Wireless Systems Security

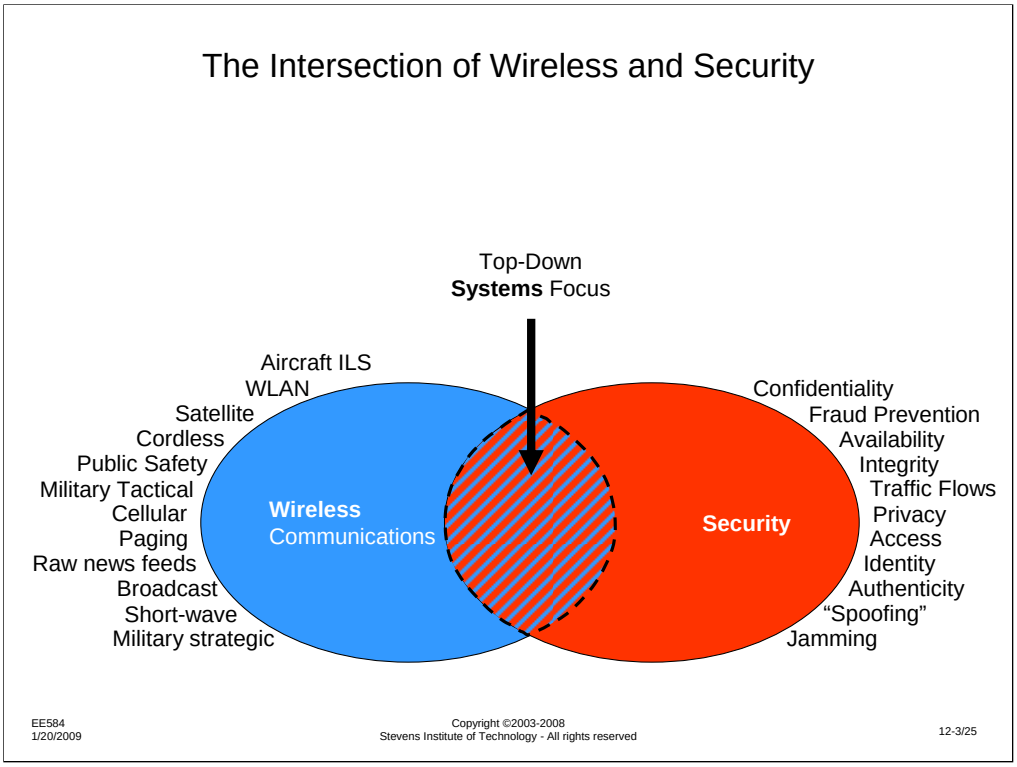
EE/NiS/TM-584-A/WS

Bruce McNair
bmcnair@stevens.edu

Wireless Systems Security

Class 12 – Wrap-up and Future Directions

At this point, we have finished all the case studies in the course. I hope the discussions around them have given you some insight into the issues involved with securing wireless systems, but any systems in general. The most important consideration is to examine the system, not from the view of the designer, but rather from the view of the attacker.

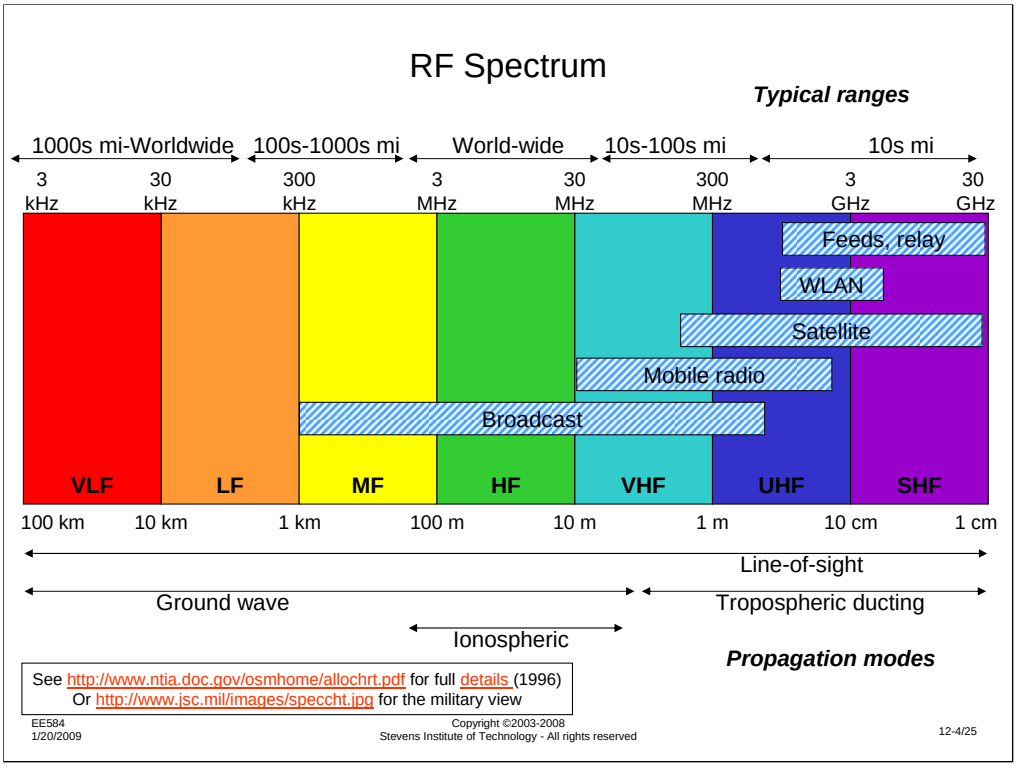


The definition of "wireless communications" is much broader than what one might generally think of, and it is getting broader every day.

