Wireless Systems Security

EE/NiS/TM-584-A/WS

Bruce McNair
bmcnair@stevens.edu

Week 5:
Still More Security Topics

In this class, we will deal with still more security-related topics, and will introduce the security assessment process that will be used for the remainder of the course.

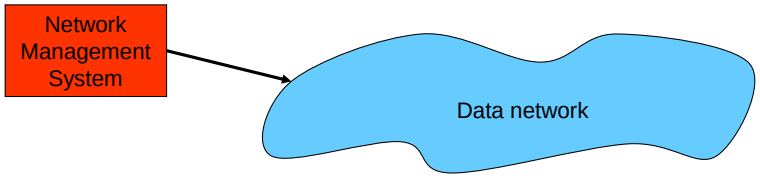
Some Important Topics in Information System Security

- Minimum privilege/minimum functionality
- Compartmentalization/Containment
  - Separation of Responsibility
  - Dual Controls
- Security Perimeters
- Trustworthiness/Design Correctness
- Single-points-of-failure/Choke-points
- Covert Channels
- Inference
- Implicit vs. Apparent Security

This is a list of several topics that are important to examining the security of a system or in designing a new system. We will discuss each of them in the next several slides.

Minimum privilege/Minimum functionality

- Network Management System



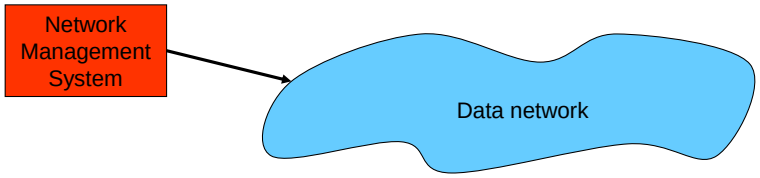
- Applications running on NMS have ultimate control over operation of data network
  1. What capabilities do users really need to have to perform their job?
     Do users need to be able to monitor traffic on the network?
     Including (potentially) sensitive user traffic?

The concept of minimum privilege or minimum functionality is an important one that is frequently overlooked. Consider the design of a network management system. By design, this system has access to the heart of the system that is being managed. It may be necessary to monitor facilities for failure, control routing in the network and modify settings of network equipment. Still, one should examine what the management system is allowed to do and what features it should have.

As an example, consider the NMS that the phone company uses to test and repair your telephone line. They certainly need to be able to access the line during normal operation to be sure it is operating correctly. Perhaps your home phone line is continuously busy for 8 hours while you are away and you want to make sure a line is not down. You might request an operator to “verify busy” on the line. However, do they need to be able to listen in on a conversation? Or would it be sufficient to know that a conversation was taking place on the line? Giving the operator the ability to distinguish between an open line, hum, data tones, or a voice conversation without actually being able to understand what is being transmitted is one example of minimum privilege. Although the NMS might be designed with the best of intentions, by minimizing the capabilities to the minimum that is needed prevents an unnecessary security hole. **Discussion topic**: Suggest some examples of minimum privilege in electronic or physical systems you have encountered, or some examples where this concept is not enforced which has or might lead to abuse.

Minimum privilege/Minimum functionality

- Network Management System



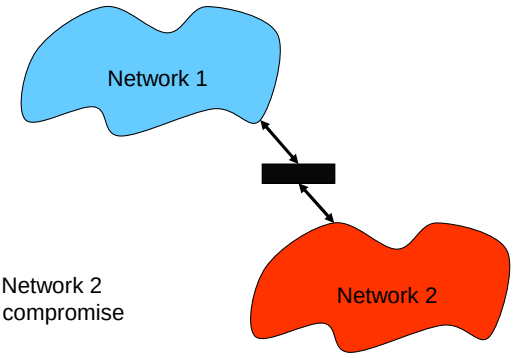
- Applications running on NMS have ultimate control over operation of data network
  1. What capabilities do users really need to have to perform their job?
     Do users need to be able to monitor traffic on the network?
     Including (potentially) sensitive user traffic?
  2. What features does system really need to enable it to operate?
     Does NMS application code get compiled on NMS or is it downloaded?

A related topic to minimum privilege is minimum functionality. Consider the production NMS that is in the field. Assume that it is built on a UNIX platform and has the application code needed to realize its requirements. Does it make sense to have a C compiler installed on the system? The operators are not the developers of the system and do not need to be modifying the code on the fly – the developers will release new versions of the application as needed. One might ask, why not include the compiler on the system – after all, it comes with the operating system. To remove it would take some minimal effort. And, just in case there was some unforeseen reason to make field modification, why not include the NMS application source code?

Consider this scenario – despite the best intentions of the designer, there is a security hole in the operating system that allows a hacker to gain access. If all that was running on the system were the NMS application, they would only have the ability to do only what the application was allowed to do. If, however, the hacker has the source code and a compiler on the same NMS platform, they may be free to modify the system to their convenience. For instance, if the telephone line monitoring function from the previous slide were masking the conversation, the hacker could modify the code to eliminate the masking, thus creating a remote wiretap capability. Without source code and, especially, without a compiler, the hacker's job is much harder. They must not only attack the system, but they have to bring their own burglary tools with them, downloading the code they have compiled elsewhere.

Compartmentalization/Containment

- Firewall



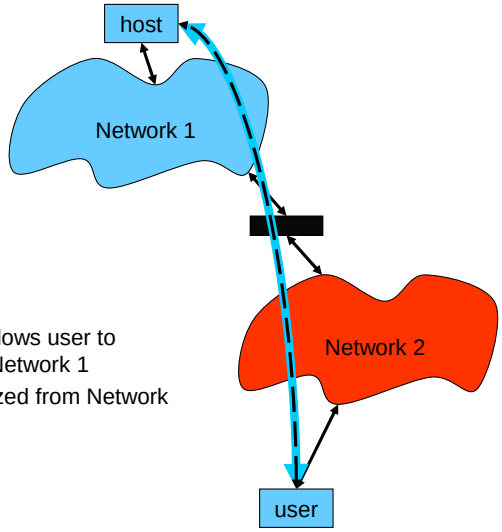
- Potential compromise of Network 2 should not be allowed to compromise Network 1
- Partitioning of traffic, namespace, services
- Entities on Network 1 may not even be visible to users on Network 2

Next, consider the concept of compartmentalization and containment. The network firewall is one example of a containment device. It is not possible to protect all networks, particularly if one of them is not under your control or if external users are allowed to access it. To protect a private subnetwork, e.g., a corporate network, from the less protected network, e.g., the Internet, a firewall limits what messages and controls may be sent across the boundary.

As an analogy, consider the protection of classified military information. Besides the well-know hierarchical separation of information according to sensitivity level (Confidential, Secret, and Top Secret), there are non-hierarchical classes of information known as Compartmented Information. Even people cleared to access Top Secret information related to, for instance, cryptography, are not necessarily allowed to access information related to the design of nuclear weapons, even information classified at a lower level. By creating separate compartments of information, and controlling who is allowed to access which compartment, the information can be more tightly controlled.

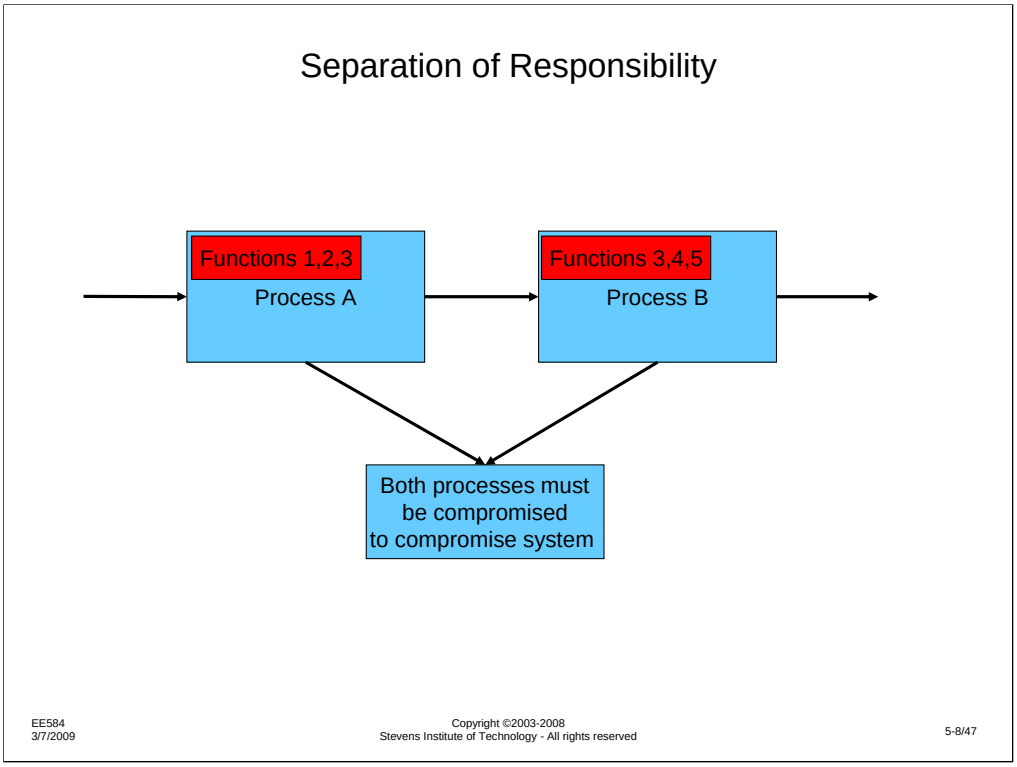
Compartmentalization/Containment

- Virtual Private Network



- Encrypted VPN 'tunnel' allows user to appear to be virtually on Network 1
- Tunnel is compartmentalized from Network 2

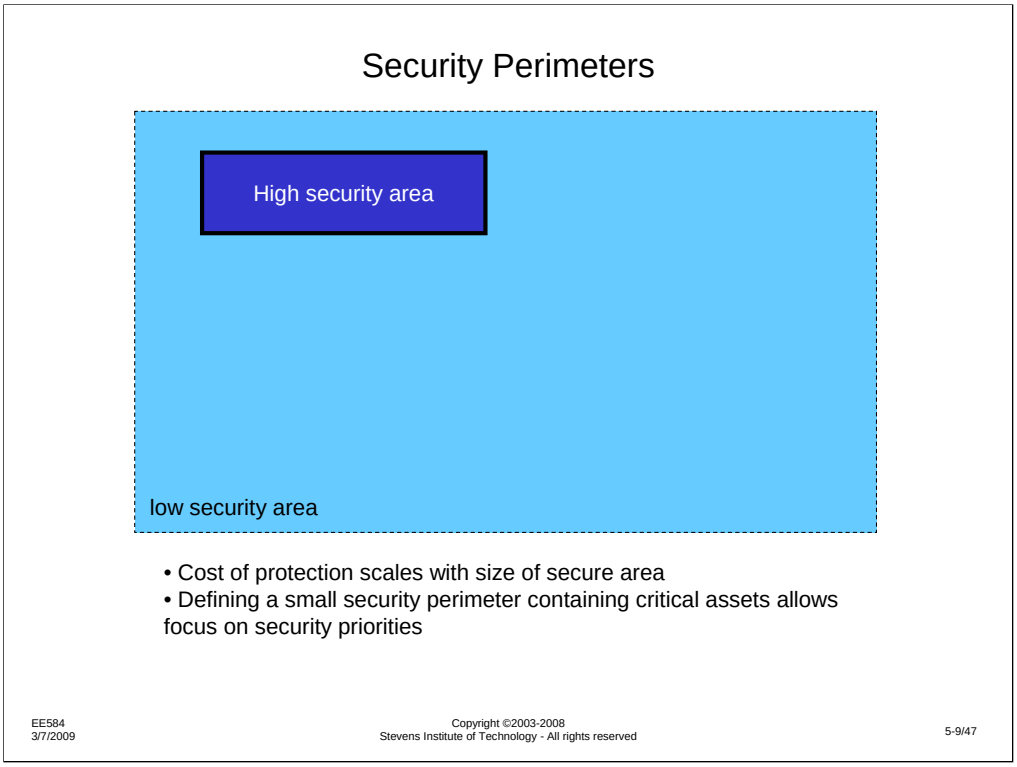
Sometimes, it is necessary to allow certain users who are "outside" the firewall to access information "inside" the firewall. For instance, a user who is traveling and needs to access information on their corporate network may need to have a means to bypass the firewall. This is the function of a virtual private network. By encrypting the information as is leaves the user's computer, a secure "tunnel" is created through the insecure network. The firewall can allow these sessions to bypass the firewall, since only a trusted user with the proper encryption can generate properly formatted messages in the tunnel.



