

ASSESSMENT OF THE IMPACT OF THE YEAR 2000 PROBLEM ON CRITICAL INFRASTRUCTURES

Report to the
President's Commission
on Critical Infrastructure Protection
1997



This report was prepared for the President's Commission on Critical Infrastructure Protection, and informed its deliberations and recommendations. The report represents the opinions and conclusions solely of its developer, the Technical Resource Center, MITRETEK Systems.

CONTENTS

EXECUTIVE SUMMARY	1
1 INTRODUCTION.....	4
2 RELIANCE OF CRITICAL INFRASTRUCTURES ON INFORMATION TECHNOLOGY.....	5
3 NATURE OF THE YEAR 2000 PROBLEM.....	8
3.1 Assessment.....	10
3.2 Renovation	12
3.2.1 Date Expansion	13
3.2.2 Date Encoding.....	13
3.2.3 Windowing.....	14
3.2.4 Encapsulation	14
3.2.5 Integer Representation.....	15
3.2.6 Commercial Products	15
3.2.7 Hardware or Firmware	15
3.2.8 Shared Data	16
3.3 Validation.....	16
3.4 Implementation.....	18
3.5 Available Assistance	19
3.5.1 Service Providers.....	19
3.5.2 Tool Vendors.....	19
3.5.3 Strategists	20

4	YEAR 2000 VULNERABILITIES	20
4.1	Limited Control over Essential Systems and Data once Access is Granted.....	21
4.2	The Risk that Repaired Systems will Fail as a Result of their Independence	23
5	YEAR 2000 THREAT	26
5.1	Effect on Physical Threats.....	26
5.2	Effect on Cyber Threats.....	26
5.3	Effect on Information Gathering Threats	27
6	TECHNOLOGIES AND APPROACHES FOR PROTECTING CRITICAL INFRASTRUCTURES	28
6.1	Software Maintenance Practices	28
6.2	Enforcement of the Legal Protections for Software and Data.....	29
6.3	Information Security Technology.....	29
6.4	National Priorities and Contingency Planning	29
6.5	Emergency Response Preparation	30
6.6	Focus on Safety-Critical and Property-Critical Systems	30
7	CONCLUSION	31

Executive Summary

As a result of the Commission's investigation into cyber threats, its attention has been drawn to the "Year 2000" problem and its potential to create opportunities to undermine our nation's critical infrastructures. Entirely new threats are unlikely to arise because of the Year 2000 problem; the list of adversaries remains the same, but the situation opens opportunities for the existing threats to adopt new guises.

Although the cause of the Year 2000 problem is arguably simple, the necessary repairs can be quite complex. Unfortunately, dates are used throughout modern information processing, so every system must be examined for impact, and possibly repaired. For many organizations, the Year 2000 issue will be the largest, most intensive information technology project they have ever undertaken. Often, this results in hiring outside help, in a manner similar to the one described in the Wall Street Journal in 1996, *"For much of the work, Consolidated Edison is turning to outside contractors, who, in turn, are using subcontractors in Ireland and India, all connected via satellite and high-speed phone links directly to Con Ed computers."*

As companies work through the complexities of: (1) assessment, (2) renovation, (3) validation, and (4) implementation of corrected systems throughout their operations, many of them will have to obtain assistance in making massive changes to their essential systems. Across the critical infrastructures, leading organizations have recognized the magnitude of this problem and devoted talent and resources to its resolution. The Year 2000, due to the scale of the problem, has spawned an entire sub-industry of service providers, tool vendors, and strategists devoted to giving organizations the assistance they need to address the problem.

The Commission is concerned that the very efforts to fix the Year 2000 problem could create critical vulnerabilities within this nation's infrastructures. Our principal concerns are that infrastructures have limited control over essential systems and data once they grant access and that there is a risk that repaired systems will fail as a result of their interdependence.

Two characteristics that protect the essential systems of our critical infrastructures now are their complexity and the lack of available information regarding their detailed structure. Successive waves of investment in information processing and automation have left each organization with a unique and highly interconnected collection of systems and applications. Adversaries are hampered by their lack of knowledge of the target systems. The natural redundancy of systems helps to mitigate the effects of any attacks that are made. When critical infrastructure organizations seek help in converting their data archives and their business processing and decision systems, they will release a wealth of sensitive information to those they hire to assist. This information will then become much more easily available as it is passed to service providers or tool vendors for repair.

With extensive access to system details, an adversary could design a subtle or comprehensive attack either to gather information or reduce system effectiveness. At the

same time, they could perform analysis to assure that redundancy does not mitigate the effects and that the attack cannot be easily detected. Some attacks could be almost impossible to detect, or at least indistinguishable from the expected impacts of a Year 2000 renovation. For instance, simply increasing the time required for date calculations (a possible outcome of Year 2000 renovations) could make a vital billing or customer service system too slow to use.

The need for interoperability of systems among business partners, along with the international synchronicity of the Year 2000 problem, also increases vulnerability. Organizations that manage to get their own systems in order still run the risk that data exchanged with business partners' and customers may not be compliant to their own. As Andrew Hove, Acting Chairman of the Federal Deposit Insurance Corporation, testified to the Senate Banking Committee on July 30, 1997:

“Year 2000 risks to financial institutions are not limited to their internal systems. Even financial institutions that have taken a proactive approach in addressing Year 2000 problems internally, nevertheless, may encounter difficulties if parties external to the bank with whom they exchange data electronically are not prepared for the century date change.”

This situation is serious enough to warrant action beyond a direct response to the original Year 2000 problem in systems. With over two years to go until the Year 2000, time still remains for Government and non-government action to reduce the impacts of these problems on our critical infrastructures. Protecting them against Year 2000 vulnerabilities begins with good system security practices to gain control over sensitive data and systems and the personnel who are given access to them. In addition to heightened security, improvements in quality assurance, verification, validation, and configuration management will help to build confidence that the renovations made to critical infrastructure systems will be reliable.

Technology is available that can make these efforts more effective. Digital signature technology can assure customers that renovated software and compliant product versions actually originated where they say they did. Encryption can assure that only the intended recipient of software can read it. Firewalls and access control systems can help to limit online access to authorized individuals. In addition vigorous enforcement of legal protections, for trade secrets or sensitive business information included in software and data, can establish a safer environment in which critical infrastructure companies can seek help with their renovations.

Finally, the Year 2000 problem is large enough and widespread enough that there are credible risks that some essential systems will fail, whether or not they are attacked. A national process is needed to identify the most critical systems in each community (i.e., safety, environmental and vital human services), presumably including many in the critical infrastructures but not limited to these, and to develop or update contingency plans to assure that failures, when they occur, will be as orderly and confined as possible. To succeed, we will have to take best advantage of all of the techniques, tools, and personnel available, minimizing exposure to new vulnerabilities and risks to the extent

that we can. We know many of the solutions, but it will still be a large undertaking to apply them across the vast range of systems on which the critical infrastructures depend. The risks to the critical infrastructures that result from the Year 2000 problem are significant and challenging. With leadership and a great deal of hard work, the critical infrastructures can meet this challenge, arriving in the new century better prepared to meet the challenges that await us there.

1.0 Introduction

The President's Commission on Critical Infrastructure Protection is charged with recommending a comprehensive national policy and an implementation strategy for protecting the critical infrastructures, which include the "framework of networks and systems comprising identifiable industries, institutions, and distribution capabilities that provide a continuous flow of goods and services" in telecommunications; electrical power; gas and oil storage and transportation; banking and finance; transportation; water supply; emergency services; and government services. The Commission has determined that threats to these infrastructures could include both physical threats and 'cyber threats' (electronic, radio-frequency, or computer-based attacks against the information processing capabilities within the infrastructure).

In investigating cyber threats, the Commission's attention has been drawn to the "Year 2000" problem. This is a name given to the likelihood that many automated and information systems will produce incorrect results when processing dates beyond 1999. The Year 2000 problem is a consequence of the manner in which these systems have been designed and programmed and is inherent in the systems themselves; malfunctions of affected systems will occur even if there is no overt attack by any malicious agent. As such, the Year 2000 problem is atypical of the issues addressed by the Commission. Nevertheless, because it is of such current concern and because it does create additional vulnerabilities in and opportunities to undermine our nation's critical infrastructures, it is an appropriate topic for Commission attention.