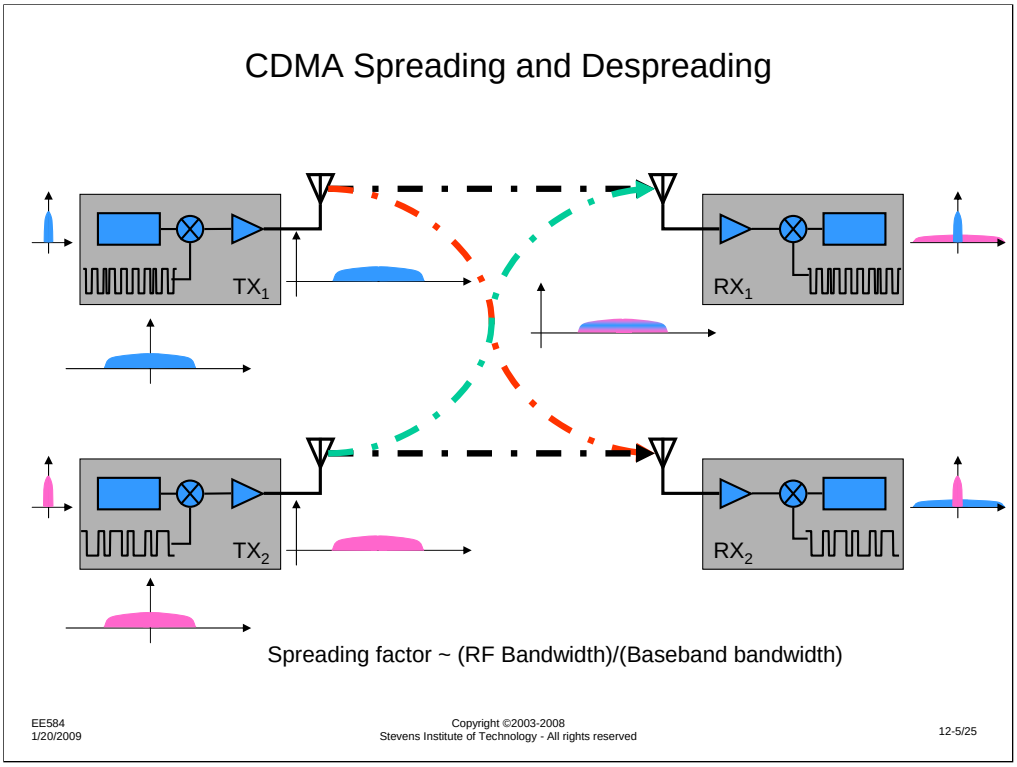
Likewise, for most people "security" first bring up the concept of confidentiality, but this too is a very narrow view of a very broad set of issues.

By constraining ourselves to the intersection of wireless and security, we are by necessity looking at a narrower field, but when the full range of wireless systems and the full range of security considerations are addressed, this becomes a very broad topic area.

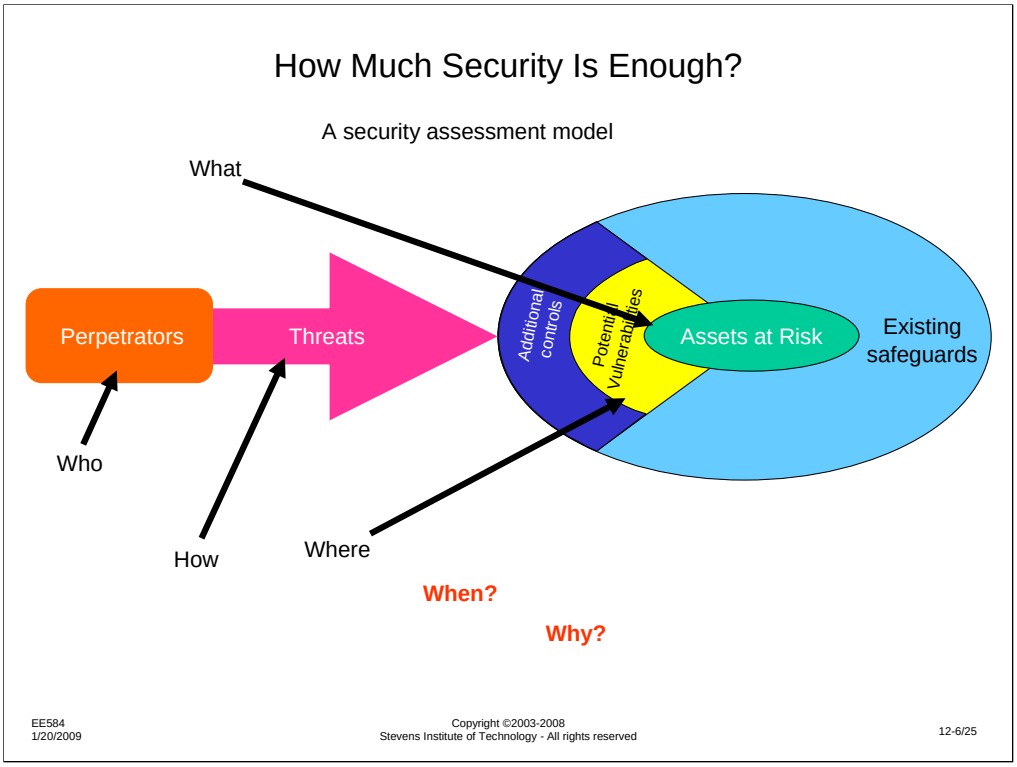
For this course, I tried to maintain a top-down systems focus on the subject area, because I think this is the only way to look at the interaction of many complex issues.



Looking first at wireless communications, we find that the range of frequencies or wavelengths for systems that are in common usage covers 7 orders of magnitude of the spectrum. There are few generalizations that can cover such a wide range, so it is necessary to break down the spectrum into distinct regions, each with their own characteristics. This is particularly important if we are discussing security about operations at the different frequencies in the spectrum. For some, propagation is intercontinental, and the points of attack can be anywhere. For others, the range is line-of-sight, so the attack must come from a distinct area. The examination of security in each extreme is different.



