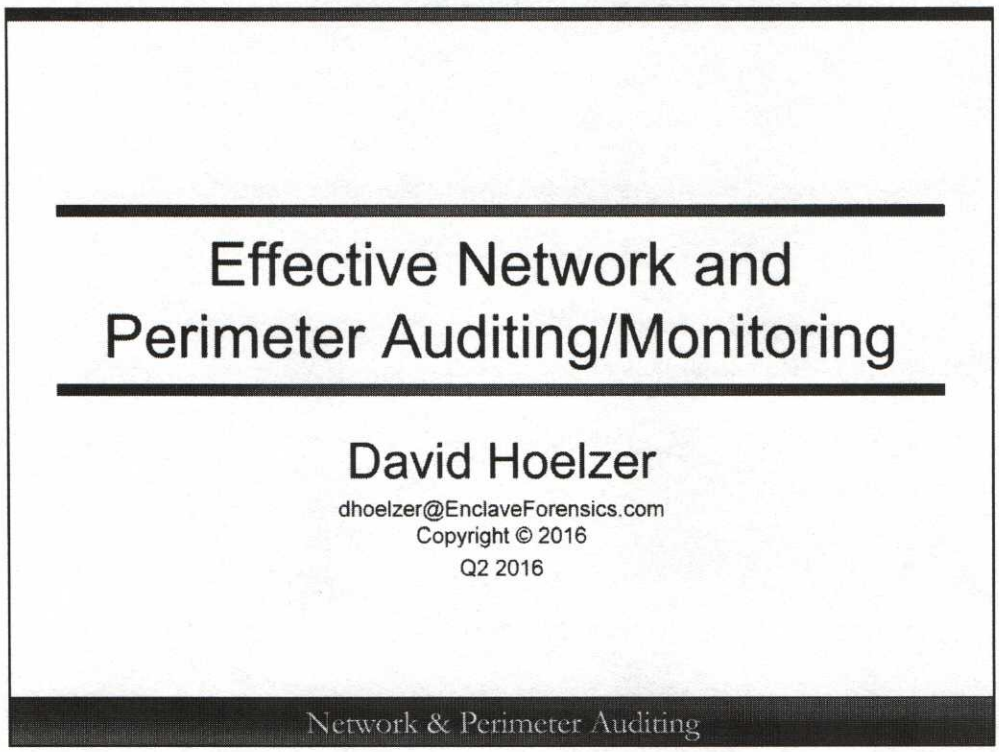
507.2
Effective Network and Perimeter Auditing/Monitoring

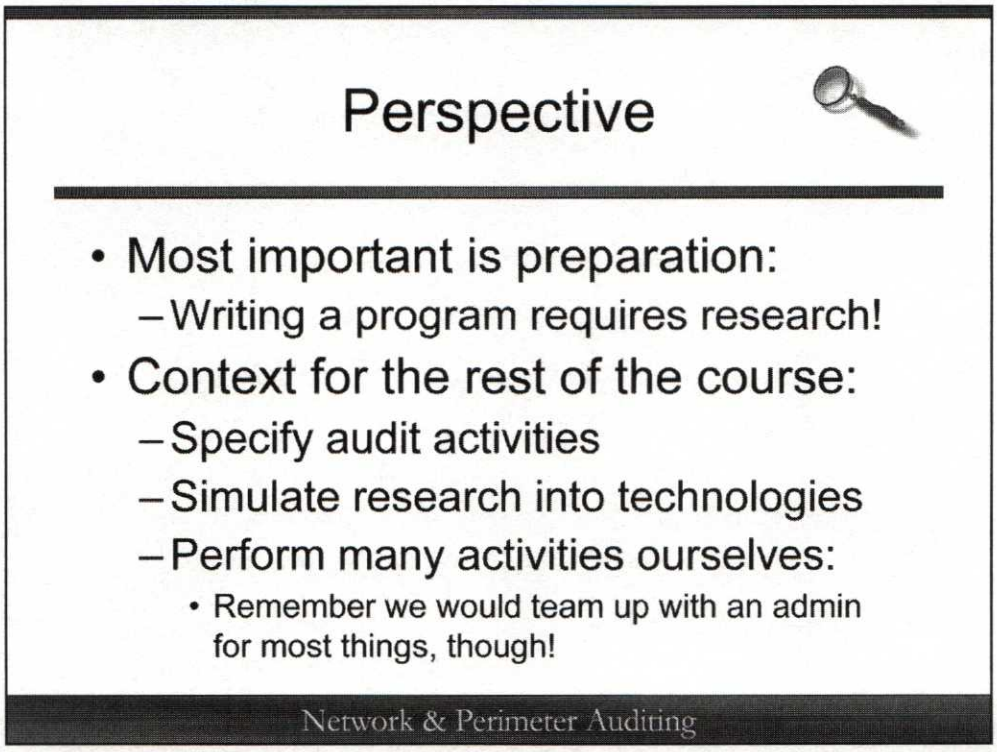
SANS
THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

ISBN 978-1-937060-07-7
Fourth edition

This page intentionally left blank.



This page intentionally left blank.



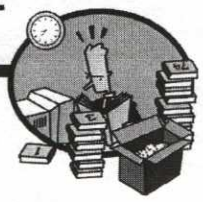
Welcome to the Network and Perimeter auditing course! Before we jump into the material, I'd like to invite you to contact me should you have any suggestions or questions regarding today's material. If you find spelling or grammar errors that exist in the material, please let me know about that, too! I can be reached at dhoelzer@enclaveforensics.com. I would encourage you to put the word SANS in the subject line so that I read your e-mail rather than move it to the junk folder.

As we cover the material starting today through the rest of the week, it is important to put it into context so that you can view it through the proper lens. You can certainly look at the information contained in the remaining course books as a collection of tips, tricks, tests, and other important activities that all relate to securing or otherwise defending your enterprise. Although this is certainly a useful perspective, our actual goal is different.

Our discussion of the audit process toward the end of the first volume in this course pointed out the criticality of preparation during the audit process. A fundamental aspect of this preparation is doing adequate research into the systems or technologies that we assess. The research does not have to be so deep that we are experts, but it does need to have enough depth to allow us to have a competent conversation with the administrator of the systems under analysis.

Therefore, as we cover the remaining material in this course, our intended purpose is to simulate the degree of research that should be performed while simultaneously sharing with you our experience in both securing and auditing these systems.

Extremely Important



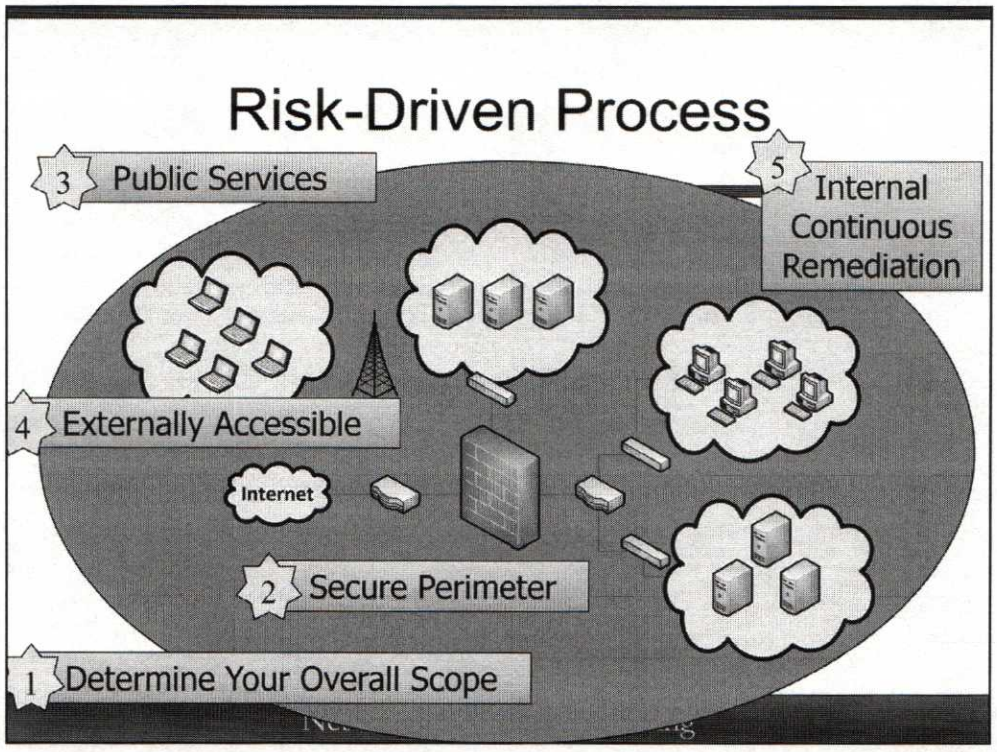
- At some point you will feel overwhelmed…remember:
  - We would never ask an auditor to configure a firewall…
    - …but you need to understand what the admin is saying!
    - …and you need to know what questions to ask and know if the answers make sense!

Network & Perimeter Auditing

Because we want to give you in-depth research quickly and for many different technologies all compressed into just a few days, you will no doubt begin to feel overwhelmed at some point during our discussion. Do not allow this to discourage you!

Always try to keep the perspective as we cover this material that we want to sensitize you to the serious issues involved, provide you with food for thought to bolster your own security program, and give you a familiarity with the technology that can enable you to act in a competent way when performing assessments or accreditations. Remember that we advise you to always have the actual administrator of the systems not only with you but also actually doing the typing!

In other words, remember that we would never ask an auditor or even the security officer to configure the router or firewall! However, that auditor or security officer should absolutely look at the configuration of such a system and ask intelligent questions about it, possibly even identify gaps in its configuration.

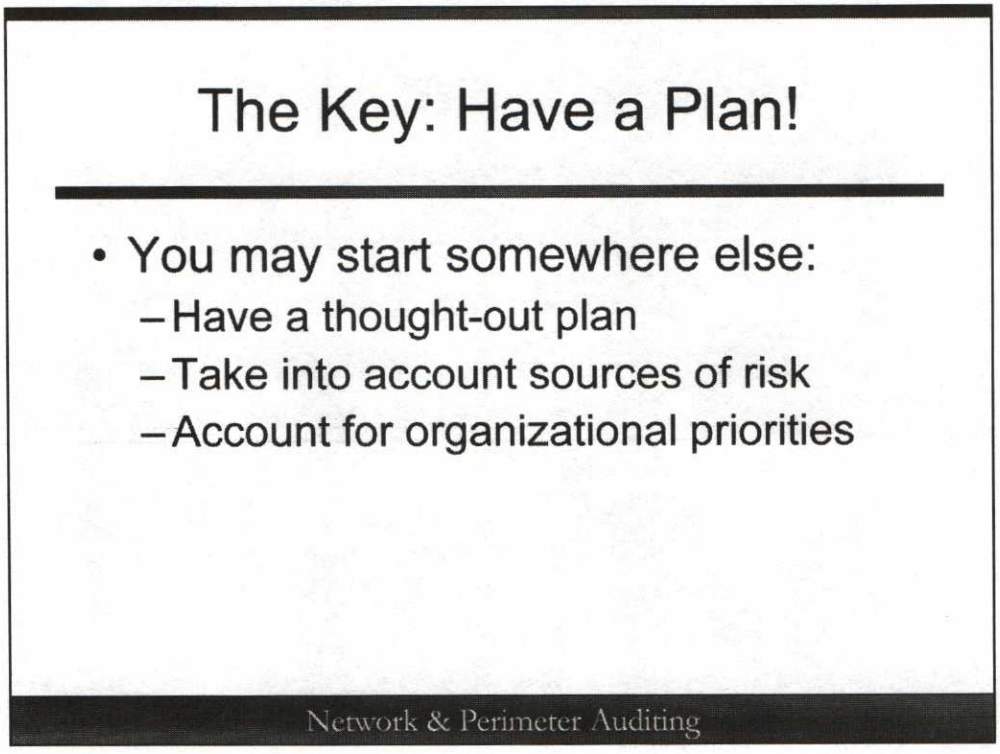


Another valuable thing to realize is that the overall design of the course is an example of the risk-based approach that we discussed during the first day. If we are charged with securing, auditing, or assessing the enterprise, how do we know where to start? Remember our discussions of risk and alignment with the business objectives. We need to understand, in the context of the business, which business systems are most critical to the actual mission critical objectives. With this understanding we can then create a program that seeks to evaluate technical systems based on their overall relationship with those business systems and objectives.

Because every organization will be somewhat unique, we take a more generalized approach in the course. After determining exactly which systems in the enterprise we are responsible for, our scope, we begin by analyzing the perimeter. That is actually what we are doing today.

We want to start with the primary threat mitigation system in our enterprise: the firewall. After the firewall is thoroughly audited and validated, the next highest risk for penetration from the outside would be publicly exposed systems such as DNS servers, mail servers, and web servers. Included in this category would be wireless access, out-of-band management and other similar entryways. After these are secured we want to analyze all the systems and services that are externally accessible to third parties such as suppliers and distributors. Not only do third parties have access to these, but also often those third parties might even have an interest in, shall we say, extending their access. With these issues addressed, we can then turn our attention to internal system security and continuous monitoring and remediation processes.

Although we target perimeters and networks today, the picture above is a macro view of the entire course. Today, we drill primarily into perimeters and network infrastructure, absolutely critical from a security perspective. Tomorrow we dig deeply into web application security, the public service in our enterprise that is likely the source of greatest risk. The following 2 days we dig into internal system security through Windows and UNIX system security and auditing.



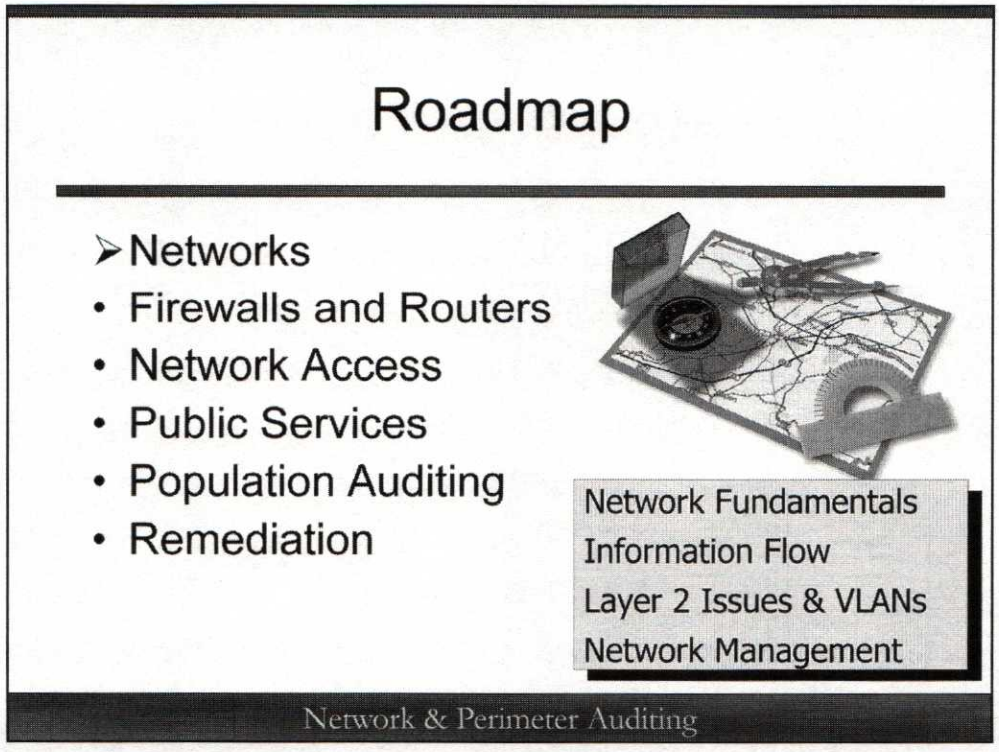
What this should illustrate for you is a critical aspect of that planning process discussed on Day 1. Having a plan is absolutely key to our success as an auditor, as a security officer, as a security engineer, as a system administrator…it does not matter what our job role is.

Take the time to think things out. Strategize. Take into account sources of risk for *your* organization or systems. Identify organizational risks and risks within your business sector. The plan that we outline in this course is such an approach, but you can customize it based on your experience and the specific issues within your enterprise.

As a basic example, I would typically advise organizations that the last thing that they should be looking for on their networks is malware. Hunting for malware often involves running vulnerability scanners in an attempt to identify back doors and other suspicious network services running on endpoints. Although this may initially sound like an important exercise, in the overall context of enterprise security, validating that the firewall is actually protecting us and that the overall perimeter is secure is far more important. Why? Because if we hunt for malware, we will absolutely find stuff! However, what we find will most often not be malware! Instead, we will be spending an enormous amount of time chasing down false positives as a result of one-off services running throughout our enterprise on ports that seem suspicious.

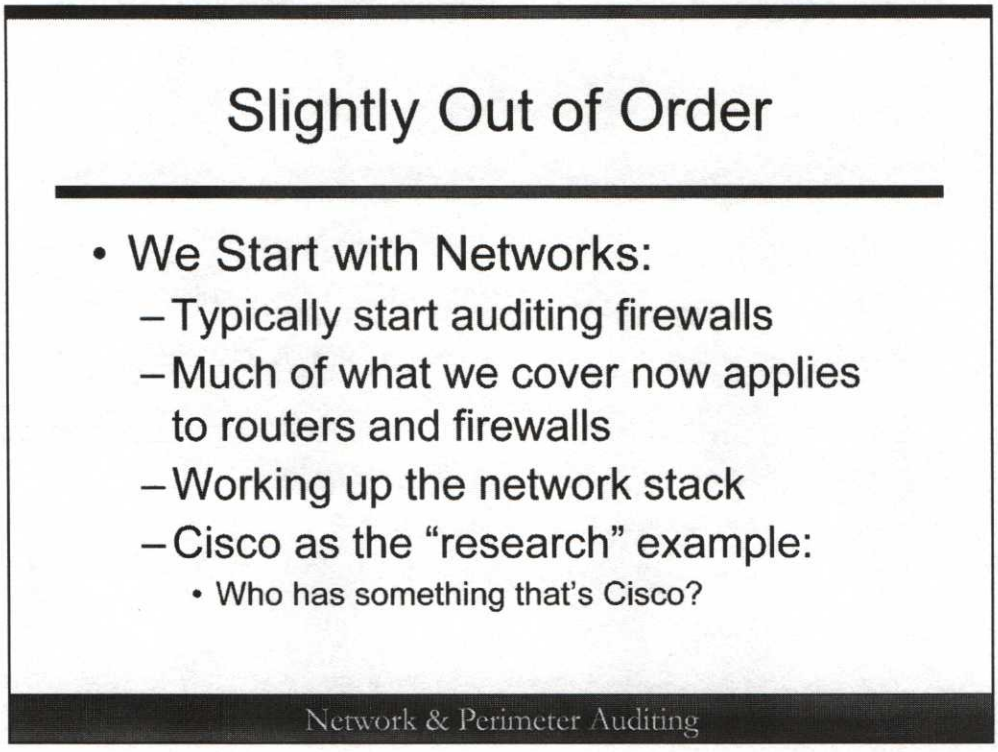
This can actually serve to hinder the overall security process because all the false positives desensitize the organization to the reality of the threats!

However, if I am engaged by an organization that has been recently compromised, the need to hunt for malware rises significantly in importance! It is still critical that the perimeter is secured first, but the next step would likely be hunting down malware rather than dealing with public services.



With this context in mind, our overall plan for today is to begin by digging into networks in general, starting at Layer 2 and then working up the stack. After dealing with Layer 2 and VLAN security, we move to firewalls and perimeter routers, those primary threat mitigation engines in our perimeter. When these are understood, we spend time dealing with other ways the network is accessed and publicly available services. Next, we dig into network population identification and maintenance and connect this to an overall enterprise continuous monitoring and remediation program for vulnerability identification.

Let's get started!



Realize that in terms of risk, Layer 2, Ethernet, and VLANs is not where we'd like to start. We'd like to begin with routers and firewalls because those are the perimeter security devices. However, because we're eventually going to have to discuss Layer 2, we may as well just start here because it is the underlying transport for our firewalls and routers.

Although portions of what we discuss in this section are specific to Layer 2 (things such as VLANs, Spanning Tree, TRILL, and other topics), there are also many things that these systems have in common with all the rest of our network infrastructure components, including our routers and firewalls. Specifically, how they are administered, what logging is enabled, how they are secured, and many other items.

As we cover the material today, we try to avoid repeating ourselves; there is far too much material here to cover any topic more than one time. This means that after we discuss the proper way to manage a generic network device (discussed in the context of switches in this section), we do not revisit that particular configuration requirement in the subsequent technologies. In other words, all the general items that we cover earlier in the day apply to all the other technologies that we discuss later in the day.

It is useful to have a specific kind of system to use as an exemplar of the concepts under discussion. For this purpose we have selected Cisco as the example. Why Cisco? Well, ask yourself: "Do we have any Cisco hardware anywhere in our enterprise?" Chances are that the answer is either "We are a Cisco shop" or at least "Yes, we have Cisco stuff somewhere." Cisco is truly ubiquitous.

Another great reason to pick Cisco devices is that it has an extremely wide range of features. You will hear more about this as we discuss routers and firewalls. They support some of the worse features in addition to having support for some of the best features in the same category. This enables us to discuss all the types of things you will encounter while researching just one type of device!

Ethernet

- You should already know:
  - LAN protocol
  - CSMA/CD
  - IEEE 802.3 standard
- What you may not know:
  - Layer 1 and Layer 2 protocol
  - Almost no security
    - You can't write Ethernet firewall rules

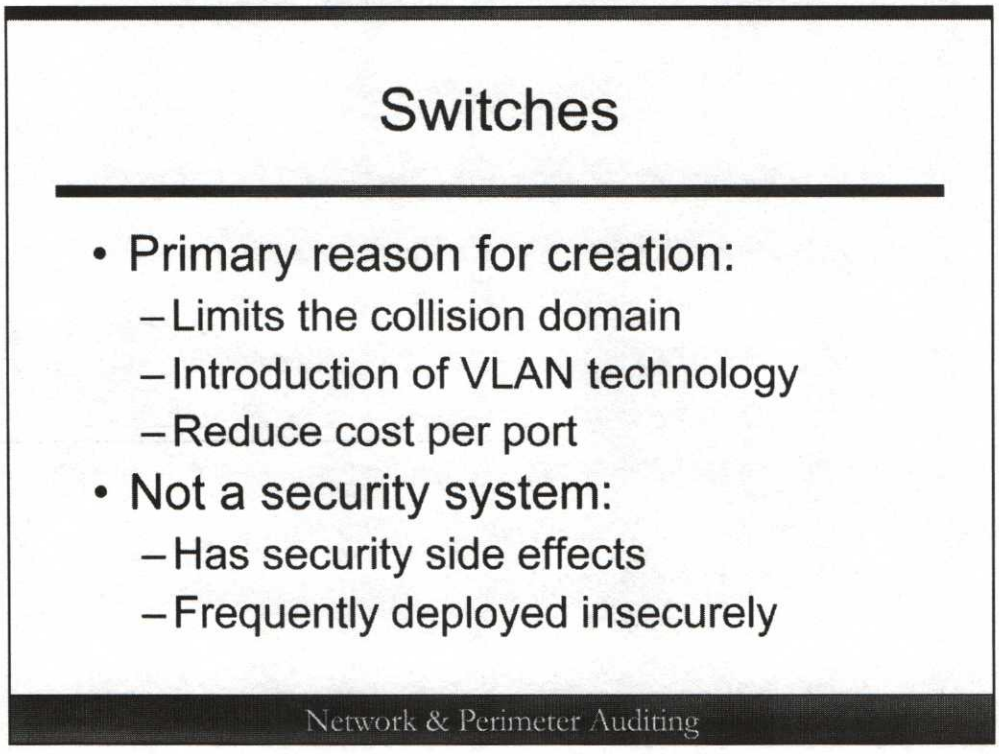
Network & Perimeter Auditing

Let's start with Ethernet and a quick reminder of what you should already know. Ethernet is the most commonly used protocol for interconnecting computer systems. It was originally invented in around 1973 by a gentleman named Robert Metcalfe at Xerox. Ethernet was revolutionary because it allowed for shared network connectivity by multiple machines, which was not a new idea, without an artificial means of brokering communication. Specifically, it allowed for more than one system on the network to have the capability to speak at the same time.

Ethernet, in fact, is defined by this capability. For something to qualify as Ethernet, it must be Carrier Sense Multiple Access with Collision Detection (CSMA/CD). The standard that fully defines Ethernet is IEEE 802.3, which is also the most common encapsulation type used for Ethernet today. There actually are other Ethernet encapsulations that could be used (802.2, SNAP, and more), and you may actually find these on networks that have been around for a while, but from our perspective they all work about the same way.

The biggest thing that we need to know about Ethernet and Layer 2 in general is that absolutely no security controls can be implemented at this layer. Now, it is true that I can configure things like Media Access Controller (MAC) address filtering on switches and other network devices to prevent a system from speaking, but that's about it.

Here's a great way to illustrate what we mean by "no security." Imagine that we configure the firewall on a host to say, "If anyone asks you for anything, don't answer. In fact, you're not even allowed to initiate connections to anyone!" Effectively, we are putting in a DENY ALL policy, both inbound and outbound. When this is configured, a remote host attempts to ping the IP address of our locked-down host. To do so, the host that wants to send the ping begins with a Layer 2 ARP who-has, attempting to determine which MAC has that IP address. Despite all the firewall rules, the locked down host instantly responds, sending an ARP is-at packet back! The firewall is like a castle gate that comes all the way down… but stops short of the ground, leaving a space for things to sneak underneath.



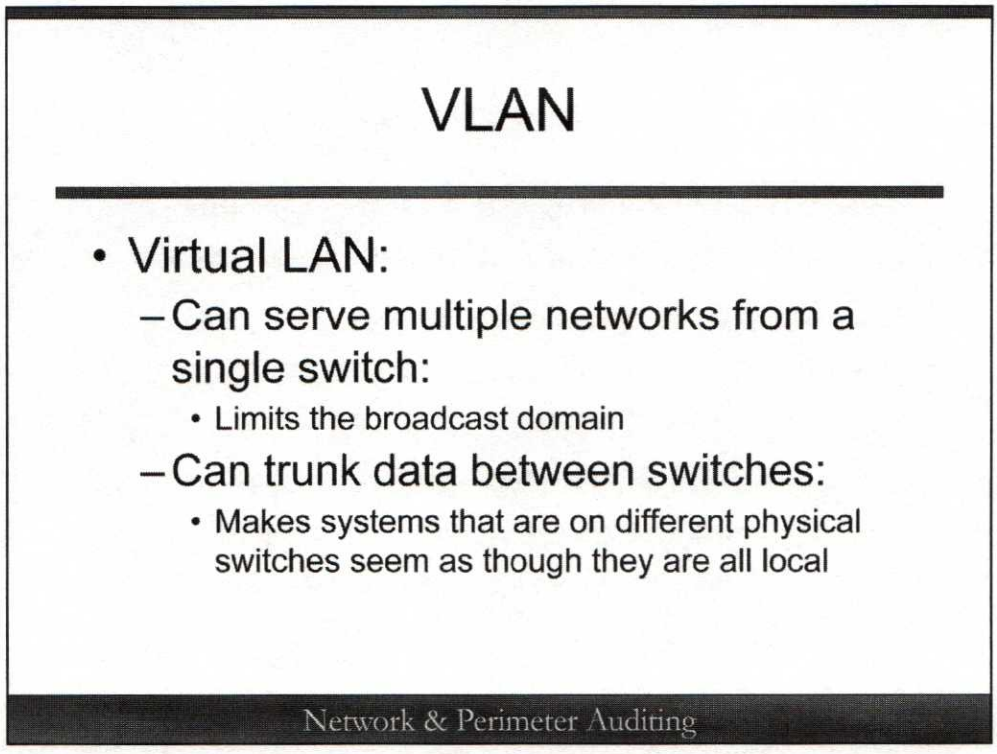
The way that Ethernet is designed, it is intended to have shared media with a shared collision and broadcast domain. Unfortunately, this is not terribly efficient when we start to have a hundred or more hosts on a LAN. While everything will still work well, especially when there is a high utilization on the network, things will get slow, dropping the overall bandwidth that any particular node experiences dramatically.

To improve this, switch technology was created. Switches operate at Layer 2 and segregate the collision domain. Much like an old-time telephone operator who connects one phone line to another with a "switchboard," creating a circuit, the switch creates a packet-switched connection between two points. As you know, this means that packets should be sent only to ports on which the destination MAC address for the packet appears.

Switches were a wonderful advance in networking technologies in the early '90s, but they were significantly more expensive than the hubs that they were replacing. To alleviate this, VLAN technology was created. This is an important point: VLANs were not created to make networks secure or to provide security features. VLAN technology was created to reduce the overall cost per port.

Cost per port is calculated for switches based on the overall cost of the switch divided by the number of ports that are actually populated or in use. If I have a 48-port switch but populate only 10 ports, the cost per port is the total cost of the device divided by 10. VLANs enable us to fully populate a switch while simultaneously segregating the collision domains *and*, between VLANs, the broadcast domain.

This certainly has positive security side effects. Because this feature was not designed as a security feature, however, VLAN technology is easy to defeat from a security point of view. Added to this, an improper deployment makes some type of compromise trivial.



So what is a VLAN? VLAN simply stands for Virtual LAN. In other words, through this technology, we can virtually make it appear that a group of physically separated systems, possibly even connected to different switches in different geographic regions, are all on the same local LAN. This is accomplished by assigning ports on a switch into a specific VLAN. As a point of interest, although we might assign names to VLANs, these names are only available at the management interface level; the technology itself supports only numbered VLANs.

Because the switch can have different ports assigned to different VLANs, at the switch level it is theoretically not possible to have data pass from a port in one VLAN to a port in another VLAN without passing out of the switch and through some type of Layer 3 routing device, like a router. This is accomplished primarily by limiting the broadcast domain, which prevents hosts from discovering one another at Layer 2.

To support the capability of a geographically separated VLAN, or simply supporting more than one switch with ports on the same VLAN, it is necessary to provide a mechanism for the switches to share data. Proprietary solutions to this problem are achieved through the use of backplane connections in a switch stack, effectively turning a stack of switches into a single large switch. All the data for VLANs that must pass to another switch in the stack is trunked over this proprietary backplane connection.

What if the switches are not physically in the same stack? No problem. All our enterprise switches also support the capability to trunk this data "in band." This does not mean that the trunked data is visible to all hosts, but in that we are actually using switch ports to trunk the data to another switch.

This trunking concept is actually important for us because an incorrect configuration can lead to a malicious user tricking a switch into trunking, or bundling, all the data for multiple VLANs to him, providing a wonderful means for data collection and attack.

How It Works

- Inbound packets are tagged using 802.1Q protocol:
  - Up to 4,096 VLANs possible:
    - 2, however, are reserved
- Trunk links between switches:
  - All packets for member VLANs passed between switches

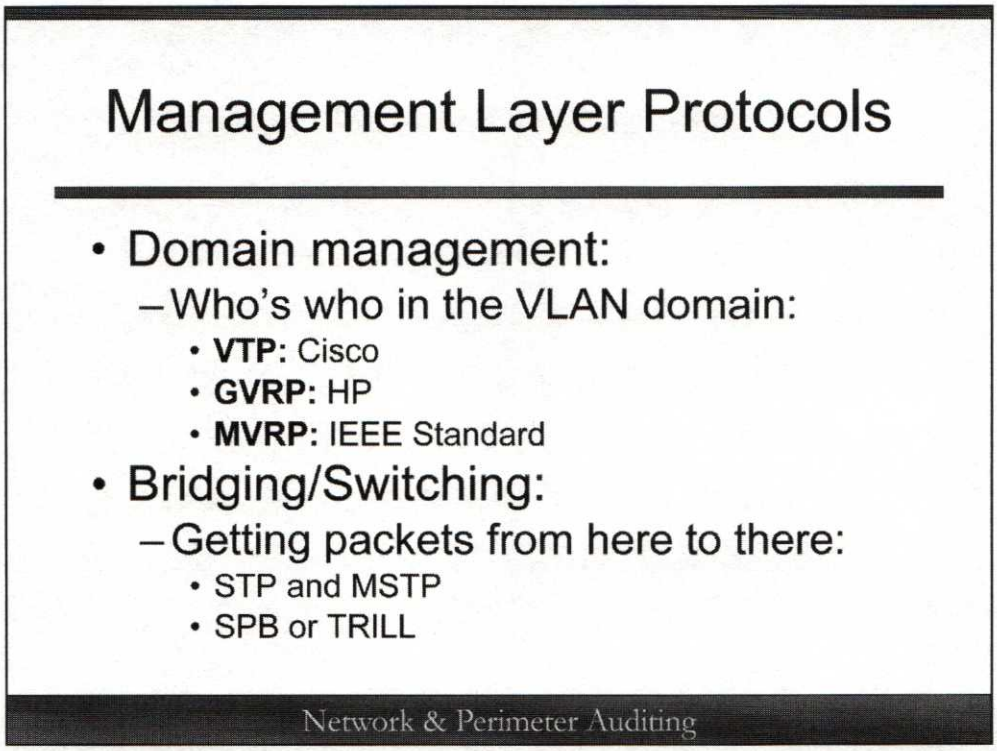
Network & Perimeter Auditing

The way that the switches handle VLANs is by tagging packets as they enter the switch port. The entering packet is tagged with the VLAN id of the current VLAN for that port. This tagging will be done by adding a tag header to the packet, not by modifying the existing packet.

In the Cisco world, the protocol that is used for this is 802.1Q. Although not all vendors use 802.1Q, they all use some protocol that essentially accomplishes the same task. One of the things that's so interesting about this mechanism is that the switch actually does not keep track of which port the packet entered the switching fiber through. In fact, it creates a potential avenue for an attacker to "hop" VLANs.

Hopping VLANs is usually accomplished by taking advantage of trunking. Trunking describes a set of ports that are interconnected between two or more switches over which all the VLAN traffic is bundled between the switches. If a user workstation manages to convince a switch that it is also a switch, it is possible for the user system to select which VLAN it is on because it can pre-tag the data before handing it to the switch. Because there is no validation or tracking for this tagged data, the user can essentially choose which VLAN to join.

Of course, if the user can convince the switch that he is a switch, he can also subscribe to multiple VLANs, convincing the switch that he has ports on those VLANs. This will convince the switch to begin trunking all the VLAN data to the attacker which allows for easy sniffing.



When we have two or more switches configured into a VLAN domain or VLAN Trunking Domain, there needs to be a single switch that acts as the master or manager of the domain. The job of this system is to keep track of who's who and to define the proper path for packets to take through the switching fiber. Why? Because using these management protocols it is possible to completely interconnect all the switches, which allows for full redundancy even if one or more switches goes down.

The primary management of the VLAN domain is done using a protocol such as VLAN Trunking Protocol – Cisco (VTP), GARP VLAN Registration Protocol – HP (GVRP), or Multiple VLAN Registration Protocol – IEEE Standard (MVRP). This protocol is used to track which VLANs exist and can be secured to require authentication or other requirements to join the trunking domain.

After the switches are connected, a secondary Layer 2 management protocol such as Spanning Tree Protocol or Multiple Spanning Tree Protocol (STP or MSTP) are used to control the path through the switching fiber. Alternative protocols that accomplish essentially the same task are Shortest Path Bridging (SPB) and TRansparent Interconnection of Lots of Links (TRILL). Of these, TRILL and SPB are the newcomers on the scene. We are waiting to see which of these two turns out to be the new "right way" to do things. Currently, Cisco fully supports TRILL with no support for SPB. However, the rest of the industry seems to be leaning toward SPB. We currently seem to be in the Blue-ray versus HD-DVD limbo, so we'd advise hardware that supports both.

The reason that we care about all the details, though, is that many organizations are relying on VLAN features to provide secure network segregation. The risk is that if there are any misconfigurations, compromising VLANs is exceptionally easy. Add to this some of the natural behaviors of switches and the problem is exacerbated.

The three most common configuration issues are a failure to configure a password to secure the trunking domain protocol, failure to remove Layer 2 management protocols (VTP, STP, MSTP, TRILL, and so on) from user facing ports, and failure to secure the configuration of the switch. The first one is obvious. The trunking domain has to have a password configured. Frequently it is not because the administrators don't see a need to configure it; their perspective is that the users can't see it or get to it, not realizing how switches and VLANs can be attacked.

The second issue is fairly easy to resolve; though, sometimes other network configuration issues prevent us from fixing it properly. For example, if we use a "Converged Network" configuration with a VOIP handset, we must have the Layer 2 management protocols enabled on user-facing ports. We'll give you a specific way to test for this shortly. We will also give you details of specific checks for the third issue.

The last bullet on the slide, however, makes an interesting assertion. Even features within switches that have been specifically designed as security features are easy to defeat. We'll show you an example in a few minutes when we discuss a Private VLAN Bypass attack, but this should raise your awareness that there are serious security concerns at Layer 2!

Audit Items

- How is it managed?
- Logging appropriate?
- Is it up to date?
- What services does it offer?
- Only required ports are active?
- VLANs properly secured?

Network & Perimeter Auditing

Now that we have some foundation, let's start dealing with auditing the device. Although some of the questions that we ask here are specific to switches, (for example, are the VLAN's properly secured?), others are applicable to all systems that we cover today (for example, is the logging appropriate?).

Before we jump into answering these questions and doing the research to see what the answers might look like, I'd like to remind you of the value of conducting interviews. Even the questions above form an excellent basis for such an interview. Let's just take one of these and examine what that interview might sound like.

We would provide the administrators who will be interviewed with a set of questions much like what is listed above. We might indicate in an introductory paragraph or cover letter that we would like the administrator to come to the meeting prepared to discuss the questions posed. An example of how to word a question might be, "If VLANs are used in the network, what steps have been taken to safeguard the network from attack or manipulation?"

The interview is not an intense grilling. It's a friendly discussion. It allows us to get a feel for the approach that the administrator and his team take toward the issues presented and can give us insight into items that we need to examine more closely. It also allows us to identify issues that we don't need to spend time on during the time spent at the console with the administrator.

What would a proper configuration look like for issues likes these? How can these controls be circumvented? Let's examine these questions.

Management

- Managed over a secure channel:
  - SSL
  - IPSec
  - SSH
  - SNMPv3:
    - Be careful here!
  - Dark network

Network & Perimeter Auditing

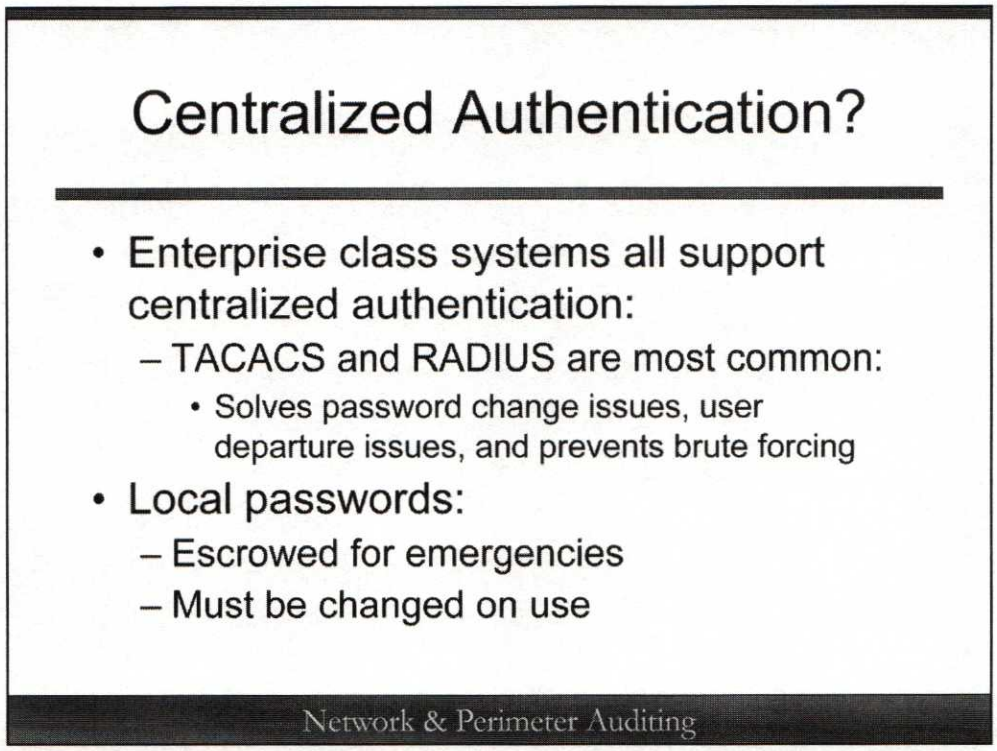
For a network device, firewall, router, or switch to be managed properly, we would expect the management to occur over a secure channel. Our preference would be to find that a strongly encrypted protocol or connection is used to accomplish this. SSH and IPSec are common solutions to facilitate this.

At times we may find that SSL is used. This may raise a red flag for you. If the device in question is something like a wireless access point, this may not be a problem. If the device is a switch or a router, it likely is a problem.

Even though these devices generally provide a web-based interface for administrators, almost all seasoned administrators will tell you that if you use the web interface to manage a router or switch, you are doing it incorrectly. The reason is that the web interface represents a potential vulnerability in addition to the fact that many of the most important security capabilities of the device are simply not configurable through the web interface.

Another possible problem is that the device is an older piece of equipment and does not support a strongly encrypted link. This issue has become more rare to see with network devices, but it remains a fairly common occurrence with legacy systems. In this case, a good approach is to create a private LAN, a dark network, that is not routed or reachable from the production network. All the management interfaces for these legacy systems are connected to this dark network along with a management workstation or gateway. The administrators can now use a secure protocol to connect to this gateway and then use an insecure protocol to perform the actual administrative tasks in a safe way.

SNMP could potentially be in use for managing devices. We will expand on how this should be configured in a little while. For right now we'll just say that there are more questions to ask if SNMP is used for management.



For managing the device, it is not sufficient for it simply to be over a secure channel. We would additionally like to find that strong password requirements are enforced and that users are distinguished from one another in the logs. The trouble is that almost all network devices on the market do not support these types of features. Even if they support the creation of separate user accounts and passwords, none of them have password strength, length of expiration enforcement.

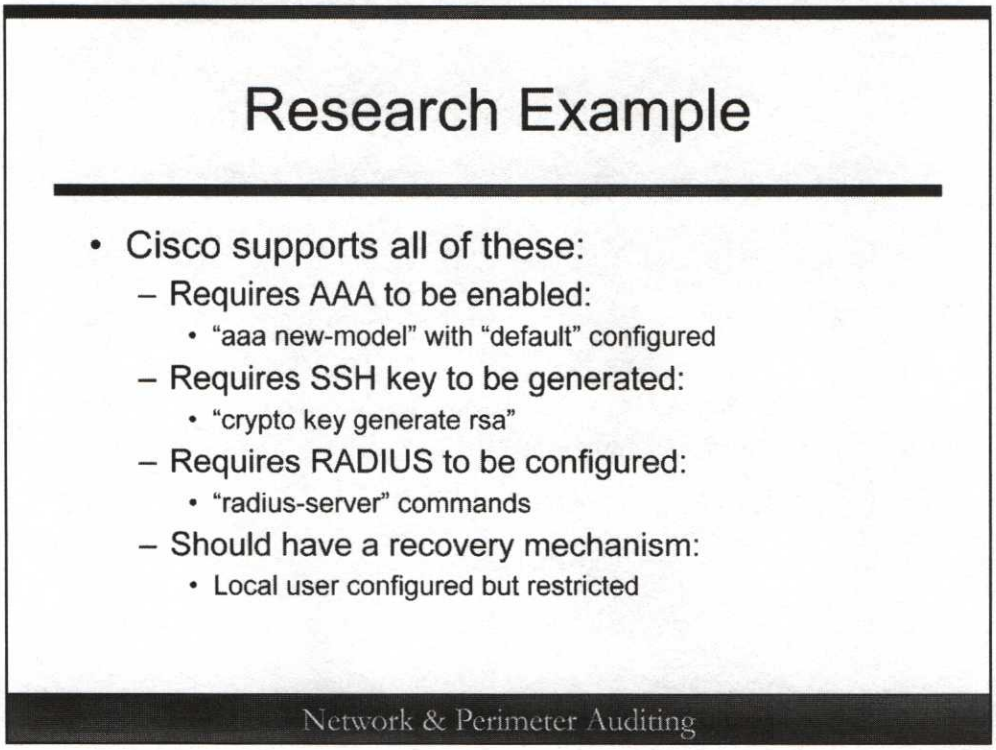
For this reason, centralized authentication must be configured. This allows us to reuse a trusted component in our infrastructure. In this case, that component will be a trusted credential store of some kind, most likely an Active Directory domain.

Using the Remote Authentication Dial In User Service (RADIUS) service from a domain controller, we can create a domain level group that is authorized to log in to routers. With this configured, and with the network devices configured to use RADIUS, each administrator must authenticate using his domain credentials when managing a switch or router.

This allows us to enforce strong password requirements and password change requirements, and vastly simplifies the process when an administrator leaves the organization. Instead of having to change all the infrastructure passwords (which is rarely done in practice), we simply need to disable his domain account.

This does not mean that there is no place for local passwords on these systems. There should be an emergency account configured with a long and strong password that is escrowed for emergencies. If this password is ever used, it must be reset and re-escrowed.

TACACS is simply an alternative authentication protocol that typically runs over port 49. Cisco has created its own proprietary extensions (TACACS+), so you may find this solution in use in some environments.

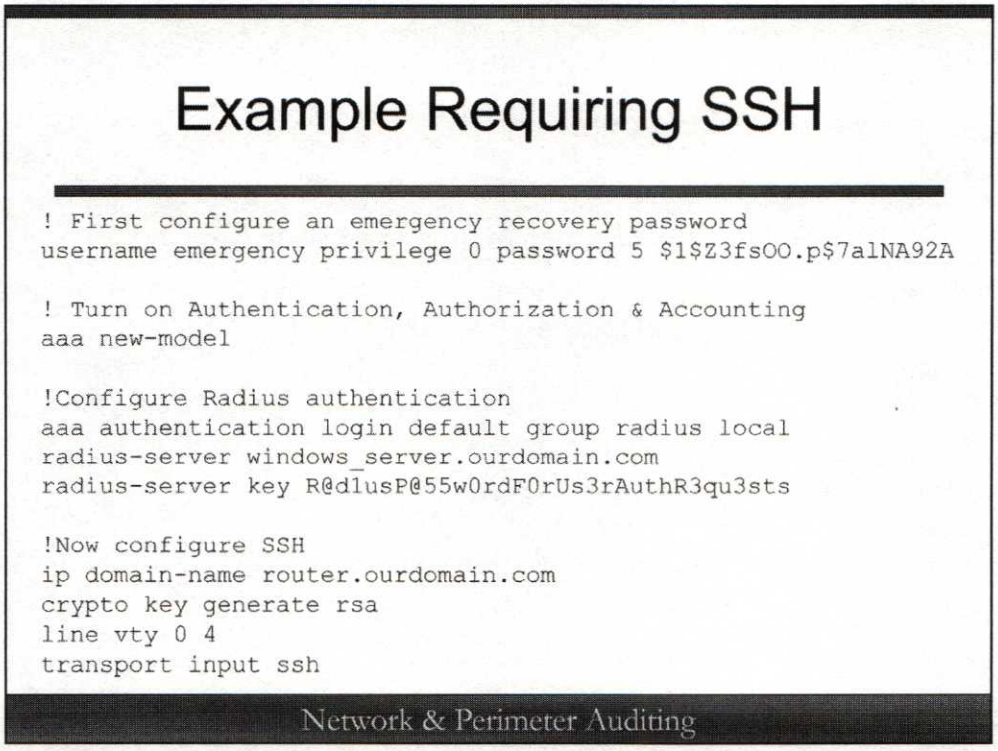


Although Cisco supports a Terminal Access Controller Access Control System (TACACS), they also support RADIUS. Because RADIUS is more commonly used for this type of management, we'll look at how a RADIUS configuration might look.

To use RADIUS (or TACACS), the Cisco device requires the Authentication, Authorization, and Accounting system (AAA) to be enabled and properly configured. For the centralized authentication to function, we need to include RADIUS configuration options as well. We'll see an example on the next slide.

Turning on the authentication is one thing. We still need to ensure that a secure channel is in use. Because the most common approach to this across vendors is to use SSH, we'll include that piece of research in our Cisco example. Using SSH on a Cisco device requires that the device has an SSH key. This can either be statically configured or we can tell the device to generate and store an RSA key for use in SSH sessions.

Finally, we would like to see that an emergency password for recovery has been configured and the password escrowed. Let's see what this all looks like.



Let's examine and explain this configuration one piece at a time. As we do so, do not view this as a memorization exercise! Instead, consider whether you can figure out the gist of what the individual elements mean without an explanation. Take a moment now and just read through the lines in the slide and see if you can figure out, generally, what is happening. Can you identify all the pieces from the last slide?

Starting at the top, the username line creates a local user on the network device named "emergency" and sets the password. This emergency account is configured at privilege level 0 (no privileges) and has an MD5 password hash (specified by password 5). In other words, the password is not readable. This is good! We should not read plaintext passwords in the configuration here.

Next, we see the AAA being enabled, one of our requirements from the last slide. Following this, the 'aaa authentication' line configures AAA to use RADIUS as the primary authentication mechanism for logins. If, for some reason, the RADIUS server is unreachable, the fallback method of 'local', the emergency username and password, may be used. Another item checked off.

Following this we see the device name configured and an SSH host key generated, allowing SSH to be used. This is not all that is required, however. Note the configuration that begins "line vty." This configures the virtual terminal, or remote login, service. Immediately after that we find the line 'transport input ssh.' This is the line that forces SSH to be used and is what disables the use of telnet.

Hopefully, you deciphered the general outline of what was happening even without the description. However, even if you couldn't, you can likely see that we have met all the requirements from the last slide! The things that help us out here are the comments. Not every line requires a comment, but we should expect administrators to document configurations. This is a primary way that this is done!

Credentials Secure

- Verify that any local credentials on the host are securely stored:
  - Cisco supports two forms:
    - Type 7
    - MD5 hashes:
      - The example on the last slide is MD5... Why?
  - Type 7 is extremely basic!
    - Can be broken by hand
    - A lot of websites do it for you instantly
  - There are some we can't fix. Like the RADIUS server password on the last slide

Network & Perimeter Auditing

Even though we use centralized authentication, we are still configuring some local credentials for emergencies. It is also common to configure the device so that logging in to it requires centralized credentials but then moving into the administrative mode with the enable commands requires that we enter a local password that rarely changes. Whatever the case, those local passwords must never be stored in plaintext. There are, however, two other options for storing these passwords, one if which is secure and one if which is not.