This paper outlines the risks of the Year 2000 problem, demonstrating that this problem constitutes a serious risk to the vital and economic health of the critical infrastructures and focusing particularly on vulnerabilities created by this situation and on special threats that may attempt to take advantage of it to attack the critical infrastructures. In Section 2, the critical infrastructures are shown to rely heavily on information technology, to an even greater degree than is found in most modern businesses. Following this, Section 3 gives a brief discussion of the nature of the Year 2000 problem, methods for repairing an affected system, and the general approaches that companies must take in making these repairs. Then, Section 4 begins to address the particular interest of the Commission, describing vulnerabilities of the critical infrastructures that are created or exacerbated by the Year 2000 problem. Section 5 discusses threats that may gain special advantage through the vulnerabilities identified and special methods of attack that threats may use to exploit Year 2000 vulnerabilities. Finally, countermeasures that may be taken by the critical infrastructures to reduce their Year 2000 vulnerabilities or hinder attempts to exploit them are identified in Section 6.

2.0 Reliance of Critical Infrastructures on Information Technology

The Commission's decision to focus special attention on cyber threats recognizes the great degree of reliance of modern business, including the major components of the critical infrastructures, on information technology. If anything, the critical infrastructures are even more reliant on information technology than most businesses, because they rely on automation in a primary role in the provision of their product or service, as well as on information systems for business applications like record keeping and billing. Not only are the critical infrastructures dependent on information technologies at many levels within their own business, but their systems have many interdependencies with each other, with customers, or with suppliers. Critical infrastructure businesses have gone as far as anyone toward establishing interfaces with their business partners and customers.

This reliance on automation in the provision of their product or service extends to virtually all of the critical infrastructures. In telecommunications, automated switches connect local and long distance calls while automated routers manage the transmission of data. Automated systems are necessary to manage and control satellites for telephone, television, and high-bandwidth communications. The satellites themselves are highly automated. In banking and finance, automated funds transfers of various kinds move trillions of dollars a day, both nationally and internationally. Important trading exchanges conduct their business via automated transactions. Payments by credit card or ATM card are a mainstay of the economy.

Modern electrical power plants are controlled by a mix of automated and manual systems, as are the switching and distribution systems that deliver electric power to users. In recent years, the Nuclear Regulatory Commission and others have given a great deal of attention toward assuring that these control systems remain safe while they place increasing reliance on automation in such important functions as monitoring employee exposure to radiation, security control, and accumulated burn-up programs.

Transportation of all kinds relies on automated systems, often for safety-critical functions. Air traffic controllers "see" the traffic around an airport through a combination of automated radars and other sensors whose data is channeled and presented through specialized computer workstations. Railroads use automated systems for signaling and for traffic control. Even modern streets and highways are often controlled through a network of automated lights and signals. Transportation of oil and gas depends on many of these common systems of transportation, as well as on its own pipelines, which are also managed and controlled by automated systems. Water filtering and disinfecting processes are monitored and often controlled by automated systems.

Emergency services of all kinds depend heavily and directly on automation. In most areas of the country, emergency '911' calls are now received and dispatched by specially trained staff supported by sophisticated systems combining advanced telecommunications and automated features. These systems provide essential functions such as location of the caller, recording of the call, and identification of available emergency resources. Police

depend heavily on critical information systems, for instance, in the identification of detainees and suspects; in response to identifying information transmitted from a squad car, these systems provide almost instantaneous information regarding warrants or criminal history.

Government services also depend on automated systems to a great extent. More-and-more, taxes are not only analyzed and tallied by automation, but taxpayers file automated returns and make electronic payments. Weather forecasters make use of automated sensors and of complex computer models to assure that their predictions are as accurate and timely as they can be. Continual computer review of weather conditions provides one important warning mechanism against dangerous weather. Both determination of eligibility for government assistance and payments such as social security, veterans' benefits, and Medicare are managed and conducted with a great deal of automated support.

Over time, the reliance of the critical infrastructures on information technology has become more complete and more extensive. Recent experience has also shown that such extensive reliance on automation sometimes makes these infrastructures unexpectedly fragile. In December 1991, AT & T 800 service experienced more than an hour outage because of incorrect software loaded into three computers, affecting several thousand calls. On December 31, 1996, a smelter in New Zealand shut down automatically because the systems did not know it was a leap year and so expected only 365 days in the year, not 366. Because of the complexity and interrelatedness of automated systems, it is not uncommon that an erroneous result in a non-critical calculation causes a significant failure of a critical device. The Electrical Power Research Institute reports that generation, transmission and distribution, and customer systems could all be affected by the Year 2000 problem.

In relying on automation for essential functions in producing their primary products or services, the organizations that make up these critical infrastructures are different from other businesses; the automated systems placed in these roles are often more complex and of more unusual design than computer systems commonly used in business for record keeping, taking orders, and making payments. However, the critical infrastructures are also dependent on these more common uses of information technology. Bills are calculated, statements sent, and information provided on demand. In many communities, even the reading of utility meters makes use of radio transmitters and automated receivers to save human time and effort. Client records are maintained in information systems that may include a national network to improve service and convenience. Like most modern businesses, when information must be processed, the businesses that comprise the critical infrastructures usually choose to process it automatically. Critical infrastructure companies are dependent upon each other and on other businesses in a highly structured, and often automated, fashion. Orders are taken and fulfilled completely on the basis of information that is exchanged electronically between partners' systems. The IRS has been encouraging taxpayers to file electronically for several years, but for even longer, employers have been expected to report withholdings electronically.

Several of the critical infrastructures appear to be moving toward a "decentralized production/centralized control" model of operation that will further increase their reliance

on information technology, making them even more susceptible to disruption of their systems. In each case, the addition of higher level, centralized control functions makes the infrastructure more efficient, but it also increases the vulnerability of the systems at a new “single point of failure.”

In electrical power, a California consortium of electrical utilities is working to establish an automated “free market” in electrical power to create a more efficient process whereby utilities with spare generation capacity can sell surplus power to utilities with unmet demand.

In communications, as competition increases for local and long-distance telephone services, reliance is growing on a number of centralized databases which must be accessed to complete a call. Demand is also growing for “number portability,” which would dissociate a telephone number from a specific location, and tie it, instead, to an individual or business. When number portability is implemented, it will result in the success of every call being ultimately dependent on the correctness of a small number of centralized databases, which could be high value targets for attack.

In transportation, the first generation of automated air traffic control systems focused control on the airspace immediately around airports; the current generation controls air traffic as it travels invisible “highways in the sky”; the next generation of systems will plan traffic flow and seek to ward off bottlenecks minutes to hours in advance.

3.0 Nature of the Year 2000 Problem

All of these systems follow recorded instructions to perform their tasks. Most often, these instructions are contained in software. The Year 2000 problem arises because some common practices in processing dates are not strictly correct and will lead to erroneous results when processing years beyond 1999. At its root, the problem is simple. In the early years of computing, storage of data was expensive and processing of extra data was time-consuming; programmers sought ways to minimize the amount of data to be stored or processed. One technique was to adopt the common idiom of abbreviating the year in dates to just two digits, storing and calculating from '77', for instance, instead of '1977'. This saved two digits per date in the file and it also saved time in calculations made using these dates. Once started, this practice perpetuated itself through habit, formal standards, and the need for new data to keep the same format as existing data. Therefore, although the problem started long ago, in an entirely different technological era, it is not confined to mainframe applications written in COBOL. Hardware and software products of all vintages have been found to be non-compliant; most custom applications must also be considered at risk. Recently, InfoWorld Electric reported that even "browsers" for the worldwide web executing JavaScript code are not presently Year 2000 compliant in all respects.

Whenever a system must process dates from two centuries—for many systems when the next century arrives—these abbreviations, and the logic that depends on them, will become ambiguous. Many instructions that worked using two-digit dates in this century will fail when one of the dates is in the next century. Algorithms intended to calculate ages, intervals, or delays, fix the order of events, or determine the day of the week may fail. Since dates have often been used as a simple check of validity of data entry, failures may occur where no critical function obviously relies on date processing. Many systems will fail well before the Year 2000. As soon as a data element or data file contains dates that cannot be processed correctly, the chance of failure exists. As early as 1969 or 1970, bank systems used for tracking and calculating 30-year mortgages encountered problems. Since then, date-dependent systems have continued to fail, at rates that will increase at least until January 2000.