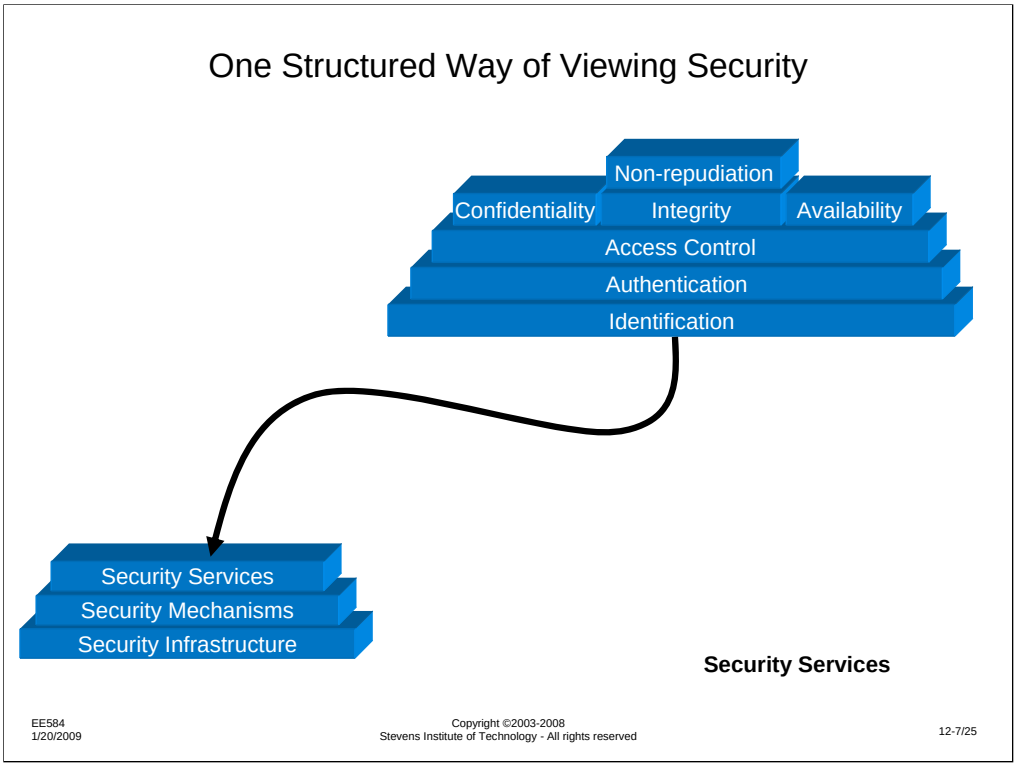
Some RF communications techniques, like spread spectrum and the CDMA variant, offer inherent advantages when one tries to secure the communications link against interception or jamming. However, while the underlying technology provides some security advantages, they must be examined in the specific context of the communications technique. CDMA and spread spectrum, for instance, do provide means to make a link more secure, but this is not necessarily a panacea. Some have argued that the spreading codes offer protection against intercept without link encryption. The fact is, CDMA uses a known spreading code and spread spectrum systems, even those using secret codes, in essence broadcast their spreading sequence. One must always examine the protection afforded in terms of the problem it was designed to solve. One cannot count on the confidentiality of a technique that was not designed with confidentiality in mind, unless some extra precautions are taken or unless one verifies the confidentiality offered.



Which leads directly to the question of how one assesses the level of security provided or required by a system.

I have spent many years assessing system security in previous assignments. On one hand, there are military-grade formal assessment processes used for systems like the Strategic Defense Initiative (SDI, otherwise known as Star Wars). These formal assessment techniques offer the best measurement of system security, but I have found them to be extremely expensive to employ. In a commercial system, one must always balance the cost of protecting the system against the cost of the system. It doesn't make any sense investing a dozen staff years in accurately measuring the security of the system if the total investment in developing the system is about the same. No commercial operation would be willing to dedicate that much effort to security.

So, in the process of looking at security in dozens of commercial systems, we came up with a much less formal, but quite effective means of evaluating system security issues, which we have used in this course. By focusing on the things of value in the system, and the people who are most likely to attempt to compromise those items, one can begin to evaluate security in a goal-oriented manner. The level of effort used for each assessment in this course is about the same level of effort that I have seen to be quite effective in real systems assessments.

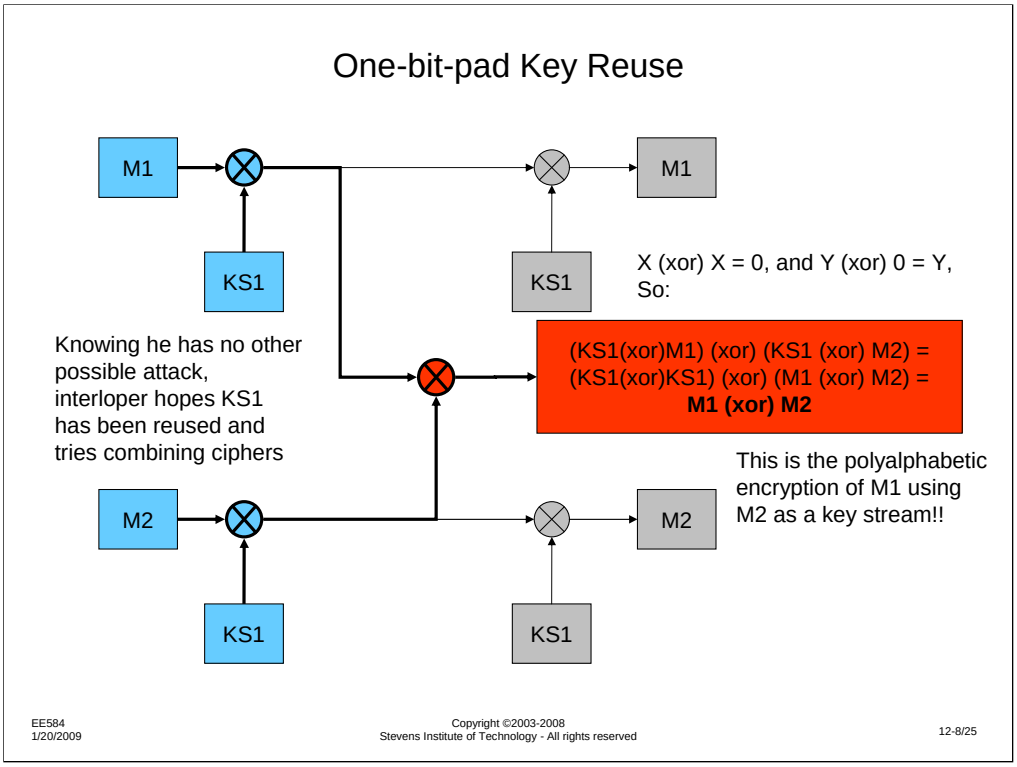


Of course, the assessment process described gives us a methodology, but we still need a structure to discuss security. Again, most people have a relatively narrow view of what security actually means. A person who has secrets to protect easily understands the confidentiality aspects of security. One who depends on a communications link's accessibility understands availability, while someone who is aware of the potential to disrupt a system by modifying data content understands integrity. It is only be bringing together all the potential security issues that a customer might be concerned about that we can get a complete picture of a system's security.