Another concept in the design of a secure system is separation of responsibility. To use an example from a business operation, consider the process of ordering and receiving goods. The person who places orders and the person who handles the receipt of orders must be different people. If this were not the case, it would be possible for them to add items to an authorized order, which they could then separate from the order when it was received. Since the person who placed the order and received the items had everything they wanted, they would probably not look more closely at what was actually paid for, and the people who do accounting and auditing would have no way to discover that extra items were being ordered.

The basic control that is working here is that, while it might be possible to have one person cheating in an organization, when two need to collaborate to engage in fraud, the chances of getting agreement and escaping detection are reduced.

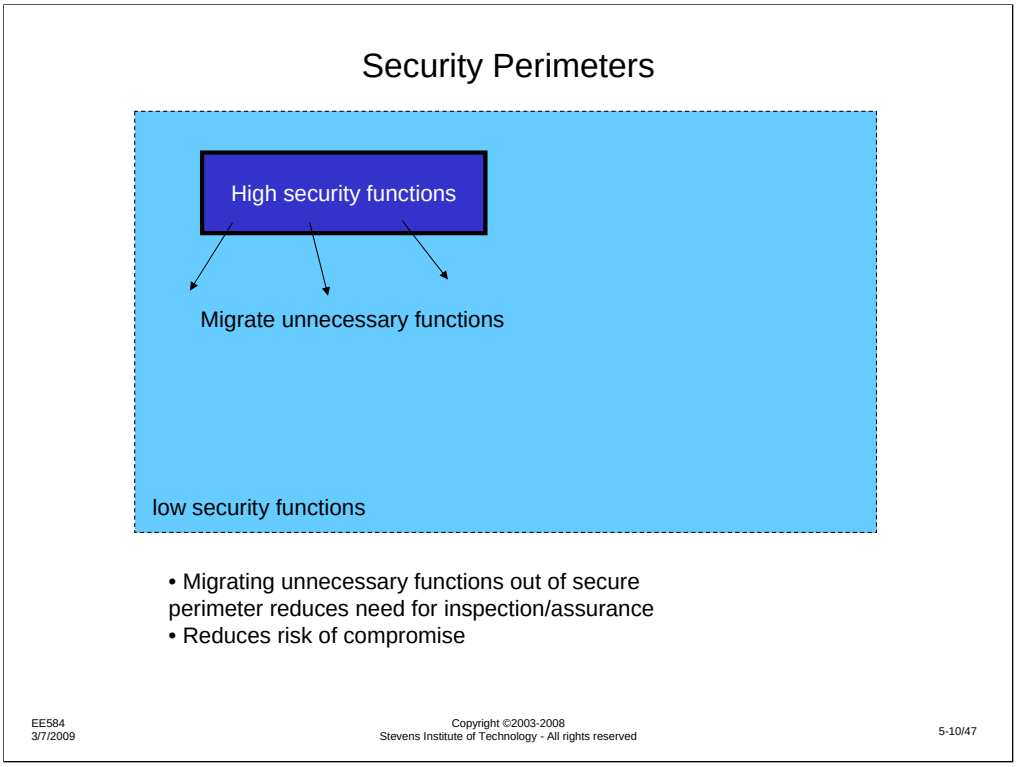
In electronic systems, any opportunity to move functions in to separately designed and controlled processes reduces the chances of designer or user compromise.



I like to look for physical analogies to abstract systems where ever I can to better understand the operation of the abstract system. I find this to be particularly useful for the security of complex abstract systems. In discussing security perimeters, let me use a physical analogy: one might have a fence surrounding their property. This is not a very powerful security control, but it is relatively inexpensive and provides *some* level of protection. Any items that are left on the property are not likely to be very valuable, perhaps some lawn tools or lawn furniture. Inside the property is a house, which has locked windows and doors, plus an alarm system. This is a "harder" target than the land itself, since a door would have to be forced open or a window broken to get into the house. Inside the house, more valuable items, like televisions, cameras, and some cash are exposed, but only to someone who can get into the house. The most valuable assets, like expensive jewelry, large amounts of cash, etc., are stored in a very well protected safe.

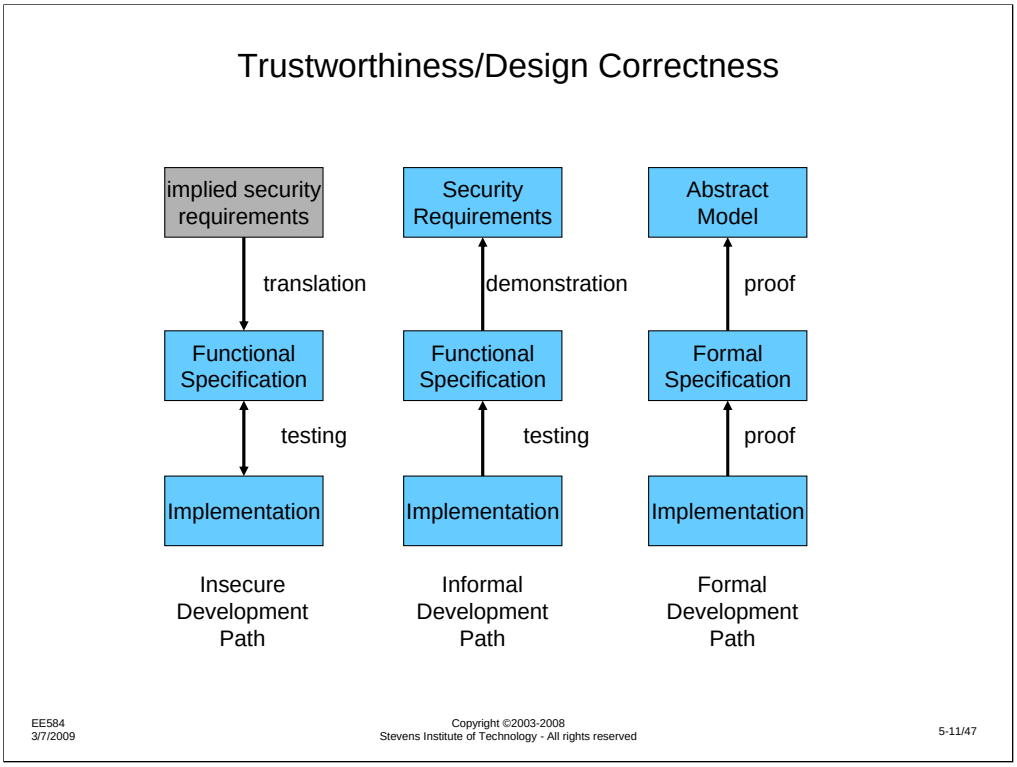
Each level of protection defines a security perimeter – the boundary of an area that has some level of protection. Interior security perimeters are smaller and easier to control, and protect more valuable assets than the exterior perimeters.

In designing a system, rather than trying to provide a high level of protection across a large area, it is useful to define small high security areas. Like the physical analogy of a home and property, the bigger the perimeter is, the more it costs to build and maintain it. There are highly secure facilities that are built like bank vaults and are the size of a house (so-called SCIFs – Secure Compartmentalized Information Facilities), but the investment needed to build such facilities limits their use to only the most important applications.



One design consideration for any secure system is to migrate functions that require less security out of the high security area. Again, using the physical analogy, I could keep my TV set in a safe if I were afraid someone would break into my house to steal it. However, this would significantly increase the size of the safe I needed. In addition, since I might want to give others in my house access to the TV, I would either have to leave the safe open much of the time or tell them how to get access. Either way, this lessens the security of the high security area.

Often the system being designed must be inspected to insure compliance with security design or operational requirements. If the bulk of the design does not need to meet security requirements, it doesn't need to be inspected, again, reducing system costs.



The mention of security inspection of a design leads directly to an important area of the design of secure software systems, known as Trusted Systems. If we consider Trusted Systems to be the extreme case of correctly designed systems, we can discuss several design methodologies and their level of security.

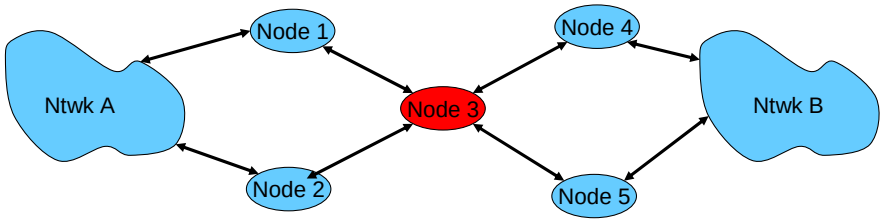
Many systems are designed with a very insecure development process. There are no security requirements or the requirements are implied by other system requirements. A functional specification for the system is written by translating the requirements from one written document to another. The implementation is supposed to follow the functional specification, but the correctness of the design is measured by testing for bugs.

A more secure, but still informal, development process explicitly state security requirements. While the functional specification is again a written document, satisfaction of the security requirements must be demonstrated rational argument. Again the implementation is tested against the functional specification.

The most secure development path, used for trusted systems, is a formal path. Here, the security requirements are stated as an abstract model of how a secure system should be have. A formal system specification is developed from the model and the correspondence of the specification to the model must be proven mathematically. Likewise, the implementation is developed from the specification, but the correspondence to the specification must, again, be proven mathematically.

This formal design process leads to a system whose behavior can be reasoned about in a formal structured environment, rather than using informal verbal assertions. While this is likely to result in a more secure system, the cost of proof at each stage limits its utility.