As an example of a typical problem, a system might determine a pensioner's age by subtracting his or her birth year from the current year, so for someone born in 1927, the system would correctly calculate

$97 - 27 = 70$ years, but incorrectly arrive at

$00 - 27 = -27$ years.

Given an age of -27, the system might detect an error and ask for human intervention, which could be quite inconvenient if it happened for every pensioner in the file; or it might fail to detect this as an error and just continue processing, finding that -27 is less than 65, so this person is not eligible for any payment, which could also be inconvenient if it happened for every pensioner.

In another example, an airline reservation system might check the validity of the planned route to assure that the return flight occurs later than the outbound flight. If one the trip starts in 1998 and ends in 1999, a two-digit comparison of the years properly shows the return as the later flight; if, however, the trip starts in 1999 and ends in 2000, a two-digit comparison will show that the return happens earlier than the outbound flight and the reservation will be rejected.

The examples given here have not been documented in any existing system, and are certainly not life-threatening. These examples are given to provide a concrete understanding of the Year 2000 problem and to illustrate an important feature of the problem: while the cause is simple, the remedy can be quite complex. Individual Year 2000 errors should be easy to detect and remedy, the number of opportunities for errors and the number of errors each could produce quickly grow to be unmanageable.

A system that processes dates in one place is likely to process many dates in many places. To assure correct processing through the year 2000, all of these places must be located and checked. Furthermore, in addition to the use of two digits to designate years, the common understanding of the Year 2000 problem has come to include a number of other, closely related problems in date processing. The challenge of finding and repairing all of these errors has been likened to that of picking up a million dollars worth of pennies scattered in the lawn. What you do when you find one is simple, it's finding enough of them fast enough, and picking them up, that makes the job difficult. Unfortunately, dates are used throughout modern information processing, so every system must be examined for impacts, and decisions must be made regarding whether and how to fix every system. The scale of this effort is what makes the Year 2000 problem hard.

A typical software system has 10 thousand to 1 million instructions, or "lines of code (LOC)"; a typical business may depend on tens to thousands of specially constructed software systems like these in the conduct of business. Many moderate-sized organizations estimate they have responsibility for tens of millions of lines of code.

As one example, the Centers for Disease Control (CDC) estimates that it has 164 systems, totaling approximately 7.4 million lines of code, which are potentially affected by the "millennium bug", and several other systems that they expect to replace, rather than repair. In CDC's case, these systems are written in 36 different computer languages, with many systems written in more than one. CDC is unusual in having a reasonable understanding of their inventory and the extent of their risks; they seem to be well along toward finding a solution. Many other organizations are not so well along. When starting out to solve the year 2000 problem, most don't know at all what systems they own. In another example, Kathleen Adams, of the Social Security Administration, says that it has about 30 million lines of code, about 20% of which is affected, and that this realization "scared" them in 1991, when they discovered it.

Professionals who have looked at the Year 2000 problem roughly estimate that reviewing and correcting each line will cost between \$1 and \$10 (though these numbers are highly dependent on the type of system, language, and design), so many organizations find that the Year 2000 problem could absorb a significant share of their total information technology budget for the next two years.

In the course of examining this problem, the software engineering community has adopted a standard view of a Year 2000 project, originally proposed by Gartner Group. This view divides the project into four active phases (plus a fifth, preliminary phase known as “awareness,” devoted to getting an organization to address its Year 2000 risks). The four active phases are

- Assessment
- Renovation
- Validation
- Implementation

Each phase contains characteristic technical issues that cause the Year 2000 problem to be difficult. The four phases and their characteristic issues are outlined in the paragraphs that follow.

3.1 Assessment

During this phase, the organization gathers and analyzes information in order to size and scope the problem. This is a critical phase and could take up to one-third of the total effort. Work starts with an inventory of all hardware and software systems. Business and information technology experts analyze the inventory to determine the relative business value of the systems and estimate the costs to bring the systems into compliance. Based on business value and costs, the organization assigns priorities and performs “triage,” (identifying some systems for early retirement, wholesale replacement, or operation without repair.)

During assessment, it is important to realize that, although the focal issue is the “Year 2000” problem, there are several related date processing issues that may cause systems to process dates incorrectly for a variety of reasons and to fail at any time, either before or after January 1, 2000. In addition to the main concern regarding two-digit years, an organization should also consider, during its assessment, such potential problems as dates used as

- special flags (for instance, the year ‘00’ or ‘99’ used to indicate missing data) or the date ‘09/09/99’ used to indicate a date not reported; and
- leap year (e.g., the fact that 2000 is, in fact, a leap year while 1900 was not; and
- restricted range (that is, the possibility that the range of dates over which the system operates correctly may be restricted by the range of a variable or the size of storage allocated to a data set).

Compounding the complexity of assessment is the idea of an “event horizon”—that failures will occur whenever systems process dates, not only in January 2000. The following table lists possible event horizons errors in several contexts.

Error / Context	Event Horizon
Year 2000	
30-year mortgage	January 1970
5-year sales incentive	January 1995
5-year forecast	January 1995
2-year forecast, cash flow	January 1998
Budget	1997 or 1998
Special flag dates	
'99' for invalid year	January 1999
'00' for invalid year	January 2000
'09/09/99' for missing date	Before Sept. 1999
Leap Year	Mar. or Dec. 2000

Obviously, this is only a selection. The point here is that date processing errors are of many types and could lead to failures at almost any time. Furthermore, there are good reasons that some systems should be repaired well in advance. Accounting systems, for instance, face a critical use in the annual “closing” of the books at the end of the fiscal year. The renovation of these systems should be completed a year ahead of time so that the non-compliant versions of them are still available as a contingency should they fail during their first annual closing. Only careful review of software and systems can determine for any organization which types of errors it has and how long it has to fix them.

For those systems where repair is the appropriate course of action, technical experts now identify and quantify the impacts of Year 2000 date changes in detail. To do this, they must examine source code and data of custom software and they must obtain information regarding compliance of commercial systems (operating systems, user interface systems, data base management systems, compilers, network services, etc.) from the vendors. Based on this understanding of impact, these experts can propose a Year 2000 solution strategy and a validation strategy implementable and appropriate for each system. The combination of strategies proposed allows the organization to devise a workable plan for managing the entire Year 2000 conversion project.

Several technical issues typically arise during the assessment phase. Often, for instance, source code for the organization’s custom software is missing or the available source code does not match the system in operation. Many of these systems are fairly old, and their code has simply been lost over time. Where source code is unavailable, scanning to find all uses of date processing logic is problematic, although recent advances in software analysis tools show some promise in this area. Even where source code is available, finding all of the dates can be very difficult. Such a scan typically starts with searching through the software for variable or function names that sound like they involve dates, so

names including strings like YEAR, YR, DATE, or DT might be identified. Some of these will not be dates at all—an UPDATE function or YEARLY TOTAL might not involve date processing at all. Other dates will not be so conveniently named; a variable called “ELIGIBILITY” might actually contain birth year data or all variables in the program might have cryptic numbers instead of readable names (e.g., D247). Furthermore, dates will be used in calculations and decisions, spreading their impact throughout the code. Finally, just because dates are used does not mean their use is noncompliant for the year 2000; someone must examine each use to determine whether it meets the definition of compliance in effect for the system, or not. So, even where source code is available, identifying all instances of dates and all components that are noncompliant can be a great challenge.

Assessment is also where the volume of code or data to be examined is first perceived. Most organizations manage information resources incrementally, updating, building, or replacing a few each year. A small staff, intimately familiar with the business, can perform much of the work. Testing is minimized because only a small part of the capability changes at any time; new errors can be readily traced back to new changes. To address the Year 2000 problem, many organizations will have to review, convert, and retest their entire inventory over two or three years. Often, neither their tools, their methods, or the size of their staff is up to such a large effort over such a small period of time. To handle the load, organizations will have to seek outside help, upgrade their automated toolsets, retrain staff, and improve procedures for project management of software maintenance. Because of the magnitude of change so rapidly put in place, formal testing may be necessary for the first time. For many organizations, the Year 2000 effort will be the biggest single project they have ever had to manage, therefore just keeping it on track will be a challenge.