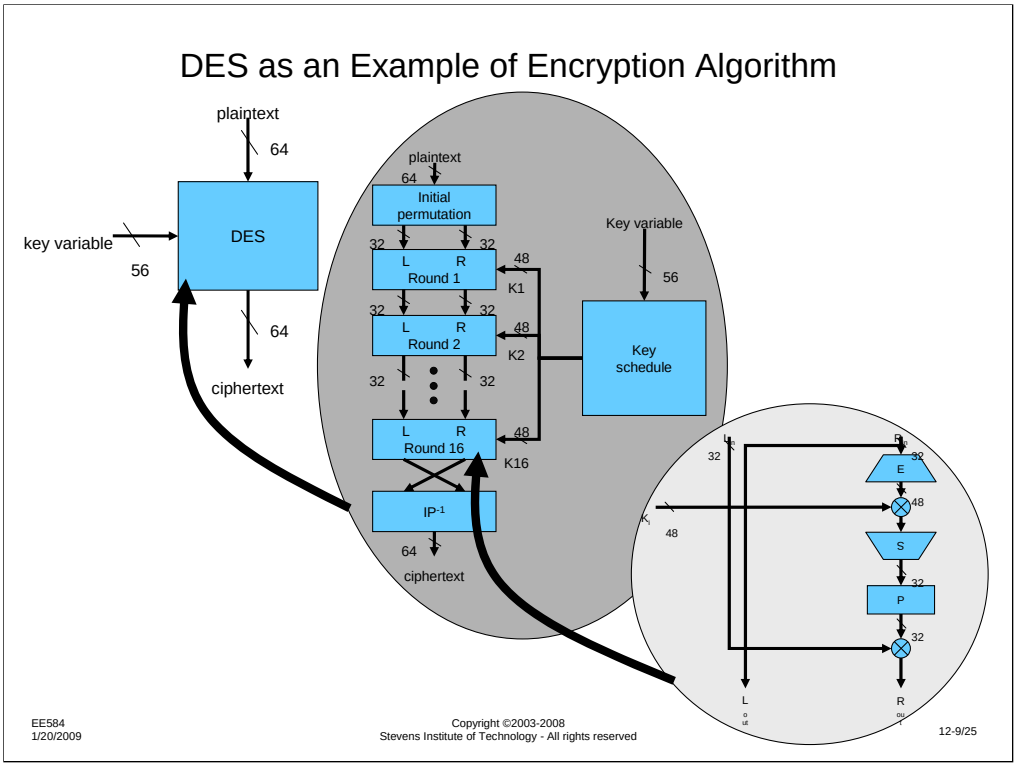
Many years ago, I took a course in switching theory where we discussed the concept of a logically complete set of functions. In that context, there are certain functional mappings from a series of input values to output values. The question comes up as to what are the minimum set of functions that can express all possible mapping. It turns out that AND, OR, and NOT form a logically complete set that can express all possible input-output relationships. It also happens that the NAND (NOT AND) or the NOR (NOT OR) functions are each logically complete by themselves. The challenge of a logic designer is to be able to use the smallest set of logically complete functions to realize a logic function, making the overall system design as simple as possible.

In this way, as I looked for a set of security services to describe all possible security issues, I was looking for a "logically complete" set of security functions. They must cover all possible issues and must do so in a manner that allow nonambiguous descriptions. It wouldn't do to have two different ways of describing a security issue that were self contradictory. And finally, the set has to be the "minimum spanning set," that is it is the minimum set of services that can cover all possibility issues. I think the set shown above meets all the required tests.

Finally, this taxonomy defines security services that are visible to an end user. Underlying the security services are mechanisms that make the services possible.



We covered encryption at some length as one security mechanism. As you saw in the 802.11 assessment, even though we have understood the potential for an unbreakable crypto system based on Vernam's one-time-pad since the early 1900s, people are still designing systems that violate the basic design principle of never reusing a key stream. It took Friedman a few years to find the hole in Vernam's nearly perfect system, but that was the 1920s that the hole was known. There is no excuse for anyone to build a crypto system in the late 20th century that is exploitable in a way that was known almost 75 years earlier. But it happened and probably issues like this continue today. Only by a careful examination of a system's security early in the development cycle and throughout the development cycle can problems like these be avoided.



I mentioned DES as one example of a modern crypto system. DES has been broken, not because it has an inherent flaw, like key stream reuse, but because, like everything else, it is subject to the rapid evolution of semiconductor technology. When DES was standardized in the late 1970s, several cryptographers said the 56-bit key variable was too short to stand up to brute force attacks for an acceptably long period of time. Nevertheless, the National Security Agency was concerned about allowing an algorithm to be standardized that would be too difficult for them to be able to break. They did not want our commercial grade cryptography to make other country's military grade cryptography stronger than it was.

As shown in "Cracking DES," published by the Electronic Frontier Foundation in 1998 (ISBN 1-56592-520-3), only 20 years after it was standardized, the exponential growth in device technology made it a simple matter to build a machine that could break DES by brute force attack (trying all possible key variables).

Today, we have AES (the Advanced Encryption Standard), which should have a longer lifetime than DES. Nevertheless, the structure of AES is quite similar to the structure of DES, showing that there was nothing inherently wrong with the design, just the choice of design parameters (e.g., key variable length).

The lesson here is that Moore's Law has been a good model for technological growth for at least the last 40 years. It always appears that it is nearing its utility in predicting the future growth, but this utility continues. It would be short sighted to assess a system security assuming that the bad guys won't have access to technology that is orders of magnitude more powerful than exists today in only a few years, based on the historical doubling every 18 months. On could argue about the slope of the curve and whether it will continue 20 years in the future, but it only takes 5 years of the historical trends to make an order of magnitude difference in performance (4.98 years to be exact).

