieve segmentation (partitioning, pulverization, degeneration, and integration).

An operator is intended to achieve one or more of the following:

- Help overcome psychological inertia
- View the problem in a different way
- Offer a solution containing an already solved problem
- Identify a resource needed to solve a problem
- Suggest an evolutionary step

To help intellectually grasp the operators, they are arranged into three groups, based on the level of universality:

- Universal: applicable to any problem
- General: applicable to many situations
- Specialized: applicable to specific situations

Examples of I-TRIZ operators are:

**Abandon Symmetry**

If an object is symmetrical, try to reduce its weight by making it asymmetrical by excluding a part of the object that does not bear the main load.

**Introduce a Strengthening Element**

To reinforce an object, use an object or material that can temporarily provide reinforcement and then be easily removed.

**Rushing Through**

Consider rushing through a harmful or risky process (or stage) by abruptly increasing the speed at which the process is conducted.

As can be seen, an operator offers a specific suggestion. We believe the set of I-TRIZ operators represents the most comprehensive abstraction of patents (measured in the millions by the 1990s) and technological/scientific history. As a result, the database of operators and associated illustrative examples represents a distillation of human innovative thought and is a knowledge base unparalleled in history.

### 3.2 The Problem Formulator™

One could generate ideas by simply browsing through the operators. However, with such a large number of operators and many inter-relationships, the space of operators is overwhelming. Practitioners need a way to intelligently find a trajectory through the operators that identifies the subset of operators most likely to be beneficial to a problem. Ideation International sells a software tool called the Innovative WorkBench (IWB) that does just this. One tool in the IWB is the Problem Formulator (PF). PF is a graphical modeling technique with a deceptively simple graphical vocabulary. To begin analysis of a system, one uses the PF to construct a cause-effect diagram. Each function or effect of the system is classified as either as harmful (undesirable) or useful (desirable) depending on whether or not the function moves the system toward ideality or away from ideality. Useful and harmful functions are represented as nodes in the diagram following the form:

```
Useful    Harmful
```

Something can always produce or counteract something else, so these relationships are encoded using:

```
Useful -------> Causes
```

Sometimes, something useful in a system causes something else useful to happen—a desirable occurrence. However, sometimes, something useful has undesirable side effects and causes something harmful to happen. (The vice-versa is true too.) These are undesirable occurrences and are called *contradictions*. One can enumerate all outcomes using this modeling technique:

```
Type 1:
A useful function produces another useful function (desirable) but also produces a harmful function (undesirable).

Type 2:
A useful function counteracts a harmful function (desirable) but also produces a harmful function (undesirable).
```

A diagram in the Problem Formulator boils down to a collection of cause/effect relationships describing various situations. Contradictions are undesirable situations and fall into three categories:
A system without contradictions would be an ideal system, since no harmful effects would be present. In reality, though, there is no such thing as an ideal system, so all systems have at least one contradiction. In fact, the reason why one would be analyzing a system in I-TRIZ in the first place would be to solve some problem with the system. In I-TRIZ, the essence of an innovative solution is to resolve as many contradictions as possible.

3.3 Software Tools

Ideation International produces a set of software applications automating much of the operator search and match activity. The tool for the IPS module is called the Innovative WorkBench (IWB) and incorporates a tool called the Innovation Situation Questionnaire (ISQ), the Problem Formulator (PF), the I-TRIZ operators, and the knowledge base consisting of case studies and examples. The ISQ is used to document descriptive information about the system under study. After completing the ISQ, the user draws the PF diagram. From the diagram, the IWB software makes suggestions leading to the most promising set of operators. By considering the suggestions within the operators, one is stimulated to think of alternative solutions improving the system.

Dating back to classical TRIZ, researchers realized there are four ways to resolve contradictions:

- **Separate in space**
  Physically distance the harmful from the useful

- **Separate in time**
  Perform harmful actions at a different time

- **Separate in structure**
  Build the system isolating harm from useful

- **Separate on conditions**
  Alter the conditions under which something harmful acts

Using the IWB, the user selects one or more contradictions shown on the PF diagram. In some cases, the software can make operator suggestions directly. In other cases, the user selects one or more of the contradiction resolution directions to pursue. A systematic IWB analysis will probe all possible avenues.