3.2 Renovation

In this phase, the organization puts the selected compliance strategies into practice, changing code and upgrading commercial products as planned. For software, several different techniques (“date expansion,” “windowing,” “data compression,” “encapsulation,” or “bridge programs”), with numerous variations on each, have been defined for bringing software into “compliance,” but these techniques differ in their permanence, accuracy, completeness, and cost of implementation. As a result, the no single definition of “compliance” can serve for all systems and no single technique can be used to achieve compliance everywhere in most organizations. The essence of a Year 2000 strategy for a system is the definition of the right combination of techniques and the right level of compliance to assure that the system performs its functions while limiting the cost and time required to fix it. The four techniques generally recognized are

- Date expansion
- Date encoding

- Windowing
- Encapsulation

3.2.1 Date Expansion

The various technical approaches for achieving Year 2000 compliance vary greatly in their cost and extent of impact on the system being modified. They also vary in the degree to which they achieve “absolute” compliance, or correct date processing. At the high end of cost and impact is a technique known as “date expansion,” in which the year representation of dates is expanded from two digits to four. To do this requires modifications to both historical and current data containing dates and to date-processing code. Date expansion requires the most extensive modifications of any of the common techniques, but it does result in a system that is “correct” in handling dates.

Unfortunately, date expansion is not practical for many systems at this time, either because it would take too long or because essential resources, such as source code, are not available.

3.2.2 Date Encoding

Where date expansion is not practical, several other methods have been suggested, each avoiding or deferring some large share of the effort, or some obstacle encountered in date expansion. One such is “date encoding.” Under the date encoding approach, additional conventions are created for representing years starting with 2000 as two characters (or dates as six characters). The table below gives examples of the many techniques which have been suggested:

Dates	Sample	Sample
	2-Digit	6-Digit
	Year	Date
	Encoding	Encoding
13 October 1997	97	071013
30 September 1998	98	080930
31 December 1999	99	091231
1 January 2000	A0	100101
30 August 2001	A1	110830

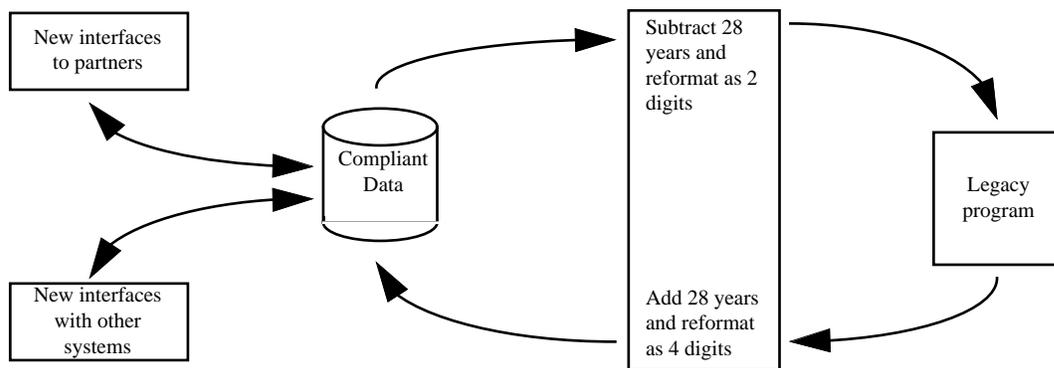
Software code that compares or calculates using the dates may then need to be changed to recognize the new conventions, or preexisting data files and archives need to be modified to use them. Depending on the situation surrounding an individual system, this might provide a great savings of time or effort. In many cases, simply keeping dates the same *size* in the date files will simplify the renovation immensely.

3.2.3 Windowing

Another, simpler, approach that preserves two-digit data is the “windowing” method. Here, two digit years are logically interpreted as belonging to either the 1900s or 2000s by reference to a boundary value; for instance, dates after ‘50 are understood to be 1900s, dates ‘50 and before are in the 2000s. Where windowing makes sense, it may not require any modification of data files, although the logic to properly interpret dates must be added throughout the software. Windowing makes most sense where a limited range of dates must be processed and a large number of records are used. An order fulfillment system would be one good example.

3.2.4 Encapsulation

The date encoding and windowing techniques discussed so far are effective at reducing the need to modify data files. Techniques have also been suggested for reducing the need to modify the software in a system, for instance where software does a great deal of date processing or where the source code is missing. The most well-known of these techniques is called “encapsulation.” In an encapsulation approach, dates in historical and current data are converted to four-digit years, but when they are to be used, a new piece of software—a wrapper or encapsulation routine—shifts all dates in the calculation back 28 years and converts them to the format required by the existing software, with years represented as two digits, as shown in the following illustration.



By doing so, the routine preserves such attributes of the situation as day-of-the-week and leap year and allows the software to continue processing as if it were operating in the 1970s, instead of the 2000s. Once a result is computed, the encapsulation routine adds back the 28 years to obtain the correct dates in the 2000s. There are technical factors that make encapsulation more difficult in practice than it may appear to be, but where it works, it can be a tremendous improver of efficiency.

These, then, are the four most commonly discussed techniques for achieving Year 2000 compliance: date expansion, date encoding, windowing, and encapsulation. Each of them brings certain advantages to addressing the problem and each of them has certain limitations. Many more inventive and creative techniques have been devised to address particular situations or more technical aspects of the problem. Almost always, date

expansion is the preferred method, used wherever sufficient information, sufficient time, and sufficient resources are available. Unfortunately, these conditions are not found in a very many Year 2000 conversion projects; more often than not, time and resources for conversion are severely limited and essential information is missing. Creative approaches are often needed to give any hope of success.

3.2.5 Integer Representation

One final conversion technique fits none of the categories outlined above, but is very important in any discussion of the problem. This involves representation of dates as integers. As it turns out, for computers based on binary arithmetic, storing dates as either six or eight decimal digits is not the most efficient choice. If dates are represented, instead, as a count of days since some base date, much internal date processing is made more efficient and there is no problem with ever “running out” of dates, since even the space of six characters is enough to represent billions of years of days. Generalizing still further, time can also be represented as a count of seconds, or milliseconds, since a given time, unifying the treatment of date and time-of-day. This type of approach addresses the root cause of the Year 2000 problem directly and so yields a solution that solves not only the Year 2000 problem, but several related problems concerning automated processing of dates and time. The one issue that remains is simply that of converting the internal representation as an integer to an appropriate “human readable” form as it is needed for each report or screen in the system.

3.2.6 Commercial Products

Even if an organization can examine and correct its own software, it will quickly find that it has a reliance on commercial products over which it has very little control. These may be fundamental to several systems; such products as the basic operating system of a computer, networking software, databases, or languages may require upgrades to achieve Year 2000 compliance. Their vendors report that even relatively new products such as JavaScript Visual Basic 3.0 and 4.0 and Novell Netware 3.12 should be upgraded to compliant versions. Organizations dependent upon these particular products are fortunate that they are still being supported and that compliant versions will be available; many organizations dependent on older technology will find they are not so lucky. Still, each organization will have to understand what compliance means for each product and then schedule and perform upgrades coordinated over a short period of time over the entire inventory of the affected product line. Then, each organization will have to determine what steps must be taken to assure that the applications dependent upon these products are also compliant. That the new version of the database is compliant simply means that the developer has provided a few convenient ways to write compliant applications using the database; there are still an infinite variety of ways in which applications can be non-compliant.

3.2.7 Hardware Or Firmware

Not only computer software is at risk. In our modern society, many seemingly simple machines contain automated components. Even components with limited capabilities for

storing data contain program logic likely to fail. It has been suggested that elevators or building security systems may fail if they store or use dates. Some instances of such risks are known to manufacturers or to the public. Many more may be unknown and uncharted, contained in program logic embedded in “chips” in electronic devices. Even where repair is simple, a very large number of repairs may be needed across the country, each requiring physical access to the internal workings of a device, and each requiring a compliant replacement part.