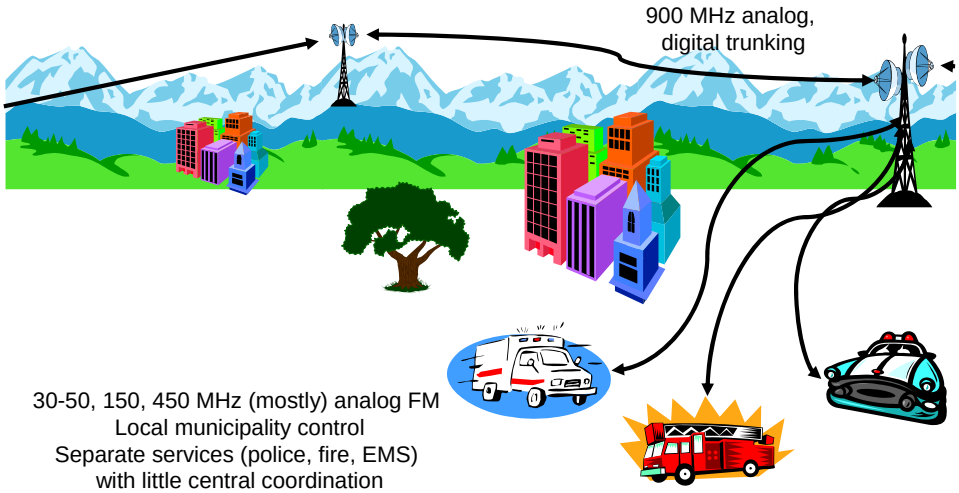
Case 1
Terrestrial Microwave RF Telephone Relay System

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

We covered seven case studies in this course. I could have chosen more or fewer systems to investigate or a different set of systems. In the subsequent slides, I'll mention why I picked each case study and what one might infer from the study.

Case 1 was a real, live system that got a great deal of attention in the late 1960s, at the height of the Cold War. This is a system where a few simple design choices (in-band unencrypted signaling) made an attack feasible that compromised a great deal of confidential information. For the time period, the sophistication of even the most powerful attacker was limited, so it was hacker ingenuity alone that allowed the attack to succeed. Today, we have much more sophisticated systems with greater levels of protection, but the attacker also has almost 40 years of technical development to rely on. Moore's Law suggests that the technology available today is more than 16 million times greater (speed, complexity, etc.) than that which was available in 1968.

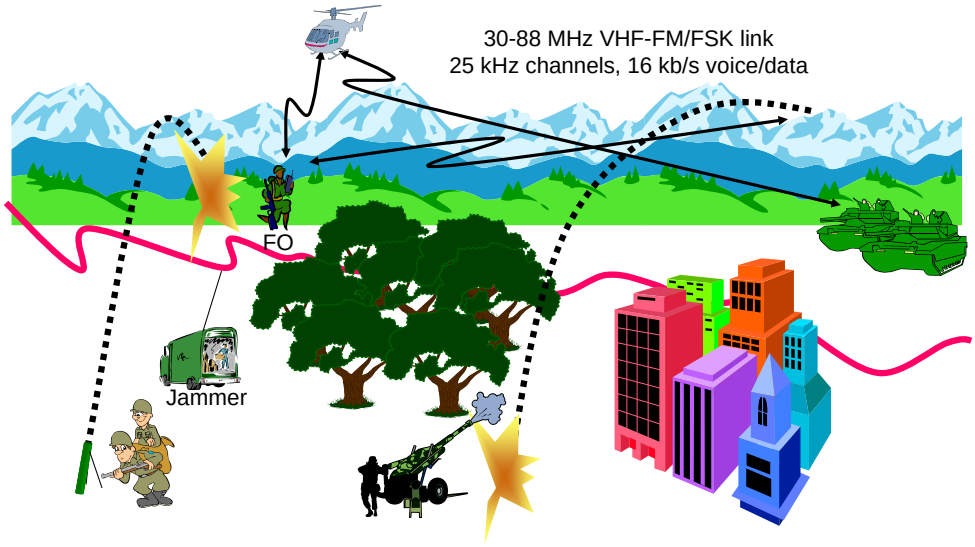
Case 2 – Public Safety Wireless Networks



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

I recently had occasion to consult with the New Jersey State Attorney General's office, the department that manages the NJ State Police. Years after the difficulty they encountered in the 9/11 attack on the World Trade Center, interoperability is as much a problem as it was then. We rely very heavily on the Public Safety systems for fire, police, and EMS, but those systems are not protected much better in the 21st century then they were in the 1920s. At least in the 1920s, the number of people with the technology to intercept and/or interrupt PSWN communications was much less than it is today…

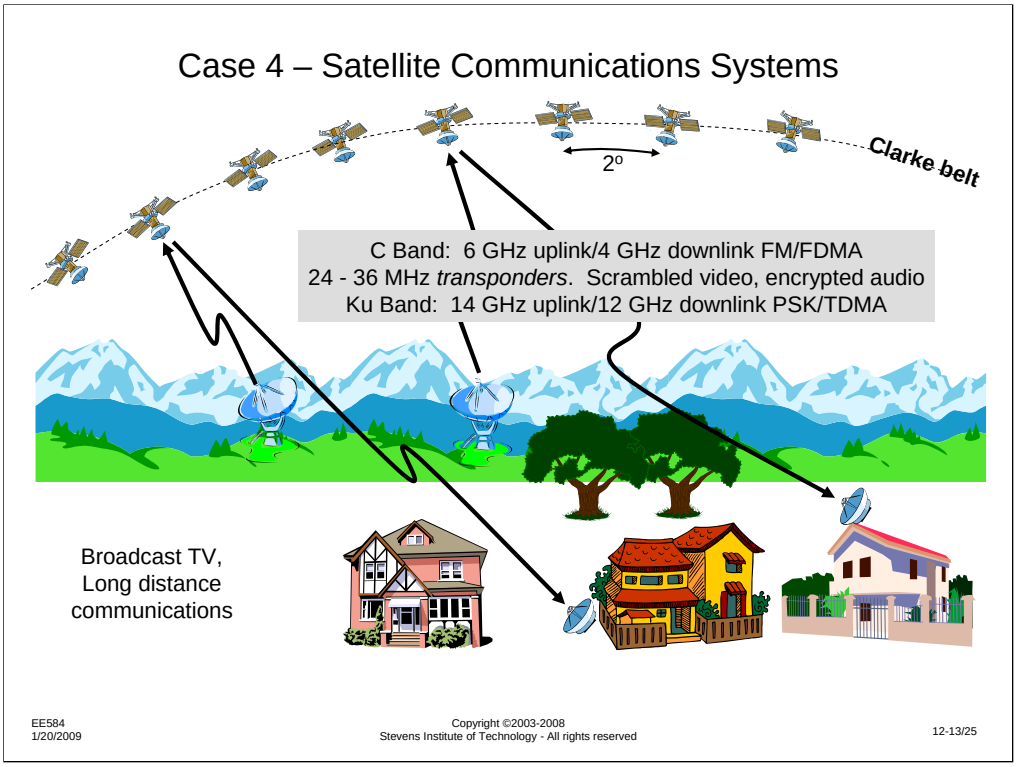
Case 3 – Military Tactical Radio Systems