Single-points-of-failure/Choke-points

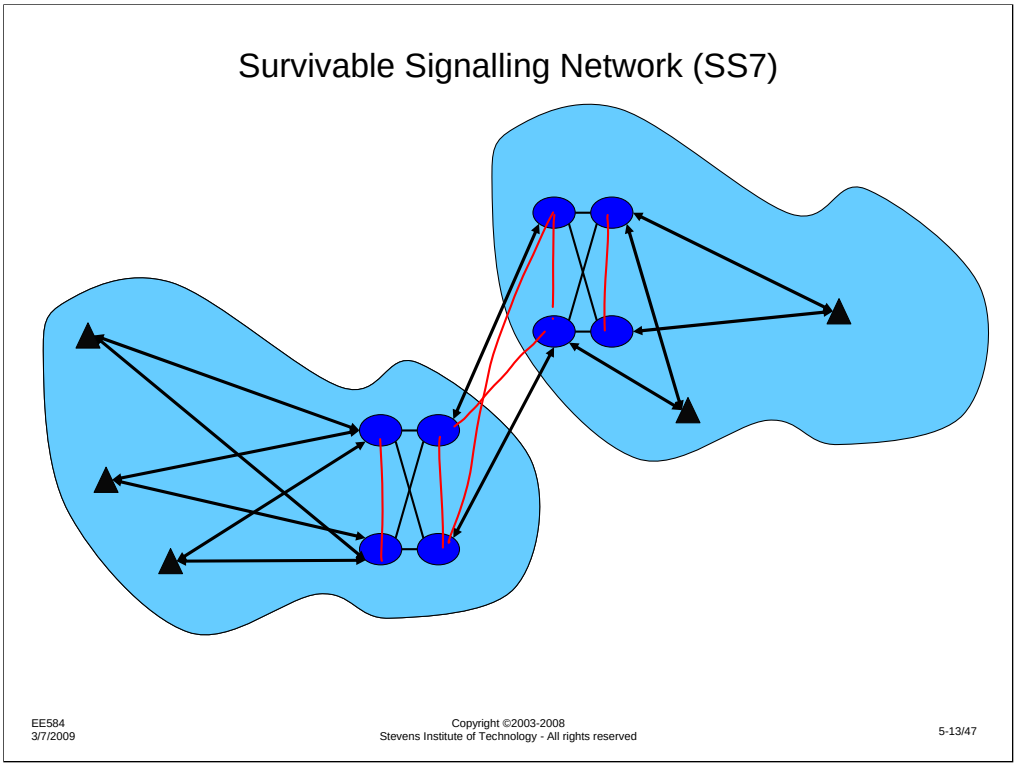


Node 3 is a single-point-of-failure (or attack) and a choke-point

Another important topic in the design of a secure system is that of single-points-of-failure, so-called choke-points.

In the system shown above, there are two paths from Network A to Network B, but both must rely on Node 3. If this node were to fail, the two halves of the system would become disconnected. If the network were heavily loaded with traffic (by chance or by intent), overload at Node 3 would impact all communications between the two halves. For this reason, Node 3 is a choke-point or single-point-of-failure. I have illustrated the case of a nodal choke-point, but it is also possible to have a link that provides a single-point-of-failure.

A classic case of improper system design existed for the control circuits for a particular nuclear reactor. While the designers obeyed the requirement to have separate, redundant, control circuits to manage critical functions (e.g., the cooling rods), the design flaw was that both control circuits ran through the same conduit. This was discovered when a technician was using a match to illuminate the conduit for inspection. The insulation on one of the wires ignited and a fire started in the conduit. Despite the separate control circuits, the commonality of their path led to a loss of control of the reactor. The lesson is that ALL aspects of a system link or node must be considered. Having two redundant controllers doesn't help if both depend on the same power supply.



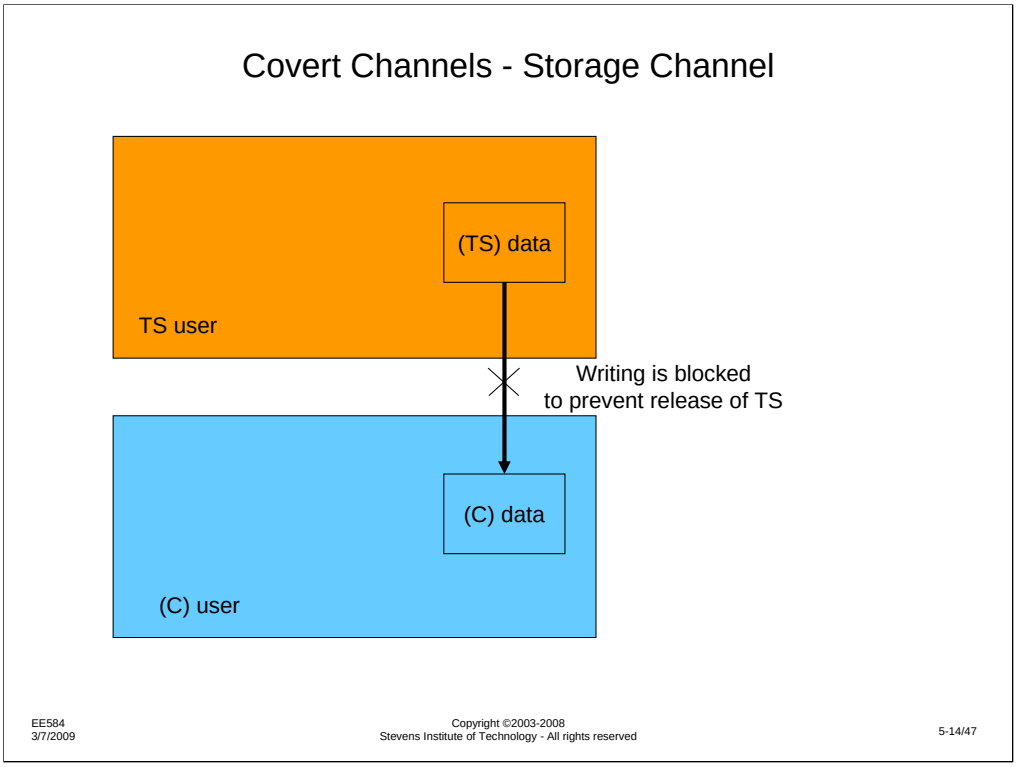
As one of the best examples of a highly redundant system, this slide illustrates a portion of the design of Signaling System 7, the control system that manages call control and enhanced features for the AT&T long distance network, as it existed in the 1980s.

The US was divided into 7 geographical regions. Two are shown here. Each region had a number of toll switches, shown as triangles. When a switch needed to route a call or request a special feature, it communicated with the signaling system through a Signaling Transfer Point (STP), shown in as a blue oval. The group of 4 STPs represented an STP complex, with two STPs in one city and the other two in a different city. Either of the two STPs in a city could handle the load for the other if it failed, and either of the two STP sites (cities) could handle all the traffic if something brought down both of the other STPs. For instance, an earthquake or other major disaster might wipe out an entire site, but since the other pair of STPs in the region were geographically separated from the first site, a common disaster would not be likely to disrupt both.

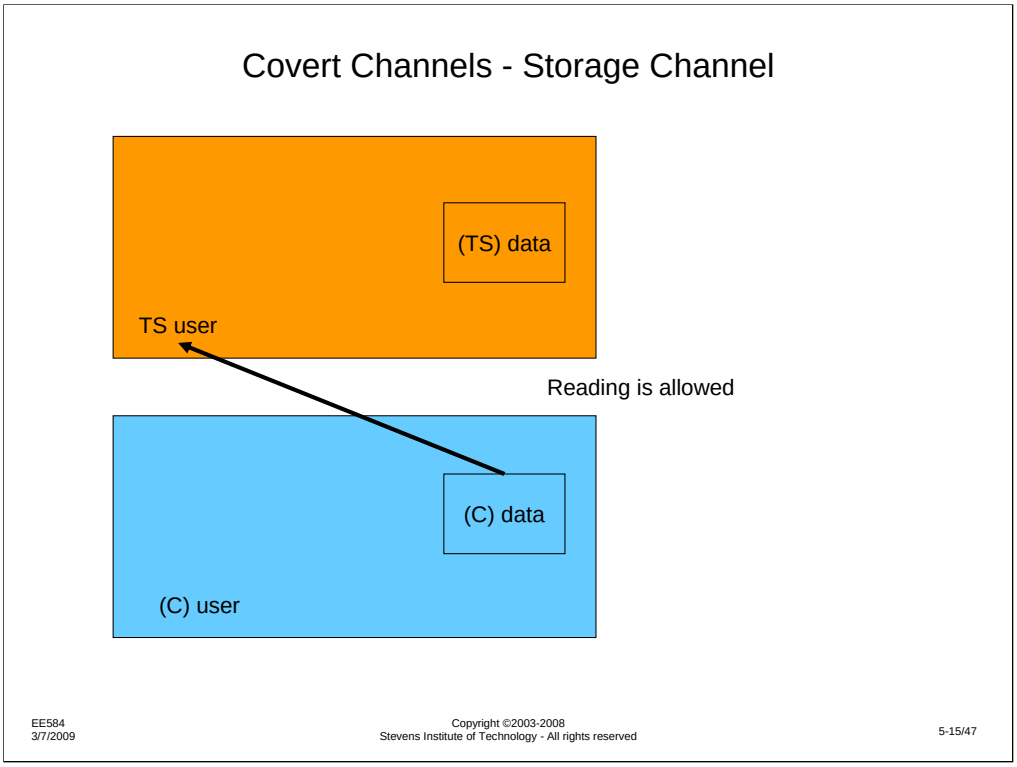
All the STPs in a region were fully interconnected and every switch in region was connected to both STP sites.

From a hardware perspective, this system provides multiple levels of redundancy, decreasing the likelihood that any individual failures or collective disasters would bring down the signaling system.

Discussion topic: On January 15, 1990, a massive failure of AT&T's SS7 due to a subtle software bug (introduced in the process of trying to make the network more robust) brought down major parts of their long distance network for several hours. See what you can find out about this issue.

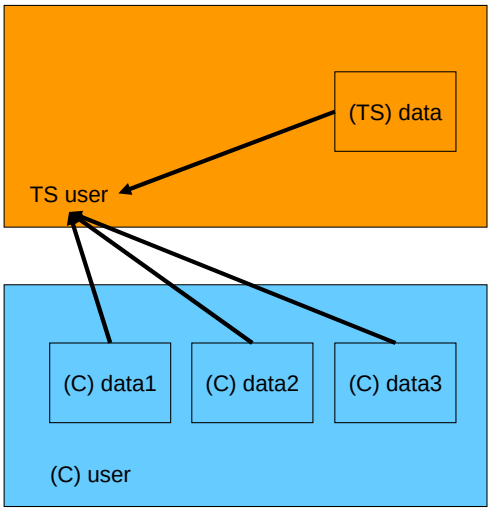


Another topic in the design of secure systems is the idea of a covert channel. Assume we have designed a system that implements a security policy that allows users at different security levels to share the system, but prevents them from compromising information. A user at the Top Secret level (the highest classification level) cannot write data to a Confidential file (the lowest classification level). If the TS user had been allowed to write any information, there would be the potential for them to compromise the security.



On the other hand, the TS user is allowed to read C data, since it is at a lower level than they are cleared for.

Covert Channels - Storage Channel



Consider a DB with record locking:

TS: Open1, Open2
C: Open1(blocked), Open2(blocked),
Open3(succeed)
Until(Open1) {}
Close3, Close1
TS: While(!Open3){}
Close1, Close2, Close3
// TS just sent a “0”

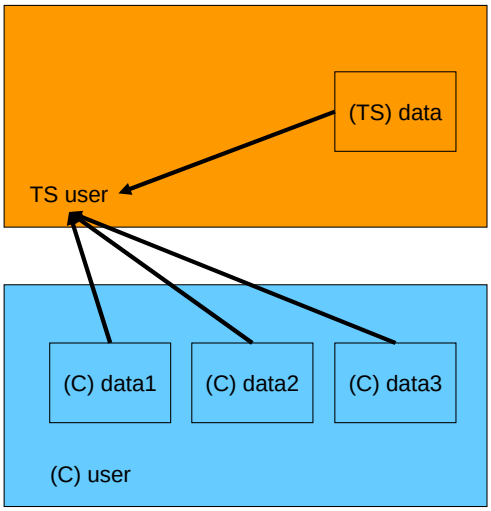
TS: Open2, Open3
C: Open1(succeed)
Until(Open2){}
Close1, Close2
TS: While(!Open1){}
Close1, Close2, Close3
//TS just sent a “1”

The covert channel exists when a higher level user can operate within their limits and, by engaging in what seems like an innocuous behavior, compromise information.