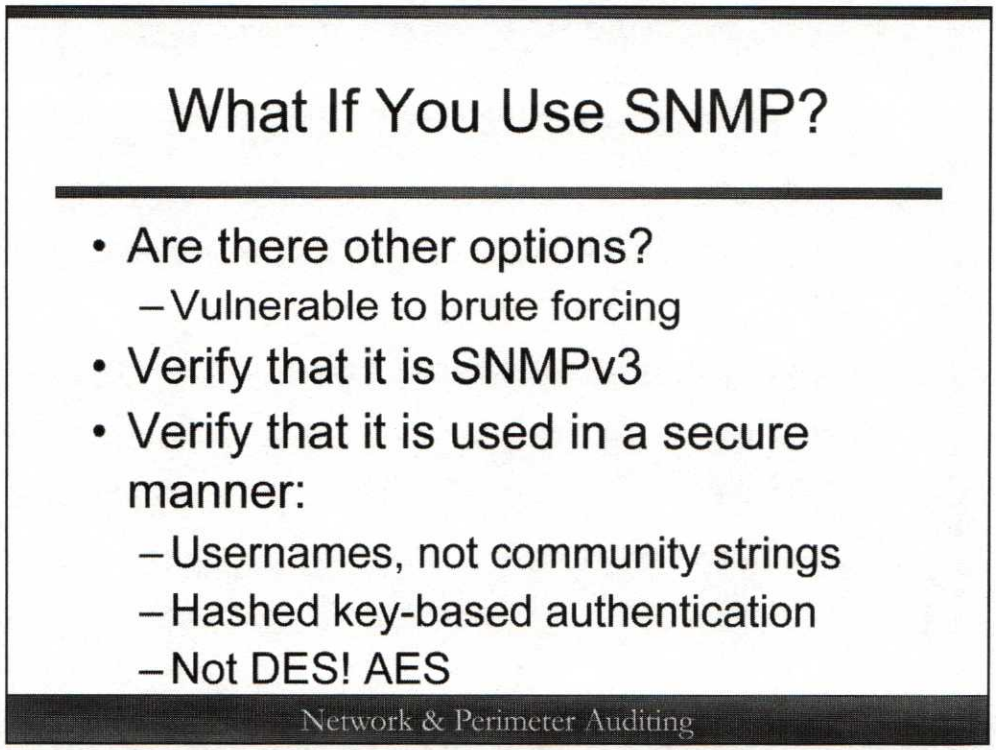
Cisco Type 7 passwords have been a part of the Cisco system for a long time. These passwords are not strongly encrypted; though, they aren't in plain text. In a Type 7 password, the first bytes are used to specify an index into a known string of characters. Using this value, the remaining values in the password are then manipulated to produce the plaintext. This is so basic that you can actually work out these passwords by hand. To make life easier, though, there are all sorts of websites out there that will decrypt these passwords instantly for you for free!

Type 5 passwords are much stronger. Rather than using a home-grown encoding algorithm, these passwords are hashed using MD5 with a salt. Although MD5 hashed passwords can certainly be broken, including the salt means that it will likely take some time for an attacker to break this password.

Of course, if an attacker obtains a copy of the password hash, we always have to change the password immediately. This is true of any system that we have. The true value here is that it will not be broken instantly!

The configuration on the last page uses an MD5 hash. Can you see how we know this? Notice that the configuration of the emergency account includes "password 5." That 5 is what indicates that an MD5 hash is in use. If a 7 appears, it's a type 7 password. If a 0 appears then the password is in plaintext.

Be conscious of the fact that there are some things that just can't be fixed. For example, in the last slide you may have noticed that we can read the password that this device uses to authenticate to the RADIUS server. Unfortunately, there is just nothing that can be done about this. The router/firewall/switch needs to speak to the server (Active Directory in this case) and to do so it must have credentials. The only real mitigating factor here is that the password that we can read isn't the password to authentication to this network device.



While we're on the topic of remote management of a network device, let's revisit SNMP. As was mentioned earlier, if there is another option for managing the device, we'd probably prefer to use that. SNMP is not an especially secure mechanism and the fact that it operates over UDP makes it much more susceptible to replays and impersonations.

If, however, we use SNMP to manage devices, verify that the systems are properly using the SNMP version 3 security features. When version 3 was formalized as a standard in 2004 (RFC 3411) it added some important security capabilities. Specifically, confidentiality was addressed by providing encryption capabilities, integrity was taken into account by adding message hashing for validation, and authentication capabilities were added through message signing. Even so, just because these features are available does not mean that they are configured. Even if they are configured, it does not mean that secure options have been selected!

For example, under SNMPv3 you could choose DES as the encryption method. As you are likely aware, however, DES is not a useful algorithm for encrypting these days unless we are going to have multiple rounds with multiple keys. Another option is AES. If AES is used, we can have some confidence that our messages are secure. Similarly, while authentication can be enabled, whether or not that will be a plaintext username or a hash-based authentication is configurable!

They key to knowing if it has been done correctly is to remember what we set out as the objective of this control. We said that remote management of the device must be protected against interception, manipulation, and impersonation. This would mean that we need all these security features enabled or it is not well secured.

Logging

- Centralized (typically syslog):
  - Timestamps synchronized
  - Informational recommended
  - Analysis and reporting
- Console:
  - At least warnings:
    - Frequently disabled for admin convenience

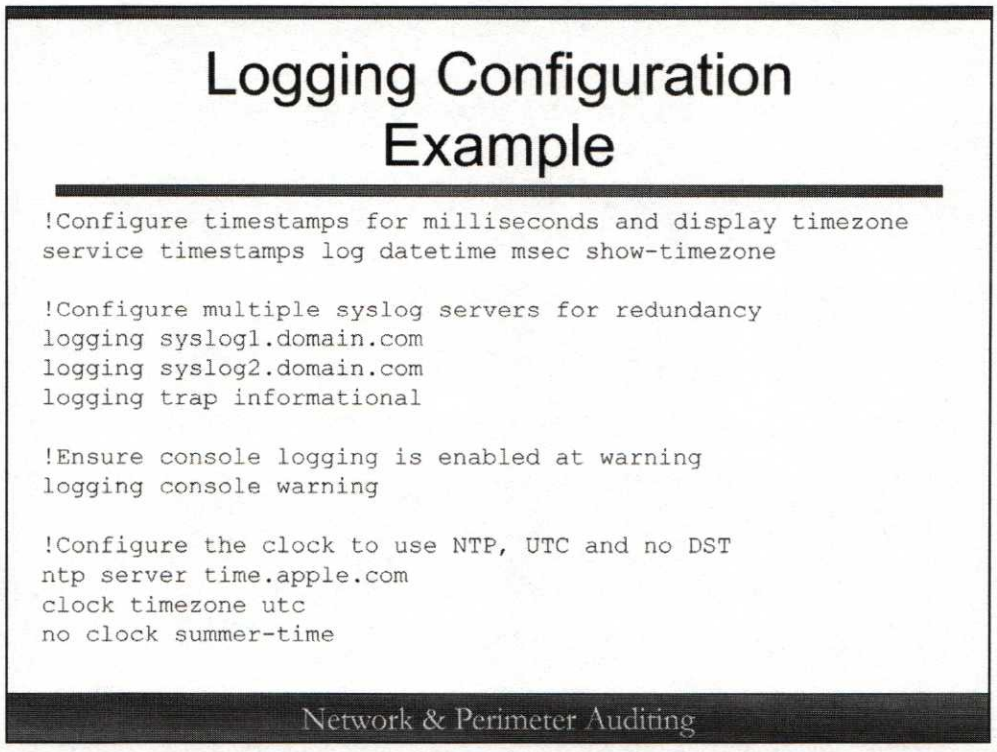
Network & Perimeter Auditing

Logging must also be enabled. Without proper logging we not only lose the ability to detect and react to faults in the logs, but we also lose the ability to account for administrator activity within the system. With logs enabled it is much easier to compare the administrator activity to authorized change control. Without it, everything must be inspected every time.

Proper log configuration would mean that the logs are being centralized. Network devices of the sort that we are discussing today all support syslog style logging. Some have additional support for more secure forms of logging. If we have these additional features and choose to use them, that's wonderful. Even without them, however, simply having the logging enabled and configured to centralize the messages is sufficient in almost all cases. Frankly, securing the logs at the network level isn't our biggest problem. Much bigger problems are getting the logging centralized and then getting someone to look at the logs! Therefore, we will want to not only verify that the logging is enabled and centralized but that there is also a system or process for regular review of those logs.

When configuring logging there are a range of log settings that control which messages are tracked. These range all the way from emergency events down to debugging events. For remote logging, Informational is a good selection. This will include important information about configuration changes in addition to warning, critical, and emergency events.

We should also verify that the console of network devices are also configured to generate logs. It is not unusual for an administrator to disable console logging to prevent log messages from cluttering the screen while he is troubleshooting or reconfiguring the device. Unfortunately, this means that if he inadvertently breaks something while he is there, he is unlikely to realize it immediately unless it affects the issue that he is working on at the moment. If this is set to Warnings, he should see events like routes dropping, adding, interfaces going down, and more.

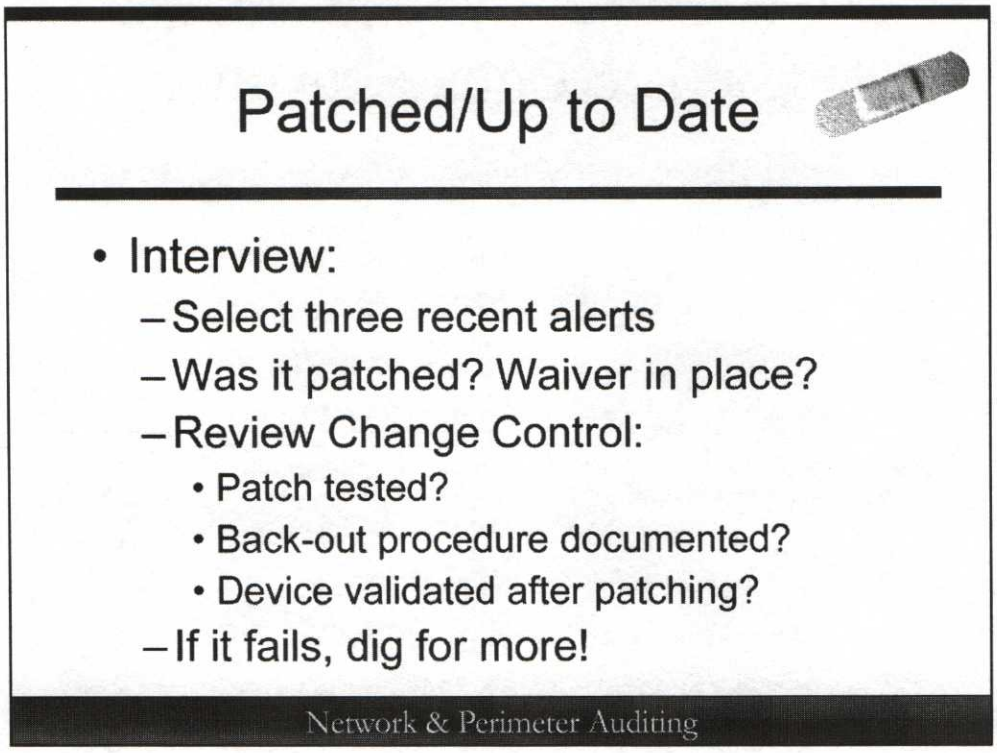


So what would all this look like within a device? This slide shows an example. Just as we did previously, start by looking over the configuration and seeing how many of the items from the last slide you can identify or check off.

First, the log timestamps are configured to include millisecond resolution and the time zone of the system they were collected on . When it comes to the timezone and the actual time, all the clocks for our entire infrastructure must be synchronized to a common source. NTP is the most common way to do this. In addition, because these are simply text based logs, we strongly recommend that you select a single time zone and have all systems configured to this. This will have no impact on users! The time that they see is localized on their desktops.

With that configured, the next several lines configure the device to forward all the log messages to two different remote syslog servers. This provides redundancy should one server be unavailable. In addition, we can see that the "logging trap" sets the system to send informational messages to these services. "Trapping" means that we are sending events. Compare that to the next configuration line. "Logging console warning" configures the console so that all warning messages appear there, exactly as we specified.

The last few lines are configuring the time synchronization and the time zone. Note the final line. For Cisco systems, this line tells them that daylight savings time is not observed. This is an extremely important setting. Without this setting, not only will our timestamps adjust automatically, but also we end up with a troublesome problem of not knowing when they will change unless we actively tell the system when to start and stop daylight savings time. Best practice is to simply disable daylight savings time to keep the logs all synchronized.



Returning to our interview process, we want to inquire about how patch management is actually done. Of course, we are already familiar with what the policy and procedure documents say; what we want to see is how well the administrators know this process and to hear whether they have followed it.

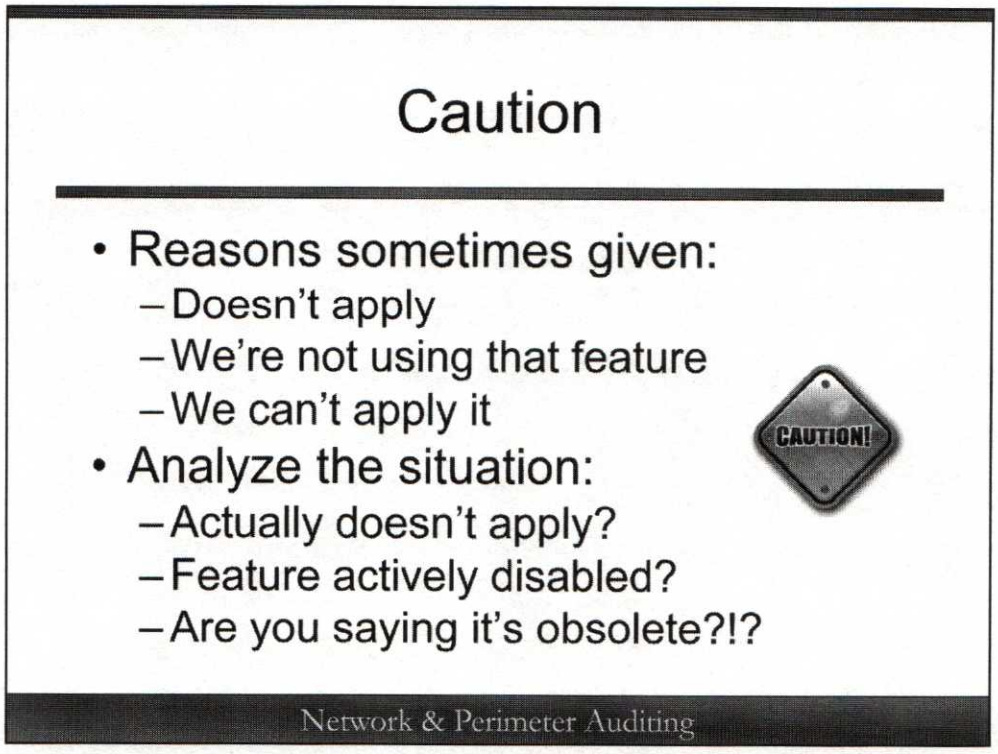
To facilitate this discussion it is good to select three recent vendor notifications that are relevant to the infrastructure and should have been addressed. Ask the administrator to bring with him to the meeting any documentation related to those specific notifications.

During the meeting we are looking to answer a few simple questions:

1. Was it patched?
2. Was it patched on time?
3. Was there a testing process before it moved to production?
4. Was there a back-out or recovery procedure in place?
5. Was it properly authorized and documented?
6. If it weren't patched, is there an appropriate waiver in place?

What if the administrator cannot produce certain pieces of documentation? If this is the case, then we would want to let the administrator know that we will follow up with questions about additional issues that should have been addressed when we work with him to validate the device. In other words, if there appear to be issues, we look for more issues. If everything looks fine, we move on to other areas.

During the technical validation we are looking for the administrator to demonstrate that the patches were actually applied. At times we have found all the documentation in place but the patches missing.

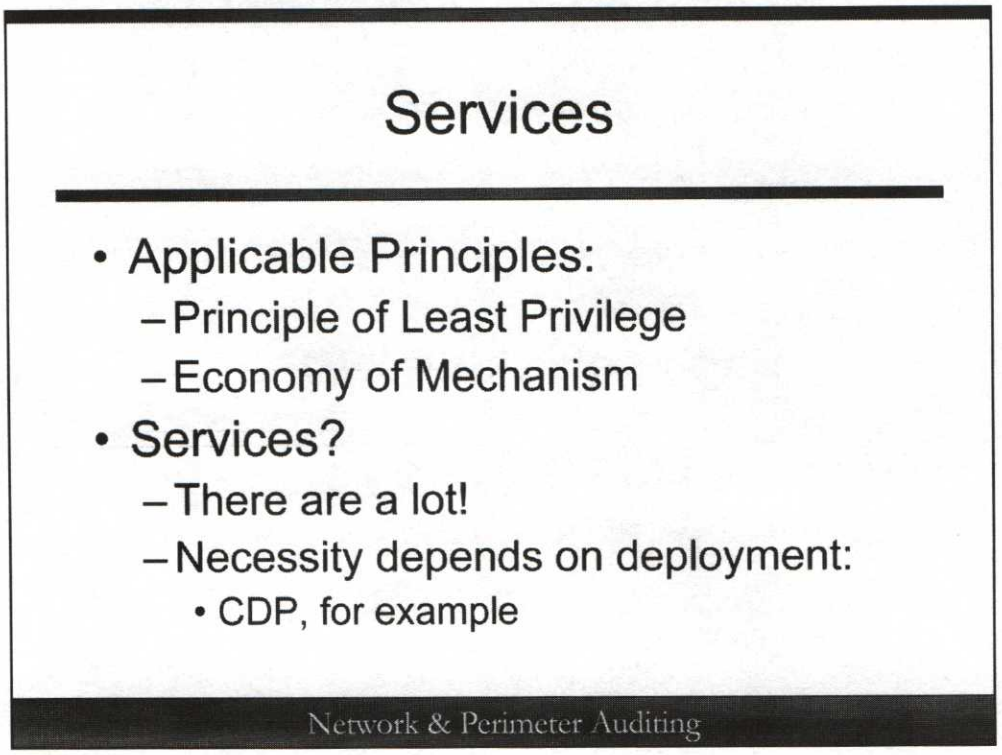


As you work with the administrator on this process, we want to caution you about some things that you might hear. To be clear, we're not saying that these are invalid claims, but we strongly suggest that you analyze the situation in context before accepting or rejecting any of these (or similar) situations.

You may hear the administrator say, "That doesn't apply to us." If he makes this claim, he should clearly explain why that is the case. For example, imagine that there is a vendor notification that the device, a switch, is vulnerable to a VLAN tagging attack in which a user could attach VLAN tags to the packet before handing it to the switch and thereby hop from one VLAN to another. We inquire about the device and the administrator says, "No, we haven't applied that patch because it doesn't apply to us." "But we have that switch," you think. What if the administrator explains that as a result of that alert the decision was made to not rely on VLANs with that particular brand of switch. Instead, the switches have all VLAN capabilities disabled and are used to physically segregate LANs. Obviously, that specific patch would not be required in this case.

A similar response might indicate that the feature isn't in use. If this is the response, we should seek evidence that demonstrates that the feature has been actively disabled. If it is not, that feature may end up being activated by some future patch.

Perhaps the most concerning response, however, is "We can't apply that patch." In fact, the administrator may even have a waiver for this patch. The waiver means that the administrator is off the hook, but we are here to advise on risk. If you perceive that this is a critical risk, dig into why it can't be patched. Is the system obsolete? Is there some other configuration or design error? Is there a way to address this or otherwise mitigate the risk? Especially if we're talking about obsolete systems, make sure that management understands that waivers don't eliminate risk! The cost of upgrading that hardware will often be far less than the overall impact of the risk.



All network devices support ancillary services. If you have a web server, no doubt there are a number of services that are also installed that have absolutely nothing to do with serving web pages. Needless to say, any service that is not required for a system to perform its primary mission must be disabled or it creates a vulnerability vector on that system.

Switches, routers, and firewalls are no different. They all support many, many services, the vast majority of which are not necessary for the system to achieve its goals. Therefore, according to the principle of economy of mechanism, all of them should be actively disabled. What if there is a service that is required? No problem. However, we would want to verify that the principle of least privilege has been applied in this case. In other words, verify that the service is running only with the rights required for it to do its job. Of course, it will work when it runs with the highest privilege, and that is the easiest way to run it, but this means that any vulnerability in that service will lead to a compromise of the highest credentials available.

This is another situation in which there is no "one size fits all," even within a single organization. We have to look at exactly how the device is used and what requirements exist. For example, the router interface that faces the server segment likely does not need the BOOTP agent running. The same would be true of the core routers. The routers facing the user segments, however, likely do have the BOOTP agent operating as a forwarder for DHCP requests.

Similarly, CDP in the Cisco world should be disabled because it broadcasts a lot of configuration and VLAN information in clear text on the network. However, if you use VOIP with a converged network configuration, you *must* leave CDP enabled to allow the phones to auto-configure. Again, everything unnecessary disabled, but make sure that you actually understand what is necessary and what is not!

Actively Disabling Services

```
!Disable unnecessary services
no service tcp-small-services
no service udp-small-services
no ip finger
no ip bootp server
no service dhcp
no mop enabled
no ip domain-lookup
no service pad
no http server
no http secure-server
no service config
no cdp run
no lldp run global
```

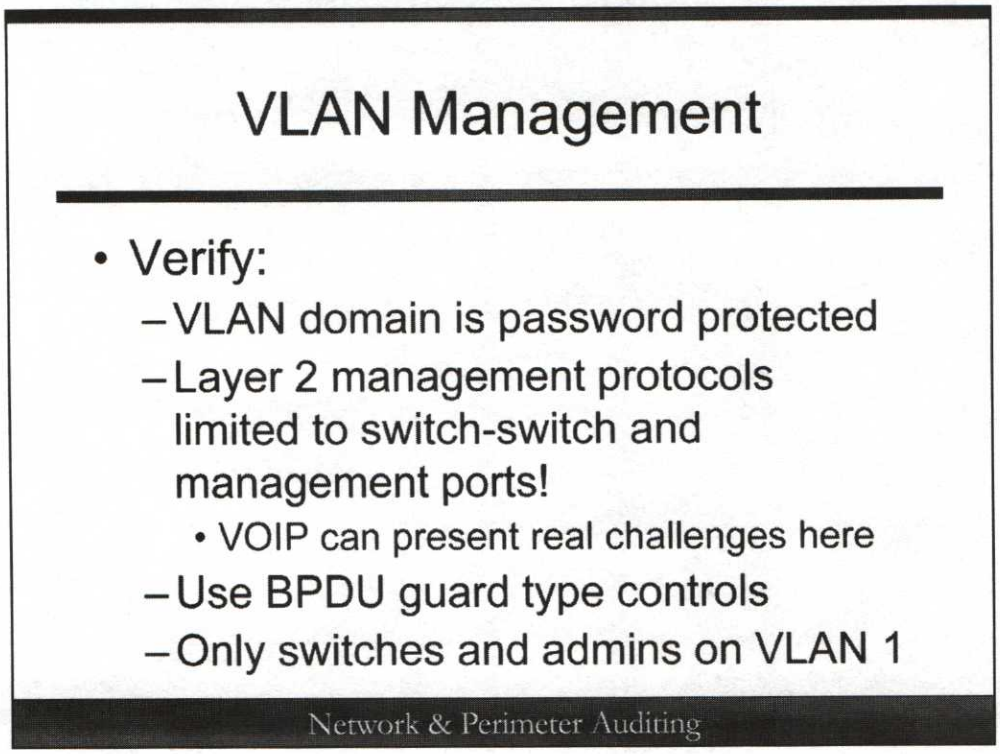
Network & Perimeter Auditing

What might this look like in the configuration file of the exemplar device that we use for our research? The slide above illustrates this. We can see that all the small services (echo, chargen, and so on) have been disabled on both TCP and UDP. The Finger service, which allows a remote user to request user information and see who's logged in, is disabled. BOOTP and DHCP are also turned off, as is MOP.

If you're not familiar with Maintenance Operations Protocol (MOP), don't worry too much about it. It is rarely found today, but in a legacy environment with, perhaps, VAX systems you may find that satellite VMS systems use MOP to perform their initial boot but like a BOOTP based system might.

For Cisco devices we also see DNS services turned off, the web service disabled, the remote configuration service turned off, CDP disabled, and LLDP disabled.

Everything that we need, nothing that we don't!



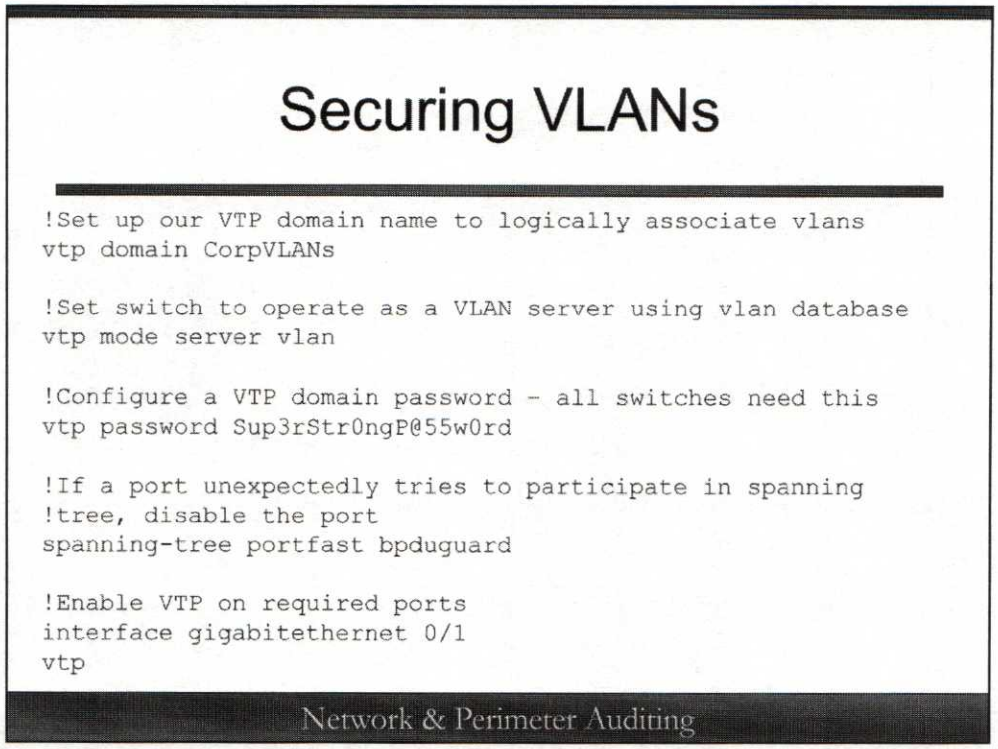
With those general device configuration issues addressed, we can now turn our attention to switch specific issues. Essentially, we want to find that all efforts have been taken to defend the device against the types of attacks and issues previously mentioned.

Recall that we said that the biggest problem is that no one has bothered to secure the VLANs. This is actually one of our first questions: Has a strong password been configured to protect both the configuration and the VLAN domain?

The other enormous issue is when we have Layer 2 management protocols accessible or enabled on user facing ports. As we said, these protocols should only be available on ports that require them. This typically means only ports that are facing or connected to other switches. This is one of the most important configuration settings to secure a VLAN environment. Consider this. What if there were no VTP domain password configured, but user facing ports had all the VLAN management protocols disabled? This would make it vastly more difficult to attack. Unfortunately, how our network is used can prevent us from doing this. Again, VOIP with network convergence will *require* that these management protocols are enabled on user facing switch ports. ☹

Especially in these cases (but even if this is not an issue that you have), it is useful to have BPDU protections enabled. Bridge Protocol Data Unit (BPDU) controls enable us to limit which switches can actually be members of the VTP domain. In addition, it can enable us to lock down the root of the STP or MSTP so that an unauthorized device cannot take control of the switching fiber. Without a control of this sort, an election can be triggered and the device with the lowest MAC address will become the new root.

An additional protection is to keep users off of VLAN1. Only switches and administrators should be on this VLAN. Switches more or less "reserve" this VLAN for themselves.



Let's see what this would look like in the real world. Again, look over the configuration and see if you can see the elements that we are looking for being configured. After that, read on for an explanation of what you see.

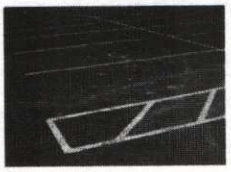
First, a VLAN Trunking Protocol (VTP) domain is configured, giving the domain the logical name CorpVLANs. Next, we configure the switch to act as a server, housing the VLAN database, within the VTP domain. In addition, a password is configured. All switches within the VTP domain need this password to exchange data. This password is used to authenticate VTP packets and serves to prevent a user from injecting any kind of configuration packet to interfere with the proper operation of the domain.

The next line configures one of the BPDU guard options; Spanning Tree, a protocol for deciding the path a packet takes through the aggregate switching fiber, needs to be defended. This line says that if spanning tree packets appear on any port that is not configured to be connected to a neighboring switch, that port should be immediately shut down. This is an effective way from not only preventing corruption, but also stopping an attack in its tracks!

Lastly, we're looking at the configuration of just one of the switch ports. You can see vtp listed, which enables the VTP on that specific port. This means that the Layer 2 management protocols that we are talking about would be enabled on this port. This would also imply that this port is connected to another switch within the VTP domain. Easy!

Further Securing Ports

- New switch: All ports VLAN 1
  - Process should be:
    - Secure configuration
    - Create "Parking Lot"
    - Move all ports to parking lot VLAN
    - Allocate ports to specific VLANs as needed
  - Disable all unused ports
  - Enable port security functions



Network & Perimeter Auditing

Let's revisit VLAN 1 for a moment. This was mentioned on a previous slide and received only a sentence or two of explanation. VLAN 1 is the default VLAN for all the ports on the switch unless you actively configure them to be on some other VLAN.

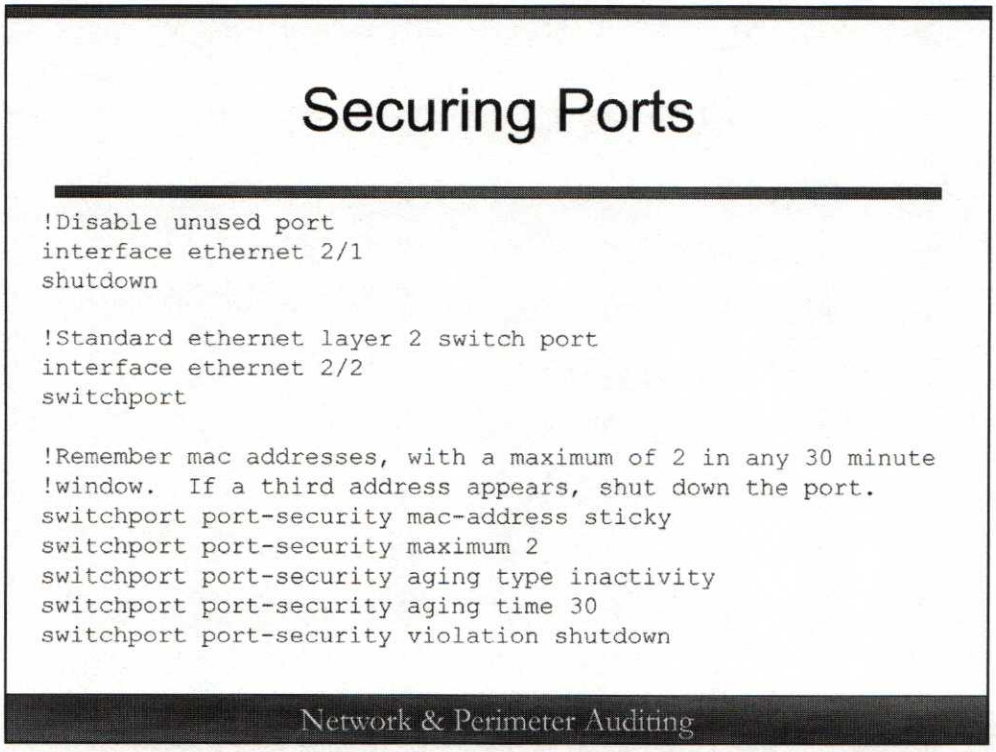
As a result of this default behavior, it is not unusual to find a switch where either all or at least many of the user ports are on VLAN 1. Why? Because when the switch was first deployed, the switch was serving users who were all on the same subnet in our network. If this is the case, VLANs are not necessary. Because they weren't necessary, the administrator did not bother to change the default configuration.

Later, additional users were added who were on different subnets and VLANs began to be configured on this switch. Rather than moving all the users off of VLAN 1, the administrator simply started creating new VLANs and left the original users where they are. Why? Because moving them is work, and because many network administrators do not appreciate the vulnerabilities inherent in our switching environments!

The correct way to configure a new switch is to immediately move all the unused ports into a parking lot and disabled. VLAN 1, by policy, is reserved for the switch and inter-switch ports. Now as ports are allocated, they are moved out of the parking lot and configured for the VLAN that they should be on.

Why move the ports to a parking lot? Isn't it enough to just disable them? Disabling is good, but the parking lot is even better long term because if that port is later enabled accidentally it should not be connected to a network that goes anywhere.

It's also excellent to enable any port security functions if these are available. Let's see what those might be as we examine an actual configuration.



If you go back two slides to the last configuration that you saw and then come back here you will realize that this is simply a continuation of the same configuration. In fact, all the configuration slides so far come from the same file.

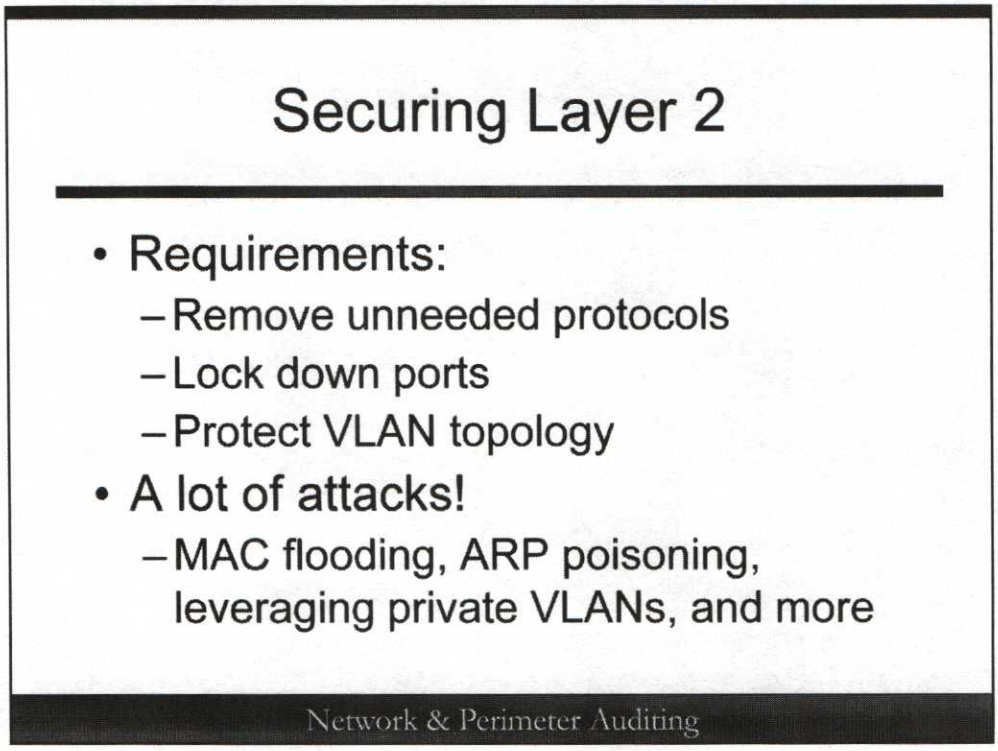
In this part of the configuration, we can see two ports being configured. Port 2/1 is being disabled because it is not in use. Port 2/2, however, is enabled and is marked as a switchport. This enables the port for use.

The elements below set up a variety of security controls for all enabled switchports. Let's explain each one in turn. The first, mac-address sticky means that the switch will remember which MAC last appeared at each port. This enables us to prevent computers from moving around, or at least to detect it.

The next option, maximum 2, configures the switch so that no more than a maximum of two MAC addresses may appear on any port. This is a great setting; though, if we are cascading switches we would need to disable this setting on any ports that are cascaded.

As wonderful as this setting is, the following three lines govern how it works and are important. Let's start with the last line first. This line means that if a third MAC address is detected the switch port is disabled. An alternative would be to just prevent the new MAC from functioning, but this is easily bypassed. Disabling the port requires help desk intervention and provides us with detection capability.

What if we need to replace a computer? No problem. We have also configured the switch so that we may have a maximum of two MAC addresses within any 30-minute window. How is that window measured? The aging does not start until the port is inactive. This gives us great port security, allows for the help desk to replace computers, and provides us with a detection mechanism if someone is "messing around."



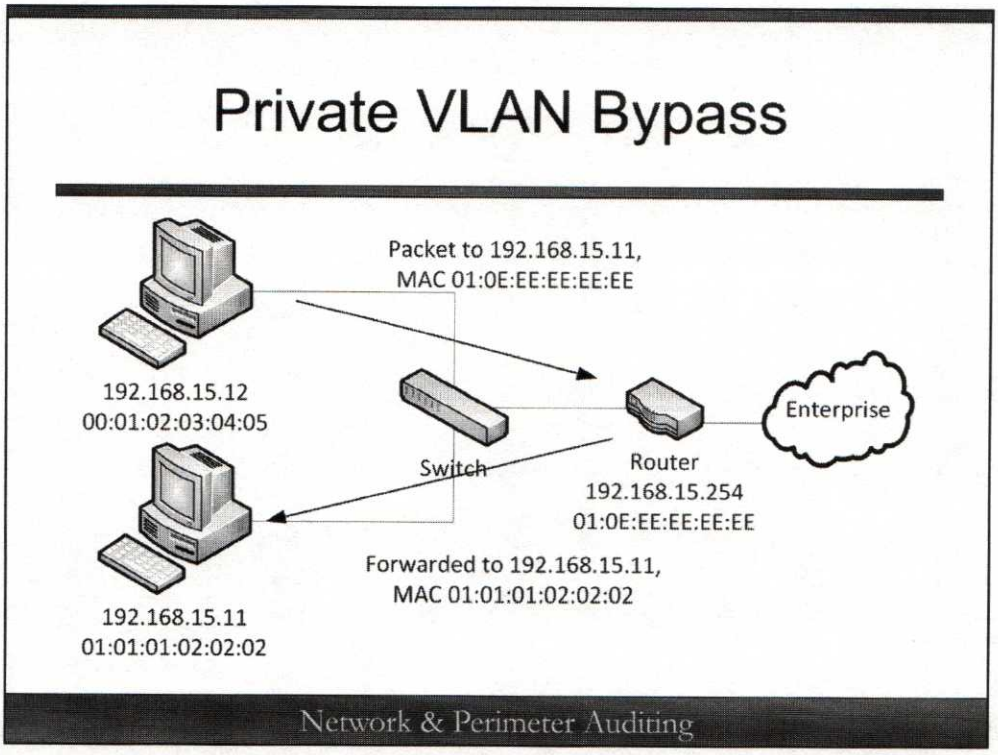
We said earlier that one of the best protections at Layer 2 is verifying that the management protocols are not available on the client ports and to lock down unused ports. We also talked about defending the VTP domain. But what are the risks?

If we don't know of the risks it can be difficult to justify these requirements to the organization. We will describe just one or two of these attacks. Remember our context, though. VLANs, although useful and although they have security side effects, are not designed as a security technology. Their development was driven by the economic benefits.

First, if we connect a sniffer to any user port on a moderately busy switch, we occasionally see packets that are neither from nor to us come out of the port. We're not just talking about broadcast traffic, here. We mean actual TCP connections with full data for two hosts, neither of which are us. We could see packets for hosts that aren't even on our VLAN! Why does this happen?

The switch has a limited amount of memory and capacity for switching data. This is often referred to as the Switching Fiber. Imagine a 48-port switch with gigabit ports. Imagine that it has 20 gigabits of Switching Fiber. What if computers on 30 or more ports are all streaming large data files simultaneously? That switch will do its best to keep up but when the switching queue fills it will flush all the pending packets out of all the ports as quickly as possible, no longer switching! Now that it has recovered, it will begin switching again. This could have serious ramifications if we have VLANs of different sensitivity levels on the same switch!

It is possible for an attacker to try to create this type of situation. Typically, this is accomplished, not by sending a lot of data but by injecting thousands of unsolicited ARP replies. The switch caches these addresses in it's CAM table, and when that fills, it may begin to broadcast packets to all ports!



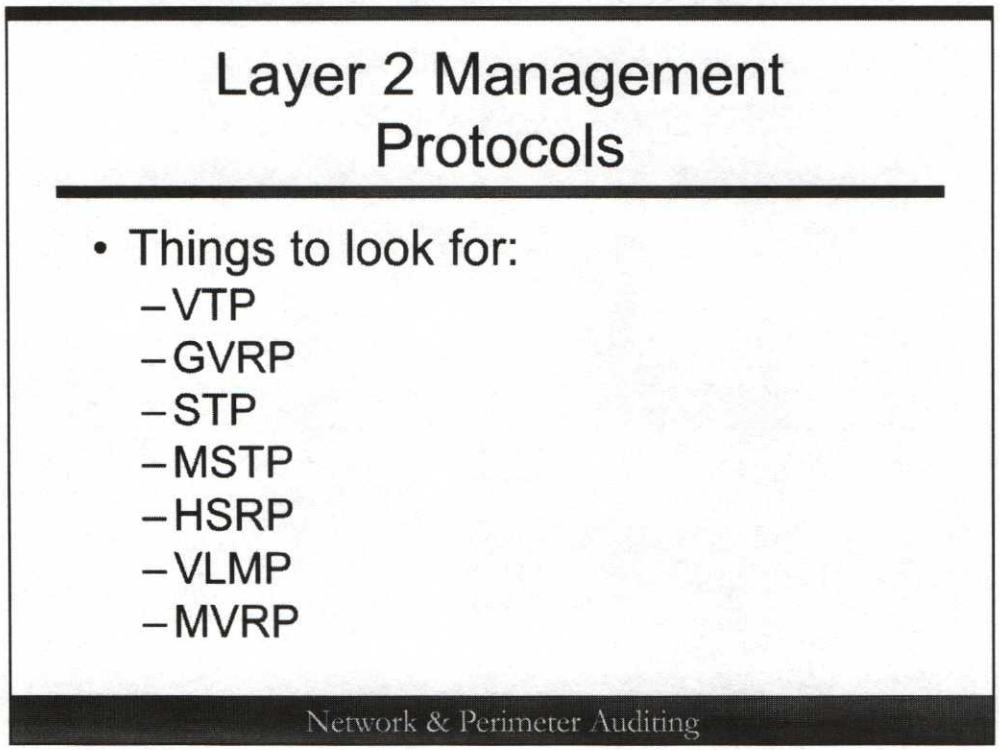
Okay, so we can see that there are ways to circumvent Layer 2 protections because they weren't intended as security features. The problem is actually worse than this. Let's look at a VLAN technology that was explicitly created as a security control: Private VLANs.

In a Private VLAN arrangement, which is supported by most enterprise switches (though the terminology used may change), we can have a collection of computers, all of which are on the same subnet connected to the same switch, but those computers cannot communicate with each other. In a sense, each one of the computers on that network feels as though it is the only computer on that subnet. The benefit is that, because I can't see or hear my neighbors, I can't attack them. Now if I become infected, it is difficult for the malware to spread to neighboring computers.

How does this work? Ports can be designated as either Private ports or as Promiscuous ports. Promiscuous ports may speak to any other port. Private ports may pass traffic *only* to Promiscuous ports. In practice this means that the user facing ports are all marked as Private and the upstream port to the router is marked as Promiscuous.

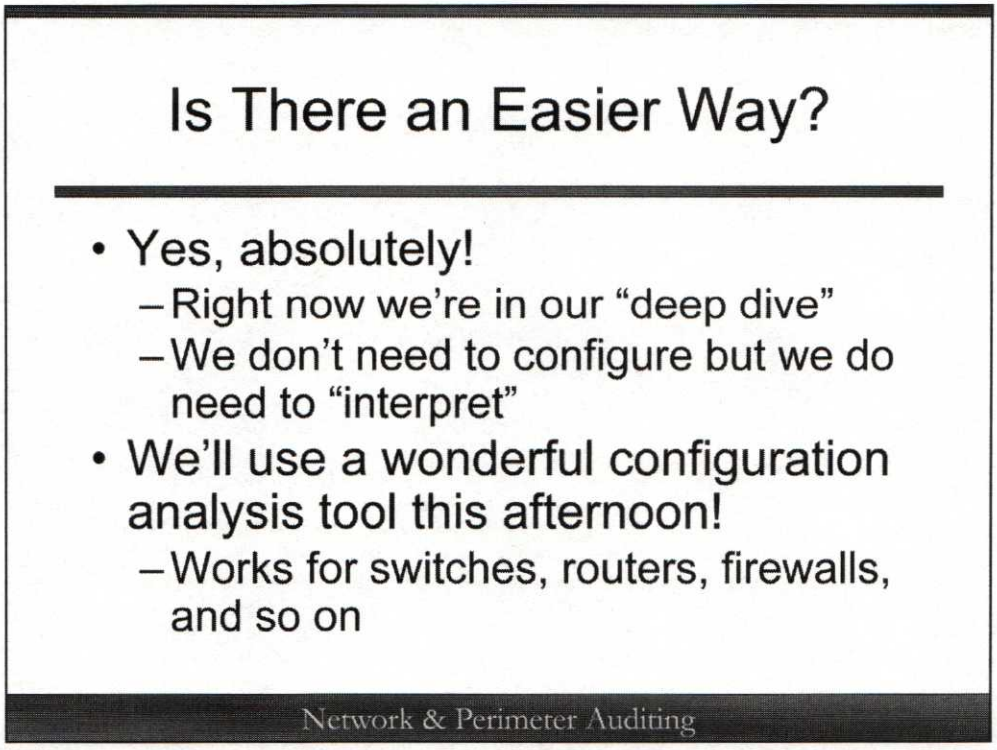
As it turns out, this is trivial to bypass. Because we can talk to the router, we can ascertain the router's MAC address. If we simply associate the target IP address of a potential neighbor with the MAC address of the router, we can force the packet to be delivered to the router. How? Using the built-in command-line tool Arp.

Now that this is configured, I can try to connect to that target host. My computer automatically sends it to the router. (And it can because it's a promiscuous port!) The router, because it is a router, simply forwards it to the actual target. It can do so because it's on a promiscuous port and may discover the correct MAC address. The best part is that the remote host will now relay its answer through the router because the router's MAC has replaced the original MAC as the source in the packet. Instant, easy Private VLAN bypass!



When it gets right down to it, the network engineers should have a clear understanding of exactly what's on the network and why it's there. For us, though, there is a good starter list of protocols that we should not see if we were to run a sniffer connected to a user-facing port. All the protocols listed in the slide are related to managing the switching fiber, managing VLANs, registering hosts into VLANs, providing failover capabilities for routers and switches, and so on.

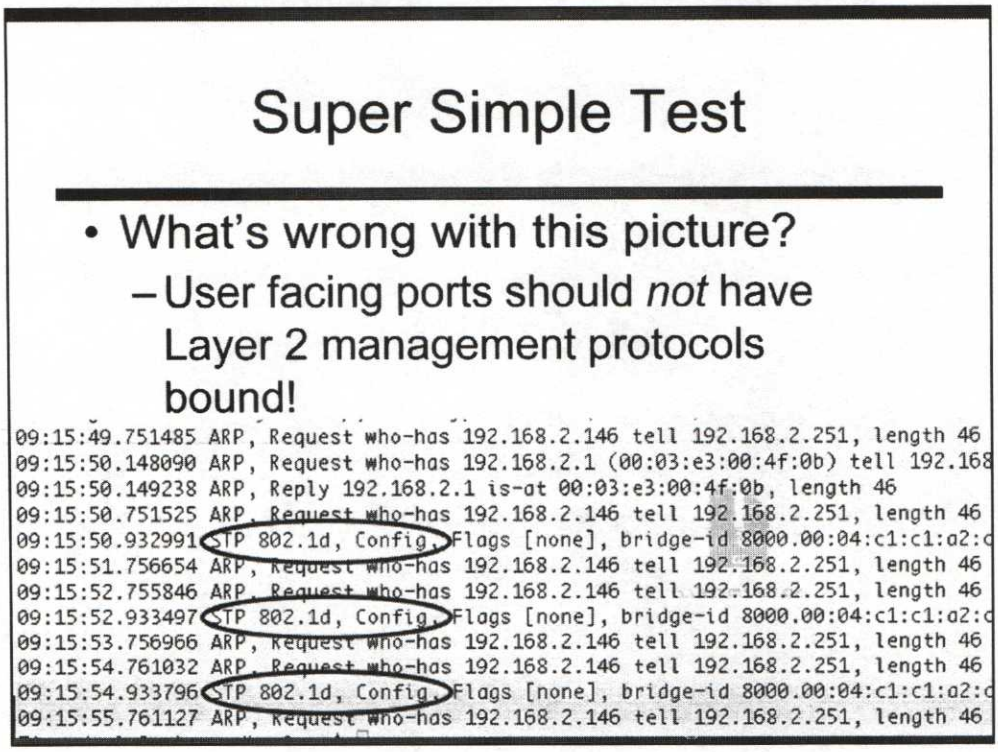
To be clear, there's nothing wrong with using these protocols. In fact, if we use VLANs, we *must* use at least one or two of them! However, these protocols should be enabled only on ports that connect directly to other switches (trunking ports) or ports used for administrative purposes. A user workstation should not see Spanning Tree packets being broadcast at it. If these protocols are user-facing, then we are saying that it is almost always an indication of an insecure and inappropriate configuration.



You might be wondering whether you need to know all this stuff. The answer is, "Sort of." You don't need to be an "Expert" in networking protocols and Layer 2 functionality, but you should certainly have enough familiarity that you can look at a configuration and puzzle out, at least in general terms, what's happening. Even better, you can go to a configuration with a specific question, like, "Is the device configured to require an encrypted management channel?" and be able to answer that question.

There is good news, however. You may have noticed (and will notice more so in a few minutes) that we haven't looked at how to test these items beyond asking questions about the configuration. The good news is that we will look at a useful tool that can be used for routers, firewalls, and switches to perform an analysis of the configuration easily!

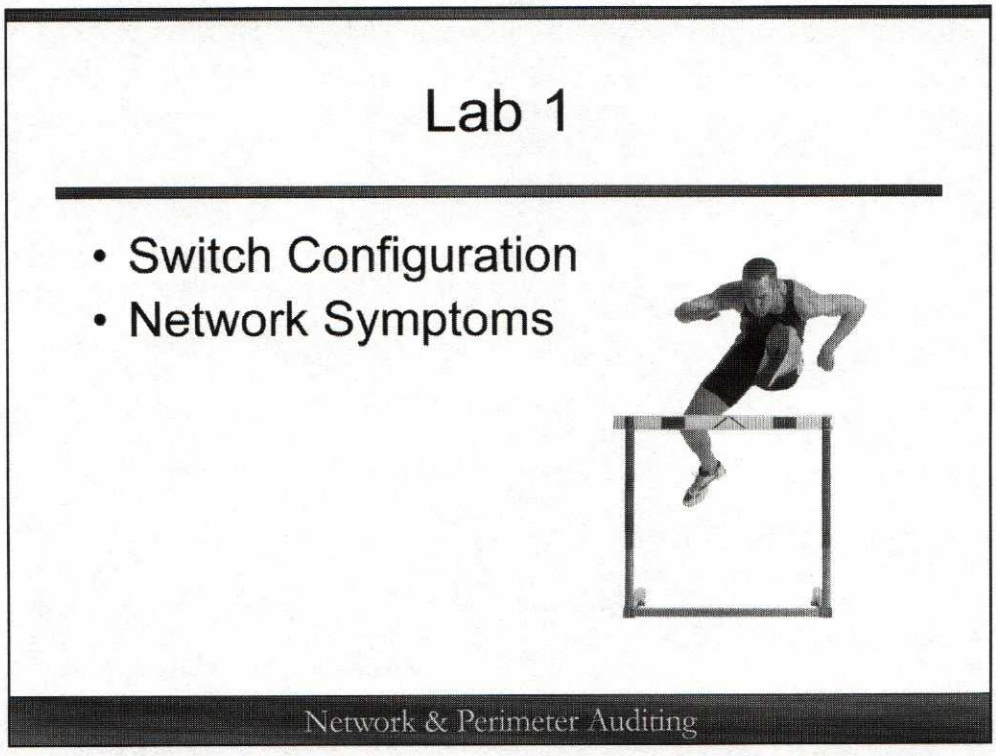
For now, just hold on and keep in mind that we're going to step back to discuss continuous management processes that should be in place. Although we will want to look at the configurations on occasion, the processes that maintain these configurations are the important things.



We've said repeatedly that the Layer 2 management protocols such as STP, MSTP, TRILL, SPD, VTP, HSRP, and so on should all be pruned from the user facing ports. We've also taken a look at what some of that might look like in the switch configuration. Is there an easy way to validate this without having to learn everything there is to know about a specific switch? Yes.