4. I-TRIZ FOR COMPUTER SECURITY

4.1 Password Security

To illustrate how to use I-TRIZ for problems in the computer security domain, we choose a toy problem whose problem formulation diagram is shown below. This diagram captures the fact that passwords being hard to guess is desirable (a useful function) because they keep intruders out (another useful function). However, a contradiction in this situation is that hard to guess passwords are also hard to remember—an undesirable side effect. In an attempt to make their passwords easier to remember, people use simple passwords. However, simple passwords are easy to guess, so counteract the desirable condition of hard to guess passwords creating yet another contradiction.

It is important to note that this diagram is only a partial diagram. In practice, diagrams of systems typically contain up to 20 elements. However, for space and brevity considerations in this paper, we consider only a portion of the diagram. A continuation of this diagram might include more detail as to why passwords are hard to remember and more detail in the area of how hard to guess passwords keep intruders out.

Even with the simplified diagram, we can arrive at several potential solutions. Below is a list of some of the operators chosen and a brief description of the innovative idea inspired by that operator:

1. **Remove an impeding part from an object**

   Require *multi-part passwords*. The idea is that one part of the password can be easy to remember and a second part can be hard to remember, but be logically connected to the easily remembered part in some way meaningful to the user. The combination is obvious to the user but harder to guess by anyone else.

2. **Use another dimension**

   Require *multi-dimensional passwords*. One idea is to use passwords that have features similar to crossword or Sudoku puzzles. Two easy to remember words that intersect (or relate to each other in some other way) greatly increases the number of combinations thereby making it harder to guess.

3. **Use the reverse side**

   Use *passwords in reverse*. Most password-guessing hackers assume we spell our words normally. Using an easy to remember word, but spelling it in reverse makes it very hard to guess (unless the hacker knows that we use reversed passwords).

4. **Nesting**

   Require *nested passwords*. In some ways, this is similar to using another dimension, but the idea here is to insert one
easy to remember word into another easy to remember word via a certain rule (such as after the first three letters). The combination of the nested words is easy to re-construct by the user, but harder to guess by a hacker.

5. Create and Use Pauses

Use the cadence of a user’s typing. The idea is to key off of how a user enters the password rather than, or in addition to, what password is entered. For example, the word password could be entered as pass <pause> word. The time between the last s and the w would be the key to the password. Several pauses could be encoded and this has the advantage that it could be implemented in software applications without requiring hardware assistance.

6. Use Interchangeable Elements

Accept any of a set of passwords. Many people have memorized multiple username/password combinations they use for various systems they interact with on a routine basis. Many have no problem remembering the passwords, but often have trouble remembering which password goes with which username and which username goes with which system. Easier would be for systems to accept multiple username/password combinations. This way, if a user inadvertently enters a valid password for a different system, they are not delayed. One can imagine combining this idea with earlier ideas and require, for example, any of a set of words, followed by any of a set of numbers (possibly even separated by a pause of certain duration).

7. Use Elements with Dynamic Features

Use morphing passwords. The idea is to have the password change each time it is used. The way in which it changes can be specified by the user. A password-guessing hacker would have to know the root, the change mechanism, and the number of morphs undergone so far. This is in some ways similar to encryption ciphers.

Several other operators suggested by the IWB software stimulated ideas including: biometrics, voice prints, fingerprints, thermal signatures, etc. However, these are already-known solutions. We chose to focus on the more innovative solutions generated. Given a full analysis, we would expect more than 100 innovative ideas like those listed above would be generated.

4.2 Physical Security

Protecting computer assets from theft is a major problem facing any organization. Recently, flat-panel LCD monitors have become widely used. These monitors weigh a fraction of their CRT-based predecessors and the thin form factor makes them easy to conceal. Not surprisingly, theft of computer monitors has been on the increase. To explore potential solutions to this problem, we first draw a cause-effect diagram in the Problem Formulator, a portion of which is shown below.