Here, the TS user is accessing information from a database that locks records when they are being read. This is often done to prevent information from getting into an ambiguous state. If the data were not locked when the TS user had it open for reading, the C user might modify the data by writing to it, something they are allowed to do. If the TS user combined two pieces of data that were modified out of order, they might reach an invalid conclusion about what was in the database. By locking records from being written while someone is reading them, the database can be made more consistent.

However, this creates the potential for a covert channel. If the TS user reads a series of records, he can signal to the C user by virtue of the order in which he reads. The pseudo-code listing at the right illustrates how the TS user can signal a 1 or a 0, merely be choosing the order in which he reads data.

Covert Channels - Storage Channel



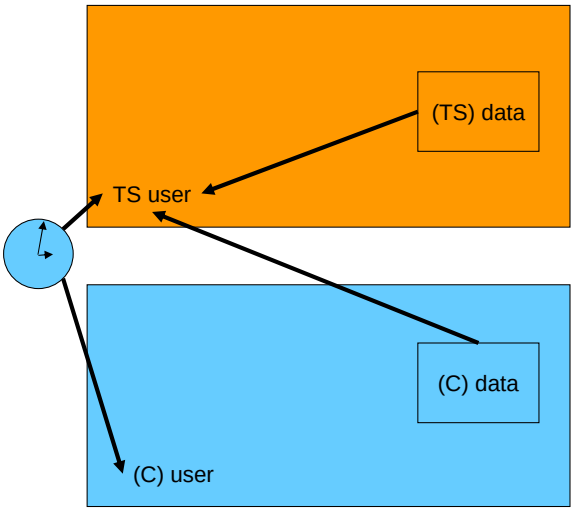
This is an obvious covert channel, with wide bandwidth (on the order of the open/close speed of a data record)

Arbitrary covert channels can be exploited with P(detection) related to utilized bandwidth.

Covert channels are generally measured in terms of the bandwidth they would allow to pass. Here, it takes several read and open cycles to pass one bit of data, so this is a low bandwidth channel, compared to the data transfer rates to the files, but a high bandwidth channel when one considers how fast file accesses can occur. But even a low bandwidth channel is better than a 0 bandwidth channel, if that is the alternative.

Discussion topic: One might be able to detect the use of this covert channel by watching the activities of the two users. How might one be able to trade bandwidth for detectability, allowing a very low bandwidth, essentially undetectable covert channel?

Covert Channels - Timing Channel



Synchronized access to lower level data is used by TS user to convey TS data to lower level user

Note: "TS user" might be Trojan Horse operating on behalf of TS user

The covert channels discussed above are examples of "storage channels." The storage of information *only* is used to communicate. There is another class of covert channels known as timing channels. If both the TS and C user have access to a common timing source, that clock can be used to simplify the protocol and storage access needed. Time could be broken into intervals of length T. Reading during the first half of the interval signals a 0, reading during the second half signals a 1. The bandwidth of this covert timing channel is 1/T. If T were 1 millisecond, this would be a 1 kbs/ channel.

Inference

- Example 1:
  - Stevens has used Social Security Numbers as Student IDs for many years. Grades were posted by SSN. Name/SSN are never displayed together publicly
  - AT&T Bell Labs (That name carbon-dates the age of the issue) switched from Payroll Account Numbers (PANs) to SSNs as employee identifiers
  - The POST employee directory was searchable by PAN or SSN, but did not display them
  - Individual privacy can be compromised by SSN fairly easily
    - How can two relatively secure systems be played against each other?

The concept of inference is related to using several pieces of separately innocuous information to *infer* sensitive information. This is easiest to see with a (real) example.

We have two individual systems that are each operated in an acceptably secure manner. The problem is that one can play information from one system against the other to compromise information that neither system would compromise by itself. The 1990s Stevens procedure of revealing complete SSNs without names in a public area did not compromise anyone's SSN, but gave an attacker a list of valid numbers. Likewise, the AT&T Bell Labs directory allowed search by SSN, which didn't compromise anyone's privacy, but did allow a search by SSN.

Inference

- Example 1:
  - Stevens has used Social Security Numbers as Student IDs for many years. Grades were posted by SSN. Name/SSN are never displayed together publicly
  - AT&T Bell Labs (That name carbon-dates the age of the issue) switched from Payroll Account Numbers (PANs) to SSNs as employee identifiers
  - The POST employee directory was searchable by PAN or SSN, but did not display them
  - Individual privacy can be compromised by SSN fairly easily
    - How can two relatively secure systems be played against each other?
      - A large percentage of part-time Stevens EE/CpE & CS graduate students have historically come from AT&T/Bell Labs
      - Obtain the SSNs of Stevens EE/CpE/CS graduate students from posted grades
      - Search the POST data base by SSN to identify individuals.
        - » Individual privacy is compromised by the joint weakness of two systems that are relatively secure separately

The compromise is due to the fact that a significant portion of the Stevens graduate school population worked for Bell Labs. If only a small handful of individuals were common to both populations, the probability of a successful attack would be rather small. If 100% of the graduate students were Bell Labs employees, the probability of success would be 1. In fact, the percentage may have been 30-50%, depending on the class, so the odds of finding a person's valid SSN could be reasonably good.

This example serves to show that when designing a system that processes sensitive information, one must examine the system operation in the larger environment. Knowing that one system in and of itself is secure does not guarantee that when combined with another system the combination is secure.

Discussion topic: Does this mean that we have to consider every system that was ever built or ever might be built when trying to decide if our system is secure?

Inference

- Example 2
  - ref: Dorothy Denning, “The tracker - inference issues in database security”
  - Database contains User names, department, ages, salary, etc.
  - Individual records are protected against search by low level users: only trusted users may read separate records
  - Aggregate database statistics may be viewed by lower level users, e.g.,
    - “Show average salary of male employees”
    - “Show number of users earning more than $100k”
  - Database security system prevents lower level user from retrieving data sets or statistics based on small number of records

As another example of the inference problem, actually the original context, consider a database that allows one to search for certain properties of data. For instance, a company might have a personnel database with specific information about every employee – name, salary, position, etc. Some people, e.g., Human Resources, can view the entire database and make changes. Others can only see summary information, and not user specific information.

Inference

- Example 2
  - ref: Dorothy Denning, “The tracker - inference issues in database security”
  - Database contains User names, department, ages, salary, etc.
  - Individual records are protected against search by low level users: only trusted users may read separate records
  - Aggregate database statistics may be viewed by lower level users, e.g.,
    - “Show average salary of male employees”
    - “Show number of users earning more than $100k”
  - Database security system prevents lower level user from retrieving data sets or statistics based on small number of records
    - The DB Inference problem:
      - Attacker creates a series of queries that have a small sample size in their intersection
      - Unless DB security system can assess sample sizes for all possible combinations of queries user has ever made, it is subject to an inference attack.
      - Even if it does this, innocent queries can be denied because they MIGHT create inference vulnerability

Now, think of the records in the database as marbles in a jar. I cannot examine an individual marble, but I can set criteria to gather information about sets of marbles. Each time I search the database, or partition the marbles into two sets (those that meet the criteria and those that don’t), I am selecting a subset of the individual components. If I can ever get to the point that there is one unique member in the intersection of the set of partitions, I have compromised the information the database was supposed to protect.

The DB security system can try to examine the queries I am making to determine what I might be looking for, but this is a hard problem. Worse, there might be cases where innocent users are making reasonable queries, but the DB security system thinks they are getting too close to one individual.

Implicit vs. Apparent Security

- User chosen passwords are notoriously insecure, often subject to dictionary attacks. Machine generated passwords are suggested as an alternative. Which is more secure?
  - Password scheme1:
    character(k) = {a-z, 0-9, !@#$%^&*() } (46 symbols)
    PW = kkkkkk
    sample passwords: a5&98!, tfhe5&, 3thp1,
  - Password scheme2
    vowel(v) = {aeiou}
    consonant(c) = {bcdfghjklmnpqrstvwxz}
    PW = cvcvcvcvcv
    sample passwords: ponihavoka, risehipeta, tojifatese

Many systems are designed to create the appearance of security without really adding much security. Let's use the example of password security to discuss this concept.

Since users rarely choose good passwords, one concept has been to assign them machine generated passwords. I have used a network that took this to an extreme – they created completely random strings of characters and assigned them to users as their old passwords expired. Obviously, these passwords were very difficult to remember.

Another system I used had a similar machine generated password, but picked passwords that were pronounceable – they were made up of alternating consonant and vowel strings, as illustrated at the bottom of the slide.

Since, at one point in my Bell Labs career, I was responsible for looking at the security of various system designs, I got into a heated discussion with the person responsible for managing the first password scheme. At one point, a comment was made like: "Just look at these random passwords. No one would ever guess a password like a5&98!, so it has to be more secure."

Implicit vs. Apparent Security

- User chosen passwords are notoriously insecure, often subject to dictionary attacks. Machine generated passwords are suggested as an alternative. Which is more secure?
  - Password scheme1:
    character(k) = {a-z, 0-9, !@#$%^&*() } (46 symbols)
    PW = kkkkkk
    sample passwords: a5&98!, tfhe5&, 3thp1,
    Total password space: 9,474,296,896
  - Password scheme2
    vowel(v) = {aeiou}
    consonant(c) = {bcdfghjklmnpqrstvwxz}
    PW = cvcvcvcvcv
    sample passwords: ponihavoka, risehipeta, tojifatese
    Total password space: 10,000,000,000
- Apparent complexity of first scheme suggests higher security, but ease of memorization of second makes passwords more secure

EE584
3/7/2009

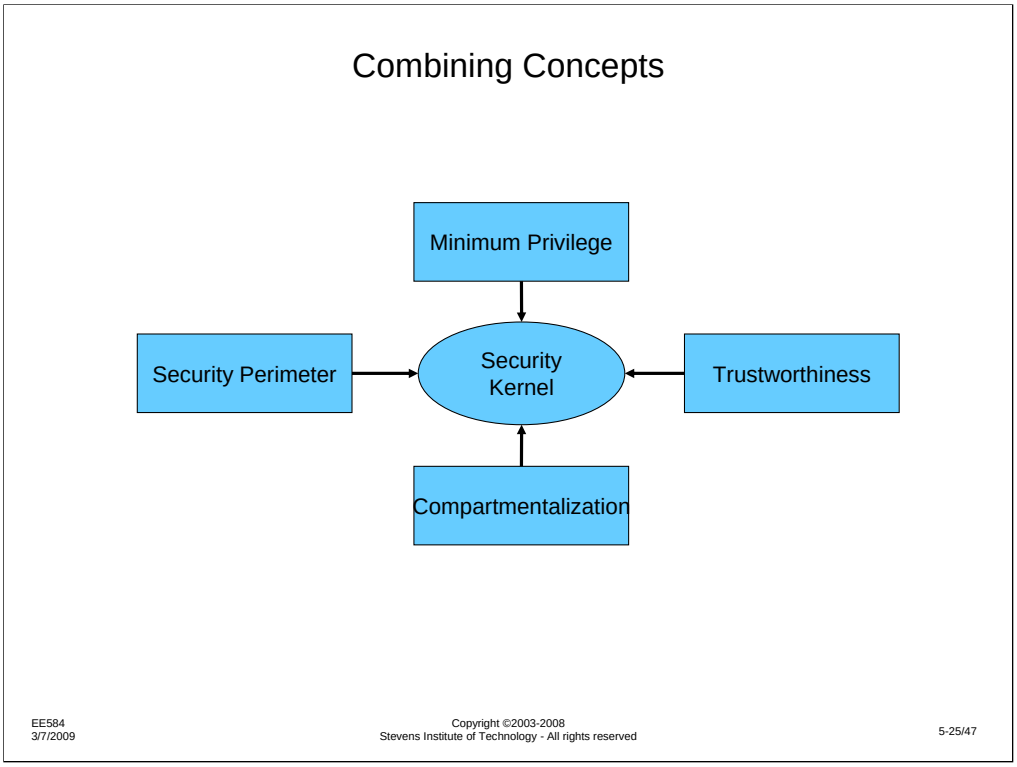
5-24/47

We can look at these two schemes objectively – which generates a larger password space and which is really likely to be more secure. As I have shown, the complete space of passwords generated by the first scheme is about 9.5 billion. Meanwhile, although they may look less secure, the second scheme has a total space of 10 billion passwords, slightly more and therefore slightly more robust against an exhaustive attack.