An extremely simple test is to simply ask the administrator to fire up a sniffer on a machine connected to a typical user port. Let the sniffer run for a few minutes. You will see ARP broadcasts, Windows broadcasts, and other sorts of typical, normal network "stuff." However, if you see Layer 2 management packets, you know the switch has at least that protocol configured and enabled on that port! This would fail the validation test.

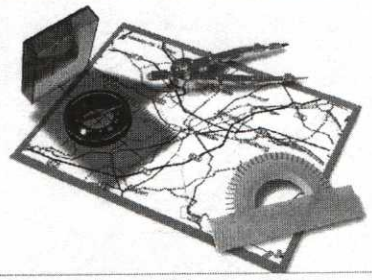
In this slide, we've given you an example of this. Although an auditor does not have to be a network engineer or a packet ninja, he should at least identify the protocols that are appearing. Doing so simply requires a little bit of time and a search engine. In this case we can see 802.1d present, which is Spanning Tree Protocol (STP). Clearly this switch port is not properly configured.



Alright. That's enough theory for now. Let's see if you can put some of this into practice! Let's take a look at some tools and some network traffic to see if you can identify potential issues in our switch.

Roadmap

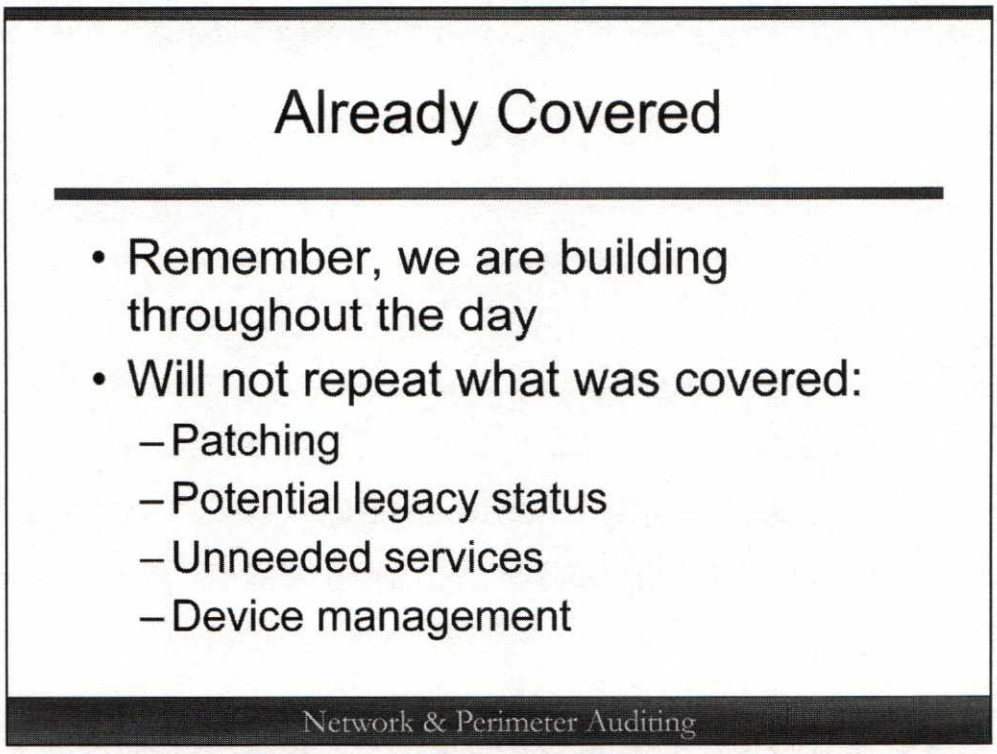
- Networks
- ➢ Firewalls and Routers
- Network Access
- Public Services
- Population Auditing
- Remediation



Firewalls: Old and New
Reading ACLs
System Management
Validation

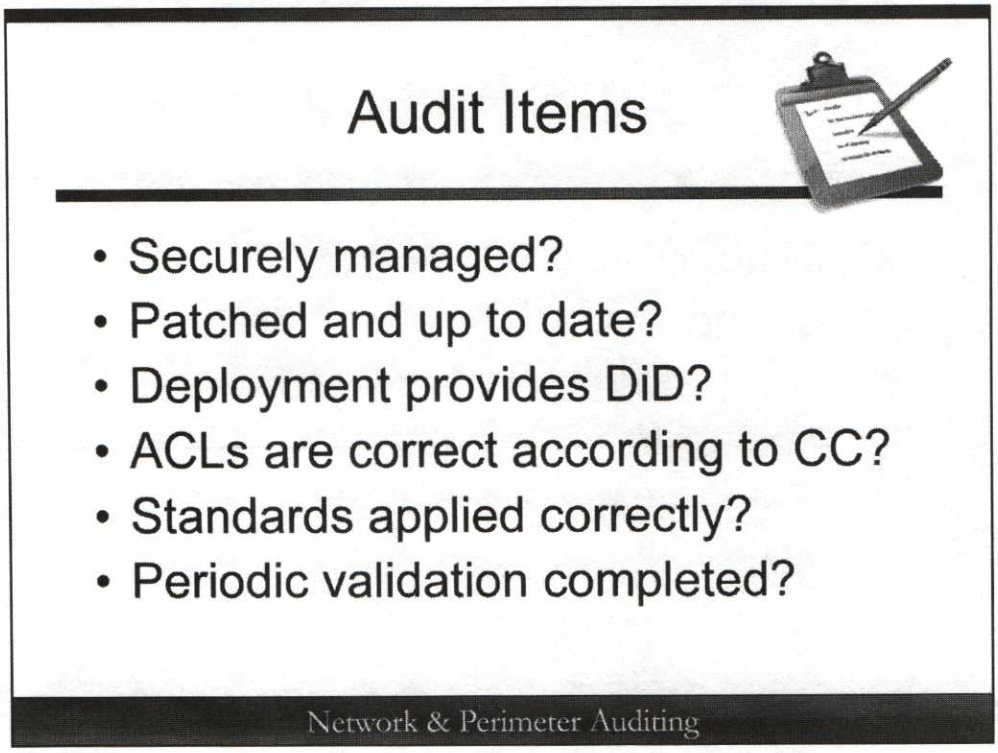
Network & Perimeter Auditing

Now that we have a firm foundation with Layer 2 and switches, let's move up the protocol stack to the systems that we would typically begin our perimeter audit with: routers and firewalls.



Please remember what was said earlier in the book. We will strive to cover any particular issue only one time even though it may apply to a later technology. For example, it is critical that we verify that our routers and firewalls are also securely managed, well patched, have clocks synchronize, have logging enabled, have centralized logging, and so on.

In this section, therefore, we assume that these issues are all addressed in the audit as well. In fact, you may see things related to these matters come up in the lab exercises. Of course, if you have any questions about how those previously covered issues apply to the technology at hand, you can always ask. If you aren't at a live conference, remember that I'm always happy to hear from you and will strive to help you out with any questions if you simply e-mail me.



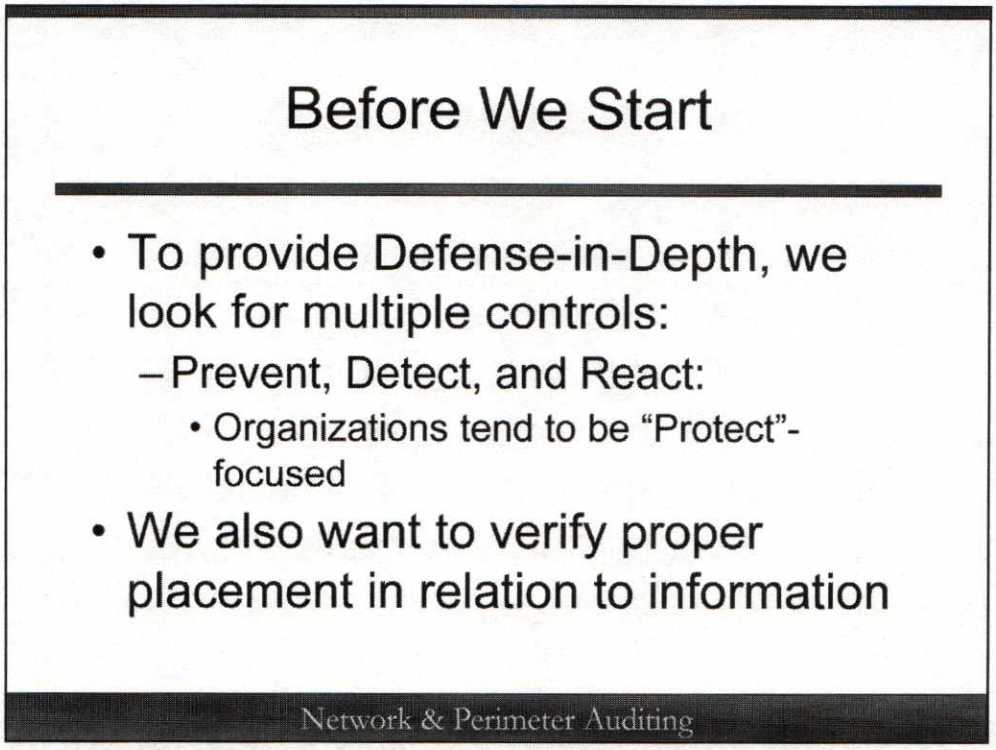
So then, our audit program and interview would include questions that dig into how the device is managed, regardless whether it is patched and other items are already covered.

Moving on to items that have not yet been dealt with, we are interested in examining the topology of the network, in particular, the placement of the firewall or router with filtering controls, in the context of the business information flows. What we're looking to do is identify that the principle of Defense-in-Depth has, in fact, been applied. Remember from our discussions on Day 1 that this does not just mean that there should be a lot of layers. Instead it means that the layers are architected in such a way as to provide for protection, the ability to detect threats, and the potential ability to react to those threats.

For this to be effective, though, we must also verify that the Access Control Lists (ACLs) that are in place make sense for this business and the security requirements of the organization. To know this we have to understand how those high level standards would be expressed in the particular technology, like a firewall. Each ACL would also have to be examined to verify that there is authorization for that rule or group of rules according to the Change Control system in the organization.

An important aspect of managing routers and firewall rules well is good documentation. Not only are we verifying that the change control for an ACL exists, but we would also like to verify that there is adequate documentation *right in the rule set*. Regardless of the firewall or router that we use, all of them allow the administrator to attach comments or notes to rules and to group rules. This is true even if the configuration is a simple text file like we find in a Cisco PIX or ASA device!

Although ensuring that the proper rules are in the device, it is equally important to actually validate that the device performs as expected. Everything defined should pass according to the security requirements. We should prove that these things do pass and that nothing else can.



So how do we know if Defense-in-Depth is accounted for? How do we know we have protect, detect, and react capabilities? How do we know if these systems are deployed in the correct places?

To answer these questions we have to look at the network topology. Before we do so, however, we have to know what the topology *ought* to look like.

Here is an important point: Network administrators are great at building networks. When they build networks, however, they are usually concerned only with building the roads. They ask about traffic volume and where exits should be so that they know how big to make the pipes and where to put the switches and routers. They almost never, however, ask what the vehicles, or packets, will carry! Without this knowledge it is easy to create a network that works wonderfully but fails to account for the security requirements that the organization truly needs to secure itself.

Information Flow

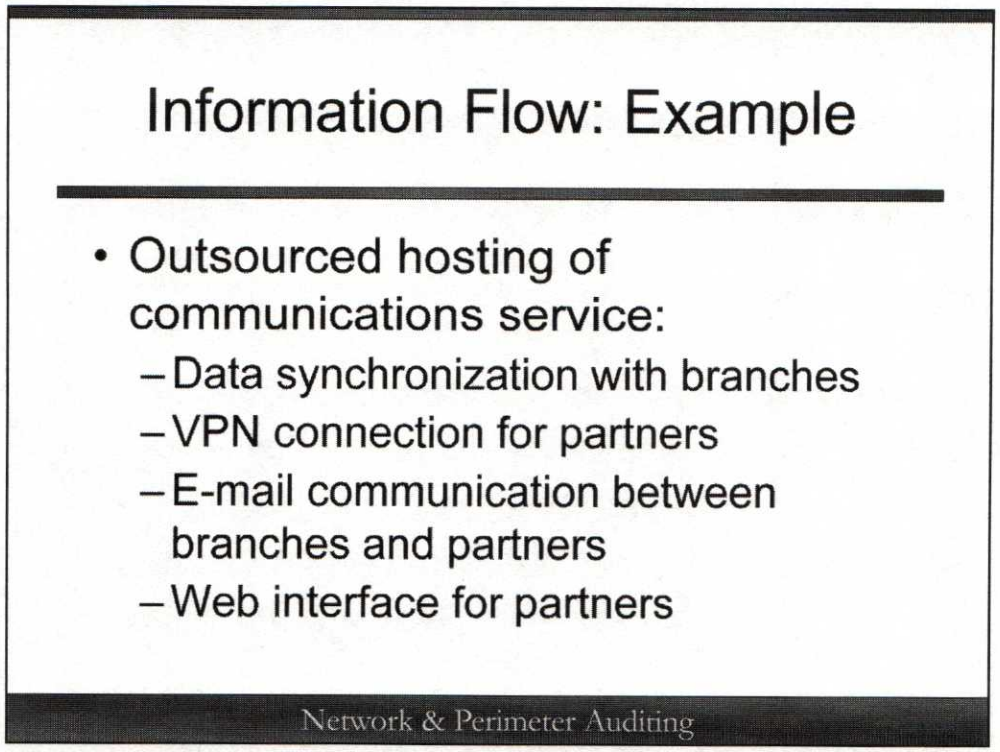
- How information gets from here to there and why:
  - Enables us to identify control points
  - Naturally reveals security zones
- Compare information flow to what the physical diagram looks like

Network & Perimeter Auditing

What we're talking about is Information Flow. If you works in the Payment Card Industry (PCI) world, you may be somewhat familiar with this idea. As a result of attending our classes, the PCI folks ended up adding a requirement for an Information Flow Diagram as a part of a PCI assessment audit. What is this and how do you draw one? We're about to do one together.

First, an Information Flow Diagram is simply seeking to illustrate where information resides, where it moves to and how. We are going to do this with an information technology and security lens, but you can create exactly the same kind of diagram for information that is maintained only in physical form. This type of approach forces to examine data movement. Data movement allows us to identify security zones and tends to indicate where control points ought to be.

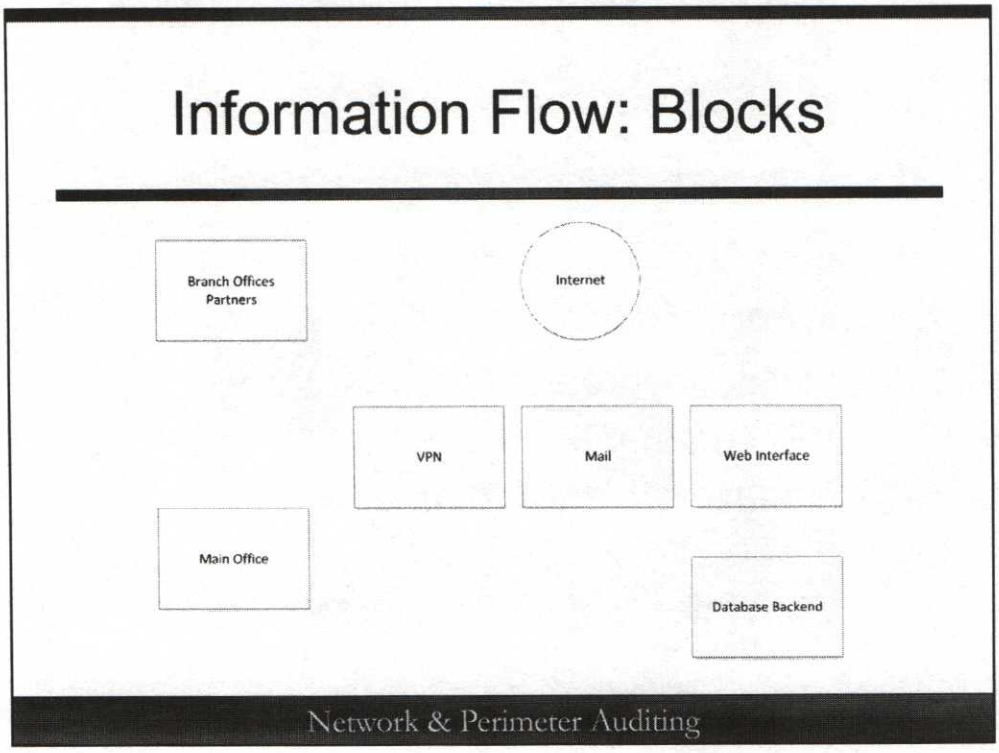
After this diagram has been created, we can compare the requirements that we find with the actual physical topology that exists. It is our hope that we find that the physical topology has all the things that it needs to meet the requirements that the Information Flow Diagram indicates are necessary.



Over the next several slides, we illustrate how this type of diagram is created and how to compare it to the physical topology. For our example we consider an organization that has decided to outsource the hosting of a communications service solution. The motivation, perhaps, is that the organization has considered how the service will be used and determined that they are unwilling to accept the added risks that come with having this service internal to their network.

This communications system will be used to interact with all their branch offices, synchronizing data with each of those branches. In addition, trusted third parties and other partners will have VPN tunnels that allow them to interact directly with a web interface that contains knowledge base style information and other support information. In addition to this web interface to the knowledge base, partners and branches will have the ability to route e-mail through the system to each either and to the main office of the organization.

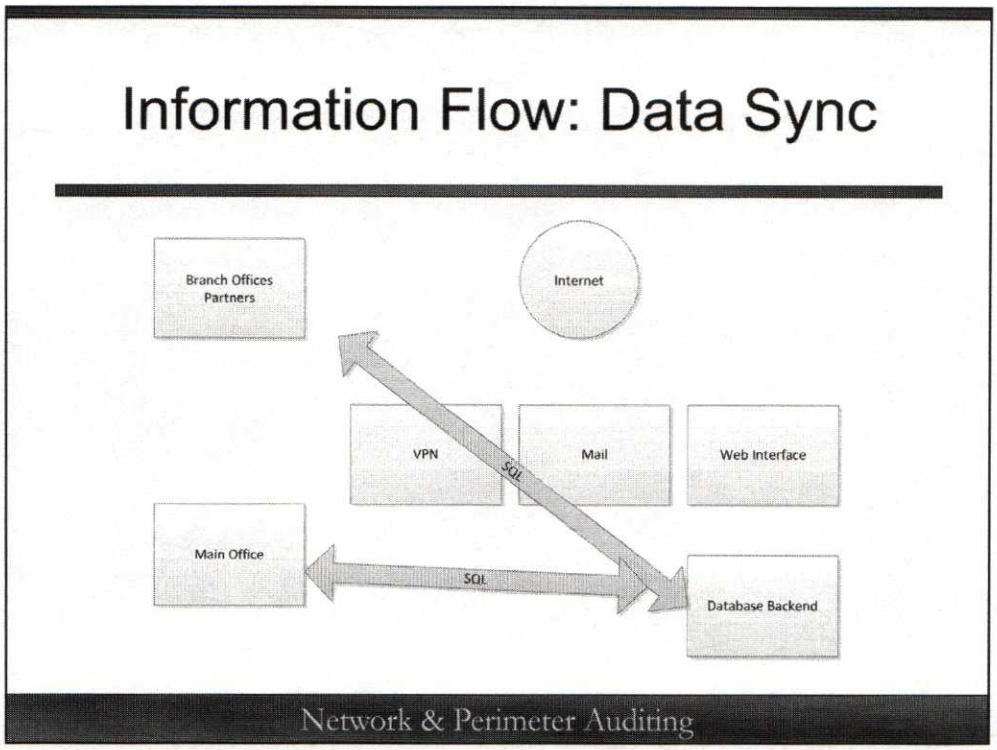
After gathering some basic information about this system, as listed here, we would next identify individuals within the organization who are stakeholders in this process. We would need to have the business operations manager who oversees the parts of the organization that utilize this system. We would need representative system administrators who actually manage systems that utilize the provided infrastructure. In addition, we would need the network engineers and managers who oversee day-to-day operations of the solution.



When we have those people in a room, we want to start by simply identifying functional blocks. At times those blocks may turn out to be servers, but there is not necessarily a one-to-one relationship. Keep in mind that we are not trying to diagram what is in place now. We are trying to analyze how everyone believes the system operates and to identify what the information flow requirements are. In other words, where is the information, where does it need to get to, and how does it get there?

In this diagram you can see that we have created a high level set of functional blocks. We have chosen to include additional details when looking at the systems in the outsourced location because we are focusing on validating the network design and, ultimately, the security of the network at that location.

As you create the blocks, simply annotate high level information that describes them or indicates their major function to the overall system.

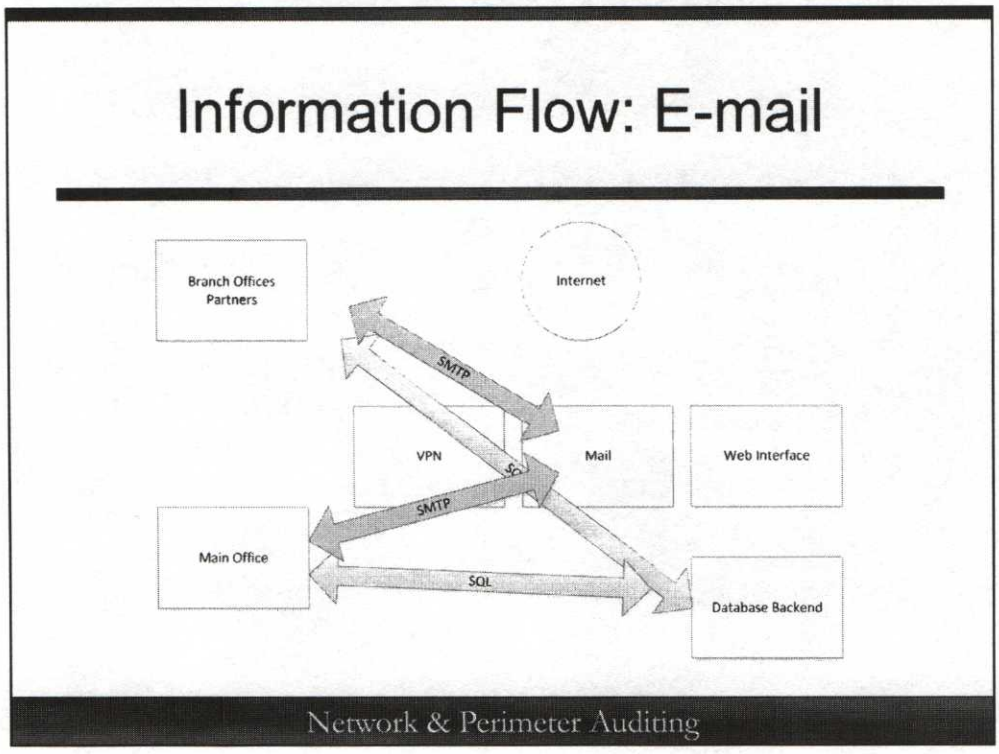


We now begin to ask questions about how things function. I'm frankly not interested in brand names, port numbers, IP addresses, or anything like that. One of the requirements that was mentioned was the need to synchronize data between the branch offices and the main office through this outsourced solution. What sort of information are we talking about? How sensitive is it? When they say synchronization do they mean that it is bi-directional? What protocol is used?

Before we continue, let's examine that last question. When we start asking questions about protocols it is easy to distract ourselves and dive down into technical implementation details. For us, all we are interested in is an answer like, "It's a SQL job and it replicates data between the branch and the outsourced site." I don't care about ports. I don't care about brands. At this point, I don't care how the SQL servers are configured.

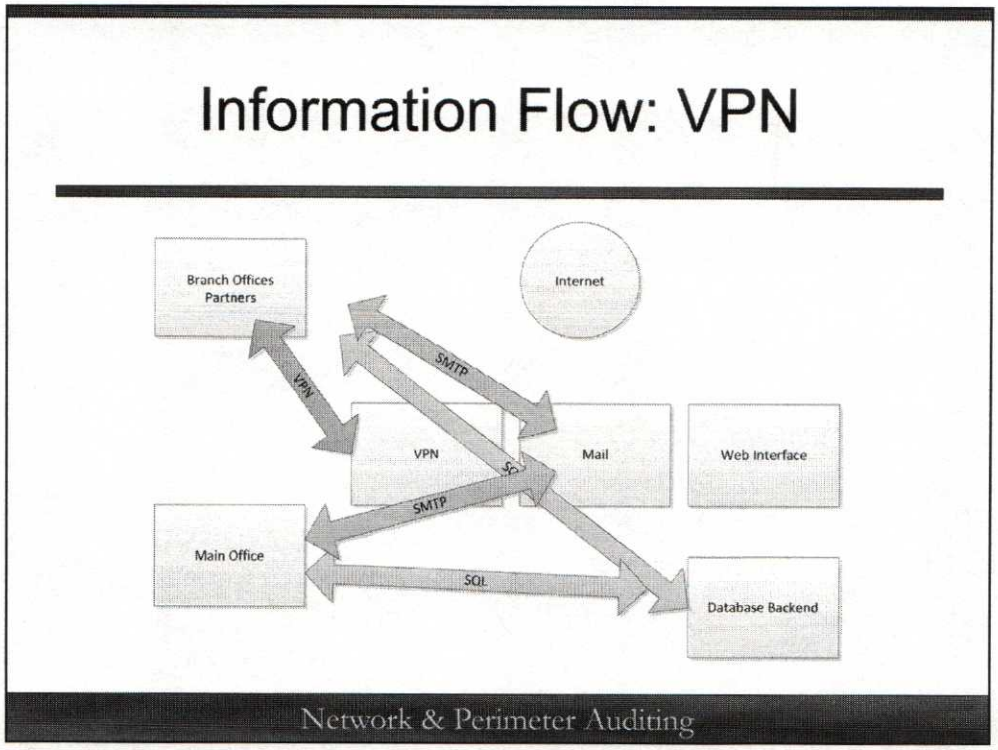
When we asked, "What protocol is used?" it is a general, high-level question. To put it into another context, imagine we're discussing backup management for disaster recovery. We ask, "How do you get the tapes to the offsite storage facility?" The answer, "Federal Express" would be an example of the answer that we are looking for.

If we dive down into port numbers, settings versions, and similar sorts of things, it would be too much detail. It would be like following up the DR/BCP answer of "Federal Express" with questions like, "How fast does he drive? Which route does he take? Which sorting facility will our data go to?" This is simply not the time for this.



Another information flow that has been identified is e-mail. When we ask about this we discover that e-mail is being routed through this location; however, there are no user local accounts. In other words, there is no POP3, IMAP, OWA, or other client e-mail solution here. Instead, they have created a private e-mail routing network between the company, branch offices, and partners.

What would the information flows look like? Our discussion reveals that we are using SMTP and that mail can route in any direction with anyone initiating connections.

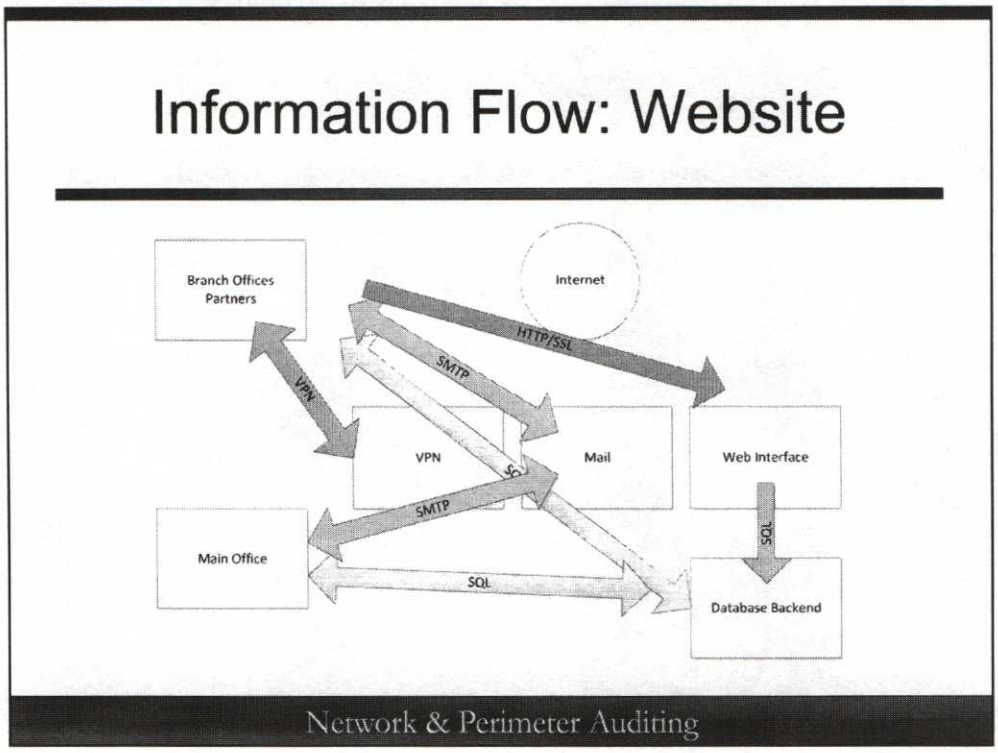


Because there was mention of connecting in branch offices and partners directly, this indicates that there is a VPN of some sort. Again, is it IPSec? SSL-based? Who exactly is connected? We just continue to add lines to the diagram, illustrating the kinds of information flow that occur.

At this point you can no doubt see that even though this is a simple diagram, it can start to become quite cluttered. Usually, I would use a whiteboard to sketch it out interactively with the group. This allows for rapid adjustments if something's not quite right. As we proceed through the discussion, however, I will ask someone to act as a secretary who will put what we are drawing into a Visio diagram using multiple layers. Each layer represents a single set of information flows that are related to a specific operational requirement.

The person who is formalizing into Visio does not need to rush. He does not need to add anything until there is consensus that the flows or objects that we have most recently added are correct. After that decision is made, he documents while the discussion continues.

By building the diagram with layers, we can choose to display or hide any particular set of flows, allowing us to have more in-depth discussions later should they prove necessary. This can also now serve as a master information flow diagram for all data flows even though the diagram would be difficult to read if we enabled all the layers simultaneously.

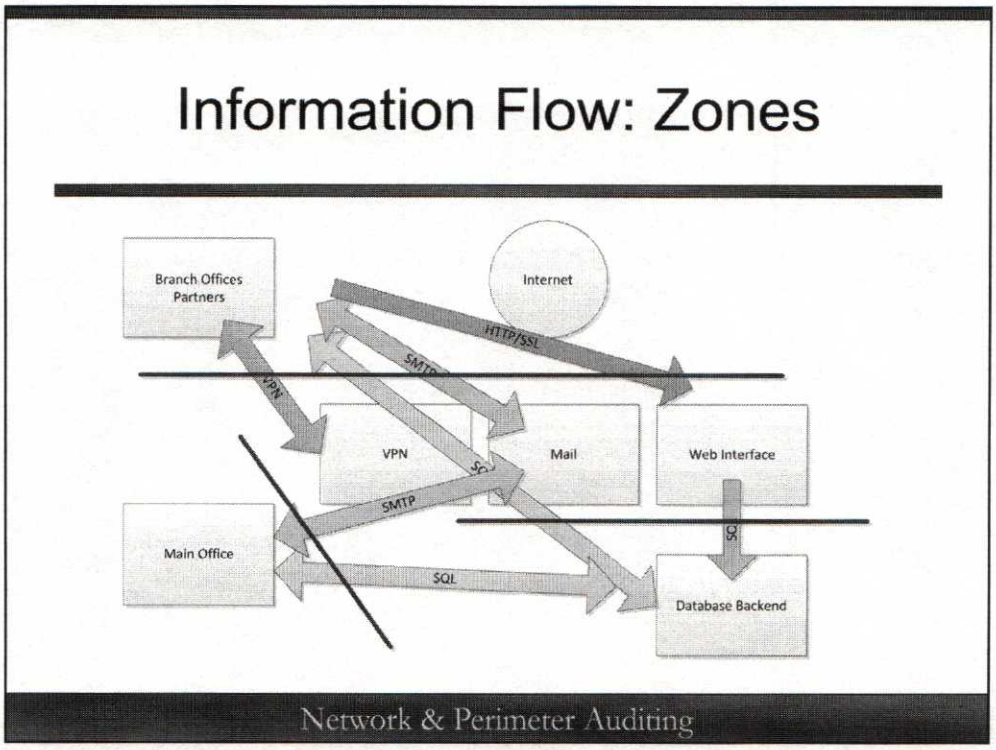


The next data flow on our list relates to web traffic. If you look at our diagram you can notice something significant that is different about these information flows. Notice that the arrows point only in one direction.

Up until this point, all our data flows have been bi-directional. The arrow heads are not trying to indicate the direction that information flows. Clearly, if there is a connection, information can flow in both directions. The arrow heads are simply indicating who will initiate the conversation or connection.

For example, when we consider the data replication model, we would expect that replication can be initiated by the main office of the company with this remote site. We would also expect that the remote site will initiate replication with the other remote sites. Not only this, but we would also expect that a remote site that has had a change will initiate replication back to the home office. For this reason, all the flows are double-ended.

Because we do not require our web server to initiate connections to anyone at other sites, these arrows are single-ended. The same is true of our VPN connections if you look back.



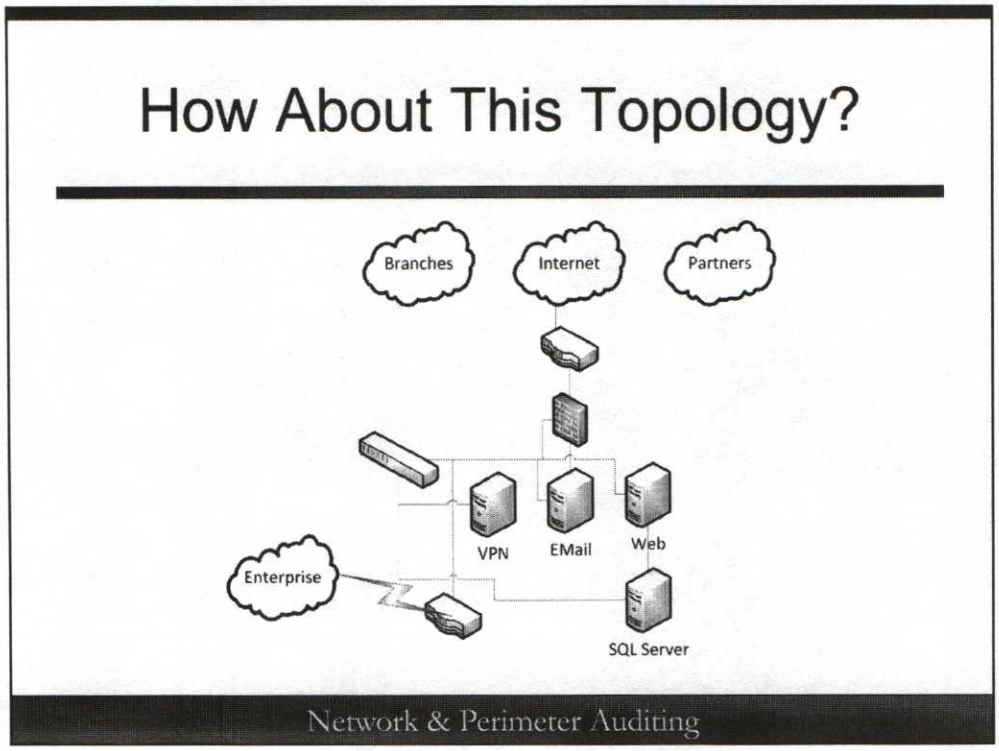
Now that we have discussed these requirements at a high level, we change the conversation. Our focus is now on the sensitivity of the systems and the trust level that we have for the different modules.

With our diagram, it is reasonable to expect that our highest level of trust and the most sensitive data is at the home office. Another trusted system with sensitive data would be the database server that is used for replication in this remote location. In fact, you may feel as though we actually have the same level of trust for this. This may very well be…but are they actually in the same security zone? Consider the concept of secure compartmentalized data. You are extremely trusted with the highest security clearances. Does this mean that you can look at any information you want to, or do we still need to consider whether you need to see that data.

In a similar way, we would say that these represent different security zones, even if we have the same trust for them. However, consider the partners who are third parties. We have a measure of trust for them, but clearly they are not at the same trust level as the database backend.

What we are trying to do is to identify the various security zones based on need-to-know and trust levels. After we have identified these, we have identified our trust boundaries. At every trust boundary we would expect to find an information flow control (like a firewall or set of router ACLs) that limits the flow of information to only what is required for operations.

This obviously isn't a pretty diagram, but we can use it as a description of what we *should* find. Let's see what the topology looks like.

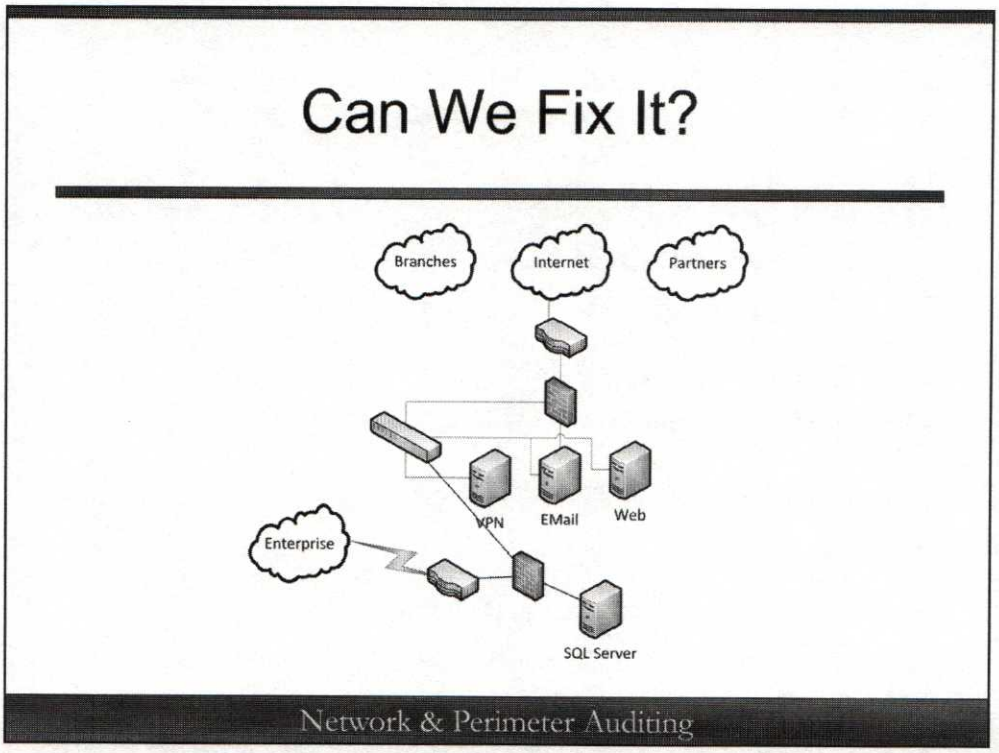


Now that we have thoroughly examined the information flows and come up with this diagram, the next step is to compare the requirements that we have identified to the actual physical topology. In the context of the information flow diagram that we just built, consider the logical topology above.

The question that we have is, "Is this topology a reasonable approach to meeting all the information security requirements for the information flows defined?"

In this case the answer is, "Not really." Although it is a nice diagram and there is a firewall, it does not appear that the firewall can adequately control the information flows. Why not? Notice that the VPN is sitting behind the firewall. Although this offers the VPN protection, it means that the firewall has no opportunity to limit information flows and data access after the data reaches the VPN endpoint. Because the data is encrypted, the firewall will simply allow all the VPN data over IPSec to pass.

The next question is, "Could we make a minor adjustment that would fix this?"



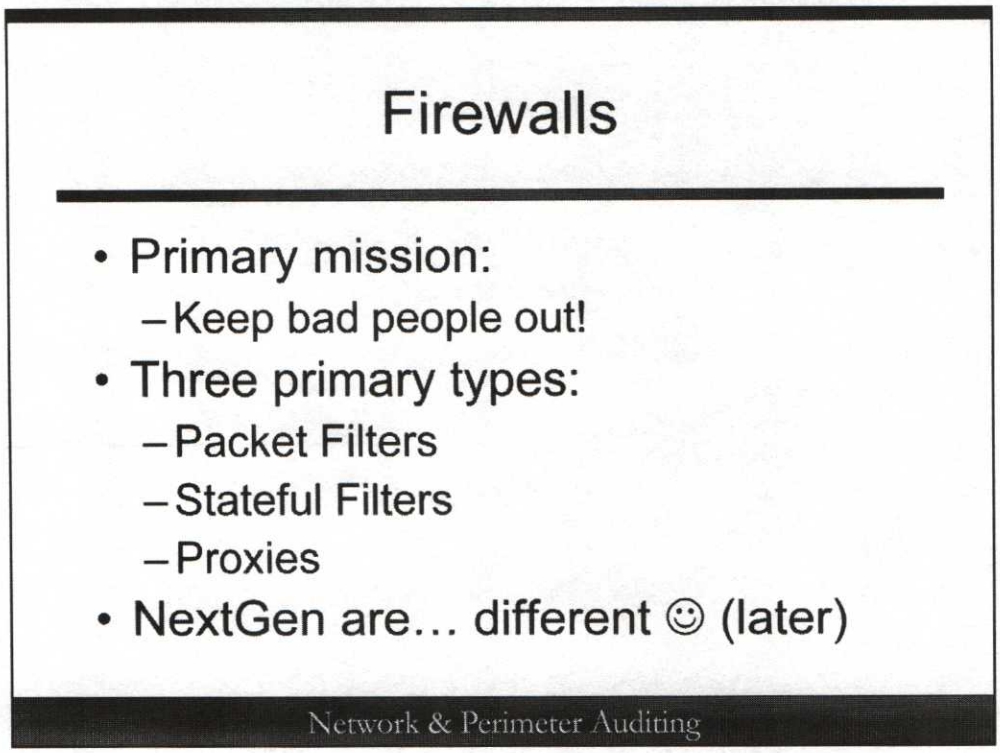
The answer is, of course, we can fix it! However, we'd like to suggest extreme caution during this process.

When you first introduce the purpose of the process that we have just worked through, network administrators and engineers may feel as though you are getting ready to re-engineer their network. Please assure them that this is not the case! In fact, it would be good to introduce the process something like this:

"What we'd like to do now is validate that the network topology that you have in place fulfills all the operational security requirements. Now, I'm sure that it does because your network seems to be working. Still, we want to use this formal process so that we can document that."

What we say above we mean sincerely. Even if it turns out that the topology is not ideal, it certainly must be close if we can conduct business. Looking at the diagram in the slide you can see that we've just made one small adjustment to the overall design, and it now enables us to meet all the information flow requirements because we now have a control at all trust borders. This enables us to enforce the information flow requirements properly.

Looking at our adjustment you might think that this requires the purchase of a brand new firewall. Is that actually true, though? Isn't it possible that we have simply added interfaces to the existing firewall? Although that does make this firewall more complicated, it is absolutely possible to meet the information flow requirements in this way.

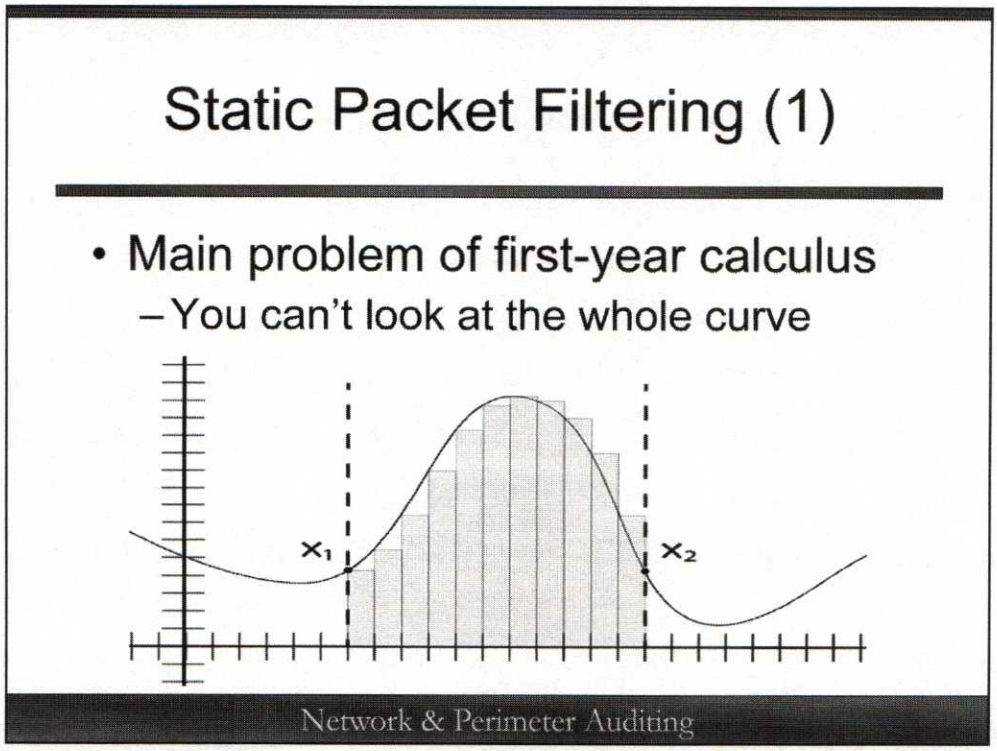


What, exactly, is a network control? Well, if we're talking about a protective or preventative control, then this is most likely a firewall or possible a router with ACLs. Obviously, we would prefer to use firewalls as controls, but often the cost involved when considered with the placement makes a router a reasonable choice, especially when dealing with interior networks.

Firewalls, however, are the types of systems that should be in the perimeter. Not all firewalls are created equal. In fact, there are three primary types of firewalls that are in wide use today. There is another type of firewall that has been emerging over the last few years, but we will deal with these next generation firewalls a little bit later.

Generally speaking, we have organized the list of firewall types in the slide according to cost. Packet filters tend to be the cheapest; proxies tend to be the most expensive. They are also organized in terms of required horsepower. Because packet filtering is very, very fast, the hardware requirements are much lower. Because proxying requires far more processing, for a slower process, much faster hardware is required. These are also organized from what are considered to be the easiest to bypass (or least secure) up to the most difficult to bypass (most secure).

Don't misunderstand. This doesn't mean that packet filters can't ever be secure. If I asked you if you'd prefer to be in a maximum security prison or a minimum security prison, you would pick the minimum security prison. This doesn't mean that it's not a prison, it just means that it's less "secure" overall. Easier to escape from, easier to smuggle stuff into. Let's dig into the differences between these types of devices.

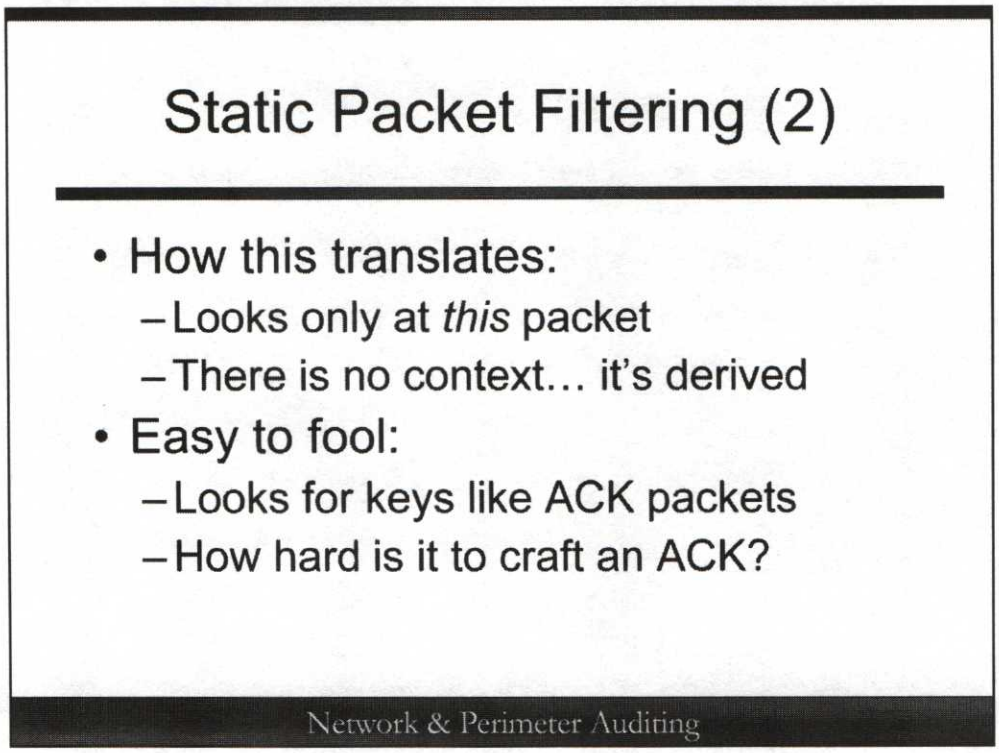


Packet filters are not sophisticated devices. They make decisions on whether a particular packet may pass by comparing that packet, and only that packet, to the rules or ACLs that are in place. This behavior is what makes bypassing them, in many cases, trivial.

Here's an illustration to help you imagine how packet filters work. If you think back to when you were in college and took calculus, the picture in the slide might bring back memories. For almost the entirety of your first semester, possibly even your first year of calculus you were focused on solving just one problem. If I graphed out a square, triangle, or any other polyhedron and asked you to calculate the area, you would have no problem coming up with an answer. The challenge in calculus is to figure out the area under a curve.

The real problem is that, because the curve is constantly changing, you can't just "measure" it. The first approach that is taught is to calculate the altitude of the curve at regular intervals, turning the curve into a bunch of rectangles. If we add the areas of all these up, we arrive at an approximation of the actual area. The smaller the interval between the altitudes, the closer we come to the actual area. This leads to limits and is called Discrete Analysis. Because we can't look at the entire curve, we instead look at discrete points along its length.

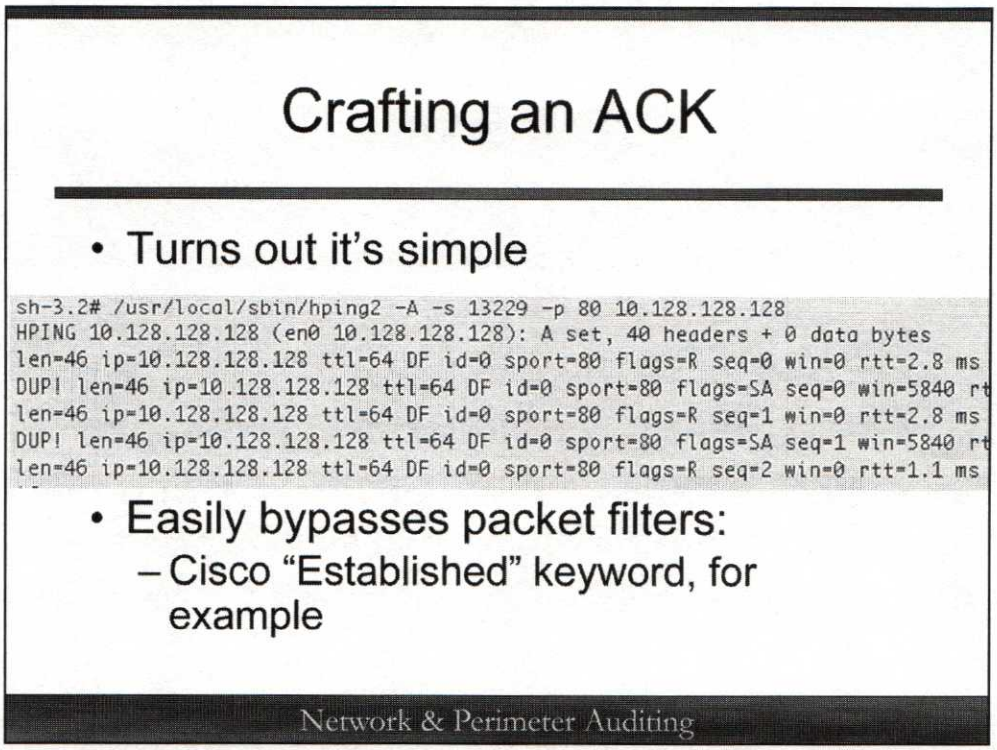
This is precisely what packet filtering devices are doing. They cannot stand back and look at the overall curve, or the complete conversation. Instead, they are limited to looking at this and only this packet and making a decision based on this single discrete event with absolutely no context.