This diagram captures the fact that monitors are flat and light-weight is a desirable feature and one making monitors popular, another desirable feature. However, these desirables lead to undesirables thereby causing the contradictions to be solved. The size and weight of the monitors makes them easy to steal. Contributing to this is the fact that monitors are placed in un-staffed computer labs. The popularity of such monitors, together with the fact that monitors cost a lot of money, and that monitors are easy to steal make monitors valuable targets for theft.

Again, this is only a portion of the diagram as it should be. But, as it is, the diagram shows three direct causes and two indirect causes of the undesirable “monitors are valuable targets.” The task is either to identify ways to alter the conditions contributing to this undesirable or to counteract the undesirable directly. Considering “monitors cost a lot,” if the monitors were to cost very little, like paperclips, people would not be motivated to steal them. Even if they did, the organization would not care about such a small material loss. However, lowering the cost of the monitors is not under our control as purchasers and users of the monitors. Considering “flat monitors are popular,” if the monitors were outdated technology, or if everyone had as many monitors as they wanted, the monitors would not be as popular. Again, this would remove any motivation to steal them but, unfortunately, this is also not under our control. If one tries hard enough, it may be possible to think of some action to achieve all or a portion of these goals in some situations. For example, one can imagine a school or an employer giving all students or employees a flat-panel monitor thereby reducing the motivation to steal them.

Counteracting “monitors are easy to steal” leads to fruitful anti-theft lines of thought; however, these appear to be traditional measures, many of which are used today. Certainly, one readily thinks of adding surveillance cameras, alarms, and staffing the computer labs to counteract “labs are unstaffed.” Bolting or cabling the monitors to the desk or CPU case, counteracts “monitors are flat and light-weight” (adding the weight of the desk to the monitors).

These are all traditional anti-theft measures and one does not need TRIZ or I-TRIZ to come up with them. For this paper, we focused on ways to counteract “monitors are valuable targets” directly. The key idea here is to find ways to reduce the value of stealing the monitor. If there is no benefit gained by the thieves, they will not steal the monitors in the first place. Browsing the suggested operators, one realizes a reason the monitors are valuable is their universality and another reason is that they are interchangeable. A thief can take a stolen monitor home and plug it into his or her home computer and be assured that it will work. If we could stop that from happening, the thief would have no benefit from stealing the monitor. With this line of thought in mind, we stepped through several of the suggested operators. Some of the ideas we came up with included:
• Alter the monitors so they work only with the lab computers. One way to accomplish this is to replace the standard VGA connector with a proprietary connector.
• Remove a key element from the monitor’s housing, for example, the power supply electronics or a portion of the control electronics, and place that portion inside the computer case or fix it to the desk. This way, if a thief steals the monitor, it will not function without the other piece. The thief will have to steal both the missing electronics and the monitor. It is much easier to secure a smaller unit than it is the entire monitor.
• Include an active component in the monitor that when triggered, renders the monitor non-functional. This can be something as simple as a programmable switch or even something more destructive such as electromagnetic pulse circuit that destroys key electrical elements inside the monitor. Some action in the theft could be made to trigger the sabotage device.
• Password-protect the monitor upon power-up. When the monitor is powered off for a certain period of time, a programmable password would have to be entered (maybe through DIP switches or a coded push-button sequence) to unlock the monitor.
• Use a magnetic inductive switch (as is used on some automobile keys). The magnetic key could be located inside the computer or fixed on the desk, or even built into the desk itself. The monitor would work only when in proximity to the magnetic key.

All of these ideas serve to reduce the value of the stolen monitor. Instead of attempting to prevent the theft of the monitor, we reduce the benefit of stealing the monitor by crippling its most useful functions.

5. CONCLUSION
We have reviewed the history of TRIZ, a 60-year effort to give us the ability to systematically innovate. Although many details of classical TRIZ are not mentioned in this paper, we have shown the connection to contemporary TRIZ-based efforts, most notably the I-TRIZ work done by Ideation International. We introduced the four modules comprising I-TRIZ and focused on one of them, IPS. We have presented some of the tools within IPS and the associated software tool, the IWB. These tools include the ISQ, and the Problem Formulator, a powerful, but very compact cause/effect graphical modeling tool. We have also shown how, using the IWB, one can leverage the I-TRIZ operators and the knowledge base to come up with new ideas in problems relating to computer security.

6. REFERENCES