EE584
1/20/2009

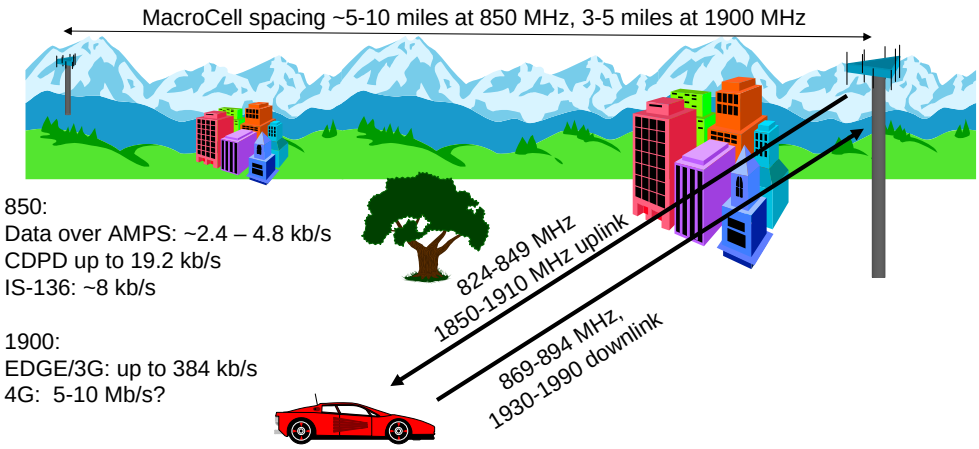
12-12/25

I use the tactical military communications systems as an example of a system that has addressed the need for security directly in the early stages of their development cycle. Part of this is because their design cycle is much longer than a commercial system and they realize that they must address the threat that might exist when the system is deployed, not the threat as it is understood today. Nevertheless, potential issues still exist in these systems and require ongoing examination.



Securing satellite communications might be the hardest security problem anyone has to deal with. The satellite, once launched, is essentially on its own. It cannot be recalled to be retrofit with new protection technology. The points of attack are nearly boundless. Potential attacks that might be launched against the satellite might target not just the electronics and communications, but could also target the operational capabilities of the satellite itself (e.g., fuel for station-keeping). And finally, the attacker can easily blend into the landscape. Unsophisticated attackers have been discovered and prosecuted, but that is not to say that more careful attackers whose motivation is to cause damage and cover their tracks could not succeed.

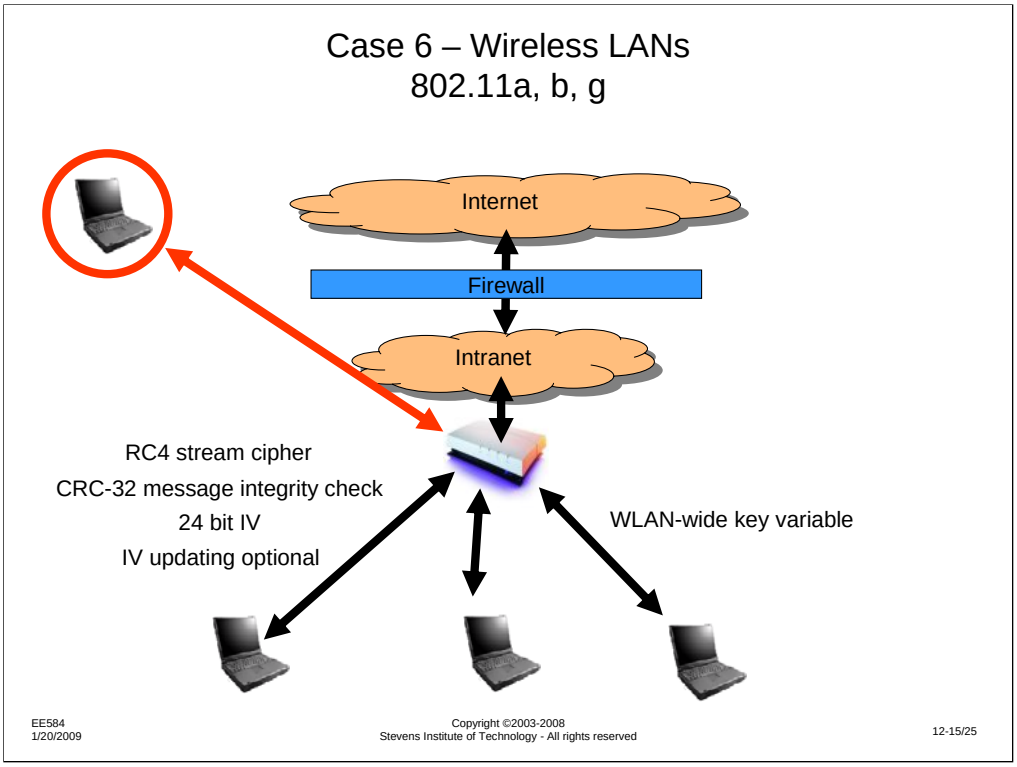
Case 5 – Wide Area Wireless Data Services
CDPD, 3G, EDGE, etc.



850:
Data over AMPS: ~2.4 – 4.8 kb/s
CDPD up to 19.2 kb/s
IS-136: ~8 kb/s

1900:
EDGE/3G: up to 384 kb/s
4G: 5-10 Mb/s?

We have seen an explosive growth in the wired Internet and in voice cellular communications. The same motivations that fuel the growth of those services are likely to fuel growth in wide area mobile data communications. However, the vulnerabilities of wireless voice, wired data, and wireless data are not the same. As we become more dependent on wireless data services for everyday commerce, the potential for attack on those systems increases, and the potential damage that might result becomes greater.