Because it looks at only this packet, this makes it simple to come up with strategies to bypass the configured rules. To combat this, most vendors provide extra "features" that try to simulate a more stateful approach. For example, Cisco devices that are operating as packet filters can be configured to permit only "established" packets.

Although many devices identify an established packet by looking for the ACK bit to be enabled, Cisco approaches this differently. Cisco looks at the problem and says, "The only time that the SYN bit is set alone is during the initial three-way handshake." Because of this view, Cisco products using the "established" keyword will pass any packet as long as the SYN bit is not set alone.

Regardless of whether we are checking for SYN by itself or to see if ACK is enabled, how difficult is this to bypass? Are there tools that can be used to generate packets that can be used to trick this device?



Of course, there are tools that can generate packets!

In this slide, we have an example command line with the output of a tool named HPing. This tool is designed not only as a packet crafting tool but also as a network exploration and experimentation tool. From the command line, we can specify exactly what we want any particular packet to look like. How is this used for exploration and experimentation? Consider these questions:

- If you send a SYN packet to a closed port, what is the correct response from the server?
- If you send an ACK packet to an open port without having a session, what is the correct response from the server?
- If you send a SYN/FIN packet to an open port, what is the correct response from the server?
- If you send a packet with no code bits enabled to an open port, what is the expected response?
- If you send a packet with no code bits enabled to a closed port, what is the expected response?

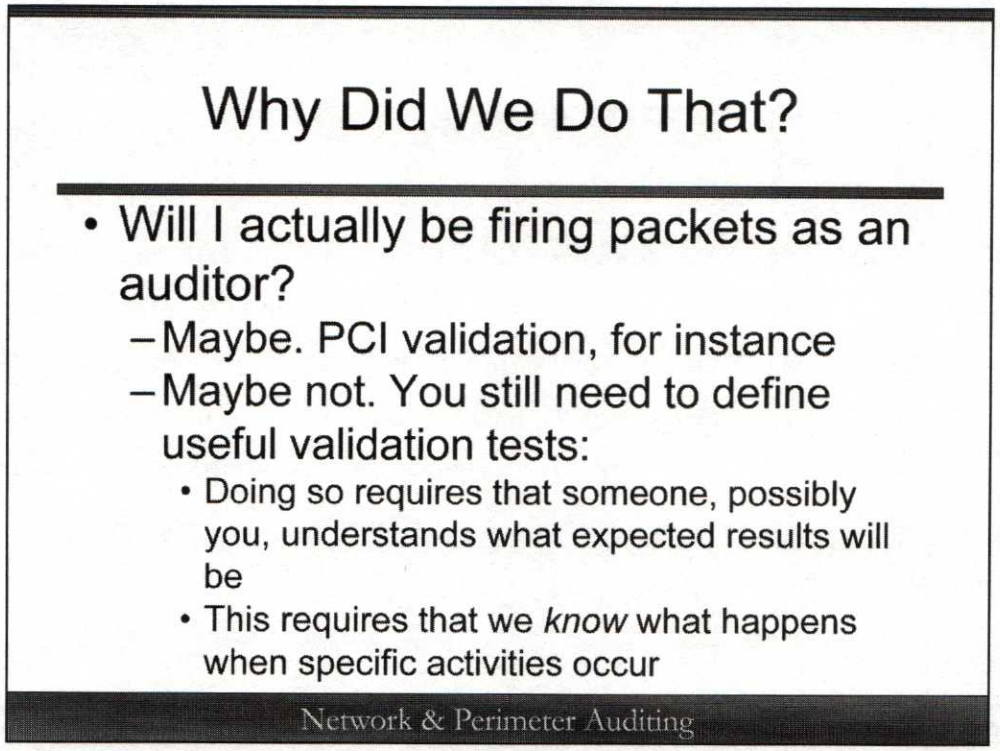
Take a moment and write down what you think the answers to these would be. Then, we'll check to see how close you came!

Lab 2: Let's Experiment!

- UNIX Virtual Machine:
  - SYN to a closed port
  - ACK packet to an open port
  - SYN/FIN packet to an open port
  - No code bits enabled to an open port
  - No code bits enabled to a closed port

Network & Perimeter Auditing

Follow along with your instructor as he works through this interactive lab. If you happen to be taking this course online or self-study, the lab is fully documented in the workbook and the Lab Videos DVD does contain this walkthrough.

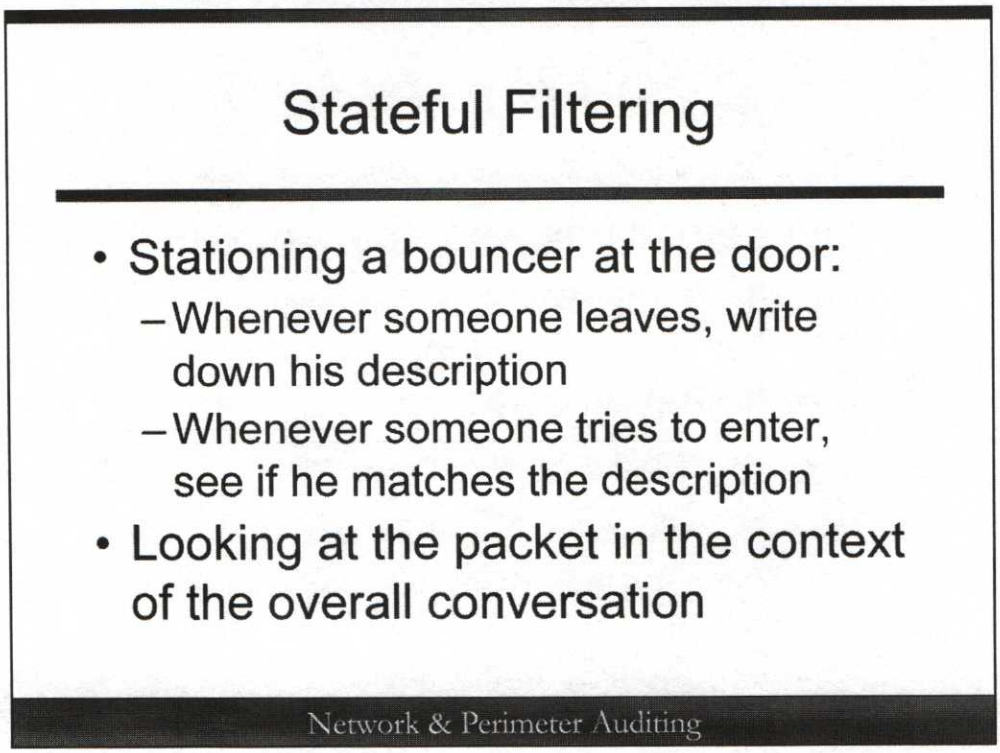


Congratulations! You're through the first lab for Day 2! However, you are likely asking yourself, "Why the heck did I just do that? When will I ever need to fire packets at something?" These are certainly valid questions, so let's give you some context.

First, remember that we are performing the type of research required to properly perform an audit or validation of a system. In this case, because we're dealing with network control systems, understanding how the network *actually* functions is important.

A second reason is that you, the auditor, might actually find yourself in a position in which you need to perform this type of activity. When could this happen? If you are under PCI and work as a QSA or, possibly, an internal auditor, a PCI ROC requires that the person performing the assessment (you) *actually validates the firewall*. This means, firing packets at it and interpreting them.

The third reason is that, even if the second doesn't apply to you, you should absolutely be involved in designing and specifying validation tests and criteria for controls within your infrastructure. This clearly includes network devices. To properly design these tests, it is critical that you understand how things *actually* work! Using this type of a tool will allow you to design validation tests and to properly understand the results!

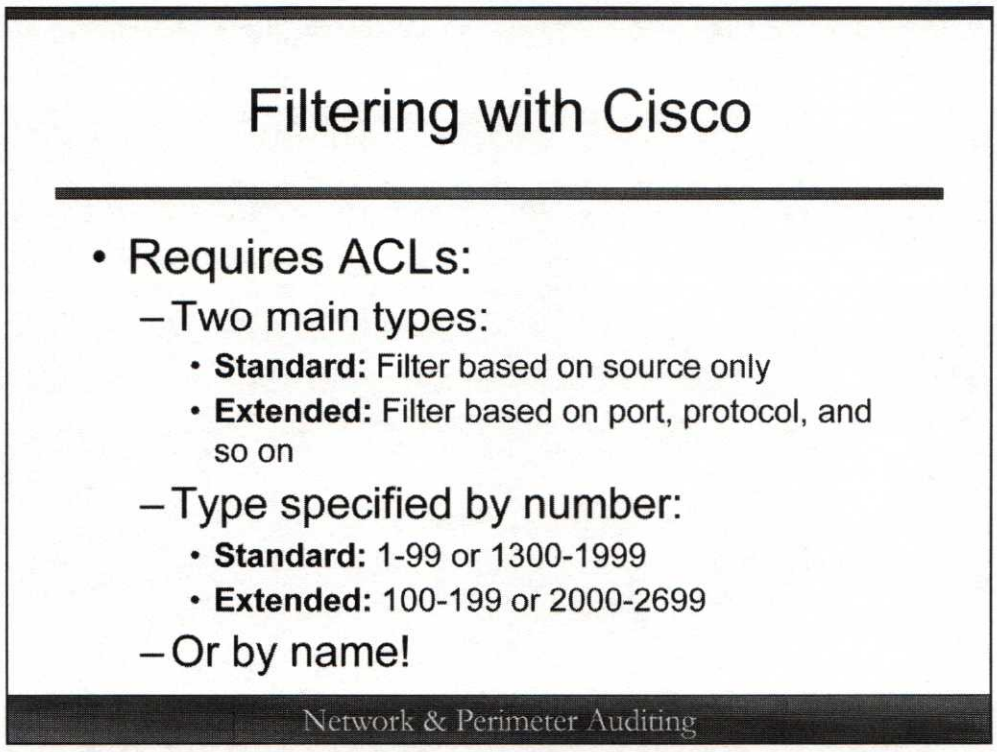


Stateful filtering is much more secure than static packet filtering. With stateful filtering we are essentially telling the firewall what kinds of things are permitted as a baseline. For example, we tell the firewall, "Allow outbound connections to port 80 and 443" so that our users can browse to web pages. With this configured, the firewall will gladly allow these packets outs, but it will also track these connections.

What this means is that when a packet comes into the firewall from the Internet from port 80 and it claims to be a response (a SYN-ACK or an ACK), the firewall will not just allow it to pass. Instead, it will consult the information that it is tracking for known connections. If this packet does not fit into any of the known conversations, it will drop the packet.

Stateful firewalls are not all equal when it comes to their capability to track sessions. Of course, they will also have limitations for the total number of connections that they can track. In fact, a denial of service against a stateful device is easy to accomplish. If we find a port that is permitted through and which is tracked for state, we can simply open millions and millions of connections relatively quickly. At some point the device will become memory starved in its state tables. When this happens, the connections which have been quiescent for the longest period of time will be dropped out, effectively terminating the connection.

These firewalls are also not equal in terms of how smart they are about state. Some are simply matching the IP port pair while others are looking more deeply, checking to see if the TCP sequence numbers make sense for the specific question being filtered.



Let's look at some examples of configuring a router and a firewall with some rules. Let's do some research to see what sorts of features Cisco has for these things.

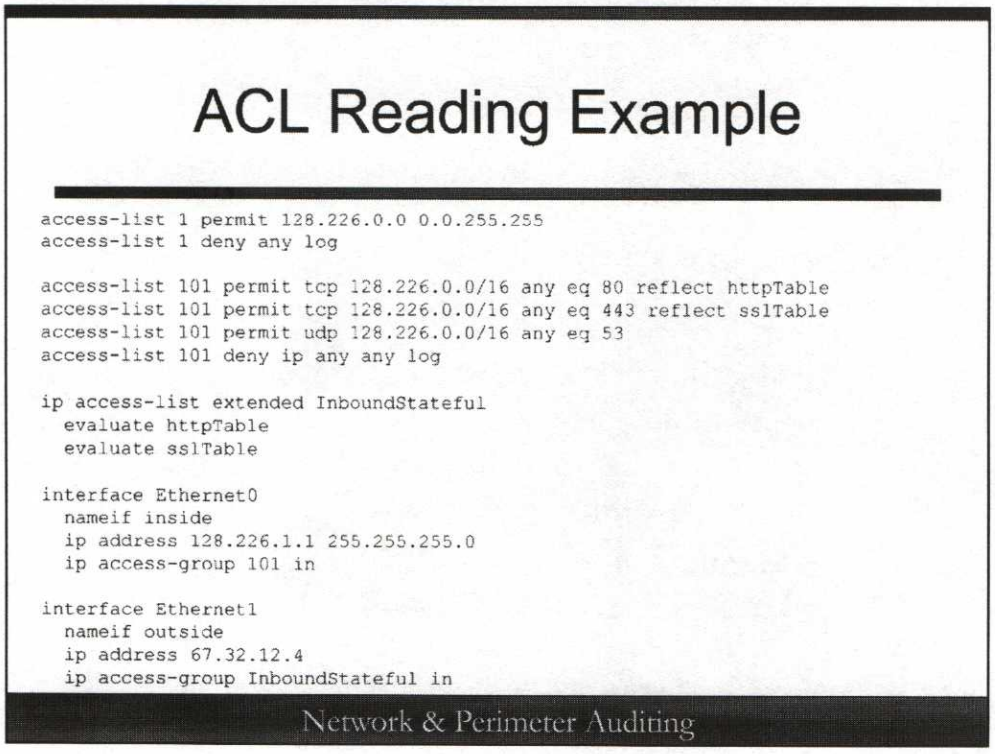
Our initial research reveals that Cisco routers and firewalls all support the creation of two types of ACLs: Standard ACLs and Extended ACLs.

Standard ACLs are rudimentary. They permit you to only write a rule that permits or denies traffic based on the source address of the packet. Even though this is simple, it is still useful. Because the rules are so simple, they are processed very, very quickly. If we are trying to block or permit something that can be defined at an address level, this is a great way to do it.

Standard ACL rules are traditionally configured into numbered access lists. Although the administrator can arbitrarily select any number that would be in range for the correct kind of rule, the number ranges are important. Access lists numbered 1–99 or 1300–1999 define standard access lists *only*.

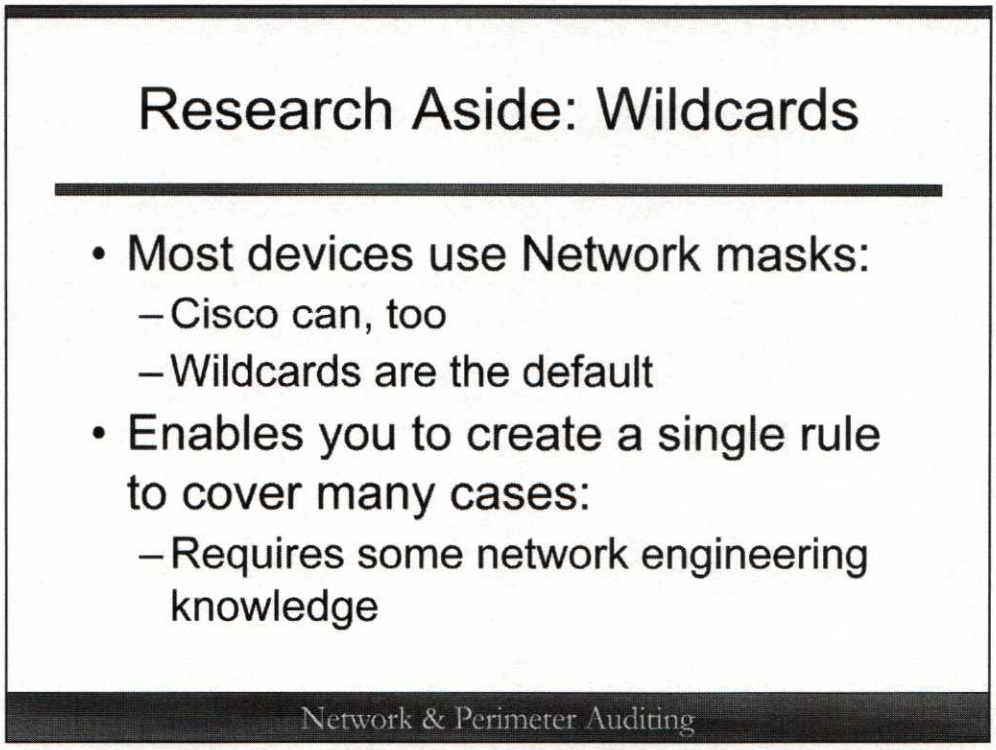
However, if an access control list has a number in the range 100–199 or 2000–2699, it would define an extended access list. Extended access control lists enable you to filter not only on the source address but also on the destination, the protocol, the port number or other options depending upon the protocol in use. It also allows for the use of an "established" keyword for TCP, implementing the "Is SYN set alone?" test that we described previously. Extended ACLs are also the basis for the more advanced features like reflexive access lists, which we will discuss shortly.

Although numbered ACLs are widely used in the Cisco world, there is a far better way to go. We can name access control lists. Not only does this make auditing the device easier but it makes administration much, much easier!



Let's take a look at some actual ACLs and see if we can understand them. Begin by looking over the ACLs in this slide. How much can you figure out on your own? After that, start reading the notes below or participate in the class discussion.

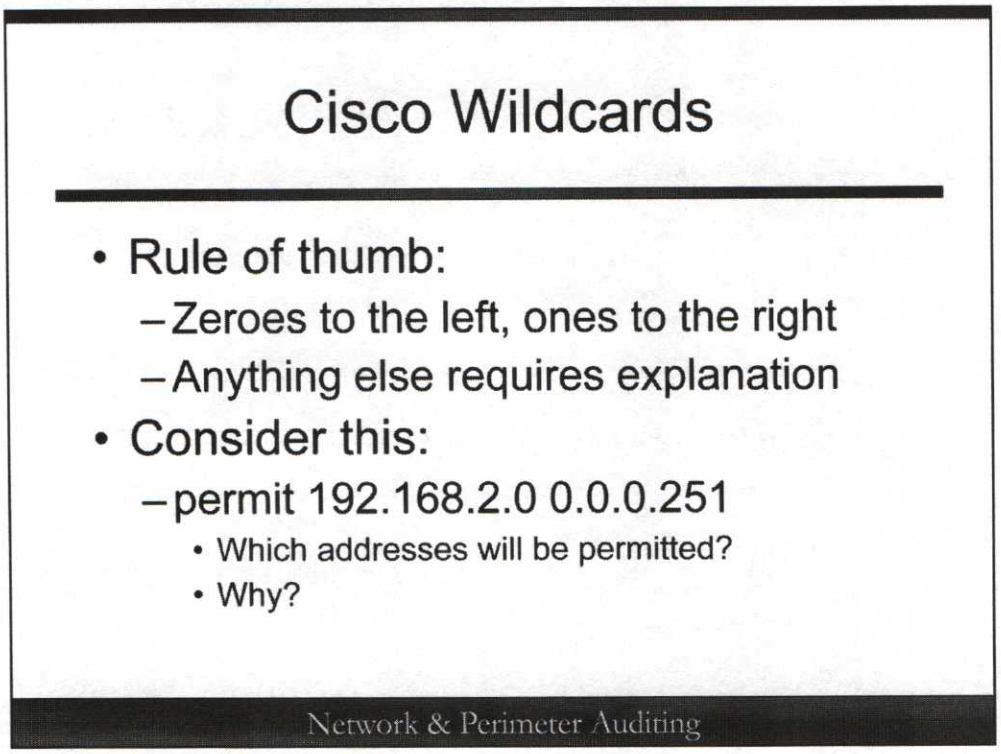
First, we find a standard access control list being created. How do we know that it's a standard list? Because it's access list number 1. This also means that we are limited to filtering on the source address. In this case, we are permitting all the packets from a particular network to pass while everything else is denied and logged.



While examining the ACLs on the last slide, you likely noticed how addresses were written out. Within the Cisco rules language, there are some built-in short-cuts for commonly used network or host addresses. For instance, we can use "all" to define an address of 0.0.0.0 255.255.255.255. We can use host 192.168.1.1 to define 192.168.1.1 0.0.0.0. But what are those numbers that come after the address? In most networked systems we would find a network mask in that location.

Cisco systems can certainly be configured to use network masks, but the default within access control lists is to use Cisco wildcards. Wildcards are simply the inverse of the network mask. In other words, if the network mask is 255.255.255.0, the wildcard that is equivalent would be 0.0.0.255.

Wildcards (and network masks for that matter) within firewall or router ACLs are used to allow us to create a single rule that covers a range of hosts. This is likely not a new idea for you. What might be new, and confusing, is how a competent network engineer might use these. He can create single, simple-looking rules that covers a wide number of hosts precisely.



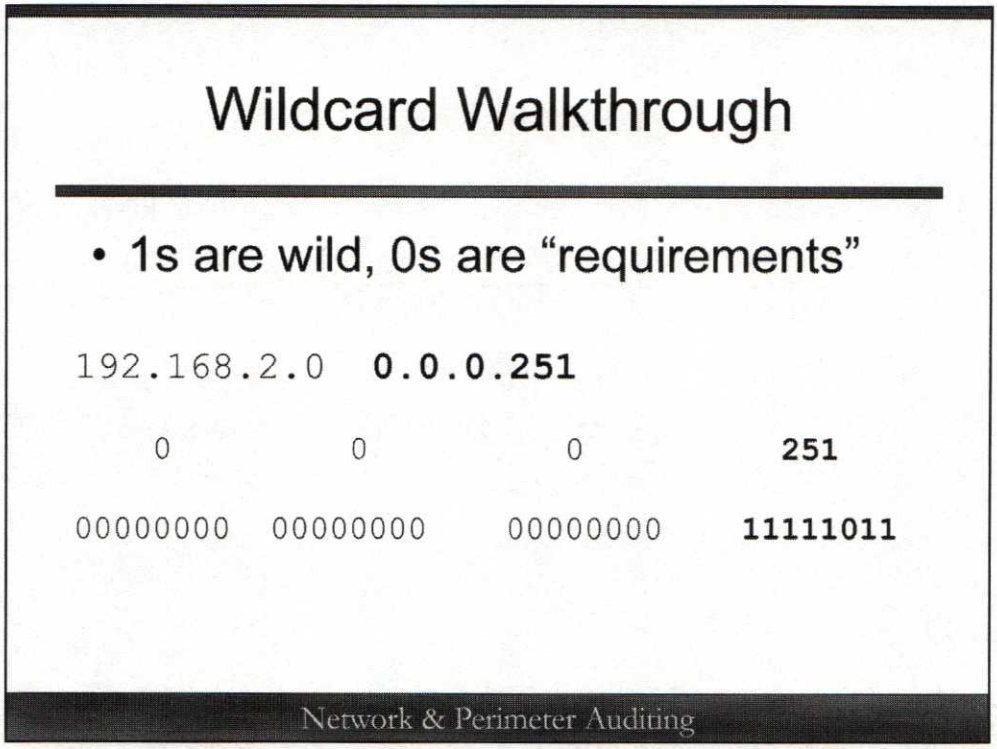
When looking at access control lists on a firewall or router, Cisco or not, we are interested in seeing what is happening with the wildcards or network masks. Truly understanding them is not actually that difficult, but it requires that we can decode them into their binary equivalents. As you likely know, when you look at an IPv4 address, it is called a "dotted quad" and when we refer to the individual values we call them "octets." Why octets? Because each value represents 1 byte in the 4-byte IP address.

The same is true of the network mask or wildcard. With Cisco wildcards, the IP address listed to the left is used as a template to compare inbound or outbound packets to. The wildcard is used to mask off pieces of that address that can be, essentially, ignored. If a binary value of zero is found in the corresponding bit field, that bit in the address of the packet being considered must match the "template" bit exactly. If the bit is a 1, however, then it is masked off, allowing the bit to be on or off and to still match.

That may sound complicated. Let's boil it down to a rule of thumb. We would like to find that all the bits that are turned off are to the left of the wildcard and all the bits that are turned on are to the right. If the system uses network masks instead of this unusual Cisco notation, we would simply invert our rule: Ones to the left, zeroes to the right.

Using some other arrangement of bits is absolutely legal but requires much greater expertise on the part of the administrator. If he has done something else, he should easily and competently explain it to us, perhaps even standing up in front of a whiteboard and showing how the bits all line up.

Consider the example in the slide. With a wildcard of 0.0.0.251, the binary values would be 00000000.00000000.00000000.111111011. What would be the effect of this wildcard? It would allow us to write a single rule that would cover a large number of hosts. Awesomely powerful but full of potential for error!

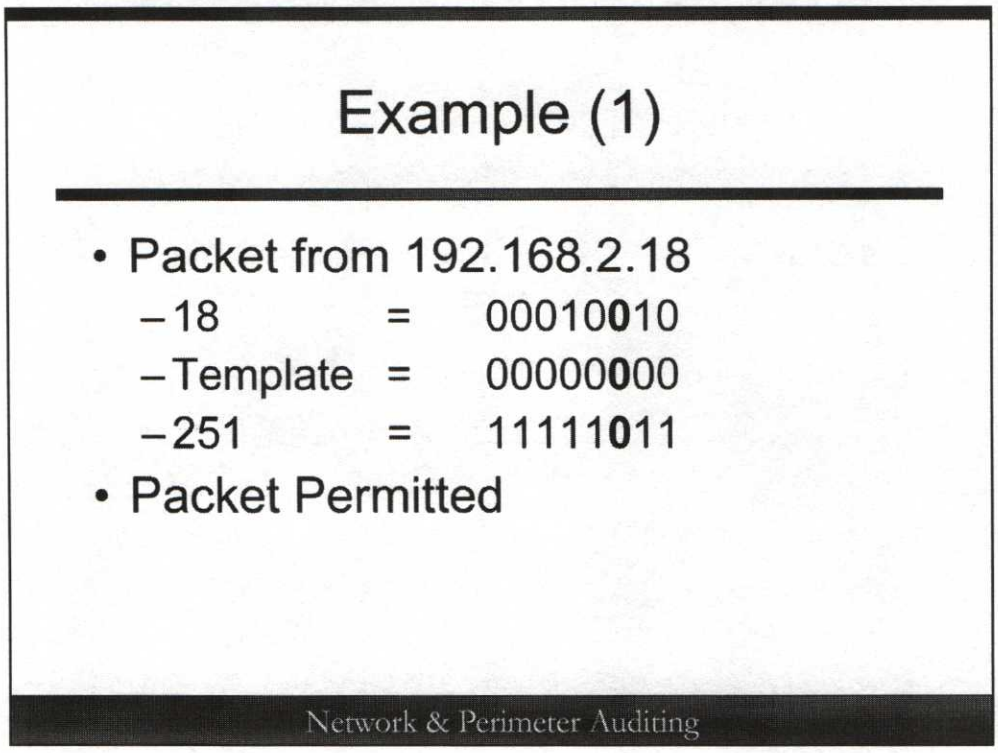


Let's just take this apart. If we were to find this kind of wildcard in use during an audit, one of the things that we would expect is that the administrator could explain it. We would listen to him for a few seconds and then, likely, interrupt him and ask him to use a whiteboard to walk us through it. What would that look like?

We would expect to hear facts such as, "Everyplace in the wildcard that there's a 1 it will match anything but anywhere that there's a zero it must match the rule exactly." What does this mean? What ones? What zeroes?

Look at the slide. Notice that we've taken the wildcard and split it into its constituent parts. Underneath that we've further analyzed it, representing each value in the wildcard as the bits in a byte. Here are the zeroes and ones.

In all the bit positions in which there is a zero, a packet being checked must precisely match the bits in the left portion of the rule (the 192.168.2.0 piece). Anywhere that a one appears the bit can have any value. Let's look at a few examples to better understand what is meant.

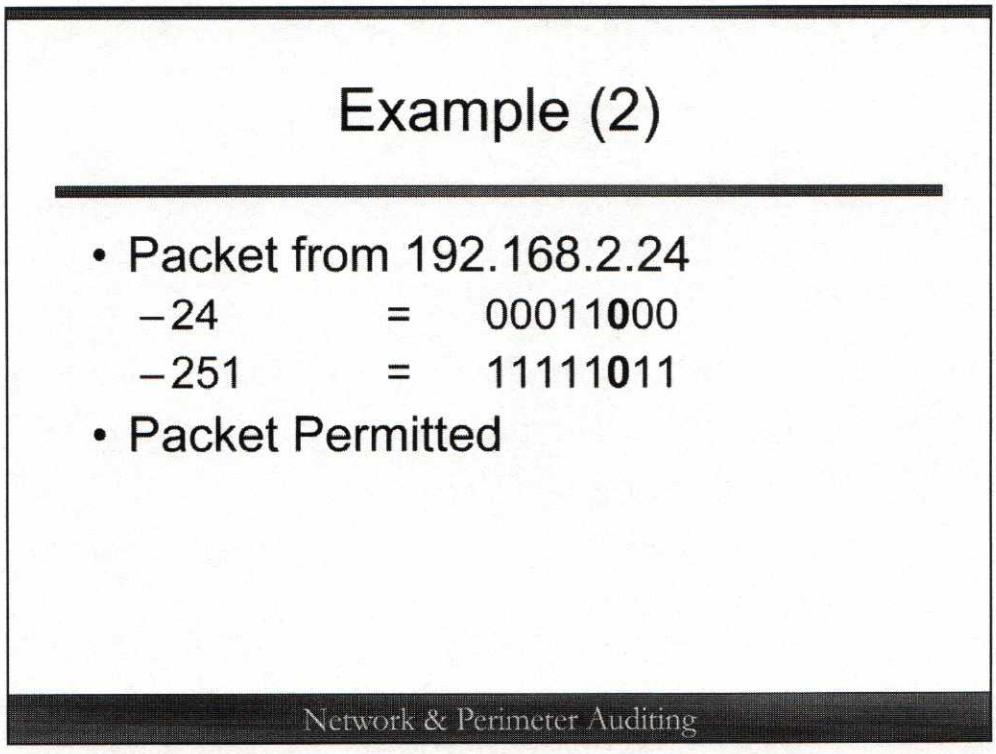


Let's take the case of a packet coming from host 192.168.2.18. We can see that the 192.168.2 part of that precisely matches the left side of the original rule, so we're going to assume that this piece matches. Let's turn our attention to the final octet.

When we convert the decimal value 18 to binary, we arrive at 00010010. Notice that the "template" (or left side of the original rule) is a zero, which is 8 zero bits in binary. In addition, we have the 251 from the wildcard shown in binary. We can see that the position in which the zero appears in the wildcard has been marked in bold. Recall that this means that the packet being considered and the template must match in this position *exactly.*

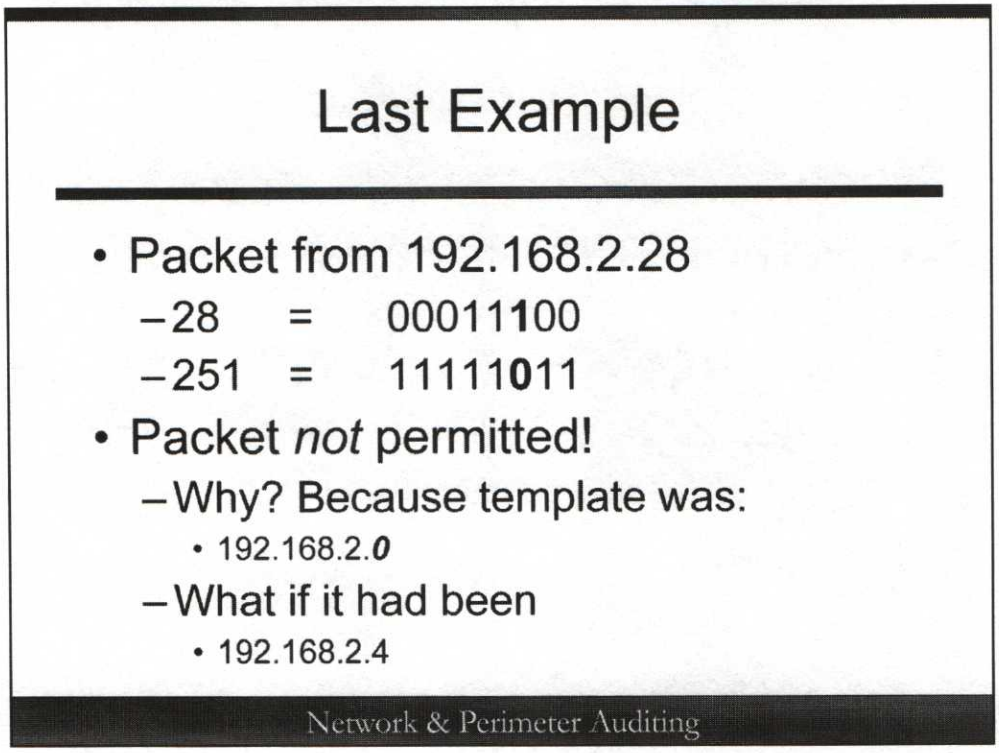
When we look at this position in the other 2 bytes, we find that the bytes *do* match. As a result, this packet will match the rule and be permitted.

It is important to understand that what allows this packet through is *not* the fact that the wildcard has a zero and the packet has a zero. What allows the match is that the template and the packet *have the same bit value in the same position where the zero appears.*



Consider another example. We've dropped the template out of the view because we know that there is a zero in the relevant position. Another packet has arrived, this time from 192.168.2.24.

The address is again dissected. This time the byte comes out as 00011000 in binary. Notice that, again, the third bit from the right is zero, matching the zero from the template. This packet is also permitted.

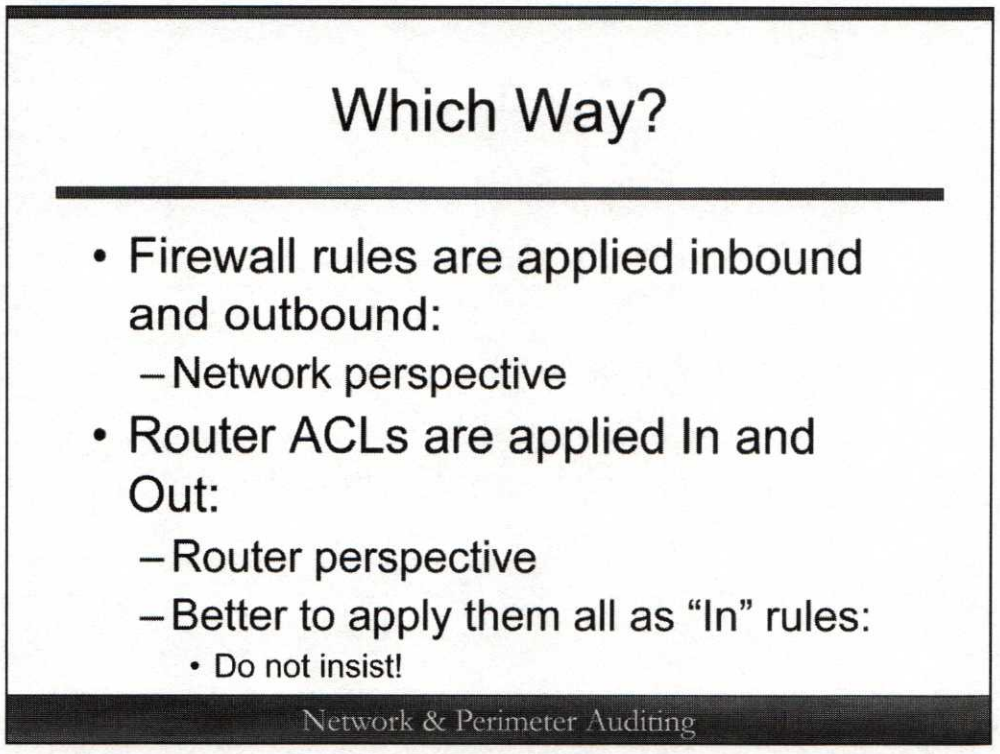


As a final example, consider the address 192.168.2.28. When we represent 28 in binary, we get 00011100. In contrast to the other packets seen so far, this packet will *not* be permitted. Notice that the third bit from the right is now a 1. Because it is a 1 and the template has a zero in this position *and* the wildcard marks this position as an exact match, the packet will be rejected.

But what if the *template* had been 192.168.2.4? In that case, the template would have been 00000100. In this case, with *exactly* the same wildcard, the packet would be *permitted!!* Can you see why?

If you said, "Because the template has that bit on and the wildcard requires that the packet have *exactly the same bit* turned on," then you are correct! Congratulations!

If you don't immediately see that this is the case, see if you can work it out on paper. Ultimately, however, bit masking and wildcards could be viewed as somewhat out of the realm of an auditor; it is not critical that you can specify or analyze these personally. What about the administrator at the whiteboard? It should be abundantly clear to us that he absolutely understands how they work and what they do and do not permit. If he can't explain it then perhaps he should not be using them!



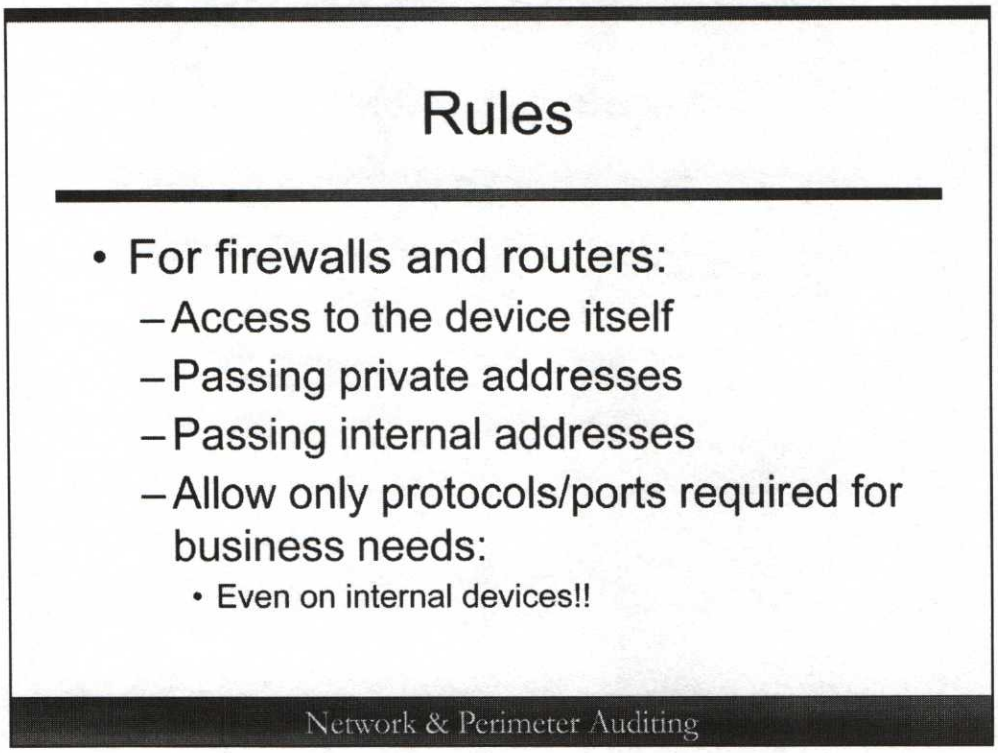
Many firewall systems, especially with GUI interfaces, automatically apply any rules that you create to an interface. This is also generally true on routers where a graphical interface is used to create ACLs attached to interfaces.

Firewalls and routers that make use of text-based configurations, however, enable you to create context-free rules. What we mean by this is that the rule can be created without having to define the interface to which it is applied. In fact, this is how rules are intended to be created! The process that an administrator would follow would be to write some rules as a part of an access control list and then, after the rules are written, to apply them to an interface.

As crazy as it may sound, you will on occasion find that an administrator has written perfectly valid rules that completely account for all the organizational requirements… but he has forgotten to apply them to any of the interfaces on the device. Sometimes, this is an oversight; sometimes it is a training or knowledge issue. This is actually one of the reasons that we take the time to learn how to perform a firewall validation.

Most firewalls have a well-developed notion of "Inside" and "Outside." This allows us to write and apply rules from the network perspective.

Routers, however, do not. They do not actually know which interface the enterprise is connected to. Instead, all the rules are applied from the point of view of the router. When a packet enters the router, whether it is entered from the Internet or from the enterprise, that packet is "inbound." Therefore, when rules are applied, we usually (and certainly on Cisco routers) have to specify whether the access list is applied "in" or "out." When examining ACLs of this type, we would prefer to find that they are all applied as "in" rules. This is the most efficient way to process the rules and packets, yet we would not insist on this as a requirement. There can be good reasons for mixing "ins" and "outs."

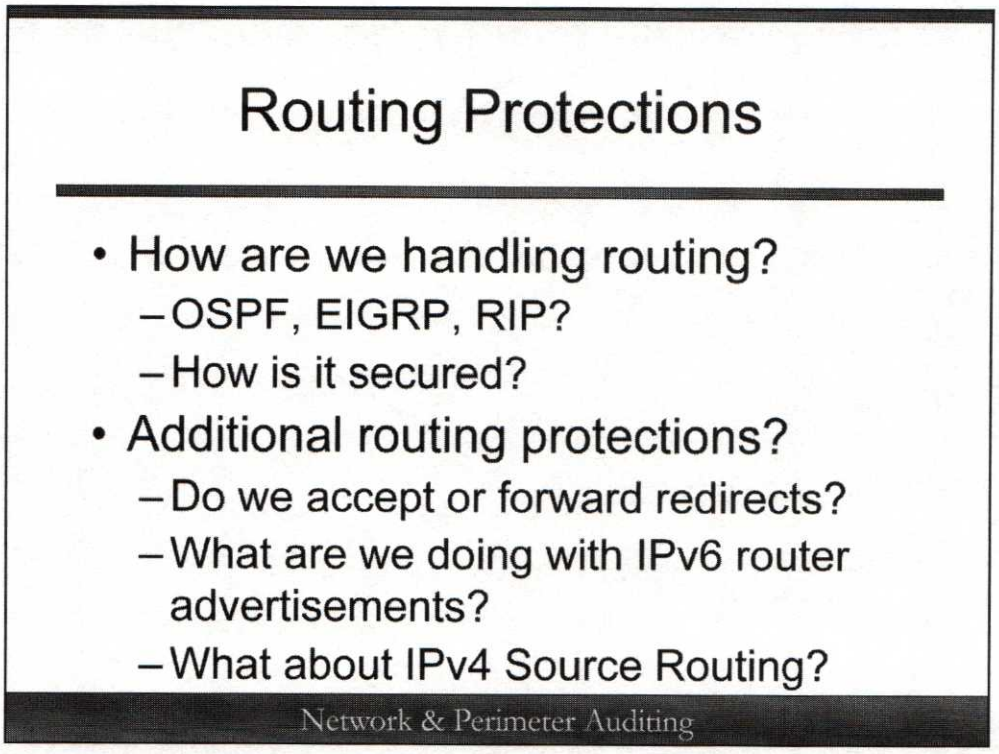


So what sorts of ACLs would we expect to find? Certainly, based on our information flow requirements discussion, we would expect to find ACLs that achieve all the information flow control requirements that have been identified. Even internal routers can benefit from this approach. Remember our discussion of Defense-in-Depth. Routers are in a wonderful position to act as network controls to limit who can talk to what. For example, if we configured the router that feeds the server segment to allow only workstations to connect to the specific services that the servers offer, even if an extra service were running, it would be impossible to connect to it from a workstation segment.

The approach that we describe here implies that our organization is taking a "deny all by default" stance not only for inbound traffic but also for outbound traffic. When we do so, especially if this has not been the approach of our organization in the past, there can be significant resistance. It is critical for the organization to understand that absolutely anything that is needed for business operations can be turned on. As long as someone can present some business case for Torrent downloads in the enterprise, for example, the security team will open up the firewall to permit that behavior. This is a far better approach than letting everything out and then trying to lock down the stuff that's causing trouble.

In addition to these rules, we would like to find rules that limit who can talk to the router, switch, or firewall. We would expect to find that only administrators or the addresses on the network segment that administrators reside on can connect to the devices.

We would also want to see that the device is configured to prevent our internal addressing from leaking to the outside, whether those are private addresses or actual addresses. It is absolutely best practice to push all our outbound network traffic through a NAT, concealing the actual internal addresses and addressing scheme.



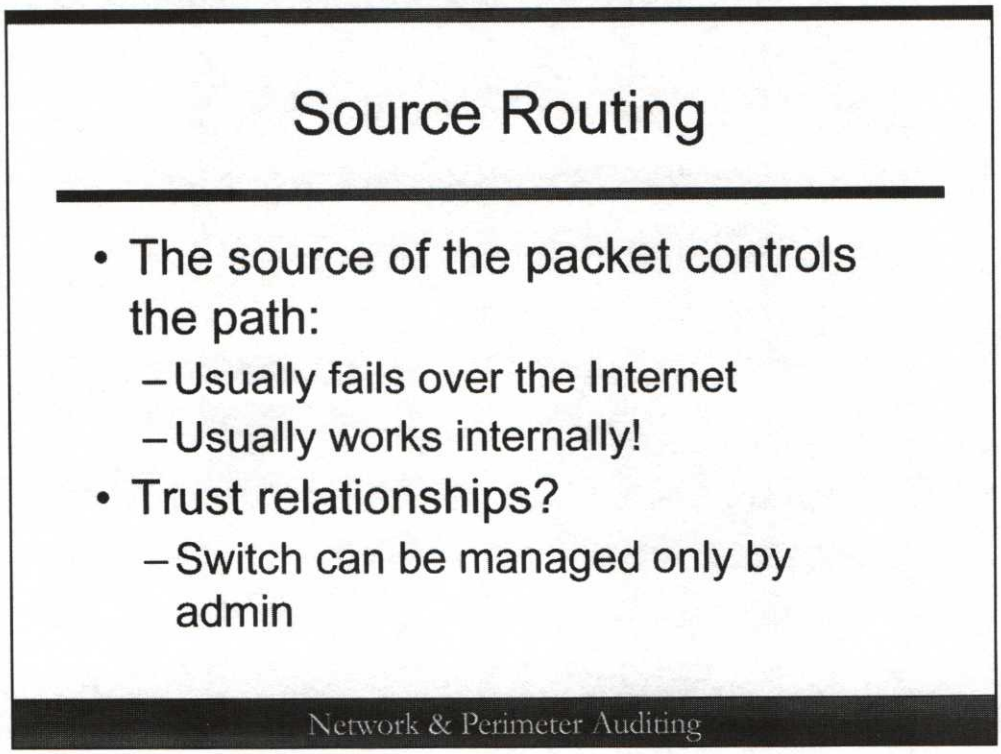
Other issues that need to be examined are not ACL-driven. We want to understand both how dynamic routing is being accomplished within the enterprise (and through the perimeter) and how that dynamic routing configuration has been secured. For example, if we use Routing Information Protocol (RIP), it will be difficult to secure the routes. Why? Because this is simply a datagram-based broadcast protocol. Each router (and possibly firewall) in the environment with RIP configured will periodically (approximately every 30 seconds) broadcast a complete list of all the aggregate networks that it knows how to reach with a cost measured in network hops. Neighboring routers or systems configured to listen for RIP broadcasts will use this information to populate routing tables.

Open Shortest Path First (OSPF) and Extended Interior Gateway Routing Protocol (EIGRP) achieve the same goals as RIP, but both of these support security features to prevent an untrusted host from injecting arbitrary routes or modifying existing routes. Just because these features exist, however, does not mean that they have been configured. We want to inquire to determine how our systems are actually configured.

We're also interested on both how these network devices react to packets that could be used to alter routing and whether these devices allow these types of packets to be passed to endpoints. Let's consider a few of these.

ICMP Redirect messages are supposed to be generated by routing devices to inform a source host that there is a better or preferred path to reach a destination host. Regardless of static routing configurations, unless it has been configured not to, that source host will accept that redirect and actually update its routing table.

IPv6 handles network discovery and routing completely differently from IPv4. Neighbor discovery is used over ICMP to find neighbors, and IPv6 router advertisements are used for routes. If someone begins advertising an IPv6 route and we have IPv6 enabled, our system will prefer this route over any IPv4 route.

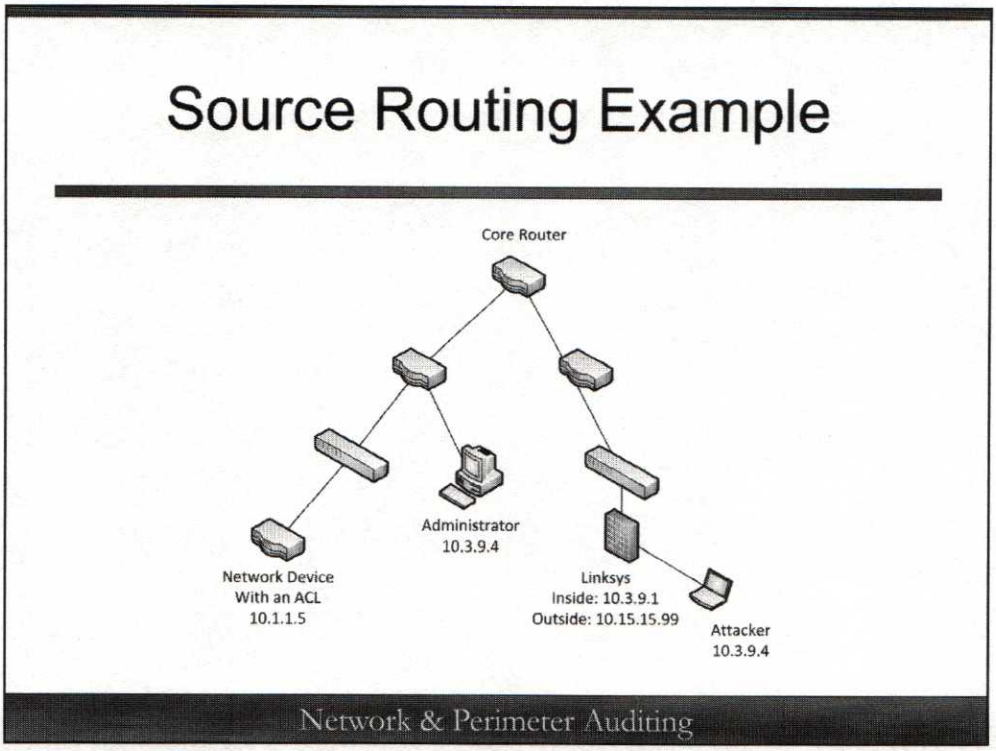


Let's dig into just one of these issues. ICMP redirects and IPv6 router advertisements are conceptually simple concepts, so we'll dig into Source Routing instead. Even though this has been around since IPv4 was first released as an RFC 760 in 1980, many network engineers and security professionals are not familiar with it. Not knowing precisely what it is, they are also not aware of how to defend against the potential risks.

At its most basic, Source Routing is an IP option that allows the source of the packet to control the path that the packet will take to get to the destination. These days, over the Internet, this is not an especially viable feature. Most Internet service providers either block, drop, or simply don't honor the Source Routing feature even though it is part of the IP standard.

On internal networks however, it is unusual to find that the organization has accounted for this particular threat. How can this be used against us? Consider this; what if we have a trust relationship within our organization. For example, imagine that we have a switch or router that has ACLs configured. Among those ACLs is a rule that restricts connections to the administration interface of the device to only specific systems on the network, perhaps the network engineering subnet.