On the other hand, there is the issue of how a user might actually deal with the two passwords. Since the human mind seems to work best with patterns and templates, the first password is hard to remember, since it makes absolutely no sense. The second password fits the pattern of "words are pronounceable and that could possibly exist, but no one has used them yet," so there is a place to store them in our memory. It is much less likely that a user will write down the second password. They have no choice but to write down the first, creating a possibility for loss.

There are many situations we encounter where procedures are put into place to create the illusion of security without really creating any real security. Likewise, there are some systems that are quite secure, but their true security capabilities are hidden from casual view.

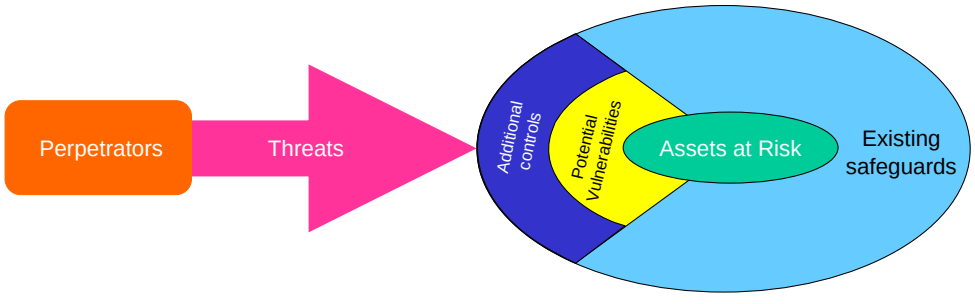
Discussion topic: Suggest some examples of this implicit versus apparent security. Hint: the current security procedures for US and certain other countries are full of these cases.



Several security topics have been introduced. If we combine them, we can discuss another concept – the idea of a security kernel. Think of the security kernel as the concentration of all the security functions in a system. By properly designing this piece of the system, paring it down to the minimum required functionality, while ensuring that all the security capabilities are controlled by or incorporated in this piece of the system, we have the best chances to design a system that will stand up to attack. As we will see later, the security threat is continually evolving, so it is best to have one place and only one place that needs to be modified to keep up with the evolving threat.

Security Assessment

- The structure:



- The process:
  - Structured brainstorming

EE584
3/7/2009

5-26/47

We have discussed the concept of security assessment in Week 3, using the structure shown above. If the concepts are not fresh in your mind, go back and review them, since we will be using these concepts for the rest of this class and for the future classes.

The process that we will be using to examine security in the context of a particular wireless system will be a structured brainstorming process. This is a free-flowing interactive process that has worked well in face-to-face interactions. I will be using the capabilities of WebCT to go through the same process in this class.

STOP HERE

READ THE NOTES BEFORE PROCEEDING

I suspect that some people may just skim through the slides without reading the notes. If we were meeting face to face, I would be able to manage this by showing each slide in order, with the proper discussion on each. Since I can't do that in an on-line lecture, I want to be sure you get the maximum advantage out of the material, so I will ask you not to peek ahead, but follow each slide as I have designed it. Since you are reading these notes right now, you obviously have followed the instruction of this slide. If you didn't follow the instructions and are reading this after you have seen all the slides, their effectiveness has been reduced.

Brainstorming

- True brainstorming occurs in two phases:
  - Free flowing idea generation without any analysis
  - THEN, analysis to weed out the useful ideas

As mentioned above, we will be using the process of structured brainstorming to discuss security concepts from this point on.

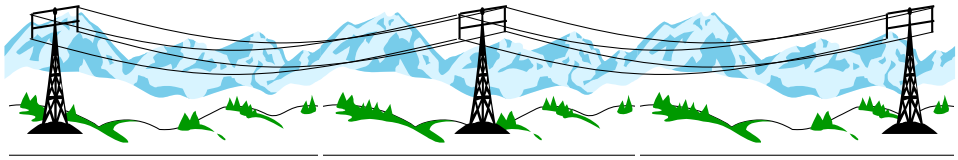
True brainstorming is an open-ended process. There is no analysis during the brainstorming phase. To do so would stifle the free flow of ideas. Instead, assume that ANYTHING is allowable during the brainstorming process. Any idea is considered a valid suggestion. Only after the ideas have been thrown out are ANY of them analyzed. At that point, we can begin to weed out the wacky suggestions and focus on the useful ones. In the next two slides, I will suggest why this process: Free flowing ideas without analysis, followed by analysis, is the best way to get the maximum exchange of ideas.

Before we start this, let me suggest why we want this free flow of ideas – security attacks are generally through holes we didn't know existed. If we knew about them, we would have fixed them, right? If we have designed a system to meet certain requirements, we are generally "locked-in" to the ideas we have implemented, making it difficult to see other approaches or problems with the current approach. For instance, it is well recognized that a person who has written a piece of code or designed a piece of hardware or assembled a series of parts is incapable of seeing mistakes they have made. They think they followed the instructions or meet the requirements, so they can't look at the result with an unbiased viewpoint. Only by involving another person, or by coming back to the problem after a distraction, can some errors be found.

In this manner, we want to examine security of system without being locked into conventional wisdom about what is possible and what is not. Assume that anything may be possible.

Brainstorming

- True brainstorming occurs in two phases:
  - Free flowing idea generation without any analysis
  - THEN, analysis to weed out the useful ideas



- Ice build up on high tension wires in cold climates needs to be removed to avoid damage due to excess weight/wind load on the wires. How to remove ice?

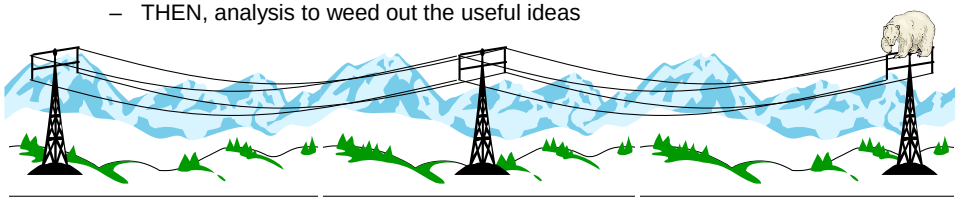
Consider the following (real) problem. In northern areas of the world, ice tends to build up on power lines. As the ice builds up, the wires get heavier and may tend to sag and either touch the ground or break. As the wind blows, the larger, heavier ice-coated wires move more and put additional stress on insulators and towers.

One power company was faced with this problem – they needed to remove the ice before it caused damage, but they needed a method that could be done easily, efficiently and economically. Because the power lines were often in remote areas, they couldn't dedicate crews to go out and remove the ice, since it was a hazardous operation and they had many miles of power lines to maintain.

It was decided that brainstorming would be used as a means to come up with a method of removing ice from the power lines.

Brainstorming

- True brainstorming occurs in two phases:
  - Free flowing idea generation without any analysis
  - THEN, analysis to weed out the useful ideas



- Ice build up on high tension wires in cold climates needs to be removed to avoid damage due to excess weight/wind load on the wires. How to remove ice?
- Brainstorming led to a suggestion to train polar bears to climb the towers to shake the wires, breaking the ice

In the process of brainstorming about a method to remove the ice, one of the participants suggested they train polar bears to climb the towers. The bears could hit the wires, causing them to vibrate and break the ice.

Let's stop for a minute and analyze this suggestion. It is obviously ridiculous. (1) Polar bears hibernate during the winter, when the issue is greatest (2) polar bears are not known for their being able to be trained (3) how could they climb the towers? (4) even if they could climb the towers, they couldn't reach the wires (5) if they were able to reach the wires, they would be electrocuted (6) even if they weren't electrocuted, they could only shake the wires near the towers – the middles of the wires would still be ice-covered.

Nothing about this idea is practical or useful. The person who suggested it has not contributed to the solution of the problem.

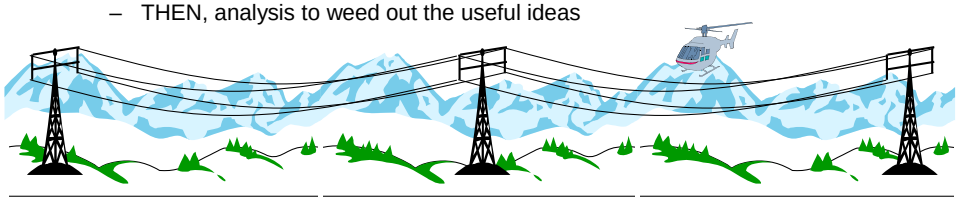
Or have they?

Remember, we aren't supposed to do any analysis at this point.

Why? - See the next slide.

Brainstorming

- True brainstorming occurs in two phases:
  - Free flowing idea generation without any analysis
  - THEN, analysis to weed out the useful ideas



- Ice build up on high tension wires in cold climates needs to be removed to avoid damage due to excess weight/wind load on the wires. How to remove ice?
- Brainstorming led to a suggestion to train polar bears to climb the towers to shake the wires, breaking the ice
- While that idea is not a sensible suggestion, it led to the idea of having helicopters fly over the wires to vibrate them, breaking the ice free.

Despite the fact that the polar bear idea did not solve the problem, it triggered another person to start thinking not of **how**, but rather of **what**. They realized that the important suggestion buried in the polar bear idea was to get the wires vibrating to break off the ice. The second person saw a solution that was practical – fly helicopters over the power lines. Their downdraft would cause the wires to vibrate, breaking off the ice.

What is important here is that by keeping the discussion open, without criticizing or analyzing any of the suggestions, the widest range of ideas could be presented, one of which led to a workable solution.

An Exercise in Brainstorming

- You are inside of a room 10'x10'x10'
- The walls, floor and ceiling of the room are solid concrete
- Embedded in the center of the floor is a steel pipe that projects 1 foot from the floor
- There is a ping-pong ball at the bottom of the pipe
- The pipe diameter is about 1/16th inch larger than the ping-pong ball.

Let's try an exercise in individual and group brainstorming.

First, consider the problem above – you are faced with a ping-pong ball that is inside a pipe, embedded in a floor.

An Exercise in Brainstorming

- You are inside of a room 10'x10'x10'
- The walls, floor and ceiling of the room are solid concrete
- Embedded in the center of the floor is a steel pipe that projects 1 foot from the floor
- There is a ping-pong ball at the bottom of the pipe
- The pipe diameter is about 1/16th inch larger than the ping-pong ball.

- In 60 seconds, think of as many ways as you can of removing the ball from the pipe without damaging it or the pipe; you should at least consider using objects you are likely to able to find in this classroom, but do not restrict yourself to those objects

As an individual exercise, try to come up with as many ways you can think of to remove the ping-pong ball from the pipe without damaging the pipe or the ball.

An Exercise in Brainstorming

- You are inside of a room 10'x10'x10'
- The walls, floor and ceiling of the room are solid concrete
- Embedded in the center of the floor is a steel pipe that projects 1 foot from the floor
- There is a ping-pong ball at the bottom of the pipe
- The pipe diameter is about 1/16th inch larger than the ping-pong ball.