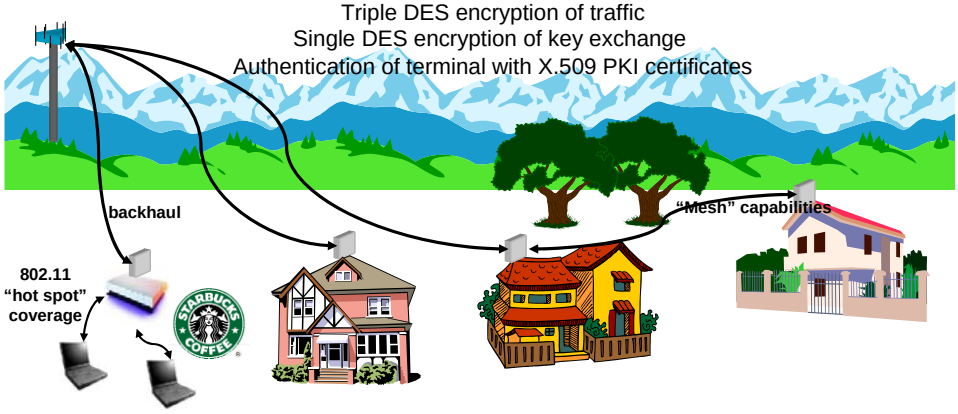
The latest success story in wireless data communications has to be the 802.11 WLAN. At the same time, the latest embarrassment in poor security design of a communications system has to be 802.11. It is amazing to see the widespread use of 802.11 WLANs in areas where there are obvious security issues. Avi Rubin has published some of the best descriptions of attacks on 802.11 networks – his motivation was driven by watching nurses and doctors in a hospital emergency room use data terminals connected via 802.11b networks. What happened to HIPPA (Health Insurance Portability and Accountability Act of 1996)? Well, apparently, they haven't figured out how to comply with the act, so lots of personal medical information is accessible that shouldn't be.

Worst of all, the items that make the 802.11 WEP (wired equivalent privacy) insecure are gross errors of secure system design. The reuse of key stream is a 75 year old known vulnerability. Simple calculations about network utilization would have shown that a 24-bit initialization vector was totally inadequate. And the use of fixed, system-wide key variables is an obvious shortcoming.

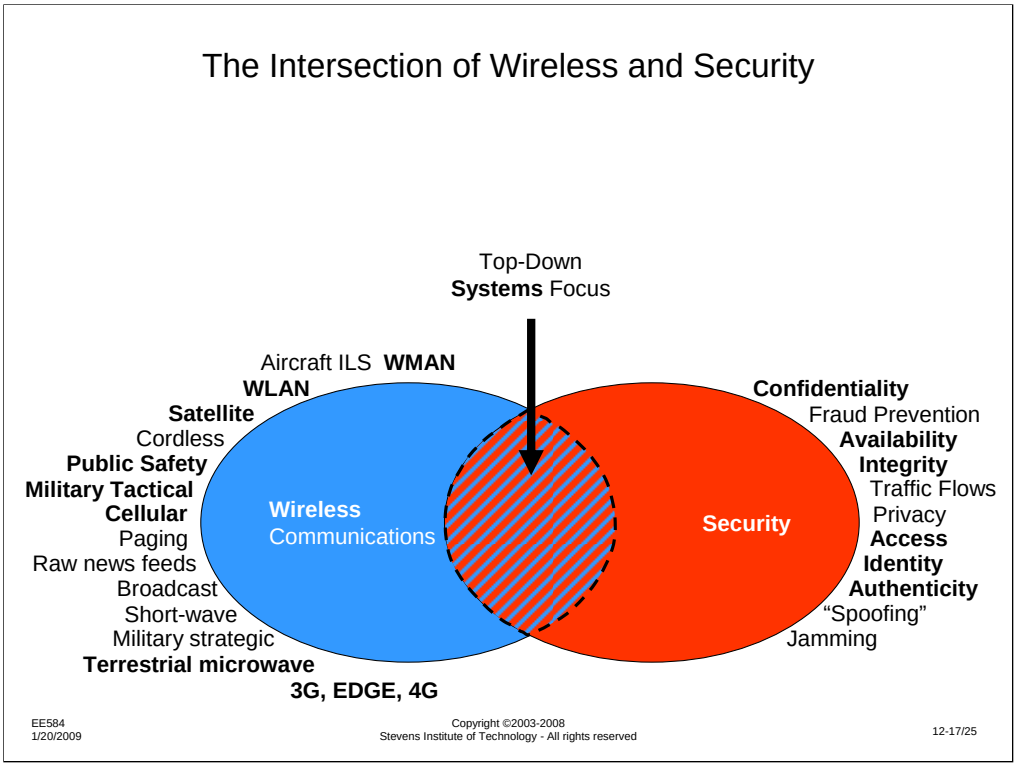
Why do these problems exist? Probably because no-one thought to look for them.

Case 7 –
Wireless Metropolitan Area Networks (W-MANs)
802.16

802.16a: 2-11 GHz 256/2048 carrier OFDM,
802.16.1: 10 – 66 GHz LOS
120 Mb/s capacity
T1+ user data, multiple voice channels, Wireless Local Loop
Triple DES encryption of traffic
Single DES encryption of key exchange
Authentication of terminal with X.509 PKI certificates



Finally, the last case study was an new evolving service. There are lots of good security controls that have been added to this system, but there is still room for improvement. The "mesh" capabilities provide means to extend the system range to users who might not otherwise be able to access the network, but create even more problems.



I generally like to end a talk or course with the context I started with. This was the first slide used in this course's material.

I have highlighted several wireless communications systems and several security items, in terms of the items we have discussed at length. The list is obviously not all inclusive. There are some topics I list but have not discussed in detail. There are obviously some items we could add to the list that are not identified here and may not have been covered in the course.

Both the topics of wireless communications and security are open-ended topics, that continue to evolve each day. Unlike a closed-end historical topic, we have to accept the ongoing change of the topic. I hope this course has given you an open-ended context to continue to study and understand wireless security topics.

Key Points

- Security:
  - is best designed in, rather than added on
  - issues must be examined in the broadest context
  - cannot be taken for granted
- The interaction between complex systems is a fertile growth medium for security issues
- Obfuscation doesn't help
  - Where does mold tend to grow in homes?
- Wireless systems are generally:
  - New designs (not much field experience)
  - Complex (interactions between varied technologies)
  - Designed with short development cycles
  - Closed systems at introduction
- Broadcast nature of most wireless systems creates issues that wired systems don't share:
  - Ease of monitoring
  - Potential for jamming
  - Attack from anywhere – difficulty in controlling access to airwaves

To summarize what I think are the most important lessons in secure system design, particularly with the unique issues in wireless systems, I have listed a few items above. I don't think I can add anything by annotating this slide beyond what is listed above.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations

Finally, I want to wrap up the discussion with a few operative definitions.