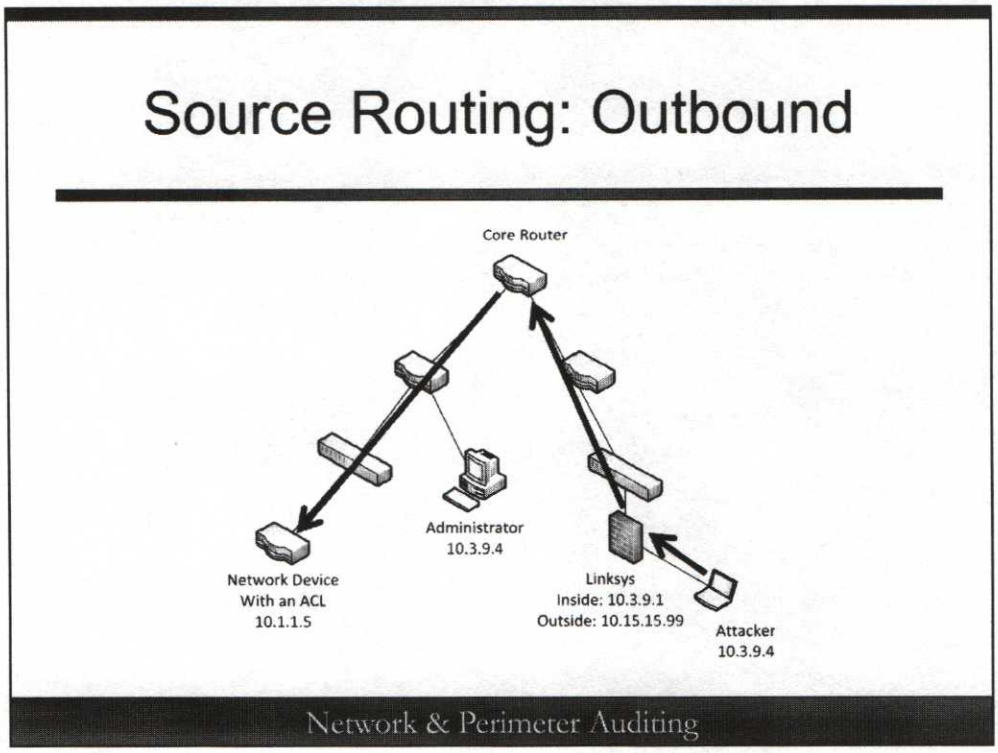
If an attacker can discover this device and can determine a likely source that would be permitted to talk to it, Source Routing can be used to interact with the device. You may be thinking, though, "Why can't you just spoof the traffic? What's the big deal about Source Routing?" Let's look at an illustration that demonstrates why.



Note here that we have a network device with ACLs on the left side of the diagram. Also in the diagram we have a network engineering segment, from which hosts may communicate with the network device. We also have various routers and switches throughout the enterprise.

On the right side of the diagram, we have an employee with too much time on his hands. He has decided that he's going to try to take advantage of the trust relationship so that he can attempt to brute force the management password for the device. To assist him with this, he has brought to work with him an inexpensive router/firewall like a Linksys or Netgear home router.

The "external" interface of the personal router will receive a DHCP address from the enterprise. The "internal" interface, however, is completely under his control. He sets this to an address on the remote network where his target resides. Next, he sets his IP address to match an address on the network engineering subnet.

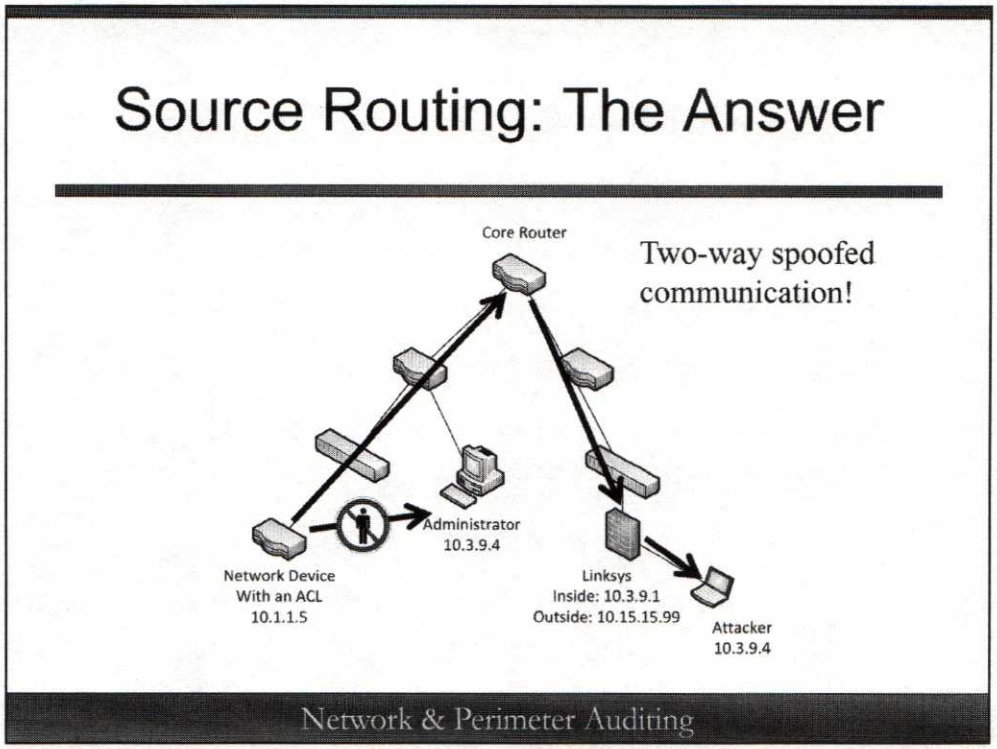


Under the hood, the destination address field within the IP header does not actually have the target host in it. Instead, Source Routing allows the source of the packet to specify up to nine hosts through which this packet *must* pass to get to the destination. To be more specific, what we are describing here is known as "Loose Source Routing." A second form, "Strict Source Routing," is more difficult to accomplish. With Strict Source Routing, the originator of the packet must specify every single host through which the packet *will* pass to reach the destination.

With this list created using Loose Source Routing, the first host in the list is actually in the destination host field. When the packet reaches this host, because Source Routing is enabled as an option, the next address in the list of hosts is moved into the destination address field. This continues until it reaches the ultimate target.

In our case, you can see that the first host through which it must pass is set to be the external interface of the Netgear device. The address after that can be any host, including the actual target host. The packet routes its way through the infrastructure until it reaches the target. When it does arrive at the target, though, note that the TCP, UDP, and application layers do not actually know that Source Routing was enabled, so from the perspective of something like TCP Wrappers, this packet came from the trusted partner.

What happens when the host tries to answer?



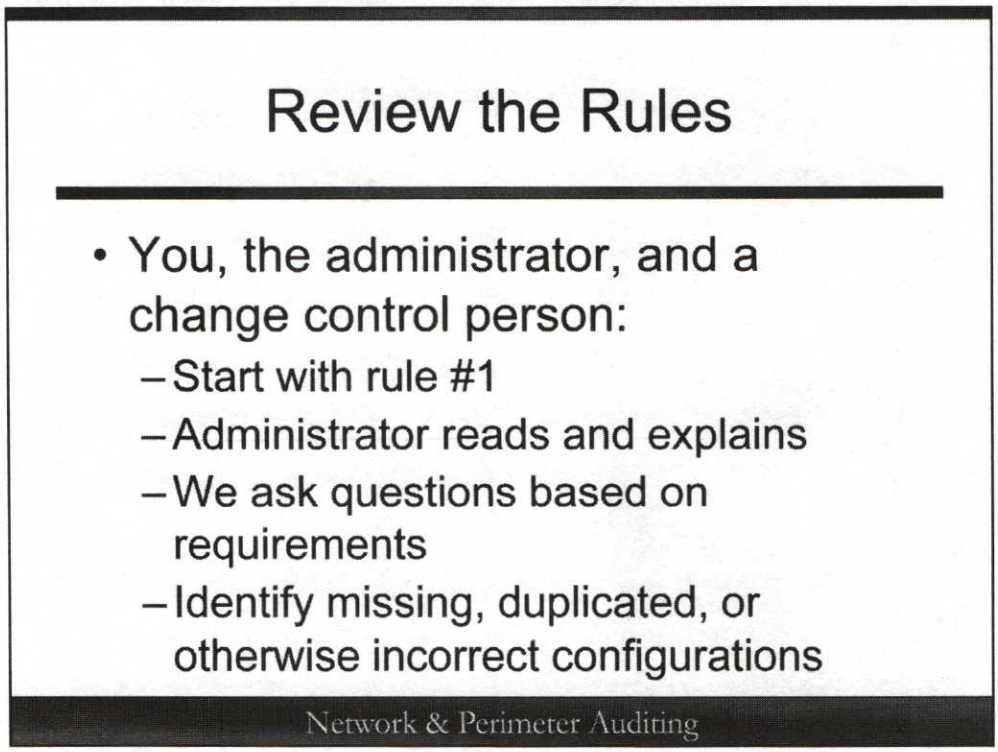
When an IP option is enabled on an arriving packet, the IP RFC requires that the appropriate response option be enabled with the appropriate options list. In this case, the appropriate answer to a source routed packet is another source routed packet. In this case, though, the options list, the list of hosts through which the packet must pass, is reversed to return the answer.

What effect does this have in our network? Well, even though the host to which the network device would like to send a response is local, to reach that destination the packet must route first to every other host in that list. Because the attacker's Netgear device is one of the hosts in the list, that packet gets to decide how to route to the final destination. When that packet arrives at the Netgear device and the ultimate destination is moved into the destination address field, the router realizes that this is a local address in its internal interface.

This means that it is trivially easy to create a two-way connection to the remote host while simultaneously spoofing our address. It also means that we can accomplish this without ever "attacking" the trusted host. Even better, if someone looks at the logs on the target host, the entire attack is blamed on the other computer! Clearly this is a powerful and dangerous way to attack an internal network.

How widespread is Source Routing? Well, it's actually pretty rare to find someone doing it. But if you do see it being done, it indicates that this individual has a high degree of competence with networking protocols and understands something of your network design. Source routing is a built-in feature to many command-line utilities such as FTP, Telnet, Ping, and so on.

Preventing Source Routing is easy. For a Cisco device you simply need to include the no ip source-route option to configure this globally on the device. With this configured, source-routed packets will be dropped by the device.



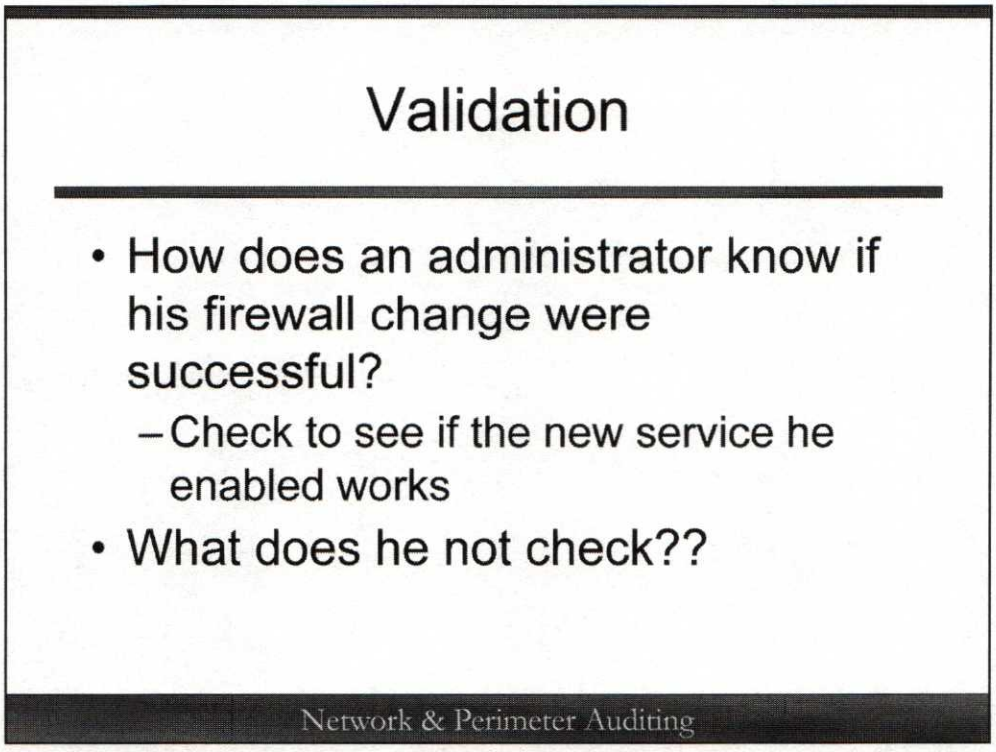
Let's continue on to how we actually audit the rules on the router or firewall. We absolutely want to have a meeting with the administrator. We would also like to have someone who has change control authority with us at the meeting, especially if it is the first time that the device is being audited and validated. In almost every case, my experience has been that the first time looking at a device with any kind of substantial configuration there will be something overlooked, usually something not documented, that needs to be adjusted. Rather than waiting longer to fix things or manufacturing findings by writing the administrator up for repairing the configuration, it's easier to have someone with us who can approve any necessary changes.

Of course, we would have asked the administrator to provide a copy of the configuration to us well in advance of the meeting. When we ask for this configuration, we always remind the administrator to redact the password hashes out of the configuration file. We can verify that these appear to be configured correctly when we meet together. We don't want a copy of the hashes, though.

Having the configuration ahead of time gives us the opportunity to do research into any configuration options that we don't recognize. We also have the time to map the rules out with the information flow requirements that we have already determined. Before we go to the meeting we already know which rules we have questions about.

When the meeting starts, we will ask the administrator to begin at the top of the configuration and begin reading and explaining the configuration to us. This gives us an opportunity to assess his overall competence with the technology, but it also provides him an opportunity to find any errors. This is always more pleasant than us pointing them out!

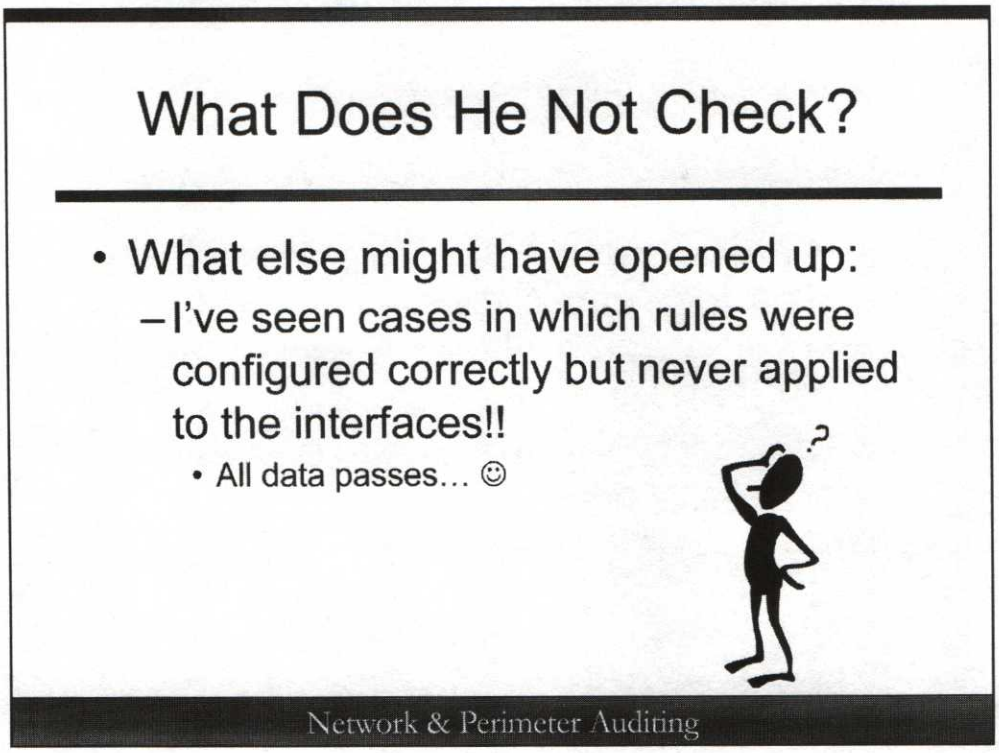
As he reads, we're looking to verify that all requirements have been met and possibly identify any duplication, missing, or otherwise incorrect entries within the configuration.



What we have just done is verified that the rules and configuration appear to be correct. You and I both know, however, that at times technical systems seem to act in mysterious ways despite the configuration. Why? Often because something is overlooked, misspelled, slightly misconfigured, and so on within the configuration. It can even be caused by bugs in the technology!

When an administrator makes a change to a firewall or router to permit some new service to pass through the perimeter, how does he know that his change was successful? How does he test it? Almost invariably, the administrator will try to use the new service and, if he is successful, assume that all has gone well.

What, however, does he not check?

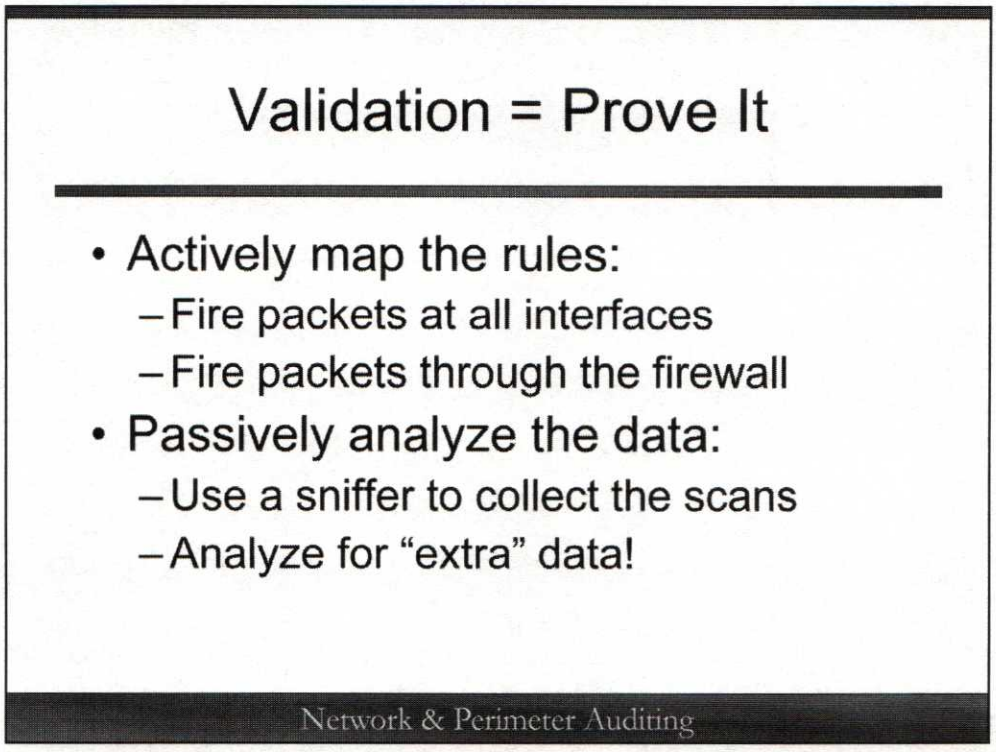


He does not check to see what other things may have been affected. Are there now protocols or hosts accessible that were not reachable before? Has this change had any side effects on things that were already permitted?

It may sound extreme, but more than one time I have found devices where the rules were all correctly written but the administrator had neglected to apply them to any of the interfaces on the system! This made it "appear" that things were correct, but in actual operation everything was being permitted to pass the device! (This misconfiguration, by the way, is far more common on routers than it is on firewalls. The reason is that rules must be explicitly applied to router interfaces whereas firewall rules are usually created in the context of an interface.)

The impact of this kind of misconfiguration can go undetected by the administrator because everything seems to work! Because he is testing to only see that things can go through rather than testing to verify that things are blocked, we can easily end up with big holes in our network infrastructure.

These issues are precisely the reasons that validation must be performed.

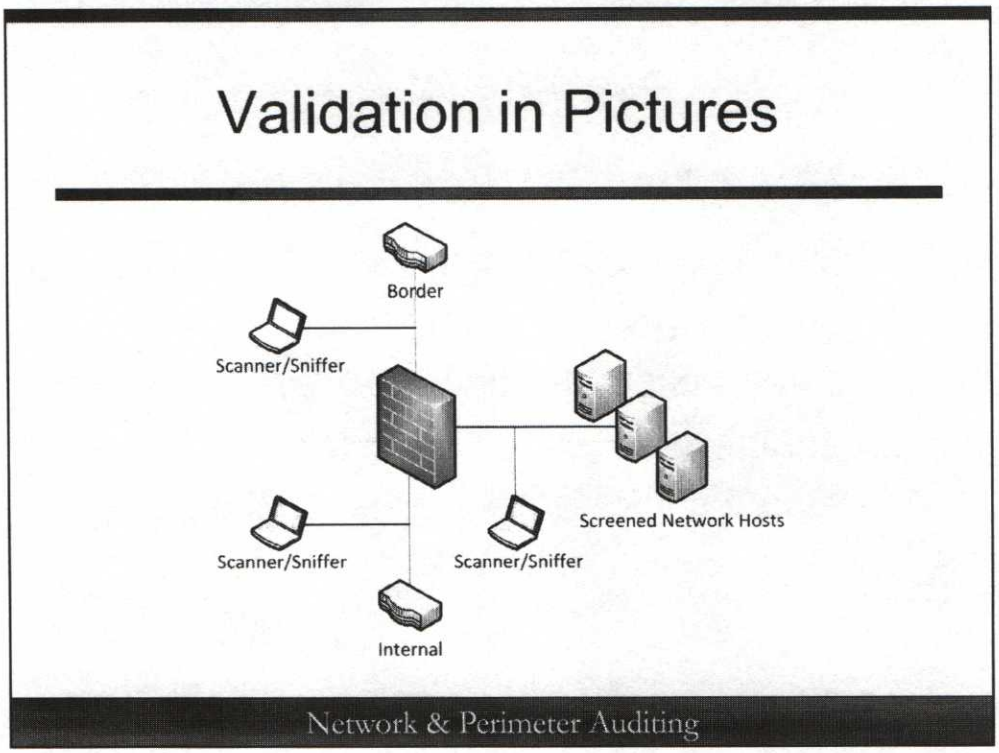


Validation in this context means that we are going to verify that the rules actually do what we think they do and that the firewall, router, or other network device is operating in the way that we expect.

To perform this validation we want to scan the firewall completely from all perspectives. In other words, we want to scan the outside interface from the outside, the inside interface from the inside, and any other interface from the network that it is on. This, however, tells you only how the firewall is responding.

We also need to try to send packets through the firewall in all directions. When we do this testing, it is not sufficient to just use a scanner. We must also use a sniffer of some kind. The sniffer would be positioned on the target network on the other side of the firewall.

The reason that the sniffer is necessary is that it may be that the firewall is silently dropping the packets, which makes the firewall appear to be closed for that service. It may also be that the firewall is forwarding the packets, but there is no host on the other side to answer. It may even be that there is a host behind the firewall, but the response is being dropped by the firewall. The problem, you see, is that all three of these cases look exactly the same from the perspective of the scanner. The sniffer enables us to figure out what is *actually* penetrating the firewall.



Here is a diagram to illustrate what we've just said. We want to scan from all perspectives and in all directions. As we do so, we relocate the sniffer and scanner systems so that we can form a complete picture of precisely what data passes through the router or firewall.

When we do this validation, our goal is to verify that the packets and services that are available exactly match the configuration settings that we have already reviewed with the administrator. What if we find something extra? The administrator will have to track down and fix the issue or create some kind of compensating control to limit the activity.

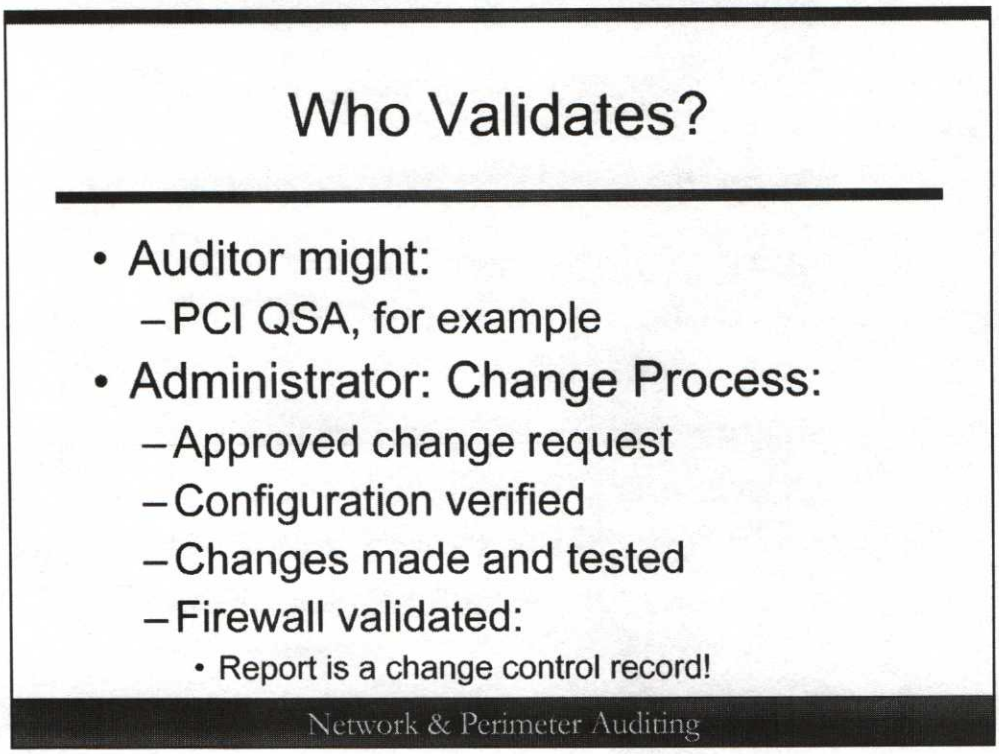
Validation How To?

- Final part of the next lab:
  - Set of free scripts that leverage open source software
  - Answer a few simple questions; scan is performed
  - When scan is completed, run another script, answer questions, and report is generated!

Network & Perimeter Auditing

So far this all sounds great in theory. The question is, how do we do this? Especially as an auditor, you may have extremely limited experience using a sniffer!

Never fear. In the next lab we walk you through this process step by step so that you can clearly see what's happening. Then, we introduce some easy-to-use scripts that automate the entire process for you, generating a report that tells us exactly what gets through the firewall.

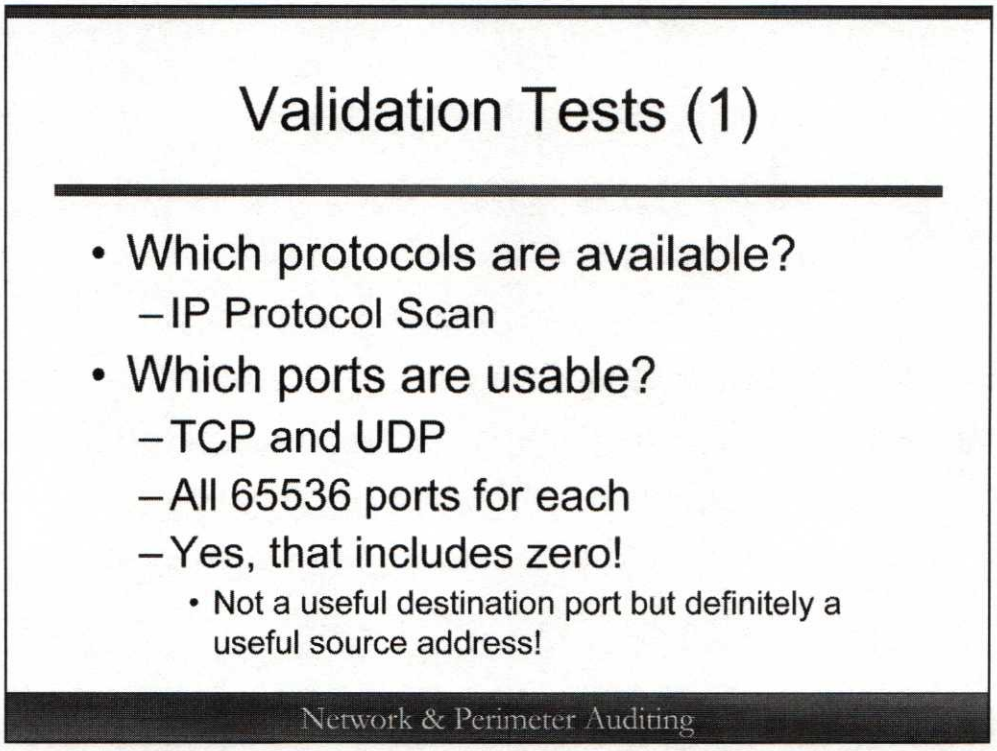


The other question that you may have at this point is, "Who, exactly, performs this validation?" Although it is true that some auditors may need to perform this test, most of the time the auditor is not the person running these tools. This is great news for us. Always ask yourself whether the task you are performing is actually someone else's responsibility.

In this case, think about it this way. A firewall or router administrator is about to make a change to the security configuration of his device. What should he do first? Of course, he should validate that the configuration has not been modified since the last approved change. Now the administrator makes his change and tests it out. What should he do as a final step in his change control? He should validate that the device is actually behaving correctly!

This means that it is actually the *administrator* who should be performing the kind of validation that we do in the lab. He should be performing this validation after every change. The report that expresses the validated configuration should then be included as a part of the change control documentation.

If you think about it, this makes our job much easier. This allows us to move up the process that manages the firewall rather than having to look at the same settings again and again.

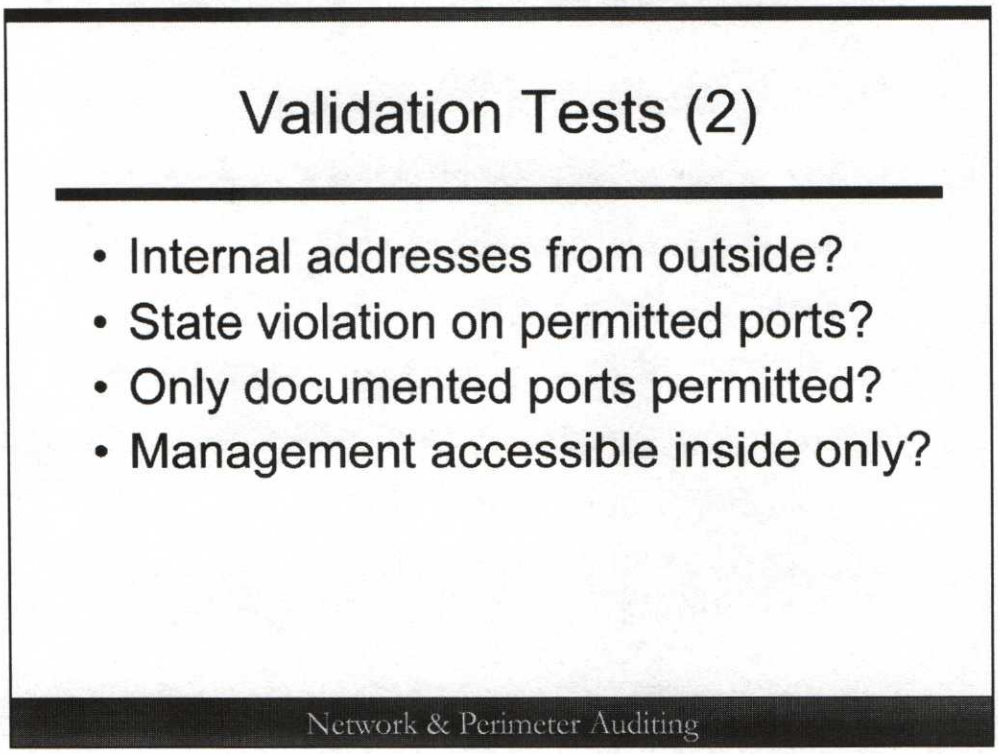


So what sorts of tests actually need to be done to validate the firewall? Everything! Let's discuss these. The tests to be performed are not listed here in any particular order. We simply need to make sure that we have done all the things listed here.

First, we want to verify that only required protocols are available on the firewall and through the firewall. The IP protocol supports 256 different subprotocols that are identified in the 9th byte of the IP header. Of these, there are only approximately 130 defined protocols. On any given network it is unusual to have more than 4 or 5 in use. These include things such as ICMP, UDP, TCP, IGMP, and IPSec. Of course, we can enable any protocol we need; the question is, are we permitting more than we need!

Next, we want to see which ports are permitted through the device using TCP and UDP. TCP and UDP support 65536 ports each. Although the highest port number available is 65535, port zero is in fact a valid port; though it is not used in normal networking. However, it is a port that is often overlooked! It's common to find rules that are blocking packets from or to ports 1–65535! This means that port zero can often slip under the rules.

Normally, scanning for UDP is incredibly slow. When performing a firewall validation, however, we do not have to wait for responses. This means that we can perform the UDP scanning rapidly, too.

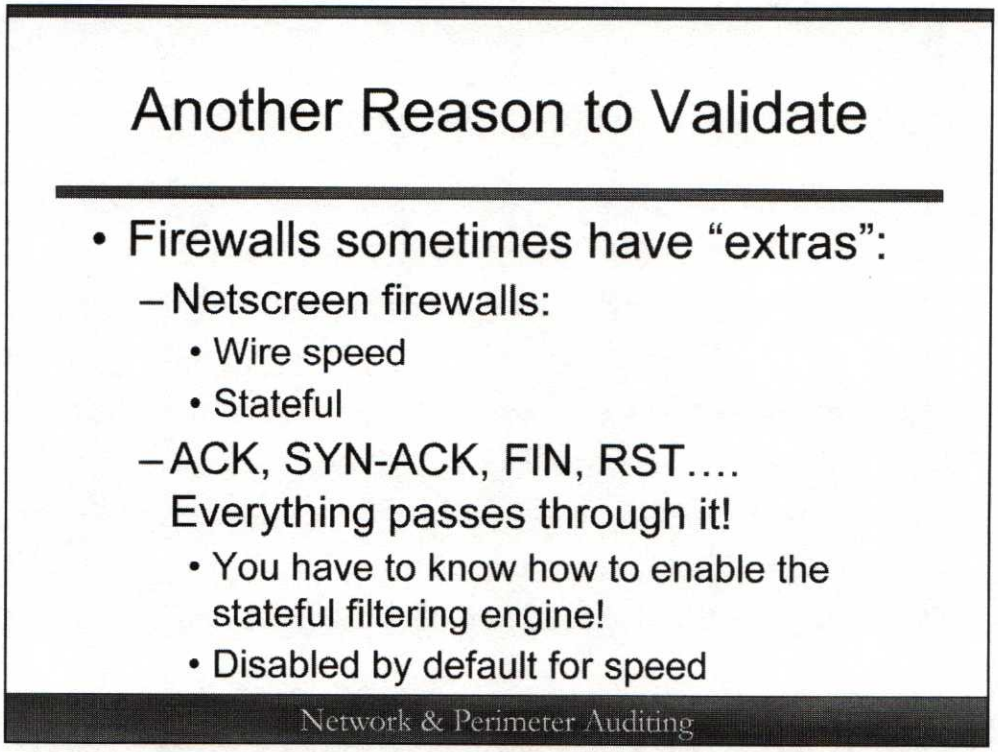


Other tests are contextual. For example, we want to see how the network device behaves when an internal address presents itself as a source address coming from the outside. This is a good test to perform regardless whether we use private addressing internally.

If our device is supposed to provide stateful filtering, we are also interested in determining how the device behaves when you send packets that violate state. Whenever you want to perform a validation of this sort, you should know ahead of time what should happen. For example, if the device is stateful and permits activity, then the only packet that should be permitted through it is a SYN. If we perform our scan with anything else, that packet should be dropped because it violates state until a connection is established.

Because we scan all possible ports in all directions over both UDP and TCP, we want to compare the results with the information flow requirements and the firewall ruleset. Obviously, only ports that are explicitly permitted should pass through the device. If you discover that there are "extras" that are not configured, this could be a limitation of the network device that you are validating. If this is the case, a compensating control would be required.

We would also like to verify that the management interfaces for the device are restricted so that they can be accessed only from the inside. If an administrator needs to perform some task from his house, he must be required to log in to the VPN *first*; then he can connect to the administrative interface from inside.



Before we start putting this into practice, let's give you another reason why a validation must be performed. Although it is certainly true that administrator error can lead to incorrect rules and leaky network controls, it is also entirely possible that the network device that we use has some "features" that allow more through than we might like or be aware of.

As an example of this, consider this experience. Some engineers find a fantastic deal on Netscreen 16 port firewalls. These firewalls are, according to the advertisements, fully stateful, and they operate at wire speed.

When vendors say that a device operates at wire speed, they are saying that there is little network latency. Essentially, if you have a 100-megabit interface, you can process traffic and apply rules and maintain full throughput on that port. Especially given the price quoted, this firewall was a fantastic deal!

Before the engineers are permitted to use a device in production on the network, they must perform a validation of the device. This is precisely what we are preparing to do in our lab. Because the device is supposed to be stateful, they define a set of test criteria based on what should and should not be permitted through the firewall.

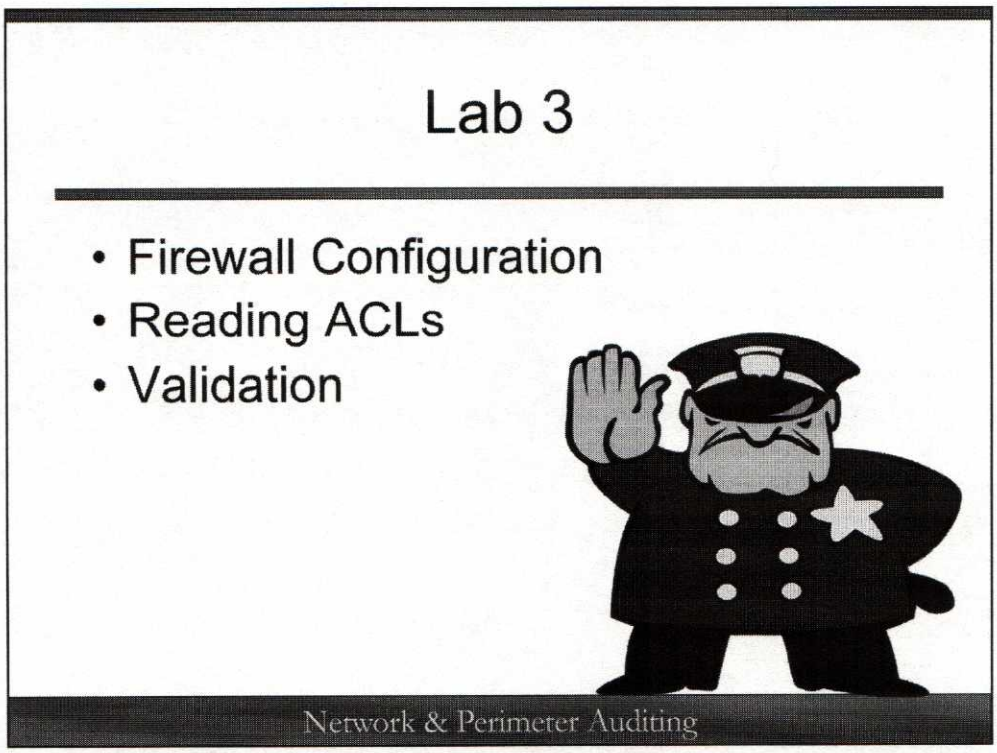
When the validation is actually performed, they discover that if the firewall had a permit rule on a port, it did not matter if the packet violated state; every packet on that port would be forwarded. Discouraged by this finding, we worked our way to third level support at Juniper. That support person explained that the device would perform stateful filtering only if we actually enabled the stateful engine. This required a multi-step complicated process, after which the device did in fact block out of state packets; however, the device was no longer operating at wire speed.

In the Lab

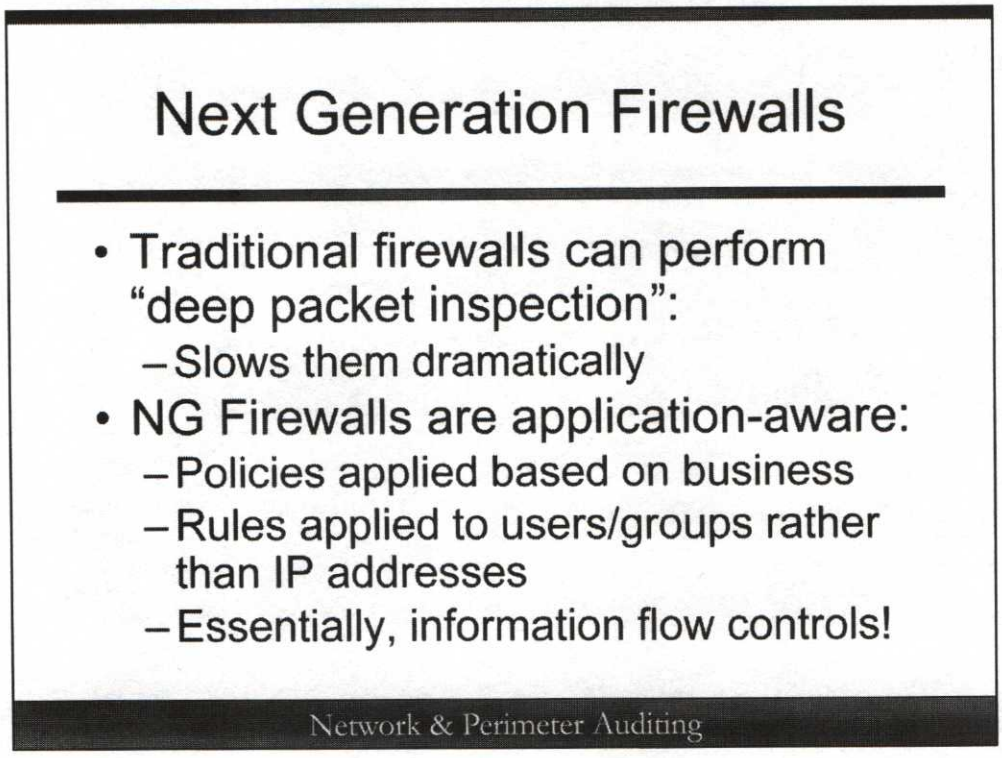
- What we have:
  - Virtual systems that can be used to perform a firewall validation
- What we will do:
  - Use some precooked captures to perform the analysis piece
- Why? Because the *administrator* should really be firing the packets!

Network & Perimeter Auditing

This page intentionally left blank.



That's enough talking for now. Let's take everything we've talked about and apply it. Open your workbook and follow along with the instructor as you work through this next lab. In it, you will have the opportunity to learn about various tools that can be used for performing these tests. You will also have the chance to use tools to perform a firewall validation in addition to using scripts to automate the entire process!



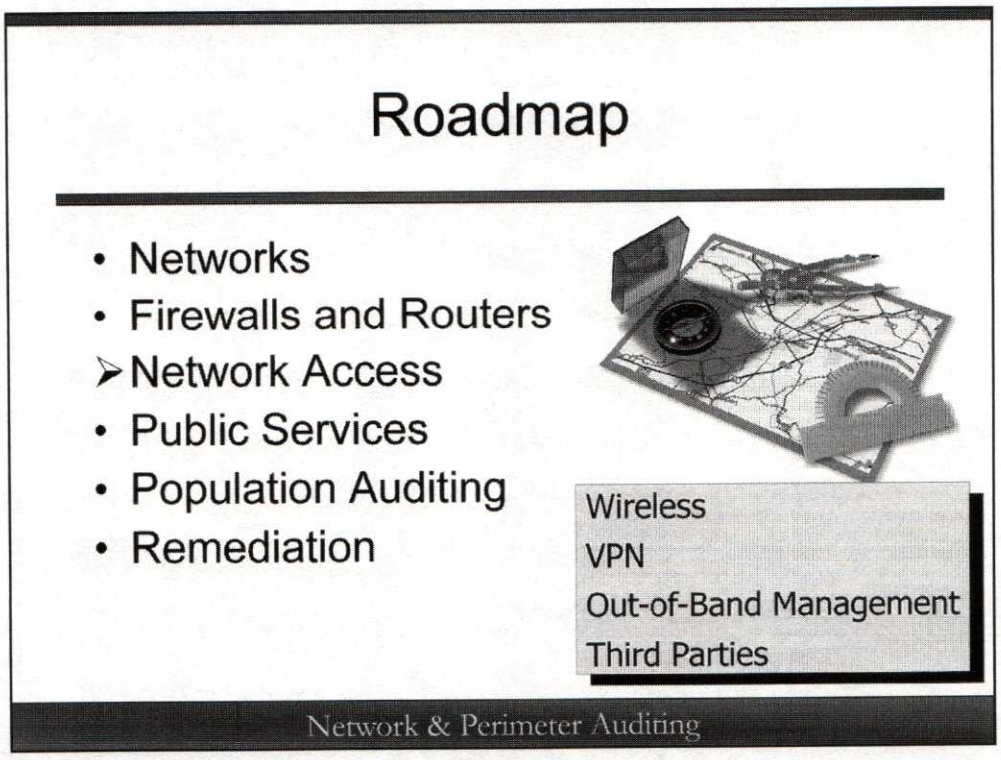
The new kids on the block in the world of firewalls are next generation firewalls. These devices represent a substantial paradigm shift for network security. Interestingly, because of this adjusted approach, it has been my experience that management of these devices can be a steep learning curve for a seasoned firewall administrator. However, it is often easy to teach someone who is *not* a firewall administrator how to configure and manage these devices!

So what changes? Many traditional stateful firewalls enable you to perform some degree of deep packet inspection. Deep packet inspection means that the device may have a rule that permits traffic on port 80, but recognizing that anything could potentially be on that port, they can look into the packets to determine if it is actually web traffic. For instance, in a Cisco firewall or router, you can add the inspect http action to a permit rule to ask it to verify that http traffic is found on that port. If something else is there, it will block the traffic.

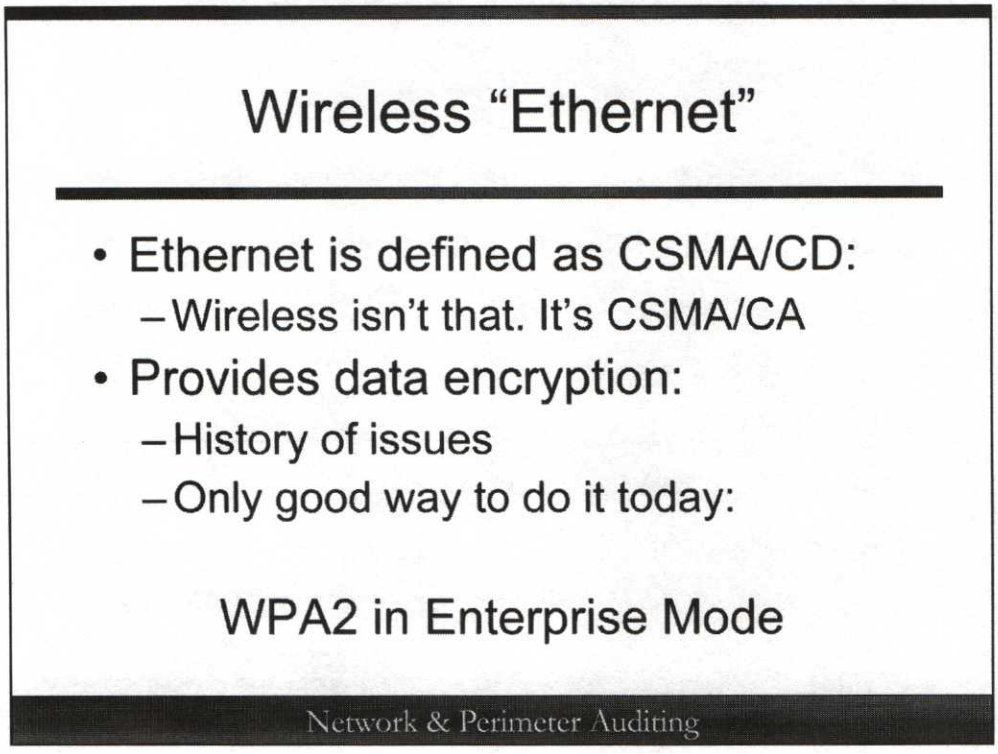
Next Generation firewalls are primarily deep packet inspectors. They are optimized to perform this deep analysis only one time rather, allowing them to operate at or very close to wire speed. In addition, although you can create rules based on addresses and ports, the primary way of specifying rules is based on users, groups, applications, and data.

In a real sense, creating rules in a next generation firewall is defining rules that match our information flow. It's true that we are doing the same thing in traditional firewalls, but because these devices "understand" the content, it allows us to much more closely model these information flows. There is a downside, of course. If we use some kind of custom protocol or something that the device does not understand natively, we are essentially reverting to a classical packet filter or stateful device.

Probably the best known example of a next generation firewall on the market today is the Palo Alto series of firewall appliances.



Now that we've covered the network infrastructure and the Internet-facing portions of the perimeter, let's take a look at "externally accessible" systems and systems that are designed to permit our users access into our network.

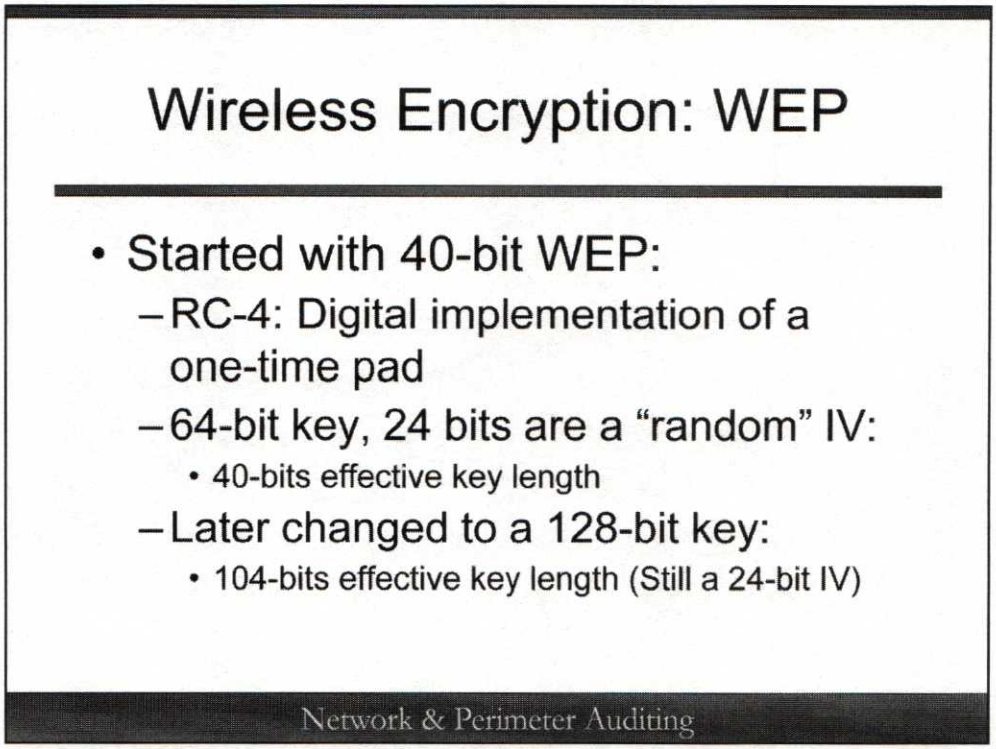


The first of these externally, or even publicly, accessible systems that we discuss is wireless technology. Wireless actually comes in a lot of shapes and sizes these days. You can take what we discuss now and extend it to other sorts of wireless systems that we have (Zigbee, for example, which is used in many building control systems).