- In 60 seconds, think of as many ways as you can of removing the ball from the pipe without damaging it or the pipe; you should at least consider using objects you are likely to able to find in this classroom, but do not restrict yourself to those objects

- Repeat this exercise using group brainstorming – start with the suggestions from the previous step

I have created a WebCT discussion group: Ping-Pong. I would like you to use that forum to post suggestions to remove the ping-pong ball from the pipe. Post your ideas and build on the ideas of the others' submissions.

Discussion topic: I will speculate that you will see many people posting suggestions similar to your own, but there will be a few that you hadn't thought of. If you get fully engaged in the process, those new ideas will generate ideas that you wouldn't have thought of on your own.

An Exercise in Brainstorming

- You are inside of a room 10'x10'x10'
- The walls, floor and ceiling of the room are solid concrete
- Embedded in the center of the floor is a steel pipe that projects 1 foot from the floor
- There is a ping-pong ball at the bottom of the pipe
- The pipe diameter is about 1/16th inch larger than the ping-pong ball.

- In 60 seconds, think of as many ways as you can of removing the ball from the pipe without damaging it or the pipe; you should at least consider using objects you are likely to able to find in this classroom, but do not restrict yourself to those objects

- Repeat this exercise using group brainstorming – start with the suggestions from the previous step

- Compare the effectiveness of the two techniques (individual vs. group brainstorming) for developing ideas

Discussion topic: I will speculate that you will see many people posting suggestions similar to your own, but there will be a few that you hadn't thought of. If you get fully engaged in the process, those new ideas will generate ideas that you wouldn't have thought of on your own.

Case 1
Terrestrial Microwave RF Telephone Relay System

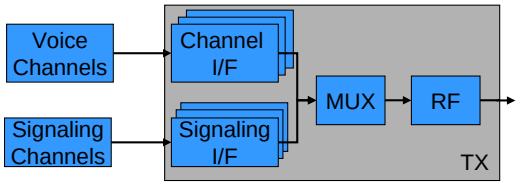
4 GHz
Analog SSB FDMA
Multichannel Voice traffic
CCS signaling
Washington, DC area

Now we begin the second part of this course. For the remainder of the course, I will present case studies of wireless systems for which security has been, is, or might be an issue. For the first case study, I will present the background of the problem and will lead the discussion through the issues. For future case studies, I will present the background and the class will brainstorm about the issues. More on this process later.

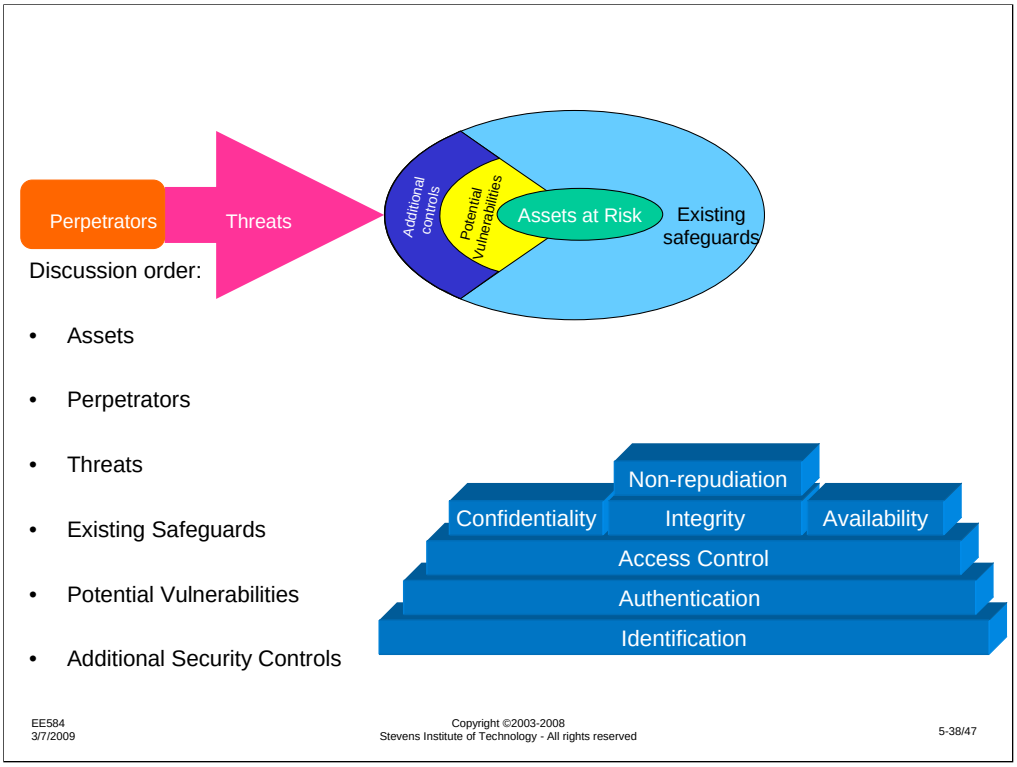
The first case we will discuss is the terrestrial microwave telephone relay systems that were prevalent in the 1960s and 1970s. Many of these systems have been replaced with fiber networks, but there are still some in use, particularly in remote areas.

The particular system of interest was a link in the Washington, DC area. It served as a major backbone link in the area of the US Capitol. The system operated at C-band, about 4 GHz, and sent multiple channels of voice traffic in analog form over one carrier. As we will see later, this was a real system with real security issues.

Network Architecture

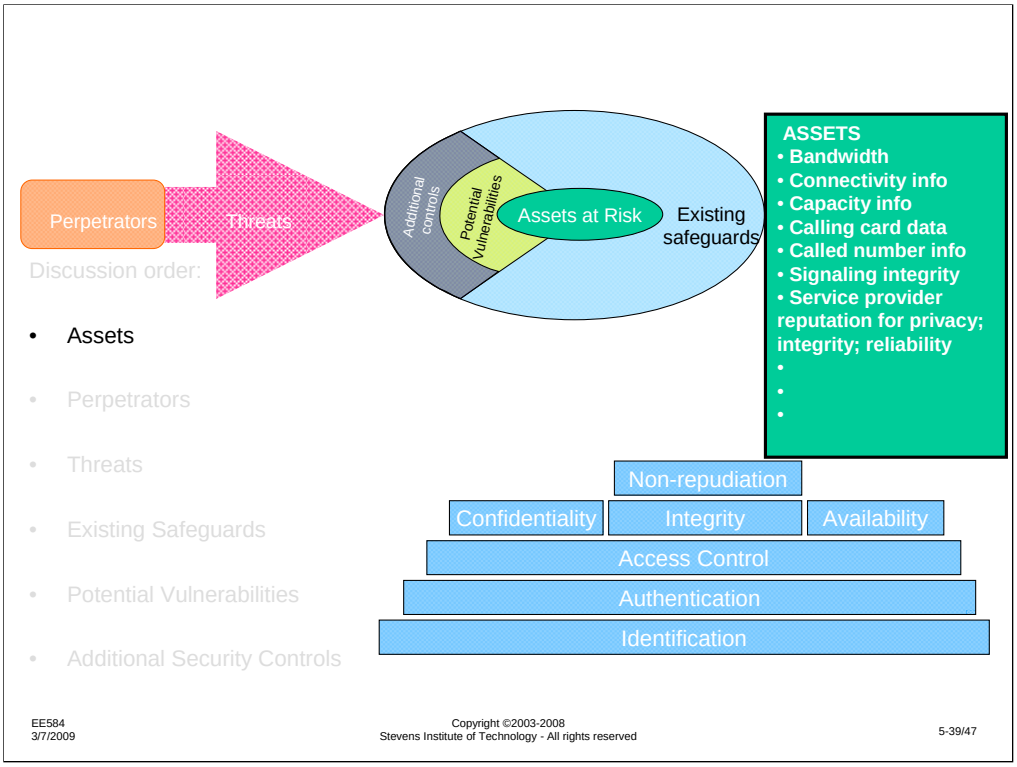


One important characteristic of this microwave telephone relay system was the signaling used. In the 1960s, the analog telephone system used a series of in-band tones to signal the status of a trunk or for call set-up. This was extended to the voice channels that were multiplexed on the microwave carrier system – each voice channel traveled together with the signaling that was associated with that channel. Today, a different architecture is used where all the signaling message travel together, but separately from the voice path (Common Channel Interoffice Signaling). As we will see, this has some interesting security implications.



As we discussed previously, we will be using the security assessment model at the top of the slide, in the order they are listed to the left.

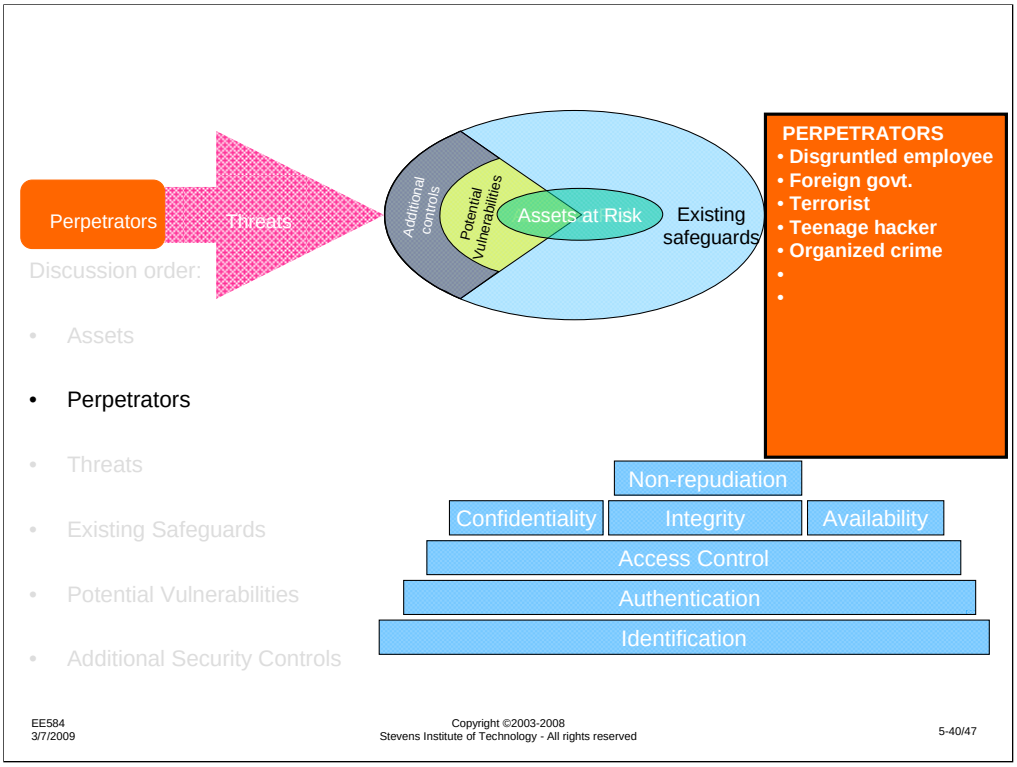
Meanwhile, it is useful to keep the security architecture in mind during the discussion. It is useful to try to find aspects of the system security that are related to each of the security services. Obviously, there will be cases where one particular security service has nothing to do with a particular system, but it is usually best to assume that each service will need to be supported until one can determine otherwise.



To begin this assessment, we will consider the Assets at Risk for the microwave telephone relay system. Since the function of the system is to convey phone calls, some of the assets are obvious: bandwidth, for instance. However, there are other pieces of information that are conveyed across this system that might be worth stealing or damaging: information on how the underlying network is connected might be revealed via the signaling. Information on network and link capacity might also be of interest to an attacker. Since the voice and signaling paths are in the clear (unencrypted analog) the voice channel content, any calling card information transmitted over the link, and the called number are all available to anyone interested in monitoring the RF signal.