The quality definition is something I came across at AT&T in the mid-1980s. I found the utility of applying it to security shortly afterwards, and haven't found the need to modify this view since.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations
- Security is meeting or exceeding customers' expectations *in the presence of the actions of an adversary*

Finally, I want to wrap up the discussion with a few operative definitions.

The quality definition is something I came across at AT&T in the mid-1980s. I found the utility of applying it to security shortly afterwards, and haven't found the need to modify this view since.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations
- Security is meeting or exceeding customers' expectations *in the presence of the actions of an adversary*

- A fool is someone who does not learn from their mistakes

My view of a fool and a genius is a bit of personal observation that I generally have found useful to understand the repeated mistakes some people (and large organizations!) make.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations
- Security is meeting or exceeding customers' expectations *in the presence of the actions of an adversary*

- A fool is someone who does not learn from their mistakes
- A genius is someone who learns from *other's mistakes*

My view of a fool and a genius is a bit of personal observation that I generally have found useful to understand the repeated mistakes some people (and large organizations!) make.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations
- Security is meeting or exceeding customers' expectations *in the presence of the actions of an adversary*

- A fool is someone who does not learn from their mistakes
- A genius is someone who learns from *other's mistakes*

- Effective quality processes do not expect perfection

There is an interesting intersection of these two concepts, particularly when you look at ongoing quality processes. ISO-9001 and the Malcolm Baldridge Award criteria are probably the best examples of quality processes, since their evaluation metrics are public information and something that any organization can begin to apply to their operation, even if they are not looking for ISO-9000 certification or trying to win the Baldridge award.

Recently, I experienced the ABET process, by which the Stevens' School of Engineering undergraduate program underwent evaluation to decide if the school should be accredited to grant Bachelors of Engineering degrees. What became apparent is that their evaluation focuses on the same methods as ISO and the Baldridge award. Perfection of the process is not the issue. Errors and misunderstandings of needs are to be expected. The important issue is whether the program has a mechanism to detect deviation from perfection and attempt to move towards it.

The same is applicable to system security. I strongly believe that it is not possible to build a system that is secure against all attacks. I also believe that it is impossible to write bug-free software. I like to state this as "In a sufficiently complex system, there is no last [security hole]|[software bug]." No matter how hard you look, and how many problems you correct, there is probably at least one more lurking in the background. This is not a defeatist attitude, however. It says that you have to be ever vigilant and try to find the next important bug and eliminate it. Security (and good software design) are not the sort of problems that can be wrapped up in a tidy little package and put aside; rather, it is necessary to continually evaluate the system against the evolving threat environment and take corrective action as needed.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations
- Security is meeting or exceeding customers' expectations *in the presence of the actions of an adversary*

- A fool is someone who does not learn from their mistakes
- A genius is someone who learns from *other's mistakes*

- Effective quality processes do not expect perfection
- They do expect continuous process improvement

There is an interesting intersection of these two concepts, particularly when you look at ongoing quality processes. ISO-9001 and the Malcolm Baldridge Award criteria are probably the best examples of quality processes, since their evaluation metrics are public information and something that any organization can begin to apply to their operation, even if they are not looking for ISO-9000 certification or trying to win the Baldridge award.

Recently, I experienced the ABET process, by which the Stevens' School of Engineering undergraduate program underwent evaluation to decide if the school should be accredited to grant Bachelors of Engineering degrees. What became apparent is that their evaluation focuses on the same methods as ISO and the Baldridge award. Perfection of the process is not the issue. Errors and misunderstandings of needs are to be expected. The important issue is whether the program has a mechanism to detect deviation from perfection and attempt to move towards it.

The same is applicable to system security. I strongly believe that it is not possible to build a system that is secure against all attacks. I also believe that it is impossible to write bug-free software. I like to state this as "In a sufficiently complex system, there is no last [security hold]|[software bug]." No matter how hard you look, and how many problems you correct, there is probably at least one more lurking in the background. This is not a defeatist attitude, however. It says that you have to be ever vigilant and try to find the next important bug and eliminate it. Security (and good software design) are not the sort of problems that can be wrapped up in a tidy little package and put aside; rather, it is necessary to continually evaluate the system against the evolving threat environment and take corrective action as needed.

A Few Operative Definitions

- Quality is meeting or exceeding customers' expectations
- Security is meeting or exceeding customers' expectations *in the presence of the actions of an adversary*

- A fool is someone who does not learn from their mistakes
- A genius is someone who learns from *other's mistakes*

- Effective quality processes do not expect perfection
- They do expect continuous process improvement

An effective security process would be one that
- continually ***anticipates*** threats,
- ***prioritizes*** the most credible threats, and
- ***adapts*** to meet those threats ***before*** they degrade the system

EE584
1/20/2009

12-25/25

There is an interesting intersection of these two concepts, particularly when you look at ongoing quality processes. ISO-9001 and the Malcolm Baldridge Award criteria are probably the best examples of quality processes, since their evaluation metrics are public information and something that any organization can begin to apply to their operation, even if they are not looking for ISO-9000 certification or trying to win the Baldridge award.

Recently, I experienced the ABET process, by which the Stevens' School of Engineering undergraduate program underwent evaluation to decide if the school should be accredited to grant Bachelors of Engineering degrees. What became apparent is that their evaluation focuses on the same methods as ISO and the Baldridge award. Perfection of the process is not the issue. Errors and misunderstandings of needs are to be expected. The important issue is whether the program has a mechanism to detect deviation from perfection and attempt to move towards it.

The same is applicable to system security. I strongly believe that it is not possible to build a system that is secure against all attacks. I also believe that it is impossible to write bug-free software. I like to state this as "In a sufficiently complex system, there is no last [security hold]|[software bug]." No matter how hard you look, and how many problems you correct, there is probably at least one more lurking in the background. This is not a defeatist attitude, however. It says that you have to be ever vigilant and try to find the next important bug and eliminate it. Security (and good software design) are not the sort of problems that can be wrapped up in a tidy little package and put aside; rather, it is necessary to continually evaluate the system against the evolving threat environment and take corrective action as needed.