3.2.8 Shared Data

Although each organization must correct its own data, organizations that exchange data must also assure that they correct the formats and standards for these exchanges, as well. Commerce these days is heavily dependent on automated messages, which must be written and read by software. If the software at one end does not write information that can be read by the software at the other end, the message does not go through. For instance, if two computers are programmed to update data in their directories periodically, they must share an understanding of time or date. So, for electronic messages, dates of concern are not confined to the message, they are found in the “envelope” as well.

Because of particular barriers, restrictions, and limitations found in every case, any actual conversion project is likely to include several of these techniques, along with specially constructed temporary or permanent “bridges” between systems in the enterprise or with external systems. Year 2000 “compliance,” for any organization, then, is likely to mean various things at various points of the system.

3.3 Validation

Estimates for the testing phase of a Year 2000 project range from 30% to 70% of the total effort, indicating a degree of disagreement over the exact value, but clear agreement that testing will be the most demanding of all phases of the project. Like conversion, several strategies for testing Year 2000 compliance of systems have been devised, but none is ideal, or even effective, for all situations. Fortunately, Year 2000 compliance does not affect all aspects of program operation or all parts of a system equally, so it should be possible to “short-cut” the testing process and still gain good assurance of the adequacy of the repairs. Unfortunately, however, definitions of compliance vary, often across systems, and records of testing procedures from the original development may be missing, if they ever existed at all, so it is quite likely that any tests that are done will have to be developed specially for the occasion. Finally, it is a well-known fact of software maintenance that any change to software brings with it a small, but non-negligible chance of introducing new errors, often unrelated to the original reason for the change. Therefore, any testing of Year 2000 modifications must include some testing of the other functions of the system, just to assure that it has not been damaged during the repairs.

Testing of software usually proceeds by “levels,” starting with testing of small fragments of code, or units, and continuing with testing of larger-and-larger software subsystems assembled out of the units, until a final test of the complete system is possible. This approach simplifies access to units buried deeply in the ultimate structure of the system

and makes it easier to identify the source of errors when they are found. It also permits automated support to much of the low-level testing.

In unit testing, tests of Year 2000 compliance are much like tests of other modifications made to software. Tests for each unit must be defined from a description of the proper functioning of the unit. These tests embody the definition of Year 2000 compliance applicable to the unit and also cover the other functionality of the unit to assure the modifications to it have not introduced errors. Tests used in routine maintenance or retained from the initial development of the system may be of great value in constructing unit-level tests, but these will have to be analyzed and verified to assure that they are still correct, given the revised definition of correct processing for the unit. Once defined, unit tests can often be executed and evaluated automatically.

As testing proceeds, automated support and use of past tests provide less advantage. Because of the complexity of software, which makes complete testing at higher levels impractical, these higher levels of testing are closely tailored to the particular changes implemented. As a result, Year 2000 tests of subsystems and systems will be different from other tests created for other purposes. Furthermore, systems under maintenance over an extended period will have changed in structure, gained interfaces to neighboring systems, added data files, and extended their functionality over time. While these changes will leave some unit tests unaffected, they will significantly change the intended operation of larger pieces of software. Therefore, tests for higher levels of testing will need to be constructed specially for the Year 2000 project.

Because Year 2000 projects usually affect systems in operation, the question arises whether a separate test environment is necessary, or whether they could be simply “tested in place.” The answer is that neither approach is wholly satisfactory. As some organizations have found by ill-conceived experimentation, introduction of future dates into the operational environment (necessary for advanced testing of Year 2000 changes “in place”) can contaminate essential data. On the other hand, constructing a test environment that sufficiently mirrors the operational context can be impossible, or prohibitively expensive. In practice, Year 2000 testing strategies rely on off-line testing to the greatest extent possible, followed by gradual introduction of the partially tested system on-line, with provisions made for roll-back, parallel operations with the unmodified system, or both. The result is that, characteristic of software change testing, the system is never fully tested until it has been operating successfully for some time.

It has been noted that the critical infrastructures depend on automated systems in essential roles where a failure could lead to loss of life, property, or inability to fulfill the infrastructure mission. When automated systems are relied upon to this extent, they are usually subject to extremely rigorous testing and other types of intensive review before they can be used. For example, when the FAA replaced its “host” mainframes in the mid 1980’s, controlled testing of the replacement hardware, and the software which had been moved onto it, also took over a year. Individual configuration and installation of these systems at 20 sites around the country took an additional two years, as is to be expected for critical control systems in such an environment. Clearly, for some of the functions where the critical infrastructures depend on automation, the appropriate time for start of testing is upon us.

3.4 Implementation

So far, we have addressed the analysis of an organizations' Year 2000 situation, the assessment of its portfolio of systems, and the modification of each system to remove Year 2000 errors. The final remaining phase of a Year 2000 project is the "Implementation" phase, in which the corrected systems are brought on-line in their intended roles. During this phase, scheduling concerns predominate. Because of the scale of the project, it is generally not possible to change over to compliant versions of all systems at once; compliant systems must be introduced gradually over months or years. One Fortune 500 company that, in normal operations, replaces a few dozen software routines each weekend, has found that it will have to replace 500 routines every weekend to implement its Year 2000 changes.

Further complicating implementation is the fact that virtually all automated systems depend on commercially furnished support software— operating systems, network management systems, database management systems, word processing packages, and the like—whose makers are, themselves, in the process of bringing their products into compliance; so, along with introducing corrected versions of its own systems, each organization must schedule introduction of compliant versions of the commercial products on which its systems depend. Since an organization's systems typically depend upon each other, on data exchanged with other organizations, and on multiple commercial products in a complex mesh of interdependencies, resolving the introduction of compliant replacements into a workable series of upgrades can be a very complex problem.

Businesses are finding that implementation of Year 2000 projects requires them to understand the interaction of their systems and their interfaces to external systems to a degree never required before—an understanding that can only be gained through extensive analysis.

Inferior performance (speed) is one impact of Year 2000 renovation that, while expected, is likely to be detected or demonstrated only during system implementation. Most of the remedies to the Year 2000 problem increase the processing load in some way. Field expansion adds two characters to every date—characters that must be repeatedly read, written, and stored by the affected systems. Windowing adds processing logic needed to resolve the century for 2-digit years. Encapsulation adds entire new routines, as does the bridging often needed during the gradual transition to renovated systems over time.

Where processing is added to the workload of an information system, the system can be expected to slow down. For a transaction processing system, the result can be a marginal or a significant reduction in responsiveness or throughput. For an automated, or control system, reduced performance can lead to incorrect behavior, since it can cause the system to miss processing deadlines.

Because Year 2000 renovation puts system performance at risk, it would be best if organizations addressed performance early in the project. Unfortunately, this would require sophisticated testing, modeling, and analysis that is beyond the capabilities of many internal information resources organizations.

3.5 Available Assistance

For many organizations, this will be the largest, most intensive information technology project ever. Its scale is what makes the problem difficult. As companies work through the complexities of assessment, renovation, validation, and implementation of corrected systems throughout their operations, many of them will seek assistance in making such massive changes to their essential systems.

The Year 2000 situation has spawned an entire sub-industry devoted to assisting organizations to address the problem before the deadline expires. Following the preexisting structure of the systems reengineering and renovation business, the Year 2000 assistance field can be divided into three major segments: service providers, tool vendors, and strategists. Often, individual companies will participate in more than one of these segments. Secondary participants are also becoming evident, in companies that specialize in training staff to perform Year 2000 work, or in brokering information regarding compliance of commercial products. Because of the short lead time available for Year 2000 work, many of the offerings available are based on standard tools or techniques, tailored or simply repositioned to address the Year 2000 market. Talk of massive expenditures and impending panic also means that the “snake oil” business is booming.

3.5.1 Service Providers

The service providers segment is led by companies that got into the systems reengineering business before the Year 2000 market expansion, such as Cap Gemini and CACI, by the technical support organizations associated with major hardware and software vendors, and by the information systems practices of the big six accounting firms. These companies are similar in that all have general capabilities in system renovation they have tailored to the new market environment. In addition to these, the Year 2000 market offers opportunities for start-ups to specialize in such services as inventory assessment or renovation of particular common types of systems (as mainframe COBOL).

These services may be performed at the organization’s site, or off-site in a “Year 2000 factory” operated by a contractor. These factories perform selected repetitive and well-structured tasks (such as assessment or code change) very efficiently. There are good reasons to do at least some portions of the Year 2000 work in such a factory, since economies of scale will support specialized tools, specially trained staff, and streamlined procedures not otherwise available. A factory can be just across town, or it can be on another continent; the type of work suitable for a factory facility can be easily relocated. The emerging software industries in India, Ireland, and Eastern Europe are home to several such factories.