Wireless, 802.11, is where we focus. Interestingly, what we call "Wireless Ethernet" does not actually conform to the standard that defines Ethernet. For something to be "Ethernet," it must be Carrier Sense Multiple Access with Collision Detection (CSMA/CD). Wireless actually uses a different strategy. Because the shared medium is the air and because these different systems that are communicating may have interference from systems on different networks or even from entirely different communications systems, they cannot perform collision detection. Instead, they use collision avoidance strategies. These include Direct Sequence Spread Spectrum (DSSS), which is a technique for not only duplicating and spreading the bits in each message out over multiple frequencies, but also switching from one frequency to another. This is the way some wireless analysis tools like InSSIDer show your wireless activity as covering multiple channels simultaneously.

802.11 wireless solutions have had a history of security issues. Recall the discussion that we had regarding time-based analysis. To properly defend a system, we have to understand what the protection is and then assume that the protection fails. In wireless, probably more so than with other systems that we have, this is an easy thing to do because there have been so many issues.

Before we dig into the history of wireless, let's just come out and tell you the right way to deploy it today. The only acceptable deployment, from a security perspective, within an enterprise today would be WPA2 running in Enterprise Mode. We'll define what this is shortly.



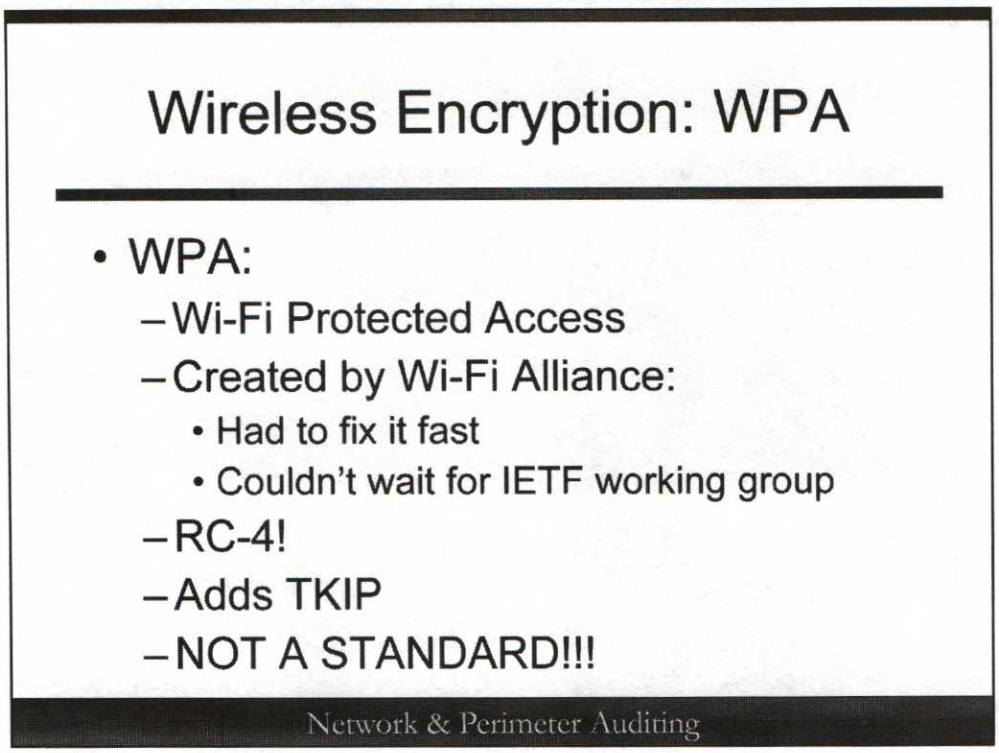
When 802.11 wireless technology was first created in the '90s, the working group developing the standard recognized that broadcasting data wirelessly could lead to a lot of confidentiality issues. For this reason they created Wired Equivalent Privacy (WEP). Their stated goal was to make your wireless communication just as secure as your wired communication. Because there have been so many security issues, some security people say that these folks were 100% successful because your wired communication actually isn't that secure.

In any event, to provide this protection they needed to select an encryption algorithm. They wanted something that would be very, very fast and very secure. They also wanted something that would allow for variable payload sizes. This led them to choose RC-4.

RC-4 is a digital implementation of a One Time Pad. A One Time Pad describes a code system where two people communicating have code pads. Each sheet on the pad tells the individuals how to encode a message, essentially acting as a key. When one of the people sends a message, he uses the top sheet on the pad to encrypt his message and then he destroys the top sheet on the pad. His partner with the matching pad uses that same top sheet to decode the message and then destroys his sheet as well.

One Time Pads provide Perfect Forward Secrecy (PFS). You may have seen this term attached to the description of your VPN. What it means is that even if an attacker could convince you to encrypt and send a known plaintext, allowing him to break that key, this would not give him insight into any previous or any future message. This is an important feature of a crypto-system.

Unfortunately, a One Time Pad becomes easier to break the more often you use a key. WEP used a 64-bit key (later a 128-bit key), 24 bits of which were a supposedly random value. The other 40 bits were a preconfigured key. Can you see the problem? The more packets we send, the more likely we will repeat the key. In fact, if we send 2^24 packets, we *must* repeat the key, allowing the key to be broken!



The discovery of the problem with WEP was made early. In fact, there are other weaknesses, called key scheduling problems, where certain keys and certain initialization vectors allow the key to be recovered almost instantly. Because the technology was still new from the consumer point of view, this represented a tremendous risk for the technology vendors. If this were not fixed quickly, billions of dollars invested in bringing these products to market would be lost.

Because the standards committee would take years to solve the problem, the industry formed a group known as the Wi-Fi Alliance. This group came up with an approach to solving the problem that came to be called WPA: Wi- Fi Protected Access.

Since WPA was created by the industry and never ratified as a standard, you can sometimes run into issues in which some WPA systems cannot talk to others. Because this is not a standard, we strongly recommend that it not be used.

How did WPA "fix" things? In reality, WPA is essentially WEP with a few refinements. The most important change is that it introduces Temporary Key Integrity Protocol (TKIP). The primary problem with RC-4 in this context was that they key was being repeated and that, as soon as the key is repeated, the strength of the encryption is immediately compromised. TKIP then forces the key to change periodically.

The second adjustment was that the initial key is not used as-is. Instead it is hashed with the SSID. This is intended to prevent a precomputation attack because an attacker would have to know the SSID ahead of time to precompute his attack dictionary.

Wireless Encryption: WPA2

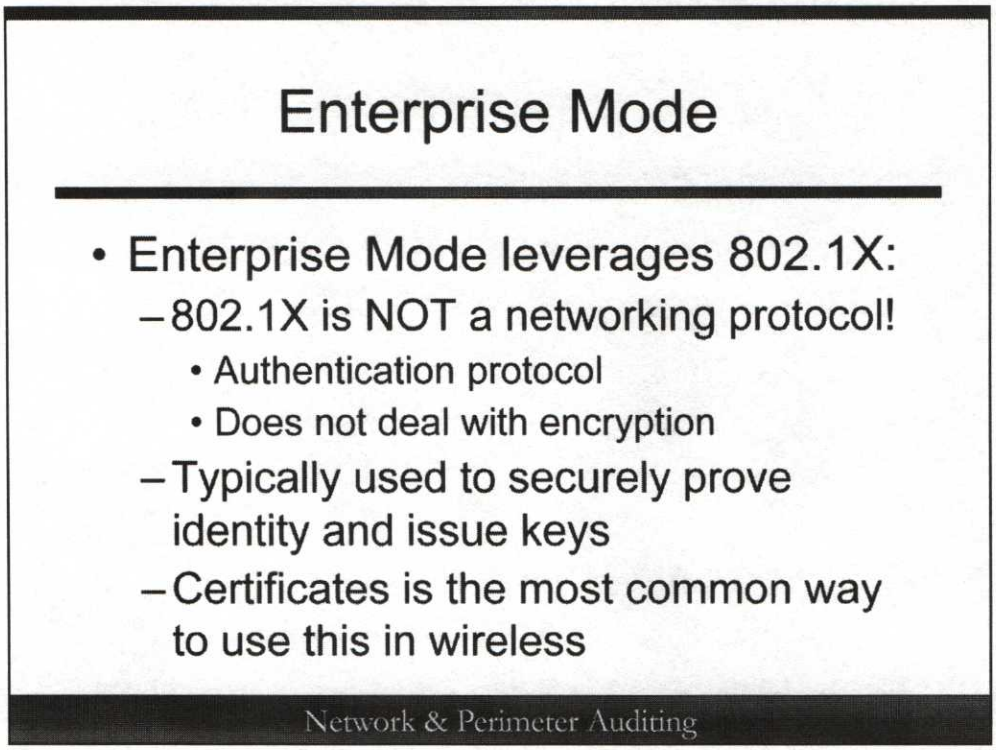
- WPA2:
  - RC-4 removed, AES is used
  - Like WPA, SSID is hashed against key to further protect security
  - PSK mode is vulnerable to attack!
    - Less than 10 minutes for most networks
  - Enterprise mode is required!

Network & Perimeter Auditing

WPA2 was ratified as a standard in 2004. It took a holistic approach to solving the security problems in Wi-Fi. Rather than just patching WEP, it completely removed RC-4 and replaced with with AES. This, it turns out, is not subject to the same sort of problems as RC-4 in this context because it is a block cipher rather than a stream cipher. In addition, like WPA, the SSID is hashed against the key to provide some protection against precomputation attacks.

WPA2 on its own is not sufficient, however! It turns out that WPA2 configured with a preshared key (PSK) is easily compromised. As a specific example, if your SSID is any of the top 1,000 and your passphrase is any of the top 1,000,000, your WPA2 can typically be broken in under 10 minutes. (For proof and a discussion of this, have a look here http://www.youtube.com/watch?v=u-dqi23oaS8.

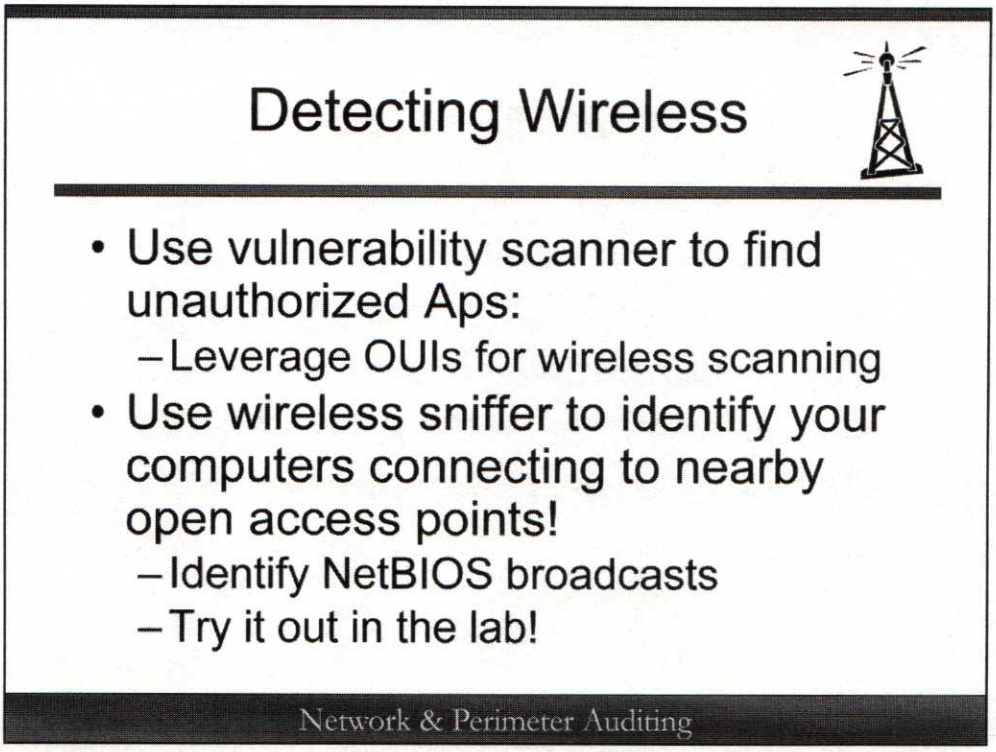
Of course, we could choose some other SSID and a more complex passphrase. This is a great idea but has limitations. One of the primary limiting factors are smart devices. Imagine trying to get your employees to key a 27-character random passphrase with special characters into smartphones!



Therefore, the best way to protect these networks today is Enterprise Mode. Enterprise Mode, typically implemented with certificates, leverages 802.1X authentication to mutually authenticate the device to the access point and the access point to the device. This also eliminates the need for the preshared key that can be attacked. Instead, the certificate-based authentication provides the mechanism for obtaining the current group key that is securing the wireless network.

It is important to realize that 802.1X has nothing to do with wireless directly. The 802 standards all define technical standards, but the numbering to the right of the dot tells you what it's for. 802.1q, for example, is a VLAN tagging protocol. 802.2 is an Ethernet encapsulation standard. 802.3 is another, and more commonly used, Ethernet encapsulation standard. 802.11 defines wireless Ethernet standards. 802.1X is an authentication protocol.

802.1X does not deal with encryption. It is only for authentication and key exchange.



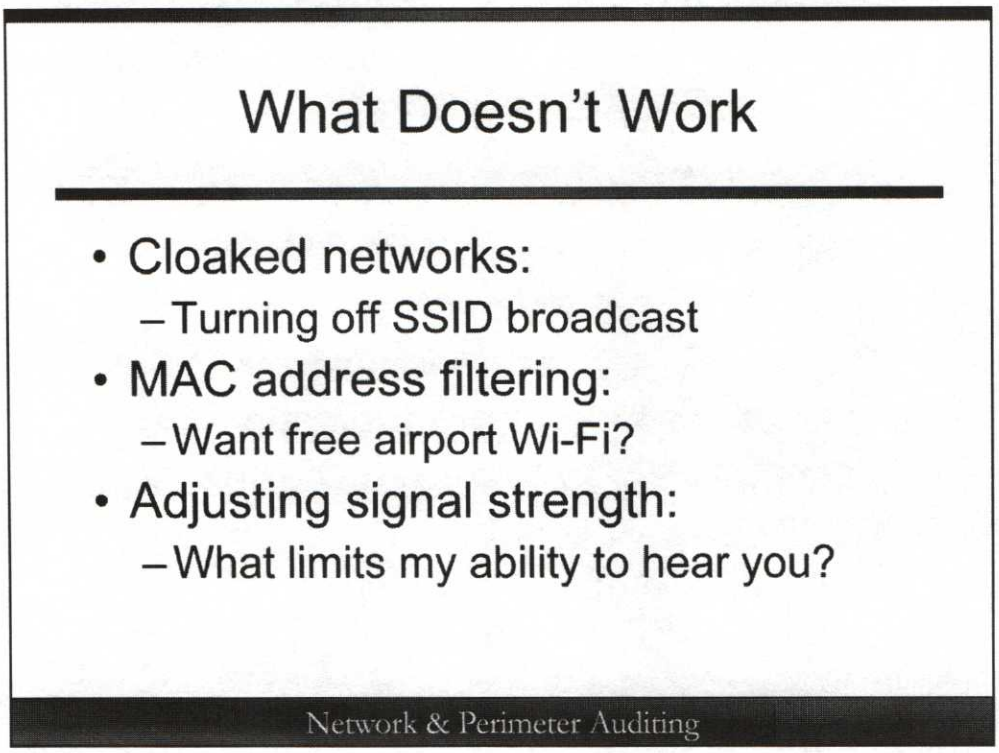
What should we be doing during an audit to identify wireless activity and verify that it is properly deployed? There are a number of activities.

First, we need to find our wireless access points. There are two ways to approach this problem. The easy way is to leverage your vulnerability scanner. All vulnerability scanners these days include signatures or plugins that identify access points. The way that these typically work is by fingerprinting and identifying the web management interface on these devices. Please note that you do not need to know the login credentials to identify these devices!

A "nerdier" way of doing this is to wander around with a wireless scanner and an antenna, trying to determine where the signal strength is highest and then poking around for an unauthorized access point. Although there is certainly some value in this, scanning the wired network tends to be more reliable. One reason is that we don't have to be concerned that the device is using 802.11n on 5 Ghz but our card supports only 802.11b/c on 2.4 Ghz. Another reason it tends to be easier to use the wired scan is that in a metropolitan or other densely populated area we will likely see a lot of access points that are not on our floor or even in our building!

This isn't to say that there is no value at all in wireless scanning. In fact, something valuable can be done. I use the vulnerability scanner to find unauthorized access points but then use wireless scanning to identify whether any of our business computers are connecting to nearby open access points. Why would someone do this? Typically to bypass Internet filtering or some other kind of monitoring.

The easiest way to find these systems is to simply walk around with a wireless sniffer collecting everything. After you do this, take the data back to your office and check to see if there is any clear text traffic containing your organization's name, domain name, or NetBIOS names from your domain. Using this data the security team can then work back to who the users were.

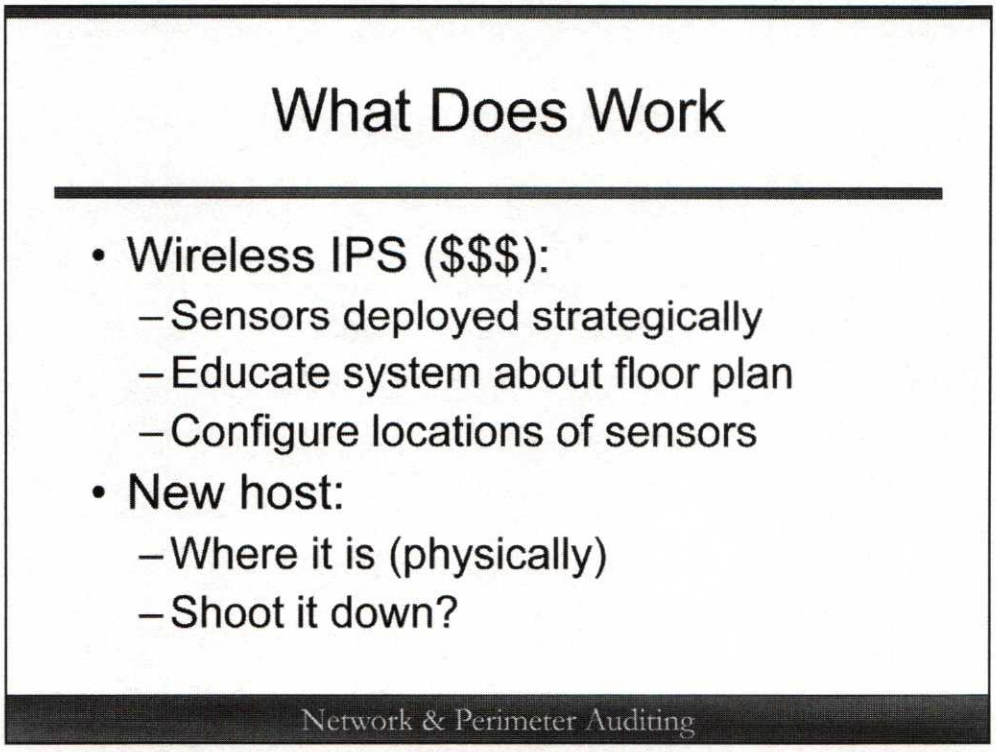


There are a variety of items that you will sometimes find in security policies or hear discussed in the context of wireless that have no benefit. Of course, if your policies require them, verify that they are done. However, it would be good to feed this information back to the policy makers if they are requiring things that have no actual security benefit.

One example of this is the idea of a Cloaked network. Wireless access points, by default, do what is called *beaconing*. This means that several times each second the access point will advertise its presence. This is what allows you to find the network when you click the wireless icon on your status bar. Many devices, however, can be figured in a cloaked mode. This means that the device will no longer send these beacon advertisements. This does not actually hide the network, though. If someone runs a network sniffer and there is any activity at all occurring on that cloaked network, the sniffer will see it. In fact, the network can be "decloaked" without doing anything active. If the person with the sniffer just waits for a host to join or reassociate with the network, the SSID will be revealed. It turns out that it is still sent in every single management frame on that network.

Another approach is to use MAC filtering. This is another approach with little value. The idea is that only authorized MAC addresses are permitted to communicate. The problem is that it is simple to reconfigure your MAC address. An attacker can simply sniff the wireless network to identify an authorized host and then reconfigure his MAC address to be the same. Connectivity will be spotty until the original user gets frustrated and turns off his computer. Now the attacker is authenticated via his MAC into the network.

The last approach that is of little value is adjusting the signal strength. If you suggest adjusting the signal strength, the administrators will want to turn it up, not down! In addition, my ability to hear your network has more to do with the size and configuration of my antenna than it does with your signal strength. One documented installation has achieved 65 miles point to point! Reference: http://www.wispa.org/dewayne-hendricks-demonstrates-65-mile-ptp-80211n-link



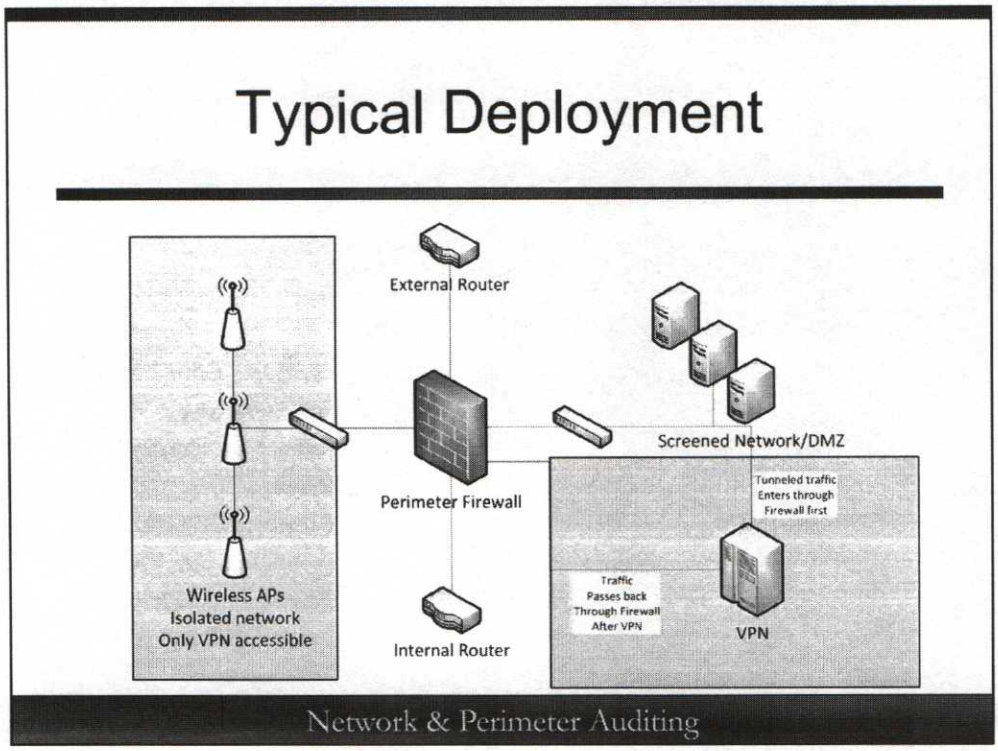
Some solutions for wireless security do work. A Wireless IPS is just such a thing.

These systems tend to be somewhat costly. Examples of these kinds of solutions are the Cisco Adaptive Wireless IPS, AirTight, AirDefense, AirMagnet, and others. These solutions will cost you at least several thousand dollars, in addition to the sensors that need to be deployed.

Typically, the way that these systems work is that you license the WIPS software and deploy a number of sensors and antennas throughout your physical premises. After these are deployed with proper overlap, a survey type tool enables you to identify where they are physically and enables you to import a diagram of your floor plan.

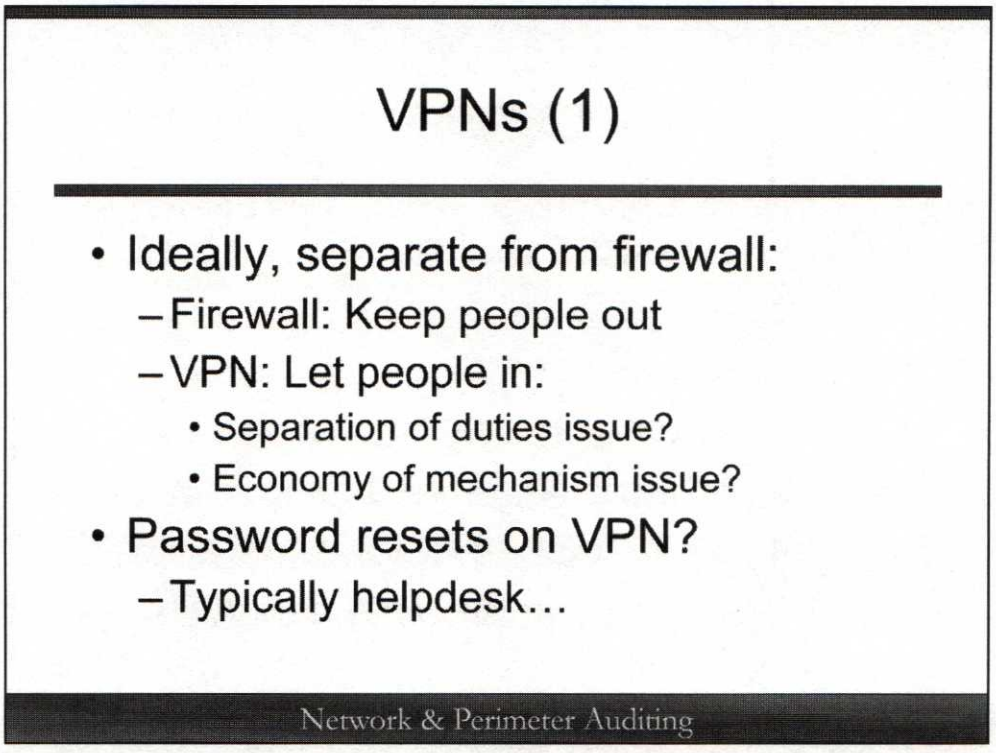
After this is all configured properly, the system can alert you to any number of things. For example, perhaps you have it configured to alert you when a new access point appears. Not only does it tell you about that device, but if you have taken the time to configure everything correctly, it pinpoints exactly where that device is within your physical building to within a few feet. This is actually incredibly valuable, especially in a densely populated area. This enables you to quickly distinguish signals that are originating from within your perimeter as opposed to signals that are outside of your perimeter.

In addition to identifying rogue hosts and access points, these systems typically have active response capabilities, enabling you to begin sending deauthentication and other types of wireless attacks at the unauthorized system, preventing it from connecting to anything.



This slide shows a deployment of a wireless system. Notice that we are not relying on the wireless encryption as our sole line of defense. Instead, any access to the Internet or to the internal network must first pass through the VPN and the firewall.

Especially given the bad history of wireless security and given that if someone is only listening to our data, not transmitting, there is no way to find that person, we recognize that we need to create more opportunities to protect and detect.



This brings us to the topic of VPNs. Many enterprises use a VPN that is included as a feature or option in the firewall platform. The primary reason that this configuration is used is cost. If the VPN is already included in the firewall, why should we buy another device?

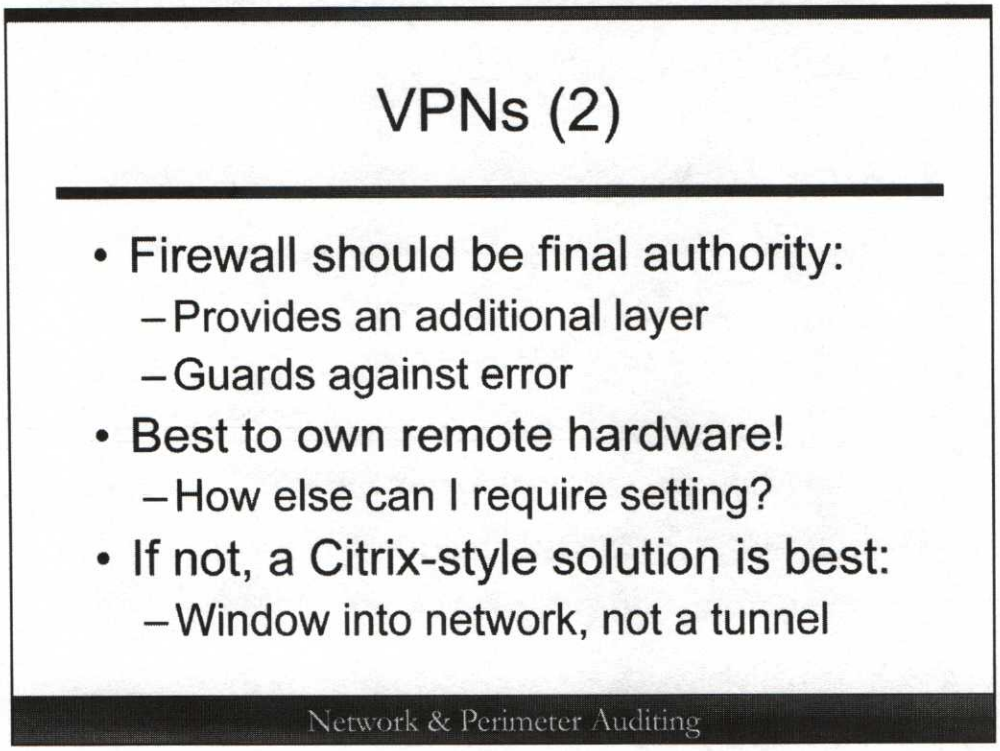
Although this is a valid argument, let's take a look at this problem from a security and risk perspective rather than a cost perspective. In the end, your enterprise may choose to use an integrated firewall/VPN solution. Still, we should understand why this might not be an ideal configuration.

First, consider the mission of the firewall. The primary task that it has is to keep bad people out. However, the mission of the VPN is to let people in. This certainly seems like a conflict of interest. The principle of Separation of Duties would seem to indicate that we should not join these services.

Another principle that comes into play is Economy of Mechanism. Firewalls are complex systems. VPNs are also complex systems in their own right. Gluing VPN technology to a firewall seems to make a *more* complex system. Because simple systems tend to be more secure, it would be preferable to keep these systems separated.

A more convincing risk-based consideration of this exists, too. Our firewall is typically managed by firewall administrators. Our first reaction would be that the same is true of the VPN. But is it actually? Unless we integrate our VPN authentication into our domain (which is a good thing!), who do VPN users call to reset passwords? I can guarantee you that it's not your firewall administrator. Those users are calling the help desk.

Here's the thing: Firewalls do not have a well-developed sense of user rights. You either are an administrator or you are not. This means that if the help desk is resetting passwords on the Firewall/VPN device, they are likely administrators on the firewall! This feels like a very risky thing to do!

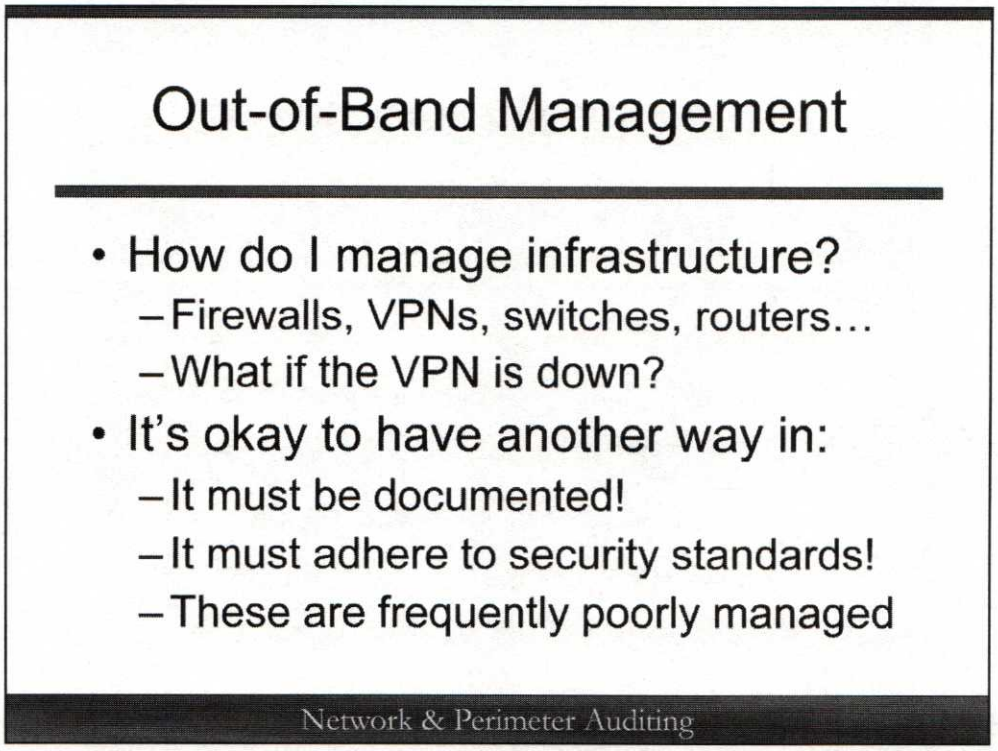


Instead, we would say that the firewall should be the final arbiter of what passes through our perimeter. This means that, even though the firewall would be protecting the VPN, the decrypted VPN data that is actually trying to enter the organization should be forced to pass through the firewall. This allows for Separation of Duties and allows a misconfigured VPN permit rule from allowing access to resources that should not be accessed from outside of the physical enterprise.

Aside from our users, VPNs also come into play when establishing business partner connections. In past years, we may have paid for a private frame relay link between us and our partner. These days, it is far cheaper to simply set up a branch office VPN tunnel. If we establish such a link, it is always best to own both ends of the connection from a hardware point of view. If we don't, how can we be sure that the settings are correct?

Exactly the same principle applies to our enterprise users! If we don't own the hardware that they connect to the enterprise with, how can we possibly require or enforce settings! Even though our VPN technologies can provide some capabilities for performing health checking of the connecting host, unless that host is locked down, it's easy to trick the VPN health check without even trying! Most of these solutions are simply using the Microsoft API to ask the system if it is patched. If the system doesn't know any better it will answer that question with a, "Yes!"

If we are not going to own the systems that our users access our network from while remote, a Citrix style VPN is likely the best way to go. We're not saying, "Go buy Citrix." Instead, it's the approach that we like. When you "connect," you actually aren't taking the data down to your computer. Instead you are viewing the data on the remote systems through your web browser using, essentially, a remote-desktop style system. Although the user could still take screen shots, the actual data passing into the enterprise can be limited to the remote desktop connection, dramatically increasing our security.



One of the other big issues to be aware of and to ask about is out-of-band management. This can be handled in several ways. Some of these have big security issues.

One out-of-band management mechanism applies most frequently to switches, routers, storage arrays, mini-computers and main frames. If we lease this equipment or have a service contract with the equipment, quite often there will be a plain old RJ-11 telephone connection plugged into a modem hooked directly up to the serial interface of the system. This is potentially an enormous security risk because these serial connections are rarely monitored and typically have no lockout capabilities. Although it is true that an attacker entering the enterprise this way would have a low-speed connection, he can typically convert that into a high-speed connection easily by simply opening an outbound connection over a permitted protocol through the firewall.

The other out-of-band mechanism is the "back door." We define a back door to be an undocumented mechanism allowing access to systems over the network. Administrators often create these systems specifically to bypass our firewall and/or VPN. The positive reason for doing this is to allow the administrator to access the system even if the firewall or VPN is down.

In reality, we don't have a problem with these types of interfaces. The problem isn't that it exists; the problem is that it isn't documented! If it isn't documented, it is rarely well maintained. Look for these things, ask about them, and make sure that they are documented and approved (or removed if they are not approved).

As we started to state on the last page, VPNs these days are not just used for our users. It is the primary way that we provide third parties and partners access to internal resources. Looking into these connections always requires that we examine the contracts to see who is responsible for what and how things are supposed to be done.

Looking at contracts is not especially fun or interesting. What we recommend is that you work to inform management, perhaps by identifying some weaknesses in existing contracts, to create a set of requirements for all connectivity related contracts. After these are identified at a high level and communicated by management to legal, our contracts overall will do a better job of accounting for security issues.

A few of these are listed in the slide. For example, who and how may the configuration of the systems and overall security be audited? Can we do it ourselves? Do we have to hire a third party? If it's not in the contract, you can't do it.

Another issue is circumstances that would allow me to terminate service. Has my partner been hacked? Are they experiencing a security event? How quickly do they have to inform me? Can I send or hire people to observe how they are dealing with it and evaluate the extent of the compromise? If we terminate service and the contract does not provide for it, we will likely become the subject of a lawsuit.

Another important item is a list, by name, of the individuals at the remote end who will have access to our data or systems. In addition, we reserve the right to refuse access to any user for any reason, such as former employees and known hackers.

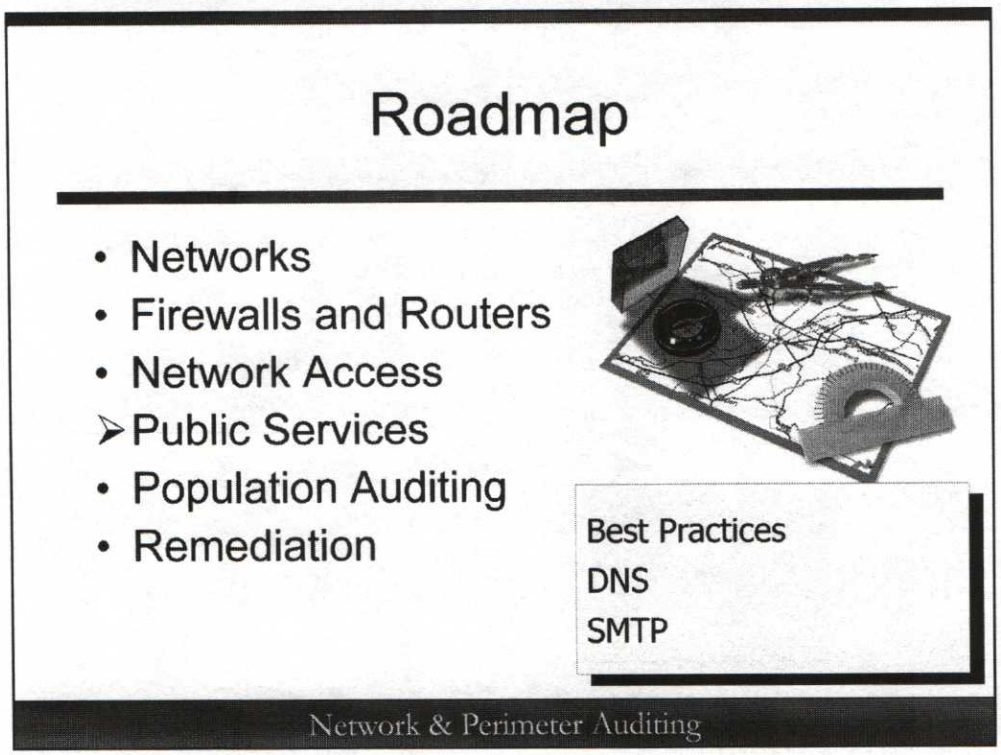
You can find an excellent resource for these types of things in the ISO-27000 standard. The sections on third-party agreements and legal/contractual responsibilities are a checklist of items to require in your contracts!

Lab 4

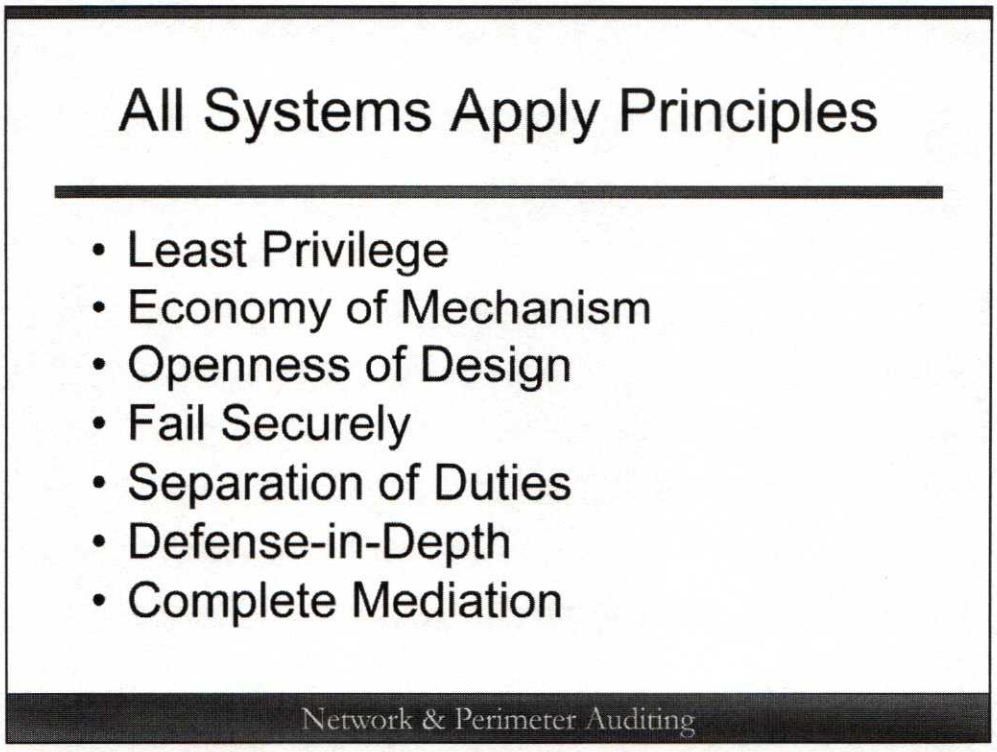
- Finding Wireless Clients

Network & Perimeter Auditing

Let's dig an and work through some exercises. In particular, let's see just how easy it is to identify whether any of our computers connect to someone else's open access points!



Again, as we move into our next section, all the issues dealing with administration and general configuration must be applied. Some of these issues are particular to certain operating systems, so we actually dig into some of those issues on Days 4 and 5 when we cover operating system auditing. For now, though, let's talk about some common public-facing technologies that we all probably have.



Although the discussion of these fundamental security principles appears here in our course, don't think that they apply only to publicly exposed services. Actually, if you were paying attention, you have heard us mention several of these already!

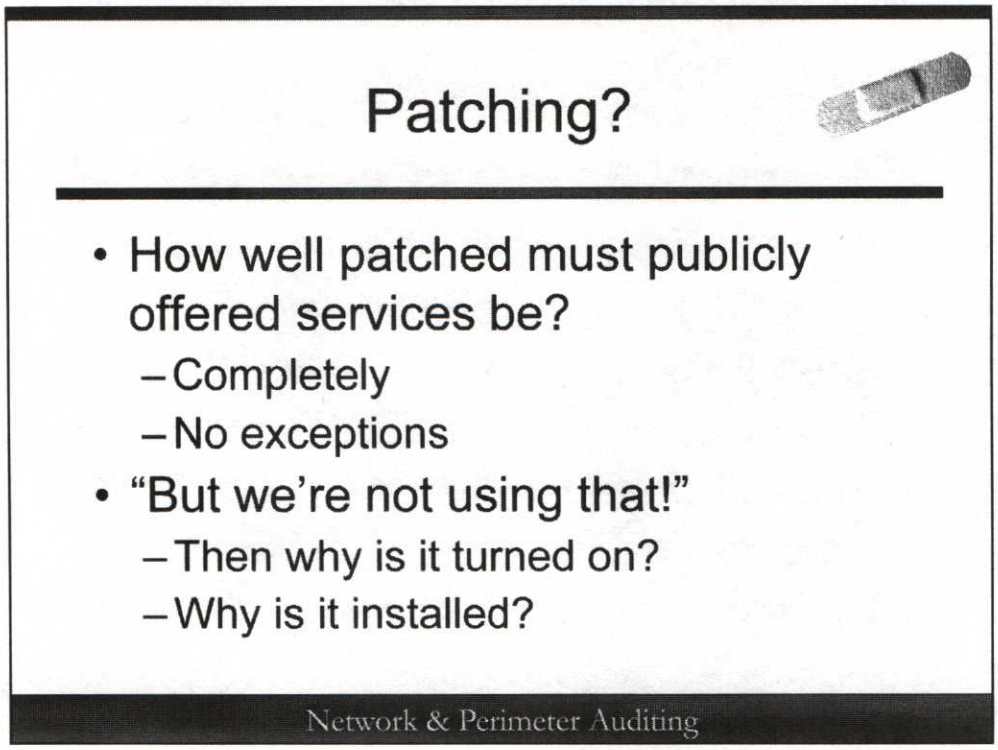
What we want to point out to you is that any system within our environment, publicly exposed or not, should be based on sound security principles. For example, if we have a service running on any host, that service should be running only with the rights necessary to accomplish its purpose. Imagine installing an IBM Notes/Domino server on a Windows host. By default this service will run as LocalSystem. If you read the documentation it does tell you that you should create a service account for this service and give it full access to the Notes data directory. Making that change is assigning minimal privileges to that service.

We have also discussed the idea of economy of mechanism; the notion that simple systems tend to be more secure. Many people know this principle by the maxim, "Keep it Simple, Stupid."

Another important principle comes from the cryptography world. Kerckhoff's principle, posited in the 1800s, is that the security of a system cannot derive from its secrecy. Instead, all components should be fully examinable. This is the principle of openness of design.

We'd also like to find that our systems fail in a secure way. Whether this means that they fail closed or fail open depends on the actual implementation. For instance, our physical doors tend to fail open during a fire, but we might prefer that our firewall fail closed to stop an ongoing attack.

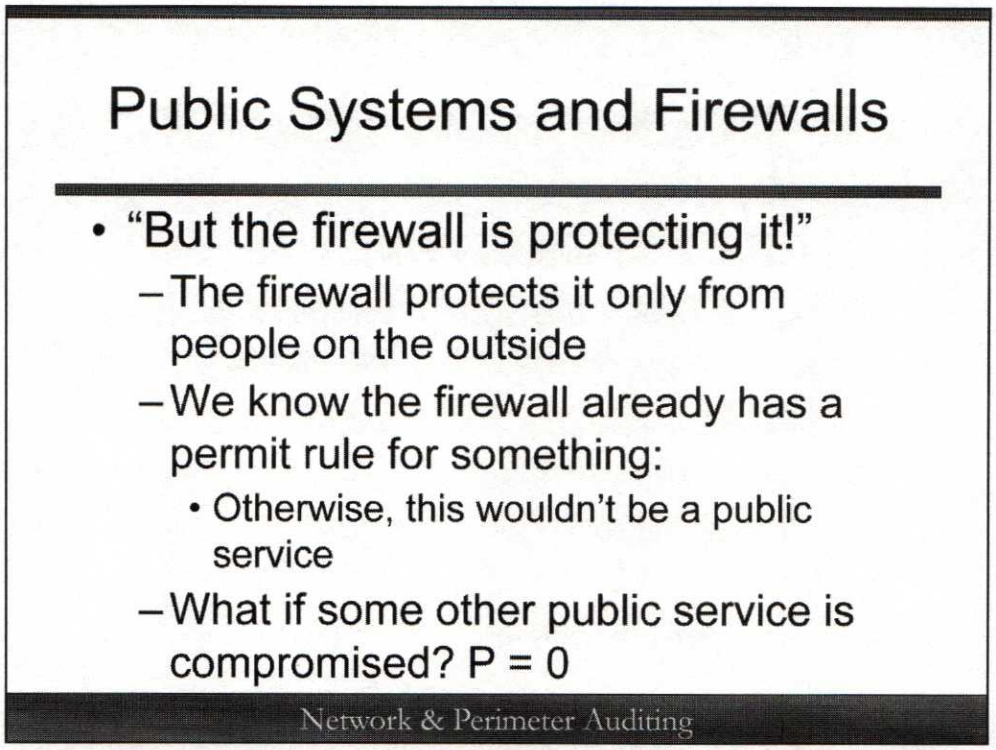
Separation of Duties and Defense-in-Depth are either well understood or have already been covered. Complete mediation, however, may be unfamiliar. This principle requires that all access is forced through a single set of common security checks or a single path.



It goes without saying that although all systems must be patched, a publicly accessible or externally accessible system must be patched even more so, or even more quickly. Patching is the primary way that we eliminate vulnerabilities in our systems.

When dealing with patch management, you might face some challenges. At times you will identify a service or system that is not fully patched; yet when you point it out to the administrator, he will say something like, "Yes, but, we're not actually using that." Another response might be, "It doesn't matter that we haven't patched that because you can't get to that from the Internet."

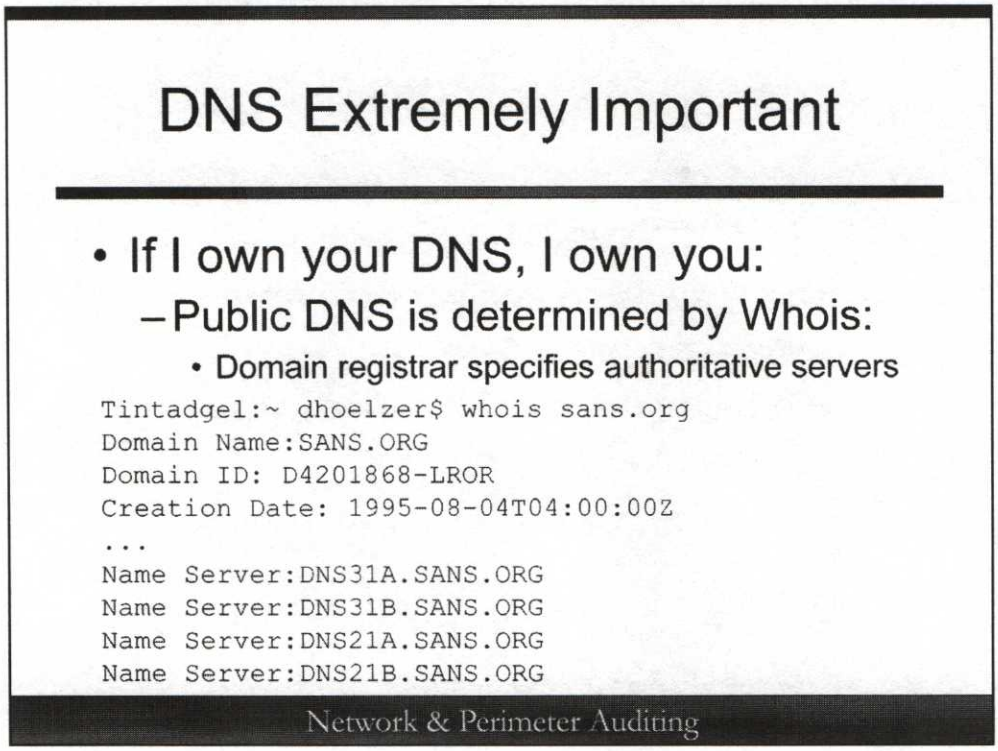
Think about these two responses. Right now, our context is publicly accessible systems. If an administrator is claiming that the service isn't actually used, what is he telling you? He's saying, "I have failed to remove the unnecessary services from my system." Economy of mechanism and least privilege both insist that these unneeded services be disabled or, better, removed! When discussing this with the administrator or with the organization in general, you can absolutely point to organizational standards and requirements. However, if these do not speak to the issues you are finding, always try to relate these issues in terms of security principles. It's not about the specific thing you're finding, it's about the principles that are involved.



If they claim that because it can't be reached over the Internet it doesn't matter, there's something else that they are missing. Recall our discussion of time-based analysis. We pointed out that we have to assume that our security fails. In fact, if we have a publicly exposed service, we don't have to imagine this. If we have a mail server listening on port 25, what is the value of P, or protection, on port 25 through the firewall? Zero! The firewall has a permit rule!

If someone can compromise any one of these systems through a service that is permitted, what prevents that attacker from going after the rest of the ports that no one bothered to patch now that he is behind the firewall?

Again, a principled-based approach to this problem enables us to see what can go wrong. In addition, this would be an excellent place to perform a formal risk assessment if the organization fails to respond to your findings. Clearly, the risk here is high!