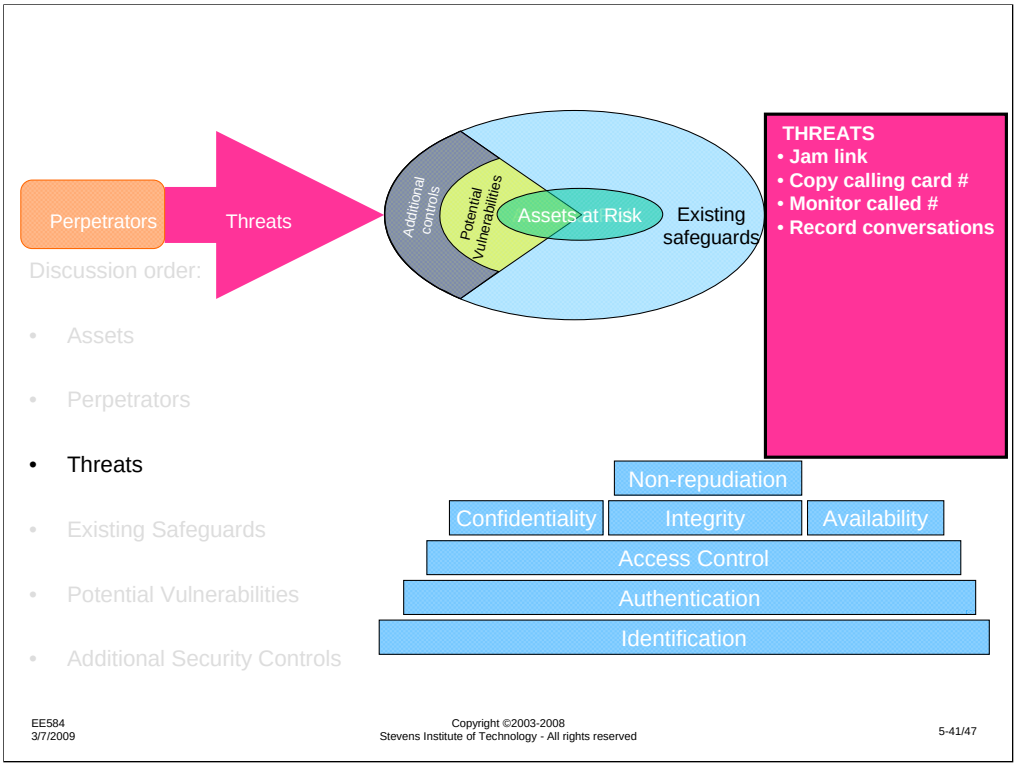
These are all more tangible assets. There are also intangible assets that might be at risk – the reputation of the service provider for customer privacy, reliability of the network, etc. might be compromised.

Discussion topic: What other assets do you see in this system? Remember, anything that is valuable to the owner of the network or might be valuable to an attacker is an asset that might need to be protected and should be identified.



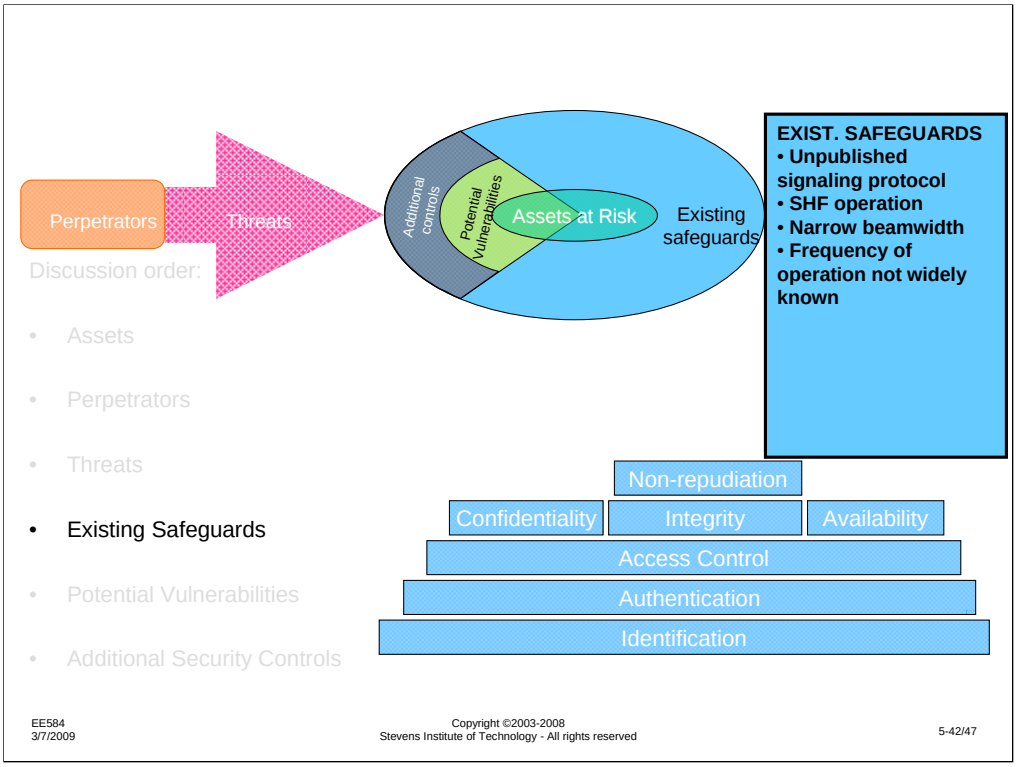
Next, we will identify potential perpetrators – knowing what is of value in the system, who might want to steal or damage it? For each asset we identified, try to think of who the perpetrator might be who would attack it. Likewise, as we go through identifying perpetrators, we should reconsider the list of assets. Perhaps we identify an attacker, but missed what they might attack. The assessment process can and should be iterative. As we get further along, we can always revisit previous items to add to the lists.

The list of people who might attack a system will vary with the particulars of the system, but there are a few common villains who must be considered. The disgruntled employee often is the first to consider. Statistics have shown that the majority of system attacks come from insiders – perhaps 80-90%. Since they have access through their job and often a high level of trust, the capabilities of an insider can't be ignored. Other common perpetrators include teenage hackers, organized crime and foreign governments. It is important to identify each of these separately, since each has different capabilities and motivations. The teenage hacker has relatively little in the way of resources (although a 3 GHz PC and a T1-speed connection is often more than enough!) compared to a foreign government, which might be assumed to have unlimited resources. On the other hand, the motivation of the two are very different. The foreign government might be looking for intelligence information regarding diplomatic or military plans. The teenage hacker may be only trying to break into a system for fun or to show off their technical prowess. Each of these capabilities and motivations need to be understood to assess what the attacker might be willing or able to do. Of course, we cannot forget the potential for terrorist attacks against infrastructure. **Discussion topic**: What other perpetrators can you identify for this system?



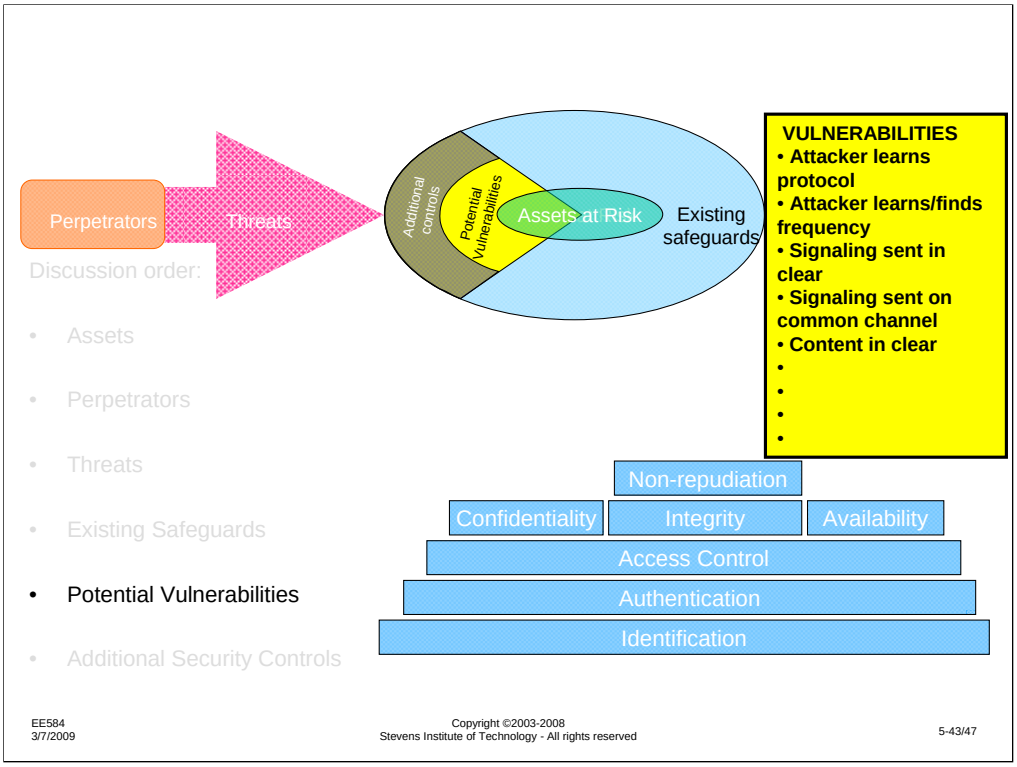
Next, we examine the threats against the system under study. What might an attacker do to steal or damage an asset? In the case of an RF system, intentional disruption of the link via jamming is certainly a possibility, one that would rarely exist on a wired system. Similarly, while monitoring of the link could be an issue for a wired system, the threat is much greater for a wireless system, since it might be possible to attack the system from a variety of places. To access a wired system, one generally needs physical access to the wire, which limits where the attack can occur.

There are several forms of monitoring, depending on what the asset is. If the asset is the voice content of the link, the threat might be recording of the voice conversation. Perhaps, the content of the call is not as important as who is being called. For instance, if one knew that Company A was seriously considering a merger with another company, but it was not known if the target were Company B or Company C, knowing that a call from the CEO of A to the CEO of C occurred, while there was no call to the CEO of B might tell an intruder all they needed to know. For this reason, interception of the called number might be a threat against a system. As we will see, this might also be used in combination with other threats.
Discussion topic: Ditto for threats



Next, we examine safeguards that already exist in the system. If the signaling protocol were unpublished, this might make it difficult for the attacker to exploit this aspect of the system. On the other hand, not only is information often published without examining the consequences, but it is generally not too difficult to monitor enough communications to infer what the protocol was. Security through obscurity is often not very effective. As an example of the publication of information without considering the consequences, we can look at the in-band signaling that was used on the system being discussed. In the 1960s, an article appeared in the Bell System Technical Journal describing how the signaling protocol had been designed. It didn't take very long for hackers to determine how they could exploit this information to build so-called Blue Boxes, allowing toll evasion. (Just to put things in perspective, in the 1960s, the value of $1.00 was an order of magnitude greater than it is today. One minute long distance calls might cost more than one dollar, compared to the few cents per minute today. Thus, the incentive to steal usage on the long distance network was at least two orders of magnitude greater than it is now).