3.5.2 Tool Vendors

The tool vendors segment, like the service providers, is led by companies who have long been in the software engineering or reengineering tool business and have simply refocused market strategies or tailored capabilities to address the Year 2000 needs. This means that many of the available Year 2000 tools have proven track records and

capabilities over decades. Companies like Platinum and Computer Associates have been building software maintenance tools for many years; their practices and history for update and user support are known. What this phenomenon also means that the state of the art for Year 2000 renovation toolsets is in line with the state of the practice in renovation, generally, tools are most available for COBOL systems, with a few available for other popular languages and platforms. Some new entrants in the tools market are finding a niche by focusing the Year 2000 problem, but the competition is stiff and the time is very short to build a reputation and capture market share.

The available tools provide capabilities of great value to Year 2000 renovation, although there is no “silver bullet,” tool to dramatically reduce time or cost of the renovations. Common capabilities include pattern matching, lists of date keywords or definitions of date concepts, language-specific date checking logic, automated browsing of source code, and checklist management linking instances found in assessment with repairs. Some tools offer to partially automate recoding to selected compliance methods. Many offer simple cost or effort estimation.

3.5.3 Strategists

The final segment of the Year 2000 market is the “strategists.” These companies generally emphasize the business implications of the problem and try to find solutions for each client that are in keeping with its business strategy or direction. This segment is led by information systems consulting groups, who often address the Year 2000 as an opportunity for business process reengineering. Although these companies may or may not offer renovation services or tools themselves, by concentrating effort on those systems of greatest business value and by assisting organizations to abandon from their inventories or “triage” systems of lesser value, they can have a great impact on the results of the effort. Because no large staff or existing product line is needed to enter this segment of the market, this is the segment where many small consulting practices are concentrating their efforts.

4.0 Year 2000 Vulnerabilities

Many of leading companies in the critical infrastructures have reported that they have plans to address the Year 2000 problem in their systems. Most of these began long ago to implement these plans and are well on their way. Almost without exception, however, these same companies warn that the Year 2000 problem is more invasive and more pervasive than they would have expected and that overcoming it on time will be a significant challenge. Tampa Electric’s reaction—that the biggest surprise in the project was that the more the project was studied, the bigger it got—is typical. Banker’s Trust estimates it will spend \$1 billion; Chase Manhattan Bank estimates \$200 million. The list of critical infrastructure leaders who have recognized the magnitude of this problem, and devoted talent and resources to its resolution, is long and distinguished, including such organizations as:

Telecommunications

- AT & T
- Lucent Technologies

Electrical Power

- Consolidated Edison
- El Paso Electric Co.
- Hawaiian Electric Co.
- Nuclear Regulatory Commission
- Tampa Electric CO.

Gas and Oil Storage and Transportation

- Northern Illinois Gas Co.
- Yankee Gas Services Co.

Banking and Finance

- Banc One Corp
- BankBoston Corp
- Banker's Trust
- Chase Manhattan Bank
- Chubb & Son, Inc.
- Citibank
- FDIC

- Federal Financial Institutions Examination Council
- Goldman Sachs
- The Federal Reserve
- HONOR Technologies, Inc.
- MasterCard International, Inc.
- Merrill Lynch
- Morgan Stanley
- Senate Banking Committee
- Visa International Inc.

Transportation

- American Airlines
- Federal Aviation Administration

Government Services

- California Department of Motor Vehicles
- Internal Revenue Service
- Office of Management and Budget
- Social Security Administration
- Department of Veterans' Affairs

Experts in each of these vital elements of the critical infrastructures have assessed the problem, and these experts have found it to be real and challenging. The Year 2000 problem poses a definite risk to the infrastructures. This "natural" risk to the infrastructures has been addressed very adequately elsewhere, however [see Newsweek, June 2, 1997, pp.52-59, for one accessible and engaging treatment], and is not the purpose of this paper.

The Commission has a more sinister concern, namely, that the very efforts to fix the Year 2000 problem will create vulnerabilities within this nation's infrastructures. The two of greatest concern are:

- Limited control over essential systems and data once access is granted; and,
- The risk that repaired systems will fail as a result of their interdependence.

4.1 Limited Control Over Essential Systems And Data Once Access Is Granted

In examining their Year 2000 situation, most large organizations have found that correcting the problem will place great demands on their existing staff. Often, this results in hiring outside help, in the process granting access to essential systems to personnel who otherwise would never have had access to them. As quoted in ComputerWeekly,

Judith Scott, chief executive and secretary of the British Computer Society said recently that “there is a risk that security bombs [software designed to sabotage user’s systems] could be placed in software opened up for legitimate purposes” such as Year 2000 renovation.

One way in which this happens is that the organization seeks additional outside help, by contracting for services or personnel, either to take on the Year 2000 project or to take over routine maintenance and operations activities. Use of outside help does mean that software previously retained under the direct control of the organization is released to outside personnel for the first time. Either the software is sent out or additional people to work on it could be brought in; this second alternative retains the greatest degree of control. Unfortunately, finding sufficient internal space and outfitting it for intensive systems maintenance work may be a long-lead item. Sending the software off site, allowing the contractor to dial in, or extending the corporate network to a contractor facility, could be much faster, but they offer less control over a greater fraction of the corporate systems. As the Wall Street Journal reported in 1996, “For much of the work, Consolidated Edison is turning to outside contractors, who, in turn, are using subcontractors in Ireland and India, all connected via satellite and high-speed phone links directly to Con Ed computers.” Given the pressure on companies to meet the immovable deadline, this must already be a common situation; and Consolidated Edison got started relatively early.

Another way in which additional personnel may indirectly access essential systems is through their involvement in the development or installation of new software maintenance tools, compliant versions of products, or specialized system components designed to address some portion of the Year 2000 problem. The vendors of these products, themselves, are addressing the Year 2000 problem, either by bringing their product line into compliance or by offering new capabilities to address the problem. In either event, they, also, will require additional staff to accomplish this goal along while they continue to pursue other business objectives. Clearly, the critical infrastructure organizations have little control over access granted to the newly acquired staff of the vendors they rely on.

Not only will the Year 2000 problem lead organizations to yield access to their systems to a greater number of people, but the extent of access granted to sensitive systems and data will be broader than ever before. This means that organizations will be vulnerable to a scale of disruption of their automated systems that is usually precluded by the slow pace and limited extent of change introduced from outside.

Unlike most software or systems maintenance activities, the Year 2000 problem will require modifications to almost all (one estimate is 83%) of the software in inventories. Because of this wholesale change, companies will give Year 2000 contractors very complete access to their software inventories. Most companies don’t understand their software inventories very well. Year 2000 contractors report that inventories they receive for analysis are often incomplete or contain multiple versions of the same program; some report finding such items as games or personal letters included in software delivered for analysis and repair.

Not only will companies give contractors access to most of their software for Year 2000 update, but the same pressures will drive them to give these contractors access to most of their automated systems during the installation phase of the project. In essence, many companies will completely overhaul their systems during a short two-year period, installing new versions of virtually every significant program and data set over this time. Compounding this risk is the fact that, by design, software is easily replicated and distributed so that a single modification can be introduced across a broad architecture or a large number of users.

Often unrecognized is the high degree of sensitivity that should be associated with software code. Software may contain sensitive data, certainly, but equally importantly, software usually contains business rules. *It is not unusual to find that software is the best record of the way an organization actually does business. When changes are made in policies or procedures, and software is involved, the software must be modified; people and documentation can wait, sometimes forever. An adversary who gained access to a company's software could understand some aspects of the actual business practices of the company better than the company's own management.*

Although the Year 2000 literature pays much less attention to it, for many organizations, conversion of data will be as great an undertaking as conversion of software. Most organizations will treat release of their business data with even more caution as release of their software; data is recognized as a sensitive resource. However, as with software, the volume of data to be converted may mean that organizations must obtain outside help, whatever the risks.