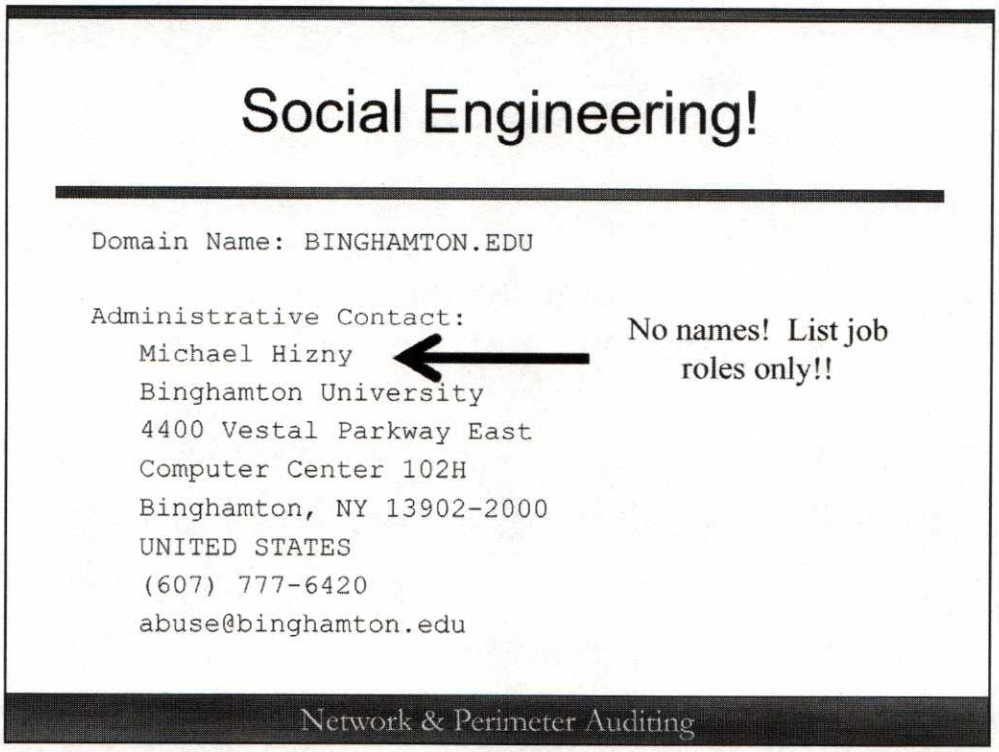
One publicly accessible system that everyone has is a Domain Naming System (DNS), which is essentially the directory service for the Internet.

When a user wants to browse to Google, he does not open his browser and type, "http://74.125.226.7/." The user tells his browser, "I want to go to google.com." Behind the scenes, the computer asks whichever DNS server has been configured. If that DNS server doesn't know the answer it will "recurse." What this means is that it goes and asks other DNS servers until it determines that the domain doesn't exist, determines that the host doesn't exist, or discovers the actual address. In many ways, the DNS service is like a phone book. We know names; the computers know where the numbers are.

The authoritative DNS servers for any domain are maintained in the Whois system. Whois, which operates over port 43, can be queried to determine where the authoritative servers are for any domain. In the slide, for example, we use whois from a UNIX command line and it tells us that SANS has four name servers configured to provide authoritative data.

For a moment, consider how critical the DNS system is to your organization. If someone can redirect name resolution for your domain to his own server, he essentially "owns" your domain. He doesn't have to get your passwords, he doesn't need to deface your website, and so on. Any time anyone tries to find you, he completely controls where the data goes. Whenever someone sends you an e-mail, he controls where that e-mail will be delivered.

Clearly, we need to make sure our DNS servers are locked down, the records are correct, and we have good procedures for keeping things safe.



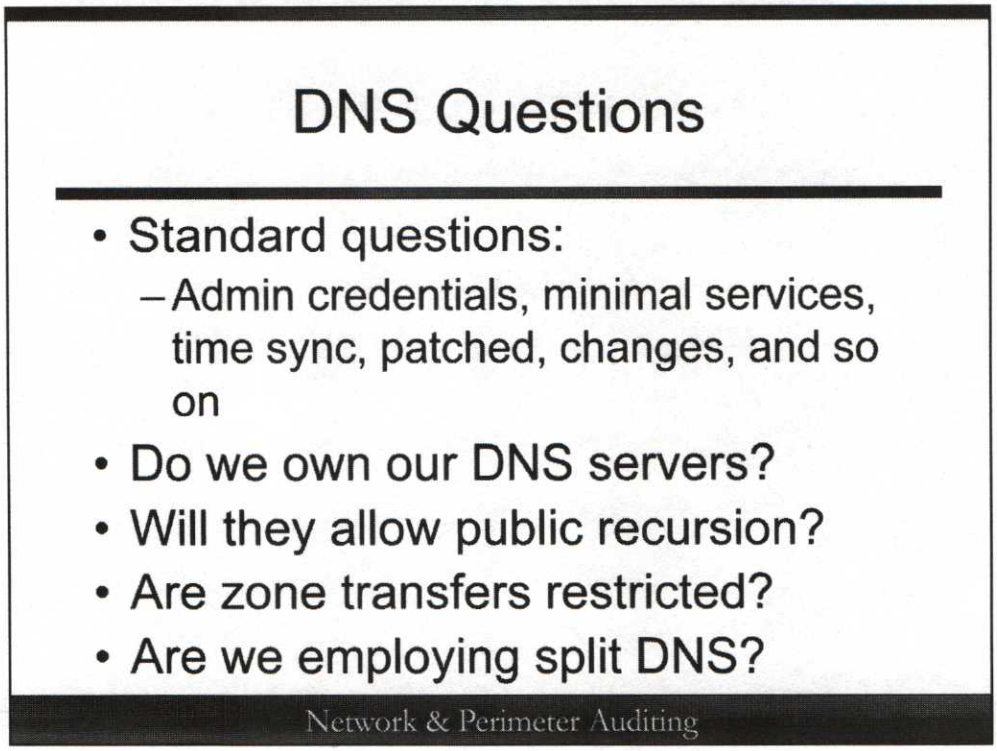
There are some important items to look for when examining DNS configurations. First, use whois to pull information on your domain. Can you see actual names of individuals listed in the registration information? As a rule of thumb, this is bad.

Looking at the example in the slide, would you agree that Michael Hizny is likely an individual with some significant level of authority? Could an attacker use this information to make some phone calls and possibly socially engineer someone into doing something that he shouldn't? We might even use this information to attack Michael, claiming to be the registrar calling to validate information with him or informing him of a new administration interface that he needs to use from now on!

There is another significant risk. If we have an individual's name here, what happens when that individual leaves the organization? Although that person no longer has access to his internal e-mail account, all that he needs to do is contact the registrar. When he does he will be asked to provide a photocopy of a government issued ID and a request on organizational letterhead asking for the address to be changed. Of course, it would be trivial for Michael to provide the ID. What about the letterhead? Do you think the registrar has a database to tell them whether the letterhead that this former employee creates is real?

Yet another thing to check on is the expiration date for the domain. We don't have this pictured in the slide, but this data is also displayed when you perform a whois. If the expiration date is anywhere in the next year, it is always good to ask who is responsible for renewing it. Often organizations pay registration fees for multiple years. When the registration comes due, it is easy to forget about it.

That leads to a final issue related to this slide. We'd like to find that there are general role-based e-mail addresses listed and that these addresses are *actually reachable!* In other words, the e-mail will actually show up in someone's inbox. Otherwise, any notifications about or domain will be overlooked or lost.



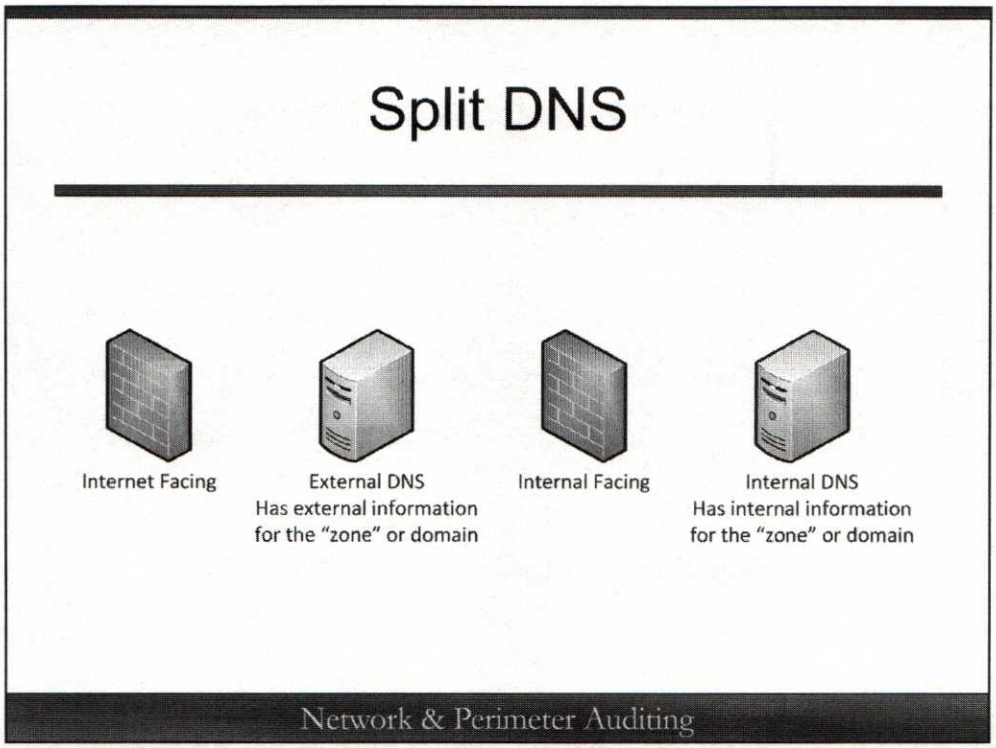
The standard questions about how the system is administered, patched, logging, and so on all apply. For DNS, though, there are some additional issues to dig into.

One of these is that we would prefer to find that we own and manage all our DNS servers. If we allow an ISP or someone else to manage our DNS entries, we are at their mercy for making emergency changes. In addition, we have no control over how well patched or otherwise well configured these servers are. Because DNS is so absolutely vital to the overall security of our domain, we should not be outsourcing the management of these services.

As for the configuration of the services, there are other important questions. For example, our servers should not be configured to allow recursion for addresses outside of our network. Remember that recursion means that the DNS server will try to find a DNS server who is authoritative to answer a particular request. If we permit recursion for outside queries, we create a configuration that is vulnerable to DNS cache poisoning. This would allow an attacker to inject invalid records for third parties into our DNS service.

We must also ensure that zone transfers are limited only to authorized hosts. Within DNS, a set of related hosts are known as a "zone." For example, enclaveforensics.com is a zone. Another zone might be the subdomain, windows.enclaveforensics.com. These related records are all stored in the same configuration file. DNS peers exchange this information through a zone transfer. If we do not secure this, anyone can ask the DNS server for a complete list of all hosts and addresses, instantly obtaining a network map without doing any scanning at all.

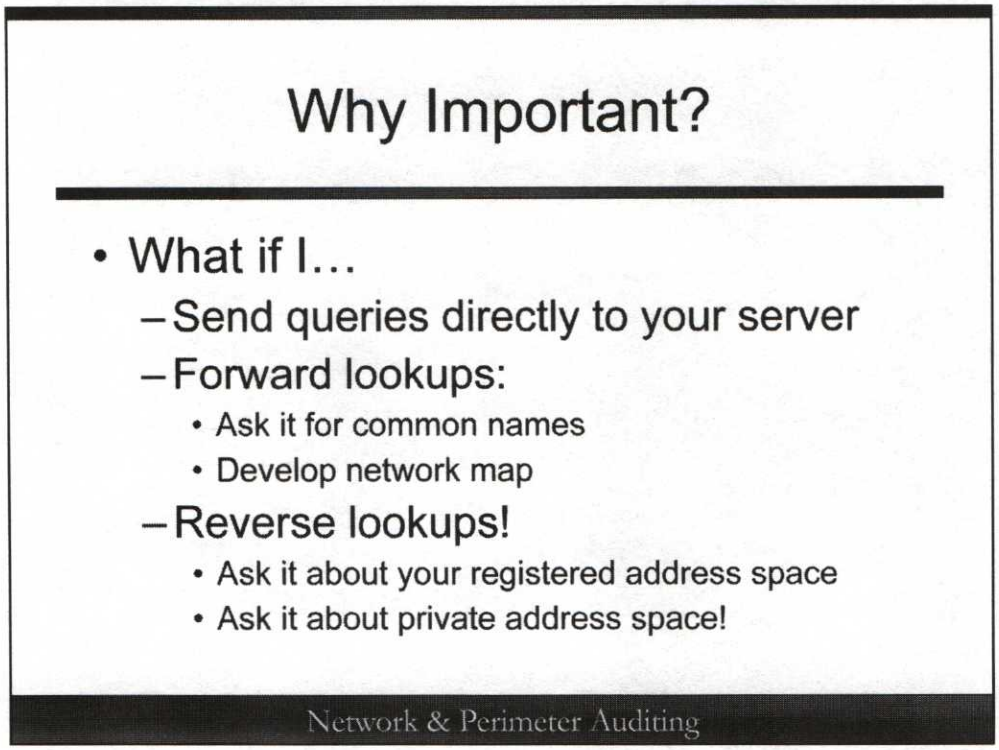
We would also like to find that we are employing a Split-DNS arrangement.



Split-DNS means that we are splitting the knowledge of the information within the zone.

In this slide, you can see this illustrated. The internal DNS server has all the information about internal hosts. It may also have information about external hosts in the zone, or it can be configured to forward requests to the external DNS server.

The external DNS servers, however, only have information about public-facing systems. The wonderful advantage is that even if these servers are compromised, it is not possible for them to reveal internal addressing details because they just don't know how the internal network is configured. As we said at the outset, we have split the knowledge of the zone.

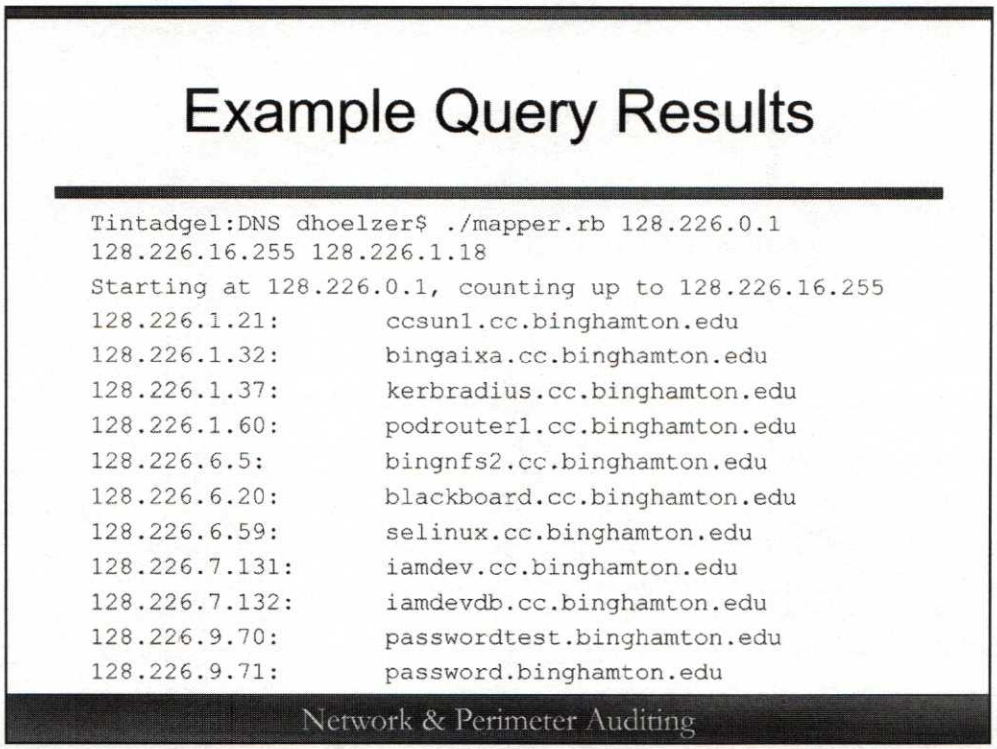


Why does this matter? Why would you want your servers configured this way? Consider the following.

What if I configured my system to send DNS queries directly to your public-facing DNS service rather than using the DNS server local to me. What if I created a list of commonly used hostnames and asked your DNS server about every one of those names? It's true that the majority of those hosts wouldn't exist; if your DNS server has information for one of these names, it will let me know! I can now create a network map of important hosts on your network without tripping any alarms!

We can do the same sort of thing with reverse queries. A reverse query is like using a reverse phone book. I have your phone number and I want to know what your name is. A reverse directory allows me to look up your number and find your name.

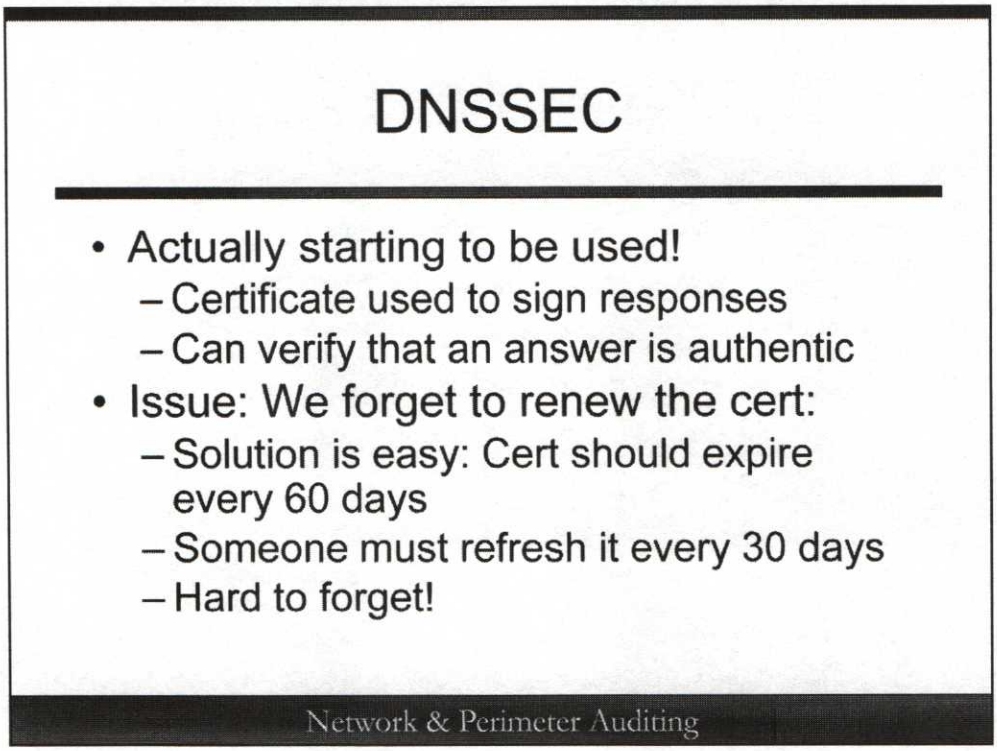
In a similar way, a reverse query says to the DNS server, "Hey, I think there's a host at X.X.X.X. Can you tell me its name?" If the reverse lookup zone is configured, the server happily returns the data. What's even better about this is that we tend to create reverse lookups only for servers that are important. This network map is even more useful! It might even be useful for discovering internal address space!



Here you can see an example of mapping out address space using reverse queries. Because we are performing only normal DNS queries, we aren't doing anything improper, unethical, illegal, or otherwise "bad." You can see how valuable this data is, though. The actual test returned many hundreds of results. I have extracted a few that tell us about interesting hosts. For example, would it surprise you if ccsun1 is some sort of Oracle system running Solaris? How about bingaixa? An AIX host? podrouter1 is likely a router somewhere.

We also see an NFS server, a blackboard (which is an online collaboration tool that has had vulnerabilities over the years), SELinux, a couple of Dev machines, and something named "password" and "passwordtest." As an attacker, all these would attract my attention.

The tool used here is available for free from auditcasts.com or from Github (https://github.com/dhoelzer/AuditcastsScripts). Another tool that can perform the forward lookups using a brute-force list is Metasploit. I have had mixed results with this particular Metasploit plugin, which is what lead me to write the script shown in the slide.



DNSSEC has been talked about since 1999 in RFC-2538, which proposed a method of storing digital certificates in DNS for use in validating records. Even though everyone seems to recognize that securing DNS and DNS records is critically important, more than 15 years later we are still struggling to get DNSSEC deployed.

As of this writing, we are actually starting to see DNSSEC used. The U.S. government, supposedly, has DNSSEC fully deployed. Google has signed DNS records available. Several other "early adopters" are supporting DNSSEC as well. Support for it is available in all the major DNS platforms that are used as DNS services today.

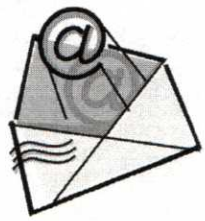
The idea is that a DNS server can know whether data presented is actually valid by examining the digital signature included with the answer. If the signature is not correct, the server knows to reject the data.

If you do choose to implement DNSSEC (and it is a good idea to implement it!) there is a potential process issue. Some who have implemented DNSSEC have experienced an issue in which the administrators forget to renew the certificate. If we create a certificate that is good for, say, 5 years, someone has to remember in 5 years to renew it. If we don't, we will not immediately realize that something is wrong. We can still get to the Internet. We can still send e-mail. However, no one who uses DNSSEC can e-mail us, get to our website, or anything else! This can make it difficult for them to let us know that something is wrong!

The right way to fix this is counterintuitive. Do not set the certificate to be good for 100 years. Instead, create certificates that are valid for 60 days. An administrator will now have a 30-day repeating task to renew the certificate. If he forgets or is out of the office for a few days, there is still time to fix it. By making this a more frequent task, it is far less likely that it will be forgotten.

SMTP

- Typically, our internal mail infrastructure is insulated from the Internet:
  - Exim, Postfix, Sendmail and more
  - Acts as a "Smarthost"
  - Relays all inbound mail in
  - Sends outbound mail out

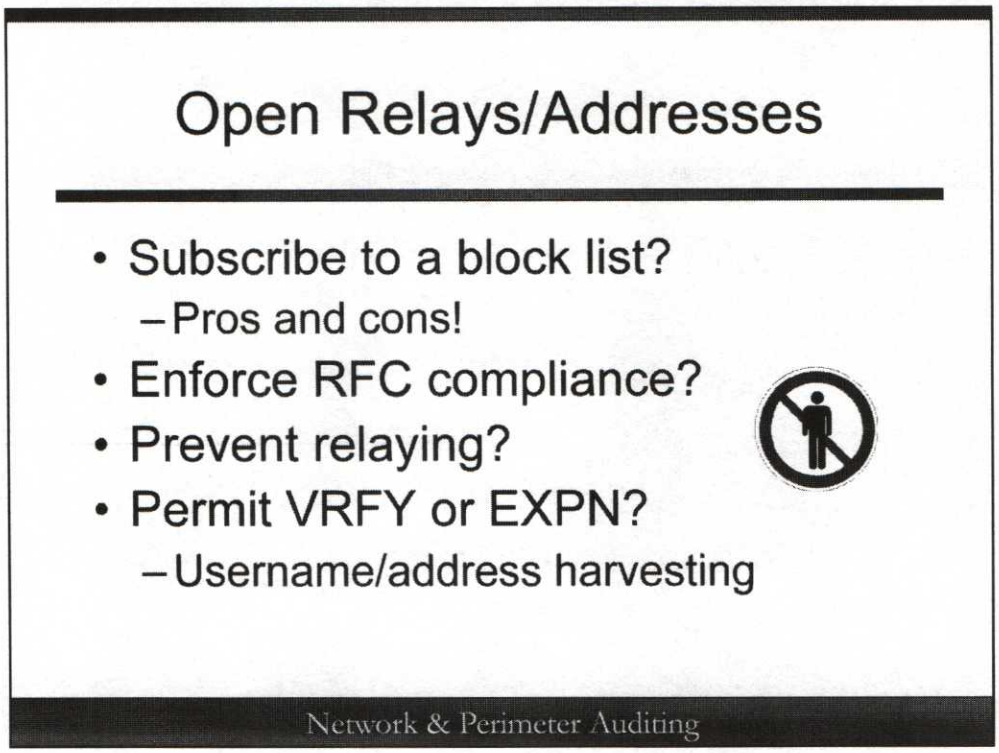


Network & Perimeter Auditing

Another absolutely vital public-facing service that we have is Simple Mail Transport Protocol (SMTP), which is a plaintext protocol used to relay e-mail messages from one system to another.

Internally, you might use Exchange or something similar, but most organizations prefer not to connect Domain-connected Windows systems directly to the Internet. For this reason, it is extremely common to find that we are running some form of UNIX mail gateway that relays e-mail in and out of our enterprise. This same gateway is used as a SMARTHOST to relay all outbound email. A Smarthost is a host that can figure out how to deliver any message anywhere.

Insulating our internal mail server from the Internet is actually a good idea. However, both the public-facing and internal mail servers still need to be configured securely.



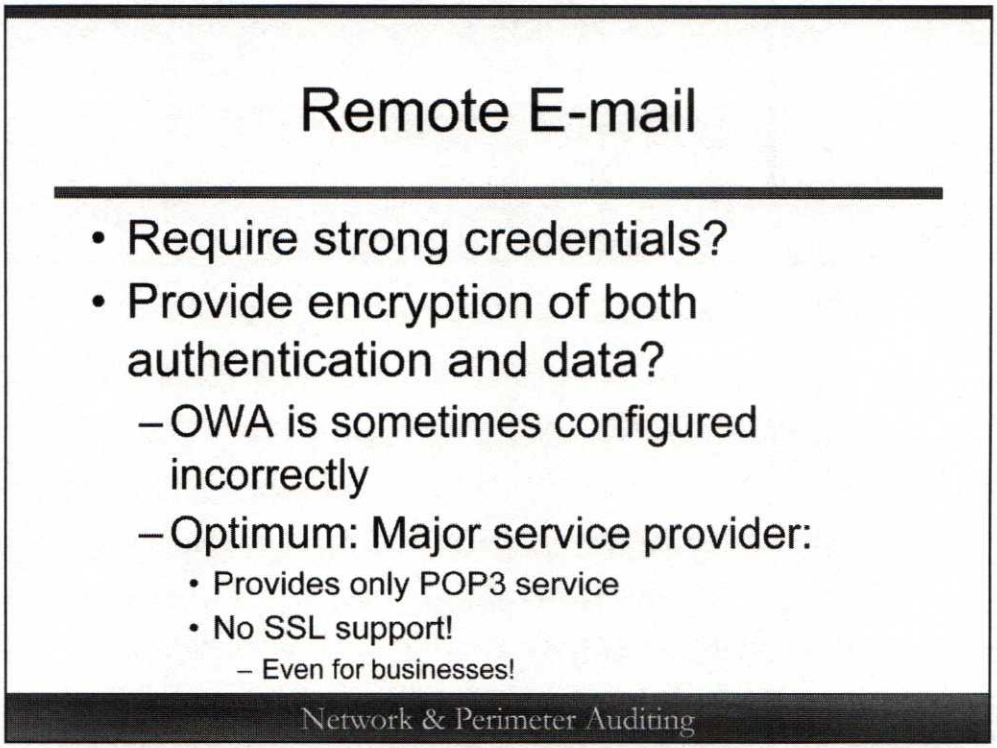
Here are some things that should be verified in the configuration of our mail servers:

Are we using a block list? If so, it can help us to dramatically reduce the amount of junk mail that we receive. A block list is a publicly maintained list of hosts known to be open relays or known to be a source of spam or Unsolicited Commercial Email (UCE). Subscribing to a block list can have ramifications, though. If we have a customer or partner who ends up on one of these lists through no fault of his own, we will reject his e-mail with an unfriendly message, accusing him of being a spammer!

Another item is whether we will enforce RFC compliance. In other words, will we require that servers that speak to us do so using the formalized standard for SMTP and ESMTP. Although this is a great feature to turn on and and it can decrease the amount of junk mail, you may find that some of your partner e-mail servers are not as well configured as yours and you may begin rejecting e-mail.

A key configuration that must be verified is that our mail server will only relay mail to us or from us. It must never permit people on the Internet to relay mail to other people on the Internet. If we do let this happen, we will be popular with the Phishing and Spamming crowd.

We also want to ensure that any extra options that are not actually in use or required are disabled. Two examples are the VRFY and EXPN commands. These are a part of the SMTP standard, but they are never actually used between mail gateways. The VRFY command allows someone to verify whether an e-mail address is valid. The EXPN command enables you to expand an address to the full name of the user. Both of these are used by individuals who are building "validated e-mail lists" that are then sold to spammers and phishers. Again, neither of these is required for our e-mail system to function properly.

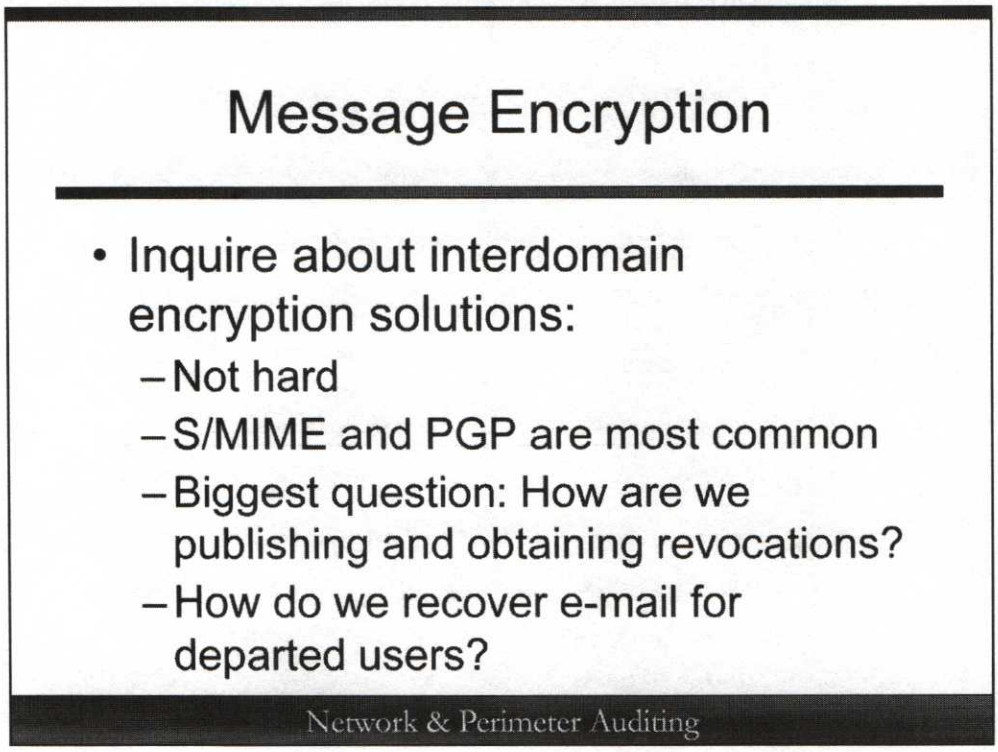


A configuration detail that is sometimes overlooked is remote e-mail access. Of course, any remote access to e-mail must require a username and password. However, it is important to know that most e-mail protocols do not support encryption out-of-the-box. Instead, someone has to take the time to properly configure it.

For example, Outlook Web Access (OWA) is sometimes used. Although this is a nice service, I have seen a number of deployments where the authentication is handled over SSL, but the content of the e-mail is all sent in clear text between the client browser and the server. Certainly, it is true that SMTP is generally not encrypted over the Internet, but there's definitely a perceived difference between it being unencrypted between servers on the Internet and unencrypted as I read it sitting in Starbucks! Make sure the entire system is encrypted.

Similarly, even large service providers get this wrong. A major service provider in the North East of the United States services tens of millions of customers and businesses. It provides e-mail services for home users and many business customers. Unfortunately, it does not support *any* method for securely retrieving your e-mail into Outlook or a similar mail client.

What we're saying is that not only is the e-mail unencrypted, the credentials used to access the e-mail account are unencrypted! Truly horrible but not unique. Make sure your systems have a secure configuration for all remote access.



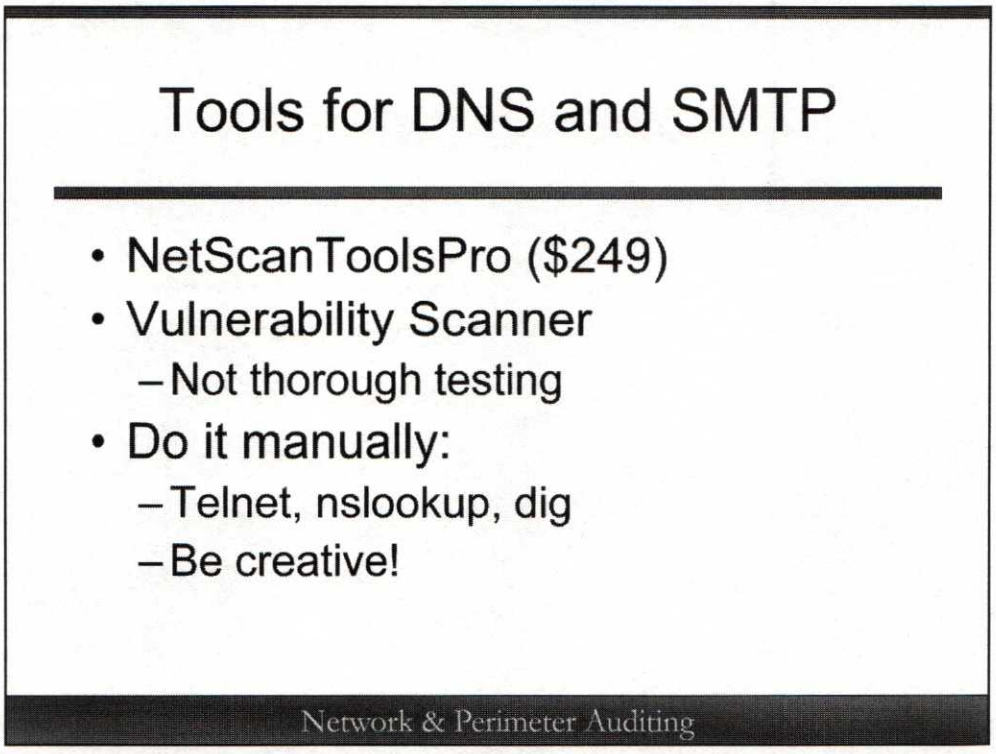
Before we move on from mail servers, there's another issue to point out. We would like to inquire about inter-organizational encryption. In other words, when we need to exchange secure messages with a third party, how are we doing that?

Encryption between organizations is not hard. The two most common solutions for this are S/MIME and PGP; though there are other providers with radically different approaches involving drop boxes and possibly even two-factor authentication. The advantage of S/MIME is that it's pretty much built in to everything these days. All I have to do is import a certificate for my e-mail address and I'm ready to go. PGP is a bit more work. We definitely have to install either a free or commercial client and get it integrated with our mail client.

For both of these, though, we need a way of obtaining the keys or certificates of the people with whom we will communicate. We also need our data published. Do we have a key server for this purpose? Are we using free Internet solutions to publish this data? How do we validate that a certificate sent to us by a third party is actually its certificate and not a forged certificate?

Another issue is, what happens when a user is terminated? How do we publish terminations? Is the CRL accessible? Is it reasonable that a remote client will even consult our revocation list? This is actually one of the fundamental problems with PKI when we do not share a common infrastructure.

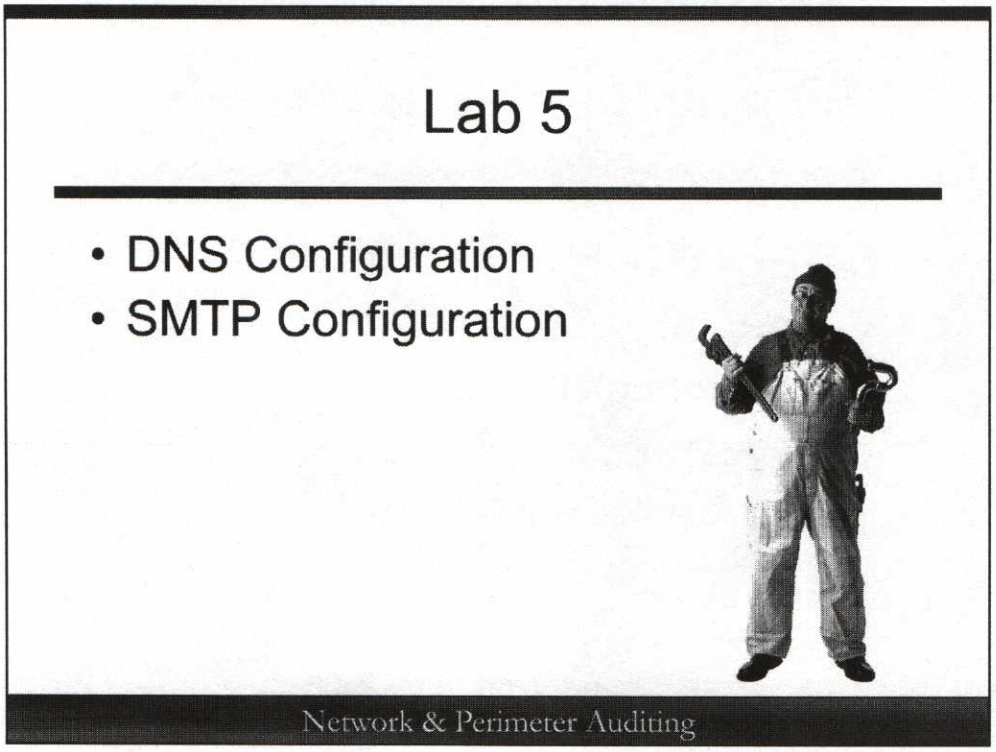
We would also like to inquire about access to the e-mail of departed users when encryption has been used. Are we issuing the certificates or keys? Do we maintain an escrowed copy of that key so that we can read the e-mail in the mailbox of a departed user? The time to find out the answer is before you need access.



There are a variety of ways to go about the technical testing of these and other network services that we might offer. One of my favorite tools for this purpose is a tool named NetScanTools Pro. This tool, available from http://www.netscantools.com/nstpromain.html, runs approximately $249 per year. It enables you to test not only SMTP and DNS for common configuration issues, but it also supports nearly any network protocol that you can name. Extremely handy!

Another approach is using a vulnerability scanner. Although these tools are fine, the trouble here is that the coverage may not be complete. These tools are certainly looking for configuration errors, but they are even more focused on finding vulnerabilities such as buffer overflows. If we are going to use a vulnerability scanner for this testing, we would want to carefully examine which tests it does. For example, if we're performing an SMTP assessment, does the tool actually try to relay mail? When it tries that, is it using internal addresses, which might actually be permitted by default?

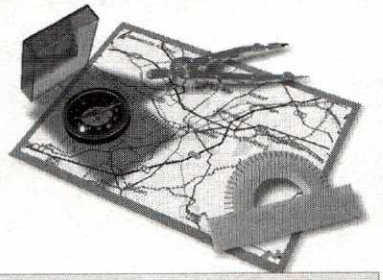
The other way to approach this is to test the services out manually! We'll take a look and teach you how to do some of this in our next lab. Be creative! Because you are familiar with your organization and how it functions, you can often come up with creative ways to compromise data that the automatic tools can't.



It's time to test out some DNS and SMTP configurations! Please open your workbook and follow along with the instructor as he works through and discusses the lab!

Roadmap

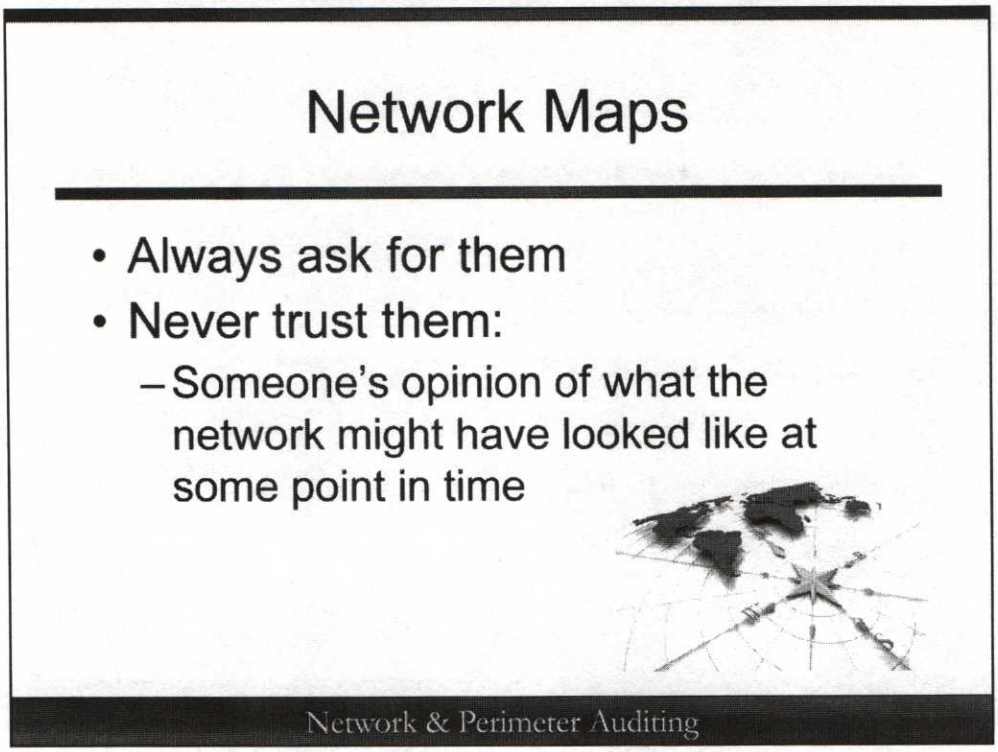
- Networks
- Firewalls and Routers
- Network Access
- Public Services
- ➢ Population Auditing
- Remediation



Discovery
Data Management
Using the Data

Network & Perimeter Auditing

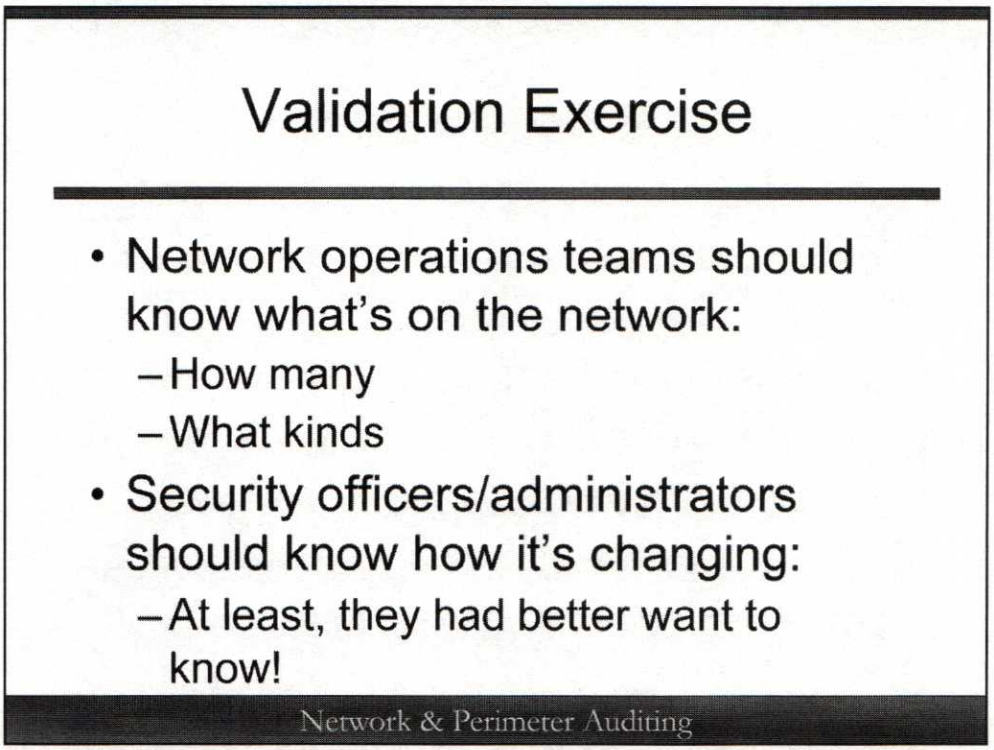
In the next section, we take a wider view of the network. Rather than looking at a specific device, we instead want to gather information about the network population. Let's see how we can effectively determine the network population, monitor it for change over time, and effectively feed back to the organization.



Every network engineering team maintains network maps. We always want to ask to see a copy of the network map, but we should never rely on it for any important analysis.

In a real sense, a network map is just one person's opinion of what that network might have looked like at some point in time. In addition, when a network engineer creates a network diagram, he is typically trying to illustrate some specific aspect of that network topology. For example, at a high level he might be trying to illustrate the interconnections between the various subnets. If he is doing so, you will likely see routers, firewalls, and VPNs. There may be pockets of the diagram that define certain servers and other important systems, but likely there will be no mention of switches anywhere in the diagram.

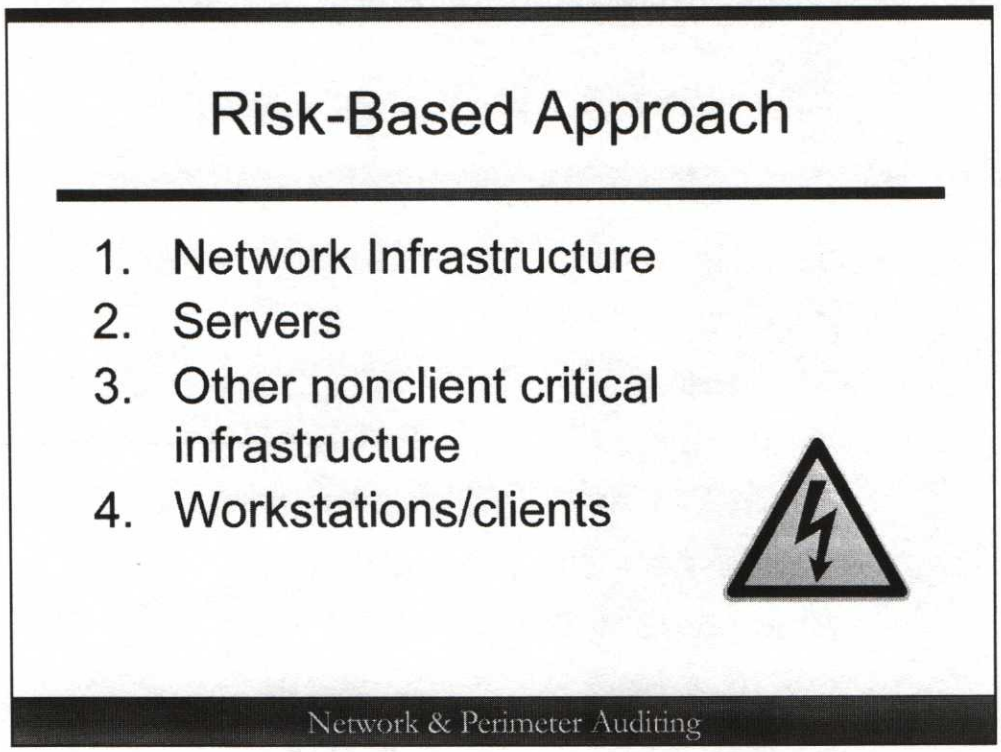
However, if the administrator is trying to illustrate how the subnets interrelate through the VLANs, we will see a lot of switches but not much in the way of routers. It's all a matter of perspective.



Yet, it's true that a network administrator or network engineer should know what's on his network. He should tell you how many and which subnets are populated, what sorts of systems you might find on any particular network, and how many systems are on the network overall. Unfortunately, his knowledge in this area is usually based on an accumulation of experience rather than any documented evidence.

Security officers and administrators should also be interested and even asking for this type of network information. They should want to know when new systems are appearing, when hosts disappear, and when hosts change. Of particular interest is information on when a host begins offering a new service that was not previously available.

In this section, we describe the steps and tools that we can use to create such documented data. We also highlight why other people will care about the data so that we can work to introduce these concepts to them. Quite often, we may not initiate these sorts of programs. If we can convince network and security administrators of the value, however, they can make things happen quite easily!

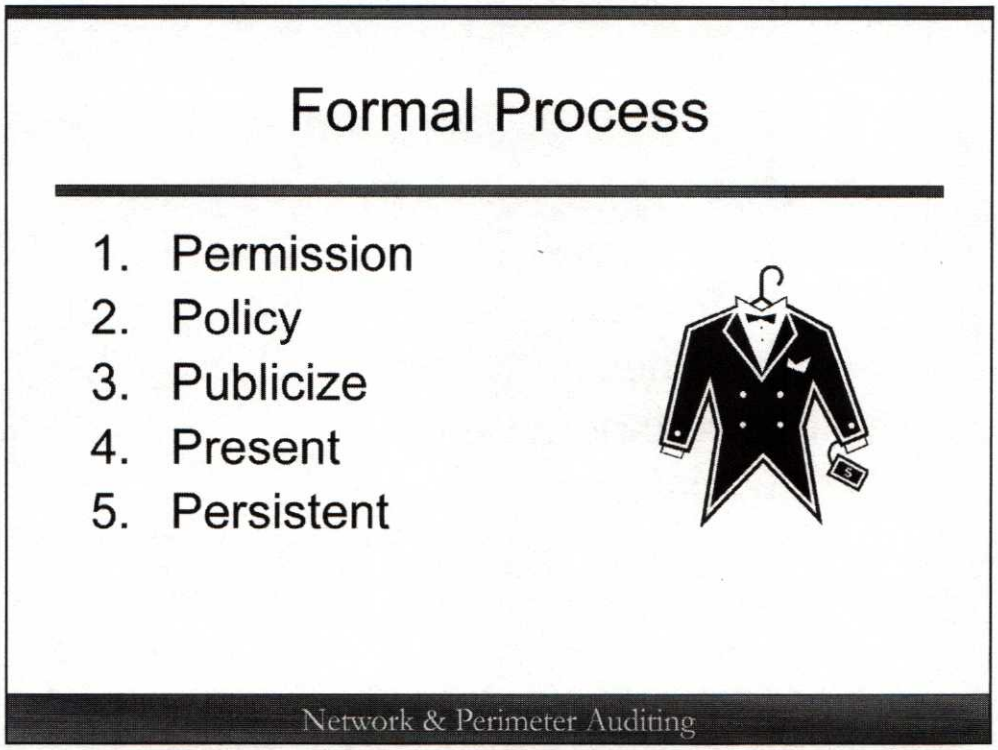


When approaching this problem, whether we do so as an auditor or as an administrator of some kind, we'd advise that you take a risk-based approach to the problem. When we think about which systems have the most critical impact on our ability to operate, we usually see that the network infrastructure and servers are the first items on our list.

This may be counterintuitive. If you're thinking of starting to do network scanning, you may be tempted to start with client systems. The reasoning is that this will likely cause the least harm.

Although this is certainly a valid point of view, it is likely also true that these systems tend to provide you the least value. In addition, there are far more systems when we think of client systems. It will likely take us a long time to get to a point in which we can reliably scan our client systems without any side effects. This, in turn, means that it will take a long time for us to get around to scanning our critical infrastructure.

Instead, starting with the servers, routers, and switches allows us to start with the most critical systems. It also means that we will be dealing with a much smaller number of systems. This allows us to move to the later phases quicker.



It is also important to have an overall process to follow. We recommend that you follow these five steps.