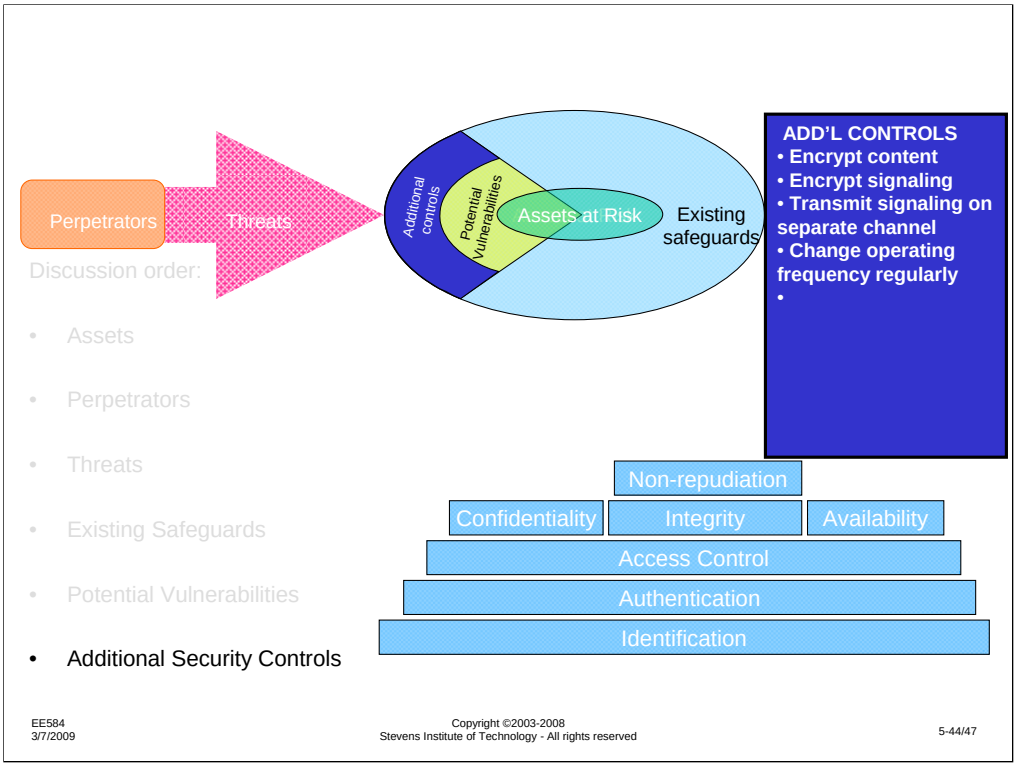
Other safeguards that might exist in this system include the relatively narrow beamwidth of the transmitted signal, making monitoring a bit more difficult. The frequency of operation, 4 GHz, certainly made monitoring more difficult than it would be at lower frequencies.
Discussion topic: Ditto for existing safeguards



Next we can identify potential vulnerabilities in the system. To distinguish threats from vulnerabilities, remember that a threat is something that an attacker could do. A vulnerability is a condition that might exist in the system. We can often state an issue as a threat or a vulnerability, and should be sure to make sure we can see the threat that might make the vulnerability an issue. Again, iteration between the two is desirable.

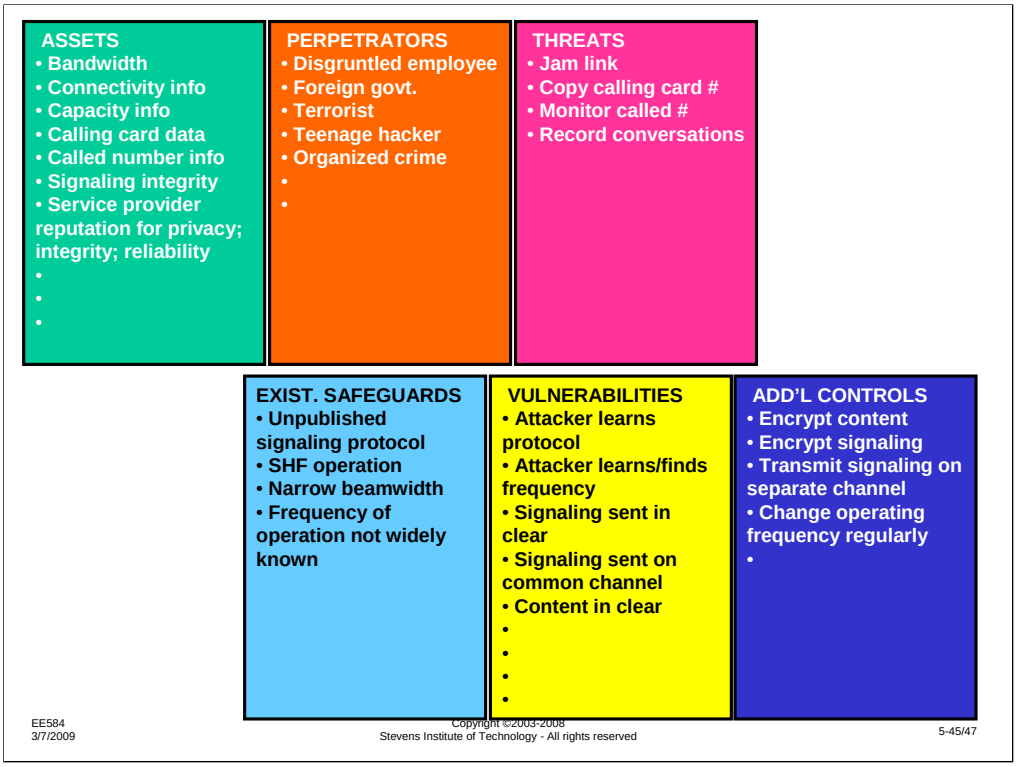
The most obvious vulnerabilities in this system are related to the fact that information is sent in the clear. Anyone with the ability to monitor the transmission can hear the voice conversations and the signaling. Further, an additional vulnerability, and one that I will focus on at the end of this discussion is the fact that the voice and signaling are sent on the same channel. If they were separated, the job of the attacker would be more difficult. Since they travel together, we will see that a very simple attack has serious consequences.

Other vulnerabilities are in the form of hypothetical issues. An attacker could learn the protocol, the operating frequency, or other aspects of the system design. What could be a barrier, albeit one we would not want to depend on, is eliminated if the attacker is able to easily find this information. **Discussion topic**: Ditto for potential vulnerabilities



Finally, we can now try to identify security controls that might be added to the system to improve the security. Remember, that this system existed in the 1960s, so not all of the additional controls we can think of today would have been practical 40 years ago. But, at the same time, these systems are still in use currently, so it might be valuable to consider how they could be upgraded with current technology to make them more secure.

Obviously, since the content and signaling are expose on the RF link, they would benefit greatly by being encrypted. As we will see, the open, common channel signaling is a major source of problems. Even if only the signaling were encrypted or moved to a different facility, the system security could be greatly increased. To make the attacker's job a bit more difficult, it might be feasible to change the operating frequency on some schedule to force him to search for the signal. **Discussion topic**: Ditto for additional security controls.



Listed here are all the security items I have identified with this system. If we were using this assessment to make a decision about what to do, it would be necessary to do a cost-benefit and a risk-reward tradeoff. When we look at the cost of a security control, it has to be balanced against what it protects. One would not spend $1000 to buy a safe for $10 cubic zirconium jewelry, but it might be worth spending this much on a $100,000 coin collection. This is one aspect of the security tradeoff, from the perspective of the owner of the system. A separate balance must be made in terms of the motivation of the attacker. The value of the asset that the owner of a resource determines is not the same as the value that the attacker might see. The owner of a gun may see the value as being what they paid for it. However, having a gun that cannot be traced to the person who steals it may make that gun much more valuable to the thief. When trying to determine the likelihood of attack, it is necessary to balance the attacker's perceived asset value against the cost the attacker may have to expend to acquire the asset.

Case 1
Terrestrial Microwave RF Telephone Relay System

4 GHz
Analog SSB FDMA
Multichannel Voice traffic
CCS signaling
Washington, DC area

EE584
3/7/2009

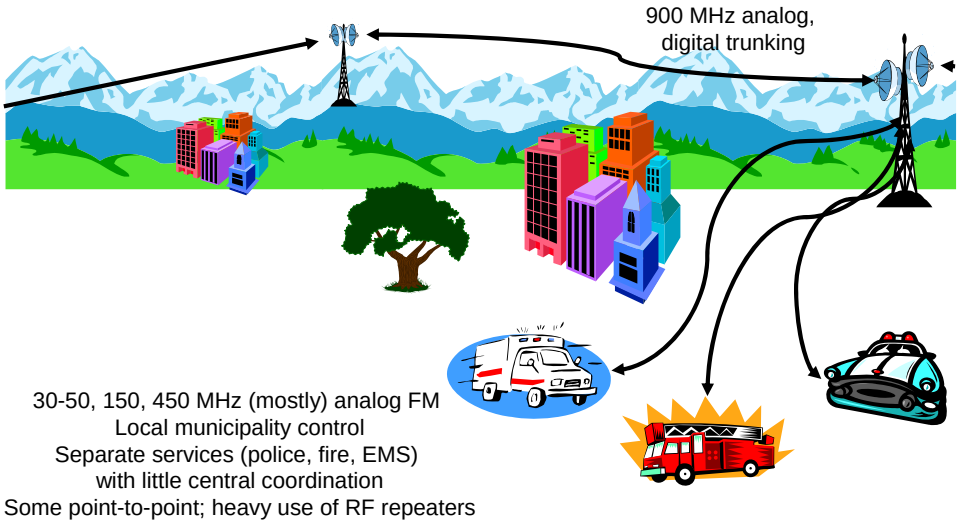
5-46/47

Let's go back to the example to discuss the security issue that really existed in this (real) system. In the 1960s, at the height of "the Cold War", the US and the Soviet Union were enemies. Either would and did use whatever means available to obtain intelligence on the other's plans. As documented in IEEE Spectrum from the early 1970s, the Soviet Union build their embassy right in the middle of an AT&T Long Lines microwave route. Using the in-band clear text signaling and clear text voice paths, a very simple intelligence gathering device was built and operated by the Soviets. A fairly simple device would monitor one of a large number of voice trunks, watching for the in-band dialing sequence. Upon detecting the sequence, the dialed number would be compared to a list of "interesting numbers." While the local shoe store was probably not an interesting number, the home and office phone numbers of members of the Joint Chiefs of Staff, probably would be interesting. Once an interesting number was found, the following voice conversation would be recorded on a cheap analog tape recorder until the call ended. At which time, the device would begin searching for another trunk with an interesting call being initiated. With minimal user intervention, a large amount of intelligence could be gathered as fast as calls were being made.

As it happened, the Soviet embassy was on a hill and received a better signal than the intended relay point!

Besides introducing the process on a real system with known security issues, I wanted to illustrate that design choices can have unintended consequences. The use of the same signal path for signaling and voice was a convenience, but made this attack simple. Current systems separate the voice and signaling. This is mainly done for performance and enhanced feature, but it increases the security as a side benefit. For critical areas, the signaling is also encrypted to deny attackers any advantage.

Case 2 – Public Safety Wireless Networks



30-50, 150, 450 MHz (mostly) analog FM
Local municipality control
Separate services (police, fire, EMS)
with little central coordination
Some point-to-point; heavy use of RF repeaters

In the next session, we will discuss the next case – Public Safety Wireless Systems. It would be useful to do some research on this topic and get some background on what exists and how it is used. From this point forward, each week, for each case example, I will be breaking the class into two halves and will be creating two WebCT discussion groups – they will be labeled as “Case N – Red Team” and “Case N – Blue Team.” The Red Team will be expected to discuss what the security issues are in the system, while the Blue Team will be expected to discuss how to protect the system. Both discussions will be in the context of the security assessment process I have presented this week. Each week, the assignment to a Red or Blue team will be random, so everyone should have an opportunity to look at the issue from both sides.