4.2 The Risk That Repaired Systems Will Fail As A Result Of Their Interdependence

Even approached with great care, Year 2000 projects will tend to reduce the ability of a company's systems to resist attack or mishap by introducing increased errors into the system, by delaying necessary business improvements, by eliminating valuable redundancy, by redirecting or diluting security resources, and by generating complexity and instability in installations that would otherwise be unnecessary. This means that organizations will be more vulnerable to all sorts of attack as a result of their Year 2000 projects.

It is a truism in software engineering that all software modification has the potential for introducing errors into the system. Sometimes, these errors are introduced at the point where the software has been modified, but they can be created anywhere in the modified portions of the code. Rates vary, but one rule-of-thumb is that three to six errors are to be expected in every thousand lines of code that are modified and ready for incorporation in the system. Organizations can expect that Year 2000 projects will introduce errors at approximately this rate, which could affect all aspects of system functioning. Testing and careful monitoring of the modified systems in use are the only means available for detecting these errors, leaving the systems in a weakened state until the errors are fixed. Until then, the greater number of errors increases the likelihood of all types of system failure—creation of faulty data, business logic errors, or general system failure. Most of the critical infrastructures are designed with some spare capacity or engineering margin to

allow for transient outages and repairs of components without jeopardizing the integrity of the whole. If the Year 2000 problem significantly increases component failures, then it may reduce the critical infrastructure below the minimum level needed for operation, resulting in failure of one of the critical infrastructures.

Not only will Year 2000 projects, like any modification, introduce errors into the system at the same time that they accomplish their intended repairs, the scale and emphasis of Year 2000 efforts will cause most organizations to delay other modifications to their systems. To gain sufficient change capacity to meet the Year 2000 deadline, most organizations will defer other work. In the Federal Government, for instance, agencies have been directed to accomplish the Year 2000 modifications within their previously allocated budgets; certainly their tendency will be delay other information technology work where they can. These normal changes include upgrading commercial hardware and software such as operating systems and database management systems, and incorporating changed business functions. We should expect that companies will delay upgrades where possible and defer business changes. This will result in systems that are less in tune with modern technology and the current business environment than they would otherwise.

One common feature of Year 2000 projects is the elimination of outdated or redundant systems in order to reduce the work required to bring an organization into compliance. This “cutting out of the dead wood” has its place and benefits, but it also may eliminate useful redundancy valuable as a double-check on automated processing. When the organization has trimmed its systems down to only one source for each type of information, it may find that errors take longer to detect, or go undetected altogether. In one famous instance, the investigation of discrepancies between redundant job accounting systems on a sensitive computer system at Lawrence Livermore Laboratory eventually led to the discovery of significant hacker penetration of security in systems at several Government labs and contractors.

Finally, during the transition of Year 2000 compliant software into production, organizations will be operating more complex installations than they usually do. The large number of commercial products to be upgraded and legacy systems to be replaced will mean that software configurations will change more frequently. In addition, at each increment there will be compliant and non-compliant elements operating side-by-side. Temporary code and data structures will be necessary to ensure these elements to work together, further increasing the complexity of the whole. Interoperability and cooperation of systems among business partners, along with the international concurrency of the Year 2000 problem, lead to another important vulnerability, that of interdependence among the critical infrastructures and between infrastructures and other businesses. Each organization may find that the Year 2000 problem is the largest systems challenge it has ever had to face, but the risks of the problem are still broader than any single organization.

Organizations that get their own systems in order still run the risk that data exchanged with business partners’ and customers may not be compliant. As Andrew Hove, Acting Chairman of the Federal Deposit Insurance Corporation, testified to the Senate Banking Committee on July 30, 1997,

Year 2000 risks to financial institutions are not limited to their internal systems. Even financial institutions that have taken a proactive approach in addressing Year 2000 problems internally, nevertheless, may encounter difficulties if parties external to the bank with whom they exchange data electronically are not prepared for the century date change.

An American Bankers Association survey reported in the *Information Week* on 15 September 1997, found that two-thirds of respondents have not completed initial Year 2000 assessments, and only 20% have started implementation. On 8 September 1997, the Bank for International Settlements in Switzerland said the change of millennium was the biggest potential challenge for the financial industry and called for urgent action to ensure all computer applications could handle it. Perhaps of even greater concern, because there of differing techniques for achieving "compliance," organizations who share data must agree on the approach taken, or at least on the proper interpretation of the data that results. Two organizations can be individually compliant, but unable to share compliant data.

Where organizations have taken prudent steps to assure continuous operation in case of system outages by providing contingency capabilities, the Year 2000 situation puts them at risk because these contingency capabilities, themselves, may fail at the same time. The critical infrastructures often play a crucial role in these contingency plans and in responses to disasters throughout the nation. When an emergency occurs, our first reaction is to pick up the telephone to call for help. Increasingly, these calls, for example, go to centralized dispatching centers for assignment of the optimum emergency resources. Of call center equipment currently installed (equipment closely related to dispatching centers), one estimate, by Dataquest, Inc., is that 25% must be replaced to deal with the Year 2000 problem. Not only might a Year 2000 problem cause the telephone not to work, but it could, at the same time, cause the fire department dispatching system and the streetlights to fail. Damage, if any, would affect the financial infrastructure as it responds to property, liability, and life insurance claims.

5.0 Year 2000 Threats

No new threats to the critical infrastructures are created by the Year 2000 problem. The list of adversaries remains the same. However, the Year 2000 problem and steps taken to alleviate it offer these same adversaries some new modes of attack and improved advantages through the use of some old modes. All three types of threat, physical, cyber, or passive (listening) are changed by the Year 2000 situation.

Entirely new threats are unlikely to arise because of the Year 2000 problem; but the situation opens opportunities for the existing threats to adopt new guises. New opportunities for cyber attacks are opened through the access given to outsiders involved in repairing Year 2000 problems in systems, or installing these repairs. A wealth of sensitive information will become available to determined competitors and adversaries as critical infrastructure organizations obtain assistance in converting their data archives and their systems. Adversaries working for Year 2000 service providers or tool vendors could also cause direct harm to the systems they are hired to repair. Ineffectiveness in meeting the increased need for systems security vigilance surrounding the Year 2000 repairs may make physical or cyber attacks on the infrastructures easier. Or, these attacks may simply be made more damaging by timing them to coincide with periods of heightened instability or weakness resulting from the Year 2000 renovations. During normal operations and maintenance, many critical systems derive a great deal of protection from the fact that they are intricate and unique and that information regarding their structure is not accessible outside of a small group of dedicated maintainers who have built them over many years. With Year 2000 renovation, outsiders may be given significant access to systems such as these for the first time.

5.1 Effect on Physical Threats

Several means exist for adversaries to take advantage of the vulnerabilities outlined above. Among the simplest is the use of information regarding Year 2000 renovations to more effectively time their physical attacks on the critical infrastructures. Timing an attack to coincide with a period of maximum confusion surrounding the Year 2000, hoping that the Year 2000 problem will compound its effects, is one possible strategy. Where specific information regarding system changes and rollovers can be obtained, this type of attack could be even better focused. During the Year 2000 renovation, spare and backup capacity of the critical infrastructures will likely be pressed into service to aid in the needed transitions of systems. An adversary who knew which capabilities would be stressed at a given time could focus a physical attack against these capabilities and so maximize its effect.

5.2 Effect on Cyber Threats

Beyond the very direct attack characterized as a physical threat, the Commission has been concerned with more modern forms of attack on the information processing capabilities of the critical infrastructures, which it has chosen to term “cyber” attacks. The concept of cyber attack recognizes the possibility that an adversary could disrupt the critical

infrastructures by jamming communications lines, flooding electronic mail, destroying or disabling a key computer, or spoiling essential data files. Within the Year 2000 context, adversaries have increased opportunities for cyber attack against the critical infrastructures, as they could masquerade as service providers working on Year 2000 renovation. During the renovation phase, this would allow them to introduce errors or clandestine changes into the systems as they work on them. During the implementation phase, it would allow them direct access to the organization's systems, where more permanent and damaging actions might be taken.

Third, an adversary may attempt to gain access to sensitive software or data by stealing it from a legitimate conversion service provider. While this might not be an ideal manner for introducing unwanted changes into an organization's systems, it is a perfectly effective manner for obtaining a copy of sensitive data or insights into current business logic.

Even providers of toolsets are not above suspicion, since they generally know the recipient of each copy of the software before it is sent and could introduce any of several types of errors into the copy sent to a critical infrastructure organization. Depending on the type of tool involved, the errors introduced could produce invalid planning or assessment data, making Year 2000 conversion success less likely, or they could introduce actual errors into systems being renovated, when the tools are used to make changes in the systems. Tool providers are also indirectly involved in that the Year 2000 problem is stimulating development of new software analysis tools and driving toward new methods of software analysis. As an example, new tools are now much more helpful in analyzing the functions of software without access to source code.

Because an increased number of data processing errors in sensitive systems may be expected during the period through the Year 2000, cyber attacks on the critical infrastructures may go undetected as attacks, even if the failures they produce are detected. Within a background of reduced reliability, those responsible for systems operations may simply conclude that the noted failures are due to uncorrected Year 2000 problems (or errors made during the corrections), rather than actual attack, and respond accordingly.

5.3 Effect on Information Gathering Threats

Both cyber and physical attacks are straightforward, usually overt, and direct and so likely to be detected by the critical infrastructure under attack. In a more subtle form of attack, the adversary simply gains access to vital information and "listens," using the advantage gained in other venues to gain competitive or political advantages. Because of likelihood that critical infrastructures will maintain reduced control over their essential systems and data during the renovations, the opportunities for information gathering against them are increased. Sensitive data files are a clear risk, but software is also a risk as it often contains the best record existing of the actual business rules of an organization.

6.0 Technologies and Approaches for Protecting Critical Infrastructures

This situation is serious enough to warrant action beyond direct response to the original Year 2000 problem in systems. With over two years to go until the Year 2000, time still remains for Government and non-government action to reduce the impacts of this problem on the critical infrastructures.

6.1 Software Maintenance Practices

Protecting the critical infrastructures against Year 2000 vulnerabilities begins with good system management practices to gain control over sensitive data and systems and the personnel who are given access to them. In addition to heightened security, improvements in quality assurance, verification, validation, and configuration management will help to build confidence that the renovations made to critical infrastructure systems will be reliable.

Protecting the critical infrastructures against vulnerability and attack in conjunction with the Year 2000 problem begins with conducting Year 2000 renovation projects with the best possible management practices, to minimize the risks inherent in the situation and because many of the measures effective against risk—as good planning and careful execution—will also be effective against attack. To do so, an organization must first understand its dependence on automation. This will allow it to correctly estimate the business criticality of its various systems, and so retain the greatest degree of direct control over those systems of highest business value in its operations. Many organizations have decided to perform Year 2000 renovations of their most vital systems in-house, preferring not to trust any outside party to give this work appropriate emphasis and assurance.

Beyond this, certain aspects of the Year 2000 renovation project deserve particular attention of the organization itself. Among these are the definition of compliance, which varies by system and organization, and so cannot be set by any outside agency. By controlling its definition of compliance, an organization can minimize the changes made to its systems, and identify portions of its systems where collateral errors are most likely to have been made. Good practices in quality assurance, verification, validation, and configuration management complete the picture of an organization intent on understanding the changes made to its systems and assuring that they are warranted and have been properly implemented.

As complexity in information systems is nothing new, there are a variety of techniques for “building confidence” in these systems as they are developed or modified. One of the most effective of these techniques is independent review of the work by another qualified party. Judith Scott, chief executive of the British Computer Society, suggests using this technique to review Year 2000 renovations. This may, indeed be effective at preventing the sabotage of systems undergoing renovation, or at least at detecting damage done to these systems before they go into operation.

6.2 Enforcement of Legal Protections for Software and Data

There is a role for Government in reducing the vulnerability of the critical infrastructures to Year 2000 failures. One measure that could be taken is to create and emphasize enforcement of software tampering or misappropriation of trade secrets and business practices that could occur. Protecting the critical infrastructures to some extent from lawsuit will allow them to proceed at the greatest possible pace to correct their systems, which is the only way to maximize their readiness for the Year 2000 when it arrives.

Vigorous enforcement of legal protections, for trade secrets or sensitive business information included in software and data, can establish a safer environment in which critical infrastructure companies can seek help with their renovations. Also, consideration should be given to the establishment of a policy to protect the critical infrastructures from lawsuits, or limit their liabilities due to failures in their attempts to correct their Year 2000 problems. This would motivate them to focus maximum effort on fixing the problem.

A related issue that must be addressed is the ownership of software that has been made compliant by a consultant. Since the organization hiring the consultant may have a need to continue modifying the compliant product in response to changing business conditions and since the consultant may have used proprietary software technology in making the repairs, both parties have legitimate ownership concerns. Organizations that hire few information systems consultants may overlook the necessity of assuring that they retain ownership of their software after it has been renovated. Otherwise, they may find that they must negotiate to license their "own" software from their erstwhile consultants, or from someone else to whom the consultant may sell its interest.

6.3 Information Security Technology

Technology is available that can make these efforts more effective. Digital signature technology can assure customers that renovated software and compliant product versions actually originated where they say they did. Encryption can assure that only the intended recipient of software can read it. Firewalls and access control systems can help to limit online access to authorized individuals. Use digital signatures to assure that renovated software comes from the renovator and that new commercial versions come from the vendor.

6.4 National Priorities and Contingency Planning

Further leadership is needed to establish national priorities and contingency plans. Understanding the reliance of the critical infrastructures on automation will help in this effort. The Year 2000 problem is large enough and widespread enough that there are credible risks that some essential systems will fail, whether or not they are attacked. The Government should establish a national process to identify the most critical systems in each community, presumably including many in the critical infrastructures but not limited to these, and develop or update contingency plans to assure that failures, when they occur, will be as orderly and confined as possible. System with safety, environmental or property protection functions, as well as vital human services, should be addressed first.

6.5 Emergency Response Preparation

For information processing, the Year 2000 has many characteristics of a “planned natural disaster.” Throughout the critical infrastructures, measures are already being taken to alleviate the widespread problems that could arise. Because the timeframe and type of failures due to the Year 2000 problem are both more predictable than in a typical disaster, this problem gives those charged with the various response and recovery capabilities an unusual opportunity to measure their effectiveness, it offers a sort of a “dry run” of such mechanisms across the country. The critical infrastructures should have mechanisms in place to support study and review of the effectiveness of their response after-the-fact. Not only response teams, themselves, but teams to evaluate the performance of the response teams, and statistical collection channels to assemble maximum information should be readied to take advantage of this unique opportunity.

6.6 Focus on Safety-Critical and Property-Critical Systems

The Office of Management and Budget, in reviewing progress of Federal Government agencies addressing the Year 2000 problem, had some disturbing news to convey. It found that the Government agencies most behind in their efforts include several that own key systems with missions to protect human safety and property: NRC (nuclear power plant safety), Transportation (FAA), Justice (NCIC), Commerce (weather systems).

The Nuclear Regulatory Commission may still have far to go in addressing its internal systems, but it has suggested to its licensees (power companies) that they examine their uses of computer systems and software well before the turn of the century and that they especially consider reviewing those programs that are used to meet licensing requirements or that have safety significance. One example the NRC cites is that computer software used to calculate radiation exposure doses may fail, leading to incorrect calculations of doses received by employees, or incorrect exposure times, and so resulting in incorrect planning of treatment of affected employees.

The New York State Government has taken action that may serve as a model for others. In assessing its Year 2000 problems, it has come to believe that hospitals, schools, power supplies, telecommunications, banks, and stock markets may all be significantly impacted in a massive disruption of the New York City infrastructure during the first weeks of January 2000. As a result, the Governor has banned all non-essential IT projects until Year 2000 renovations have been completed. Other government and industry bodies with responsibility for protection of life and property might do well to take heed of this leadership.

7.0 Conclusions

The risks to the critical infrastructures that result from the Year 2000 problem are significant and challenging. We know many of the solutions, but it will still be a large undertaking to apply them across the vast range of systems on which the critical infrastructures depend. To succeed, we will have to take best advantage of all of the techniques, tools, and personnel available, minimizing exposure to new vulnerabilities and risks to the extent that we can. With leadership and a great deal of hard work, the critical infrastructures can meet this challenge, arriving in the new century better prepared to meet the challenges that await us there.