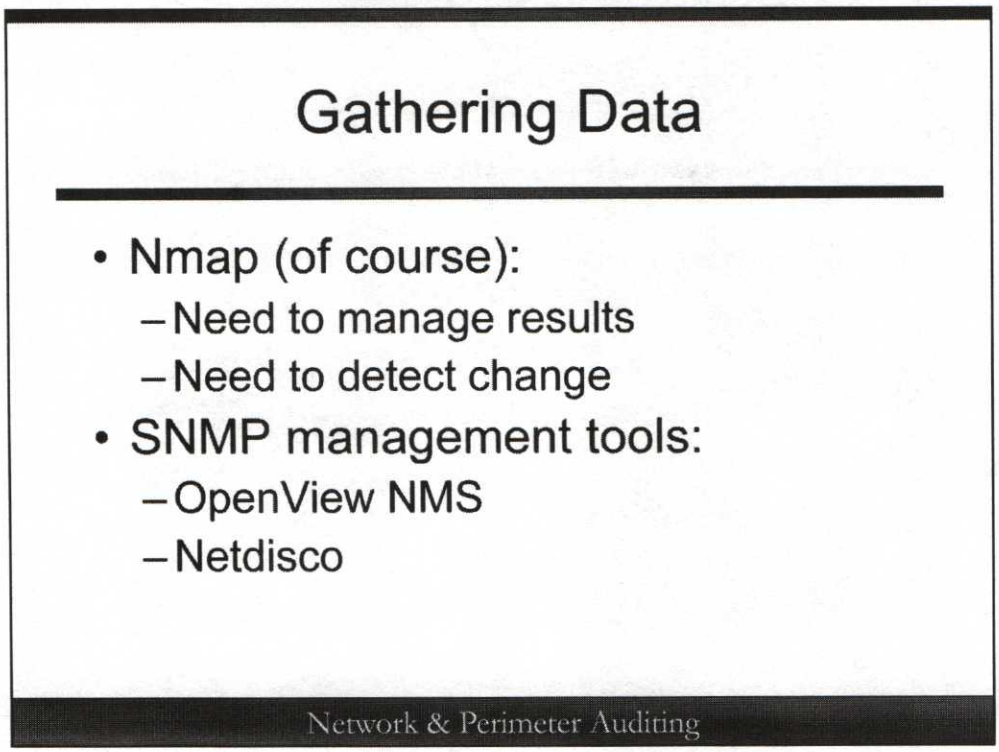
First, obtain proper permission. Of course, we want the administrators to perform these tasks, so it is these administrators who would need to obtain permission. Whether it is us or them, always make sure that you obtain permission from individuals who actually have the authority to grant that permission. Also, it's best to get that permission in writing.

One way to get permission in writing is to formalize a process. The process would include who can scan, what can be scanned, how frequently, with which tools, or perhaps detailing the intensity and type of scan.

Next, we would like to publicize the scan. You may at first think that publicizing is a bad idea. After all, if you tell people you're going to scan, they'll turn the stuff off that they're not supposed to have! Just hold your horses. We'll deal with that in a later step. We want people to know the scan is happening so that if the scan starts interfering with operations, people will realize that it might be the scan.

When people realize that the scan is causing issues, they need a way to get in touch with us. We should remain available the entire time that the scan is running, perhaps even several hours afterward. We should include contact information including our land line, cell phone, and e-mail address. It would be terrible for the mail server to go down during the scan, preventing people from contacting us if that is the only mechanism that we provided!

Finally, we want to be persistent. This means that we will not be satisfied with one scan. Instead, when a subnet can be scanned without any harm, we will stop announcing the scan. The scan will become a periodic and automatic process. This allows us to discover anything that may have been hiding from us during step 3.



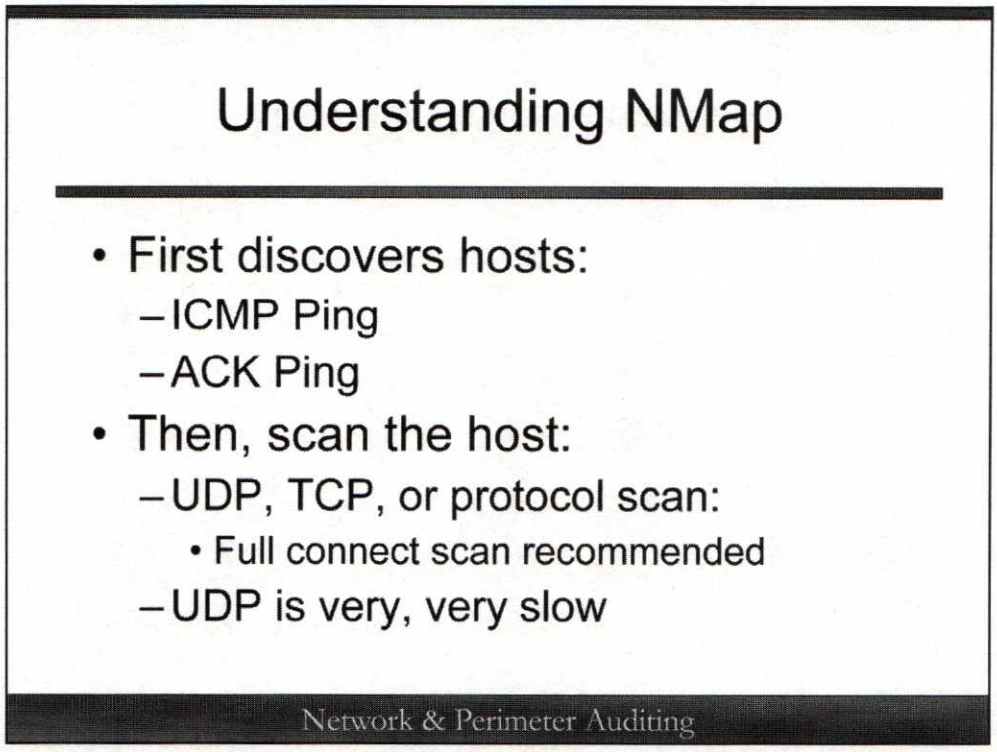
How do we do the actual scanning? You can actually use any tool that you'd like to, but Nmap is a wonderful tool to get this work done. Our discussions and labs largely revolve around how to use Nmap effectively for this task.

The big challenge that you may perceive is that Nmap tends to provide verbose output and that output is fairly unwieldy because it's just text. We need to find a way to turn that into a manageable solution. We also need a way to detect changes over time.

This isn't to say that Nmap is the only way to go. For example, an extremely useful piece of information to baseline is the aggregate routing table for our internal network. There is absolutely no way for Nmap to get you this data. This isn't something you "scan"; it's something that you have to ask for from the routers. We may also be interested in tracking hosts moving around the subnets.

SNMP monitoring tools can be quite useful here. OpenView Network Management System from HP is quite good for this type of stuff. It's actually not a security or audit tool, so there's no automatic report for the stuff we're talking about. Even so, it's easy to use a tool like this to extract the relevant data.

Another tool we could try out is Netdisco. We'll take a look at this tool in a virtual machine during our lab. It is a bit different than OpenView. OpenView is about health monitoring. Netdisco can do this, but we can also use it to aggregate a lot of interesting information about our network, router, and switch configurations.



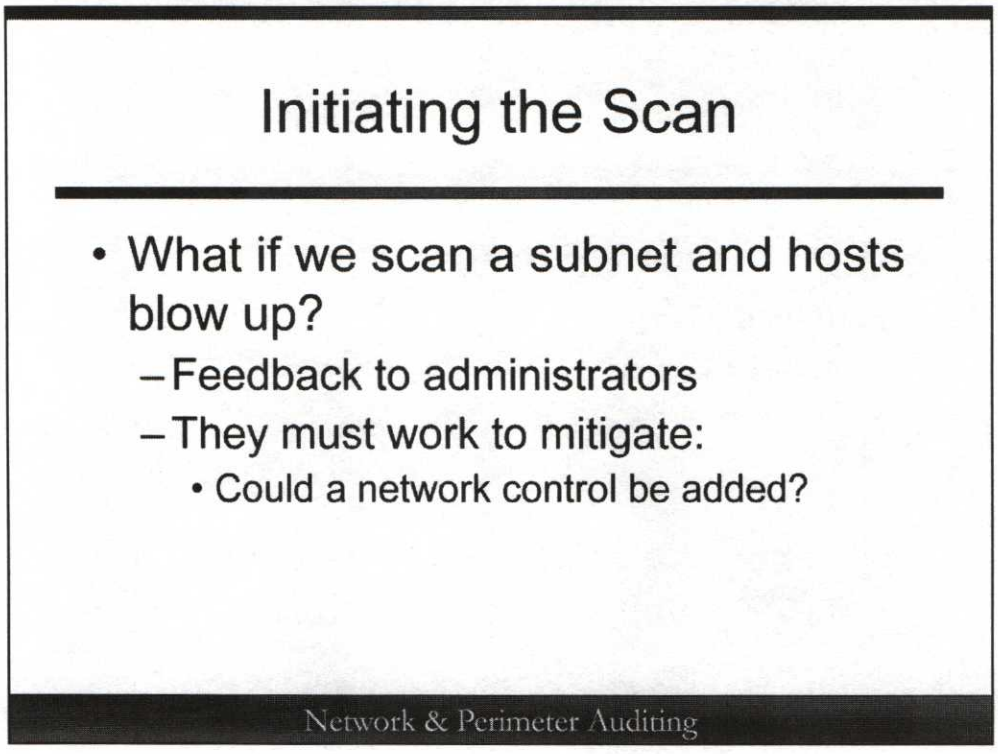
Let's take a minute to explain exactly how Nmap works. Even though we have already used this tool today, the only thing that we've used it for is scanning the firewall. Scanning one host with Nmap is easy. Scanning a lot of hosts can be extremely time-consuming if we don't understand how the tool actually works.

In its default configuration, Nmap begins any scan by checking to see if each target host is online. Nmap has several ways that it can determine this information, but the default is to perform an ICMP ping (echo request) and an ACK ping (to port 80). The only exception to this would be if Nmap detects that the target address is on the local subnet. If this is the case, it instead uses an ARP who-has (mentioned earlier) to determine if the host is up.

If Nmap determines that the host is online, it then begins scanning for TCP ports by default. You can, of course, specify which ports to scan for. You can also control how the scan is performed. For our purposes, the most interesting scan type to use is a full connection scan. This means that Nmap complies with the typical three-way handshake approach to connections.

This scan type is important to use, especially with legacy systems, because it is much friendlier for the target network devices. This gives us the most reliable way to scan hosts while causing the least damage. In addition, we may want to look at the speed with which the scan runs because this, too, can create issues on a fragile network.

Nmap can also perform UDP scans. For the next lab that we do, we will not perform any UDP scans. The biggest reason is that they are incredibly slow. Remember that if a UDP service is listening, it will not answer a probe. This means that we have to wait for a *lack* of response. As you can imagine, this takes a long time.



What do you do if you run a scan and it crashes hosts or services? We absolutely need to make the scan results available to the administrators who are responsible for the systems that were scanned. Those administrators must take the data and verify first that all the enabled services are actually required. Next, if a service is required and fails for some reason during the scan, the administrator is responsible for patching the host or otherwise fixing the issue.

What if the system in question cannot be patched? Perhaps, you have a legacy system and no patches are available, yet the system is critical to your operations.

Consider whether it's possible to introduce an additional network control to mitigate the problem. Let's go with an extremely simple solution. Could it be as easy as installing a $30 Linksys router onto the network between the legacy host and the rest of the network? This Linksys device can be configured to perform port forwarding for any required service, but everything else on that legacy host is now completely insulated from our network and, as a result, from our scan.

The whole point here is that we *must* scan the network. It is simply not acceptable for an administrator to request a long-term exception to scanning his system if it is connected to the network.

Persistence

- We will not always publicize!
  - When the network can be scanned safely, we no longer announce the scan!
  - We now find people who are trying to hide stuff

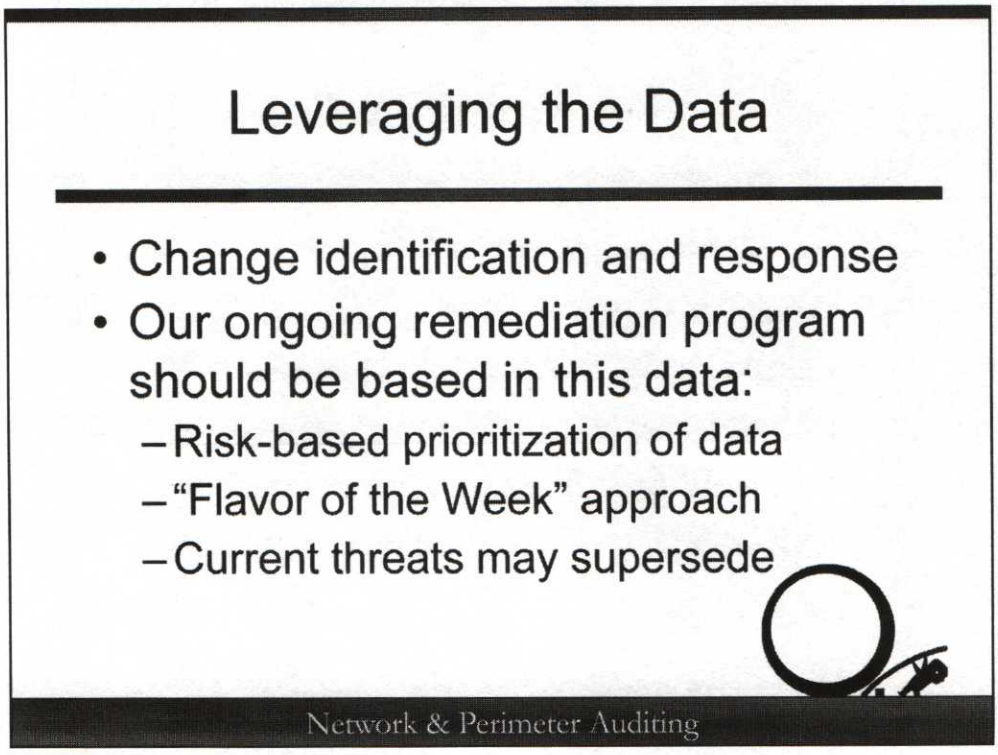
Network & Perimeter Auditing

After we get to the point in which we can scan each network with impunity, we are now ready to worry about that persistence element.

This raises the question of how often the network should be scanned. The answer depends on the size of your network and what you scan for. Under ideal circumstances you need the scan to run every night and scan your entire address space. If your network is only a few thousand nodes, you can likely scan through this entire space every night with no problems.

If your network is significantly larger than a few thousand nodes, you need to break the scan into several pieces. Even so, think about how you do this carefully. I would still prefer to find that the critical systems and routing/switching infrastructure is scanned as frequently as possible.

The other thing that can stretch this out is UDP. If we choose to include UDP in our baseline, each scan can take an extremely long time. In most enterprises in which we have chosen to include UDP in our scanning system, we typically let the scanner run continuously, allowing it to work nonstop. When we take this approach, we also tune the scanner using the timing options so that it is not trying to scan too quickly, leading to other network issues.



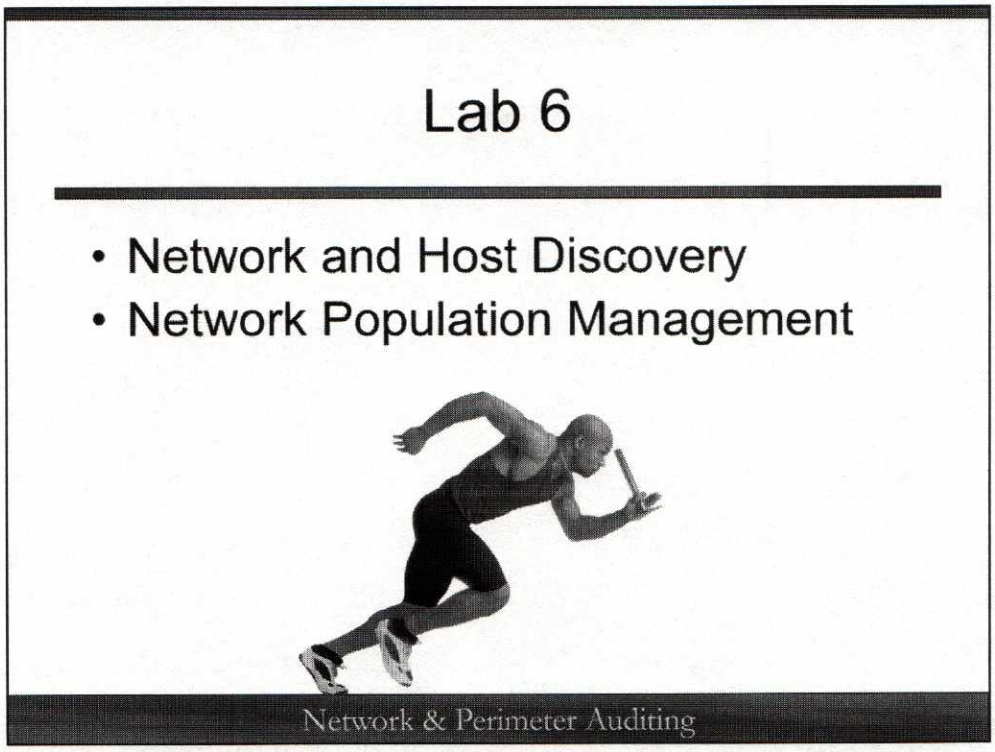
Now that we have all this data, what do we do with it? Well, network engineers and security analysts would likely want to monitor this data daily for change. We'll look at an extremely easy-to-understand reporting tool for this, yandiff, in our lab.

We also want to see how the security and administration team handles continuous remediation and monitoring. Actually, I expect to find that the security officer is overseeing and driving this process. When I have worked as a security or chief security officer, I typically call this the "Flavor of the Week" process.

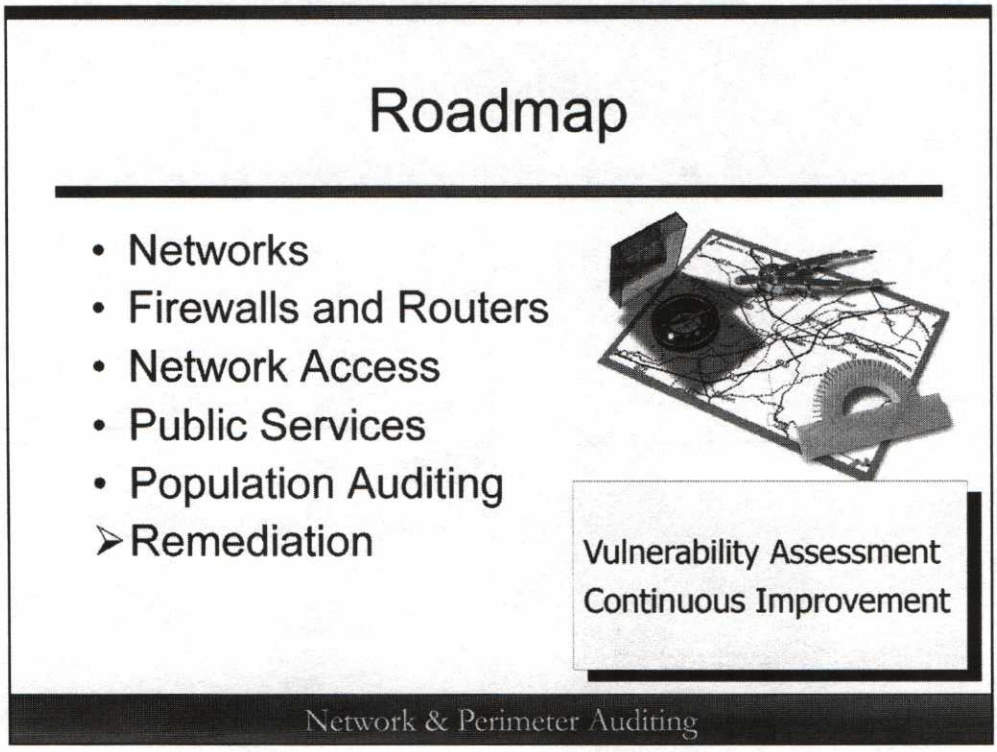
Each Monday morning when the security officer comes to work, he begins his day by reading about all the threats discovered over the weekend or last week. If there hasn't been anything that is particularly important for his enterprise, he consults a prioritized list of services within his organization that he is working through. Where does this list come from? It is an artifact of our network scanning! He doesn't have to wonder; he *knows* what's running.

After one of these ports is selected, he then leverages his vulnerability scanner to look for issues with that service on all the hosts that, according to the network mapping system, have that service running. This report is then sent to the relevant administrators asking them to determine if the service is necessary and to either disable it or patch it.

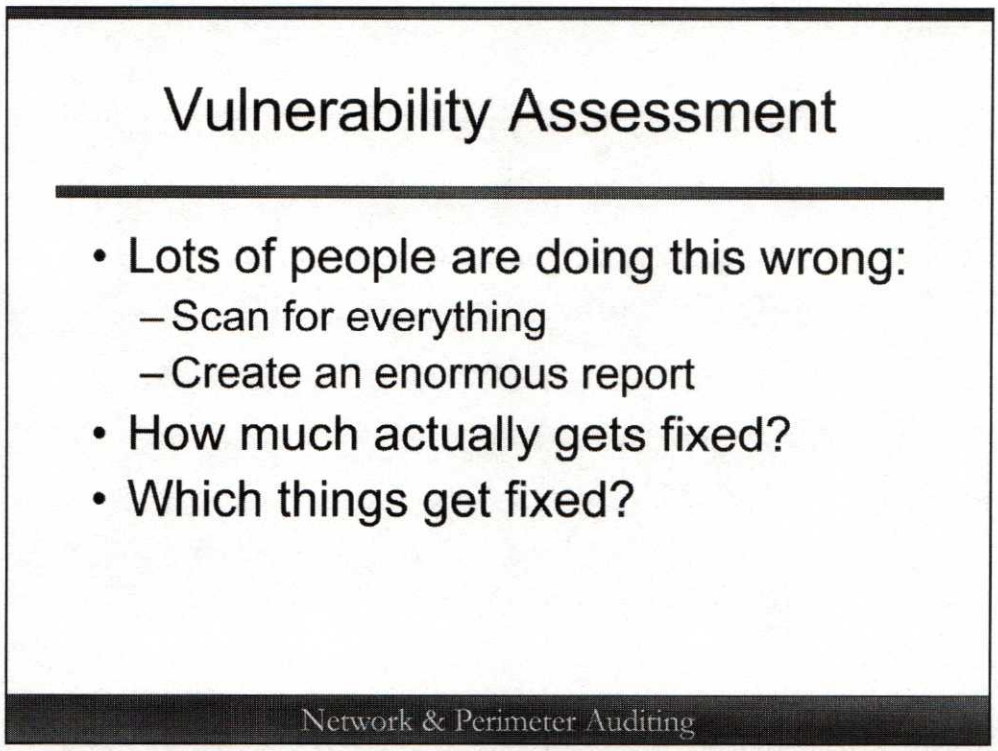
After checking in once or twice during the week, if he has not heard back from the administrator and the issue has not been resolved by Friday, the next step is to terminate service to the offending node. Of course, if the administrator is out on leave, the security officer would hold off acting until the administrator has an opportunity to resolve the issue. As an auditor, if I found a security team following this process, I would be extremely satisfied with its continuous monitoring and remediation program!



Enough talking! Let's jump back into the workbook and see some tools that we can leverage to accomplish these tasks! Please follow along with your instructor and feel free to ask questions!



The last thing that we'd like to cover today is vulnerability assessment. Let's talk about how to do this right and discuss some important features that you'd like to find in a vulnerability scanning tool within your enterprise.

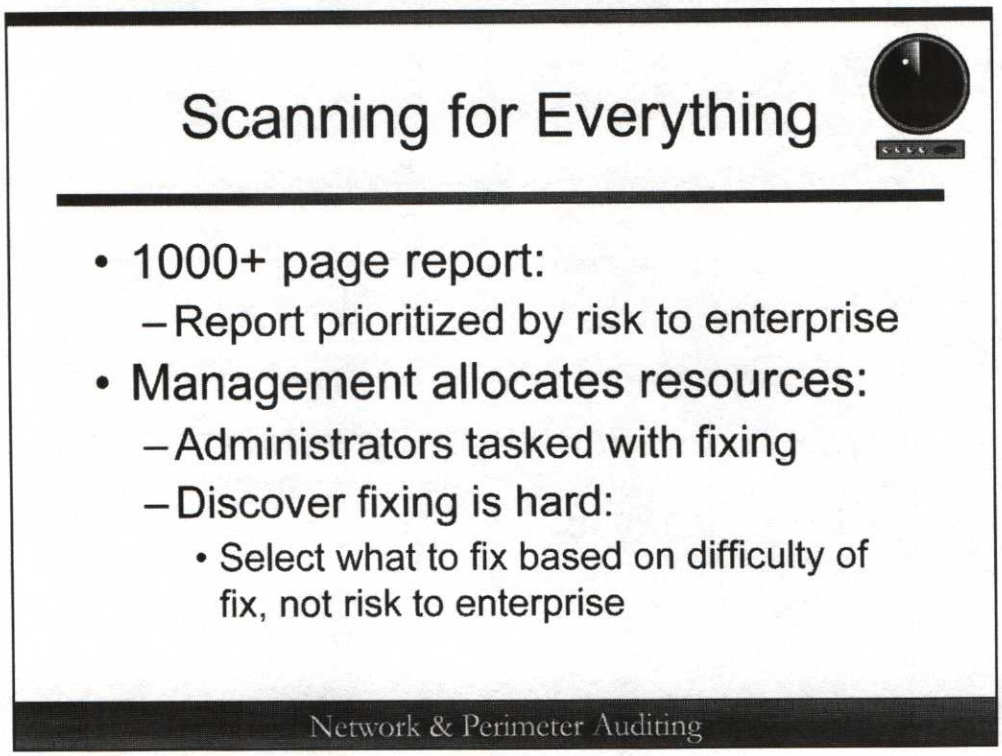


Vulnerability assessment is an important activity, but we find that most organizations are just plain doing it wrong. This may sound puzzling because the tools are not terribly hard to use or configure. Think about this question, though: If you have ever run a scan to find everything that's wrong, be honest…how much of that actually got fixed?

When management asks to have a vulnerability scan performed, there is almost always some kind of initiating event that sensitized them to the need for a scan. However, the more time that passes between the initiating event and the ongoing scans, the less investment management is willing to make because it is far less sensitive to the issues.

Think of it this way. If you, like me, had to get onto an airplane and fly somewhere October 2001, you likely were not particularly disturbed by an airport security person wanting to search you and your bags. Today, more than decade later, however, we are far less sensitive because nothing especially bad has happened with terrorists and airplanes recently.

You can understand this, but how can we communicate this effectively to management? How can we help it get the kind of scan that will actually give what it wants and needs?



I usually approach this by discussing it frankly with management. When management asks me, "Can you do a vulnerability scan for us?," I typically respond by saying, "Sure. What is it that you'd like me to find?"

Because of my question, the response I usually get is, "Everything…?" My question has made management unsure, but that's okay! I now follow this up by explaining what will happen if I scan for everything.

After the scan is complete, I'll generate a report. That report will, for argument's sake, be 1,000 pages. You may think that's big, but if you've done this before and if your network is of any reasonable size, you can attest to the accuracy of my claim. In any event, I explain to management that I will organize that report based on risk and then I will come in and scare them. I'll scare management so badly that it will tell the administrators that their one and only job is to fix everything in the report.

The administrators don't mind this because they recognize that there are a lot of things that they'd love to fix; they just haven't had the time. The administrators now begin fixing issues. They discover quickly that fixing things is actually a lot of work. In fact, after fixing one or two issues, they begin leafing through the report. Rather than looking to fix things based on how much risk they represent, they are now choosing what to fix based on how hard it appears to be to fix.

As time goes on, their energy wanes. In addition, management's attention, too, becomes unfocused. Within a few weeks, everything is back to business as usual.

Time Passes

- Within a month or so, no more fixes:
  - Remediation follow-up report pages:
    - New vulnerabilities
    - Fixing things uncovered other f
  - Nothing bad has happened
  - No more resources

Network & Perimeter Auditing

Now I come back to the enterprise a few months later and scan the network again. This time the report isn't 1,000 pages long, it's 1,100 pages long. "How could it be bigger?," you wonder. "We fixed things, didn't we?"

Although it is true that you have fixed things, it is also a fact that security is not a stable system. Not only have new vulnerabilities been discovered but in the process of fixing things, previously concealed flaws have now been exposed to the light of day.

When I try to talk to your management about my findings, it may pay lip service to how serious it is, but it will not actually devote any resources to it. Why? Because it sees no return on investment and nothing bad seems to have happened.

Clearly this process doesn't work.

What Works

- High-level risk assessment
  - Objectives tied to systems:
    - Which systems matter most to mission?
    - How could they be taken down?
  - Top 10 or top 20 assessments
  - Resources allocated
  - Three months later, how much is fixed?
- Now expand the scope!

Network & Perimeter Auditing

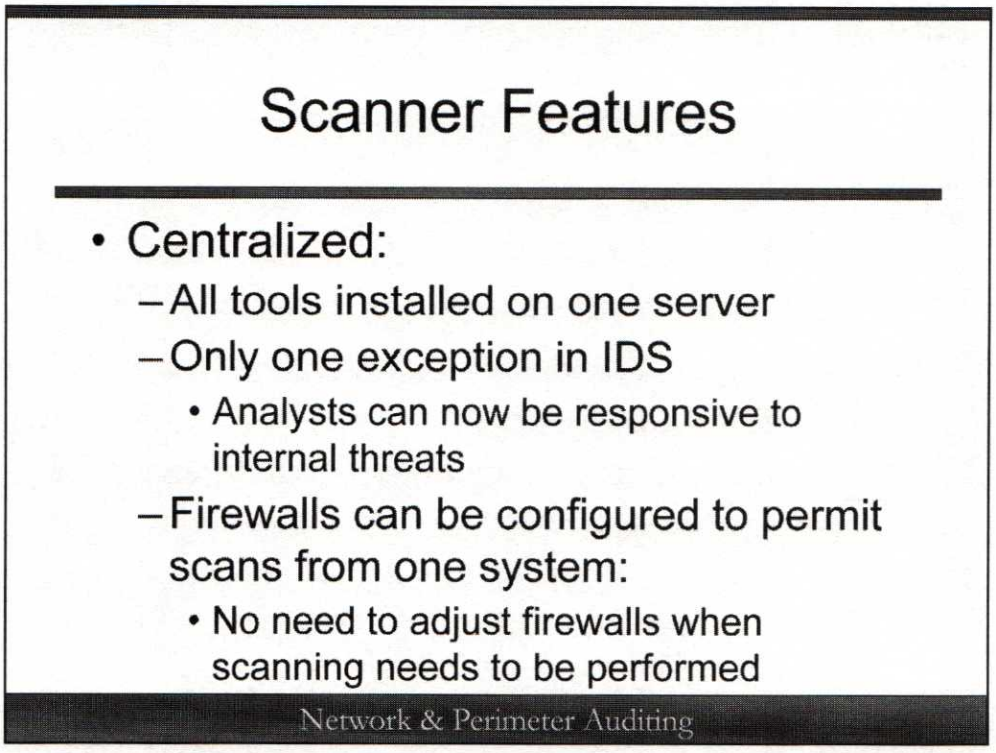
What does work is to approach the problem in an entirely different way. We begin by performing a high-level risk assessment, working to identify the organizational mission and identify which operational components most directly influence the ability to achieve that mission. With that knowledge we now work to identify which systems support those operational components.

Now that we are at a system level, we take the time to identify the top 10 or 20 risks that might exist in the context of those systems. This is a great time, by the way, to have a look at vulnerability lists and resources such as the 20 critical controls.

With these top 10 or top 20 items identified, we now perform a vulnerability scan for just these things. The report is now only approximately 100 pages, and it's actually important that all these items are fixed. The administrators now have the same 3 months to work through remediation.

When I return in 3 months to check on remediation, we still have findings, but the report is only 10 or 20 pages. When we report to management now, it can see clear improvement and real return on investment. It actually feels more secure, and it has good reason for feeling that way!

Given that we've made progress, funding will likely continue. We can now expand out to the top 25 or the top 50 issues, progressively working through issues based on actual risk rather than arbitrary vulnerability.

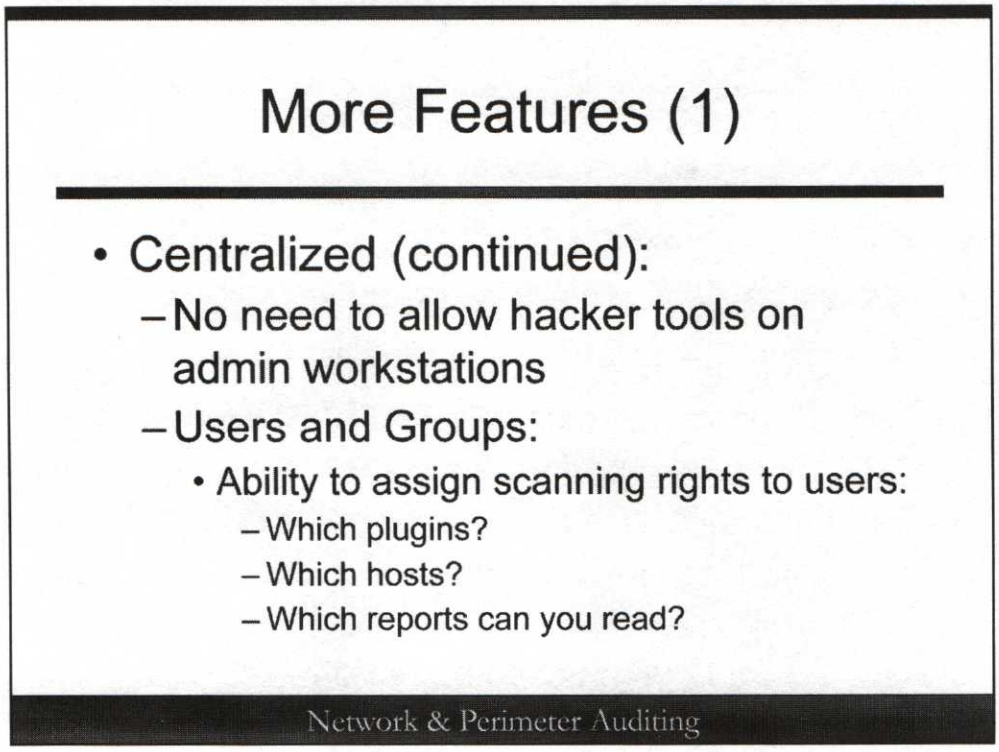


Now that we understand the process that works, let's talk about features that you'd like to find in a vulnerability scanner. There are many scanners that can fit the bill. There's no single vulnerability scanner that we would recommend over all the others, and trust me, if there were one that I preferred, I'd let you know!

One of the more important features that I'd like from the perspective of a security officer is a centralized scanner. I don't want something that I have to install onto endpoints. I'd greatly prefer something with a web-based interface that I can deploy centrally in my network. This means that I no longer need to allow administrators to install hacker and other security tools onto their systems. They also don't need to be local administrators. Instead, the location of the vulnerability scanner becomes, essentially, the approved security scanning host. If there's some tool that someone needs, it can be installed there and that user given the access needed to use it.

With this centralized, I can also make better use of my IDS/IPS. If you don't have a centralized scanning device, IDS analysts eventually become desensitized to internal scans because administrators run them from all over the place. With the centralized server, any scan coming from anywhere else is unauthorized, and we can deal with it as a security threat.

Another advantage is that it will not be necessary for administrators to turn firewalls off and on to conduct scans. Instead, systems can be configured to allow the authorized scanner to conduct scans, but no other system is permitted. This all becomes manageable.



Another feature that I'd like is the ability to distinguish users and groups within the scanning system. In particular, I'd like the ability to assign rights and restrict features from various user groups.

Imagine that you're a Windows administrator. If this is the case, you should certainly have the ability to run scans against the servers that you administer. However, you should *not* scan anything else. Furthermore, although you are allowed to scan Window hosts, I may want to restrict you from using certain "dangerous" plugins that are known to cause issues. All this should be configurable by the administrator.

Don't overlook the ability to restrict which reports any particular user can view. Certainly, the security officer and auditor can view any report. That Windows administrator, however, should view only the aspects of reports that actually apply to the systems that he manages and nothing else.

More Features (2)

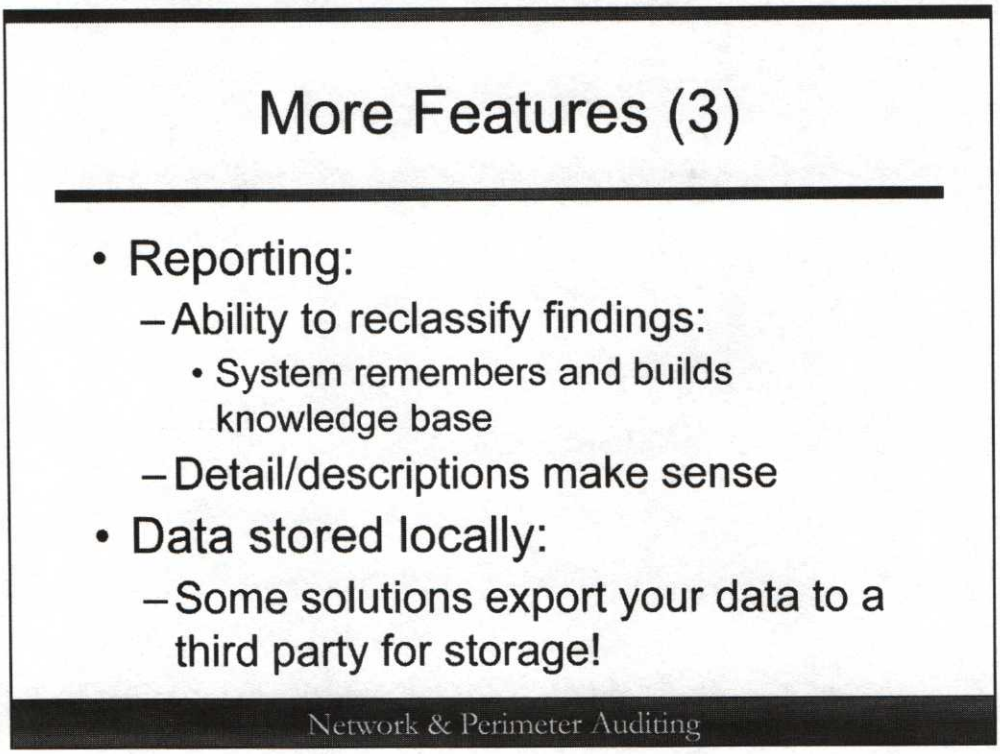
- Plugins:
  - History of keeping tool up to date
  - Ability to write our own tests
  - Good documentation of plugins

Network & Perimeter Auditing

This brings us to plugins. We want to make sure that whichever scanning tool we use, it has a good history of updates from the vendor. Many vendors begin quite well intentioned, but some just can't seem to keep up with the vulnerabilities that are discovered every day.

We'd like to find, too, that the vendor provides good documentation with the plugins. It is frustrating to find something in your report as an important finding, but when you try to dig into it and figure out what it means, the tool has only the most rudimentary data that is essentially meaningless.

Another important feature is the ability for us to create our own plugins or signatures. This is especially important if we develop software or hardware within our organization. Now, when we test our own internal stuff and find issues, we can build plugins to test for these things automatically. This allows us to leverage our existing vulnerability testing infrastructure rather than having to build some other tool for every test that we need to automate.



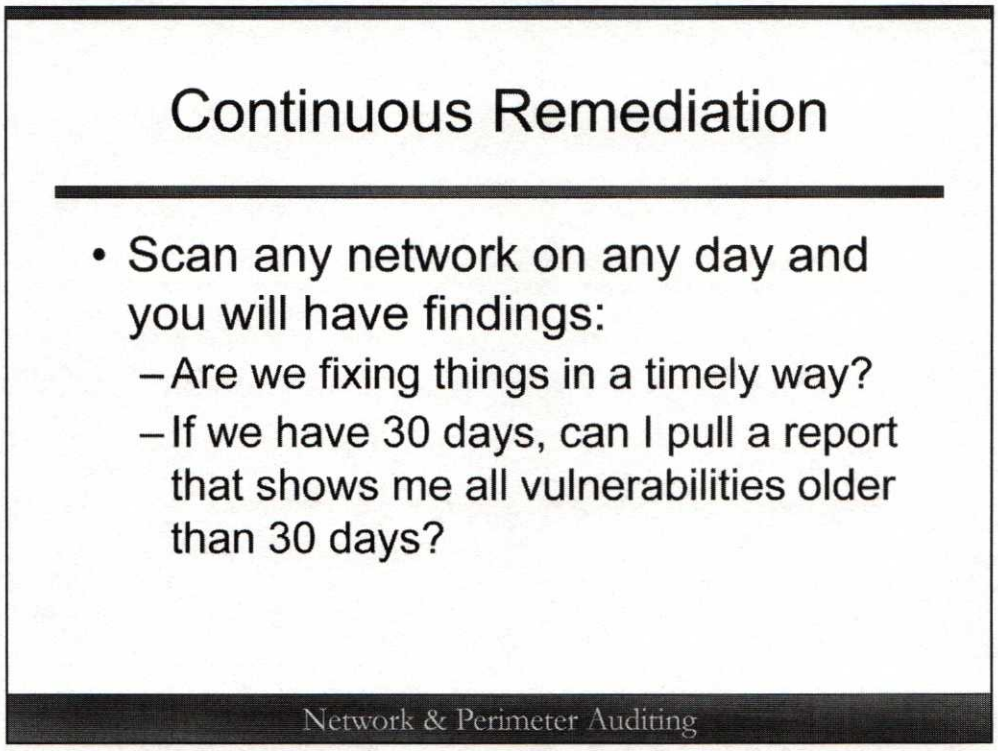
For reporting, you should look for a couple important features. Obviously, the reports should be easy to read. More than this, though, we'd like the ability to reclassify findings. We'd also prefer that the system remembers how we have classified a particular finding for a particular host so that all future scans will take this into account.

Understand that we're not just talking about the ability to mark and exclude false positives. We also want the ability to simply reclassify something as unimportant or as important regardless of how the tool feels about that issue.

You also want to be sure that all the vulnerability data and reports are stored locally. We point this feature out because there is an otherwise wonderful vulnerability scanning tool that provides what some feel is the best reporting in the industry, but all your report data is sent out to its site for management. When you want to view a report about your system, you must log in to its site to retrieve the report.

In all fairness, this vendor does claim that your data is all encrypted and it doesn't have a key that can read it. Even if that is the case, I am extremely uncomfortable with my vulnerability information being stored anywhere other than inside of my network.

I'm sure you're wondering who this vendor is. Don't worry, I'll tell you in a couple slides.

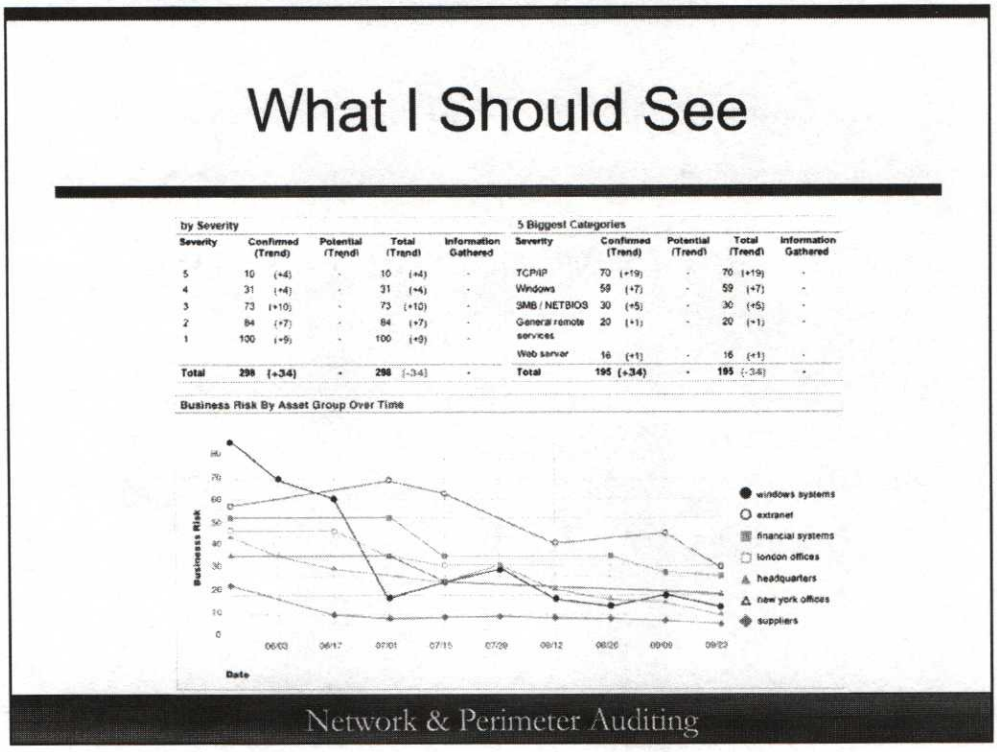


When working to validate the security of networks, keep in mind that if you scan any network in the world on any given day, you will absolutely find vulnerabilities in it. I'm not saying that an attacker can just walk right in, but I am saying that there are patches missing or configurations that are not ideal.

This problem is not a result of a poor security budget or a lack of focus on security issues. This is because networks are constantly evolving, vulnerabilities are constantly being discovered, and we must go through testing and patching processes. This takes time. Add to this the sheer number of issues that must be fixed every month, and it's a wonder that we are keeping up as well as we are!

What we would like to do is look at the patch requirements that the organization states. If organizationally patches are supposed to be applied within 30 days of release, then I should be able to look at a vulnerability scan that includes everything older than 30 days and it should be pretty clean.

This gives us another feature that we'd like to find. Can the system's reports show me how quickly we are fixing things? This is sometimes called the Mean Time To Repair (MTTR). If we find that our MTTR is within the threshold specified, we wouldn't get too excited if there are one or two issues that took 40 or 50 days to resolve because of difficulties in applying the patches.

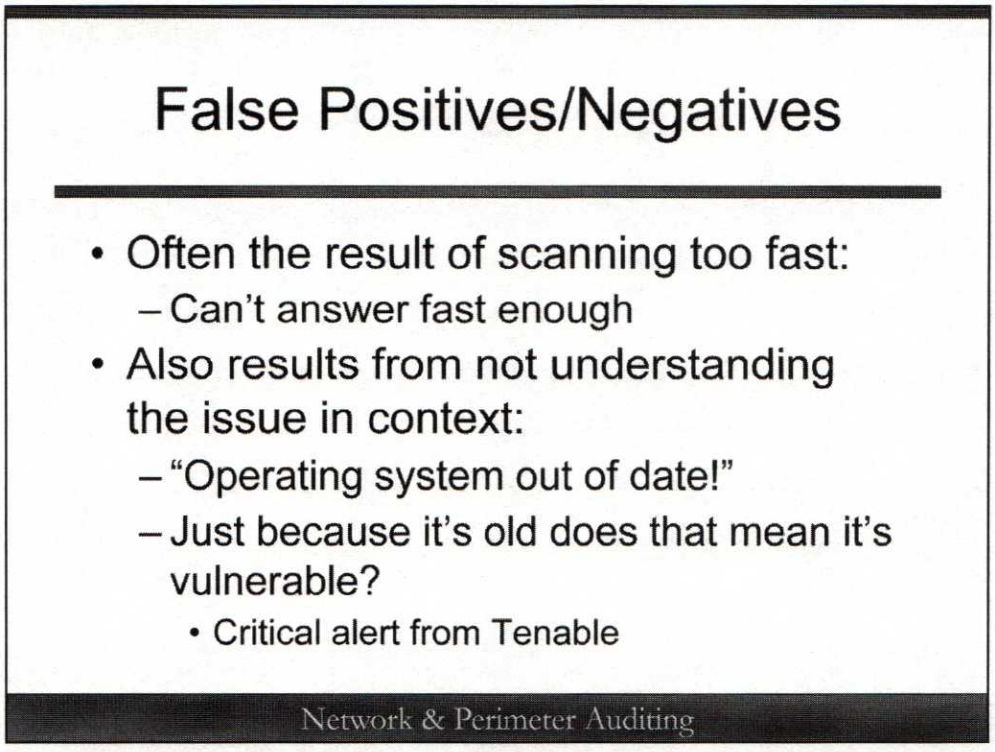


This slide shows the type of reporting you'd like to see. This report comes from Qualys. The QualysGuard tool is, in the opinion of many people in the industry, one of the best for reporting. It doesn't just tell the techies what they need to know, but it also provides useful dashboards that executives can leverage in managing an overall risk response strategy. The only big downside to Qualys is that these are the folks that we mentioned two slides ago. All the information on your vulnerabilities is actually stored out in the Qualys servers.

Notice the bottom chart in the slide. Here we can see overall trends for vulnerability over time. In the slide, it is broken out by type of system, but this is an arbitrary classification. You can actually configure it to produce the report in just about any way that makes sense to you.

Reference:

*Screenshot from http://www.cn.qualys.com/products/screens/?screen=Business+Risk+Assessment



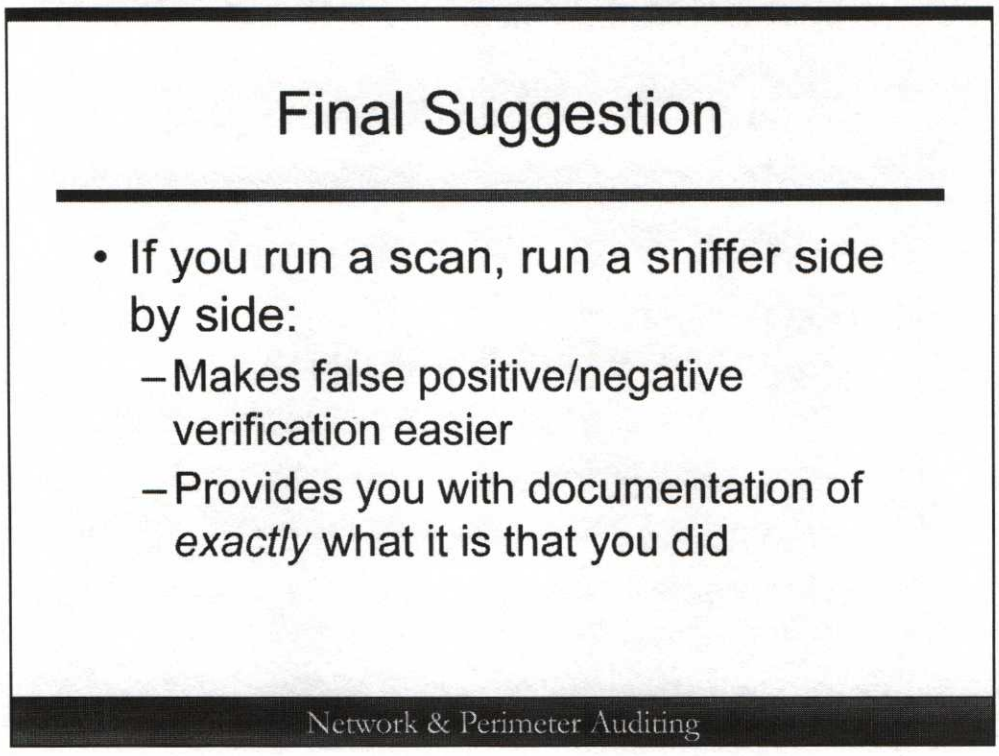
False positives and false negatives are a fact of life in any vulnerability scanning system. A false positive occurs when the tool tells you that there is an issue when, in reality, there is nothing actually wrong. A false negative is when the system fails to identify an issue even though you do, in fact, have a problem.

There are a few common causes for both false positives and false negatives. Probably the most common cause is misconfiguration of the tool. Everyone is always concerned about running scans as fast as they possibly can. Although we can understand this desire for immediate gratification, the faster the scan is run, the more likely that the tool will miss things. This can be because the system being scanned is getting loaded down and just can't answer fast enough. It could also be because the system is becoming overwhelmed and services are failing, the scanner begins to assume that a compromise has actually occurred when it hasn't.

Another major cause, especially of false positives, is poorly written signatures. This is why manual validation of any important findings is valuable.

Yet another cause is that we don't understand what the tool is actually telling us, or the tool is simply classifying something as a serious issue arbitrarily. Let me give you a specific example. I have a VAX cluster running OpenVMS 7.3 in my enterprise. OpenVMS 7.3 was released in 2001. "Wow, that's old!," you might say. In fact, there are newer releases of OpenVMS. The most recent release was for version 8.4 in 2010.

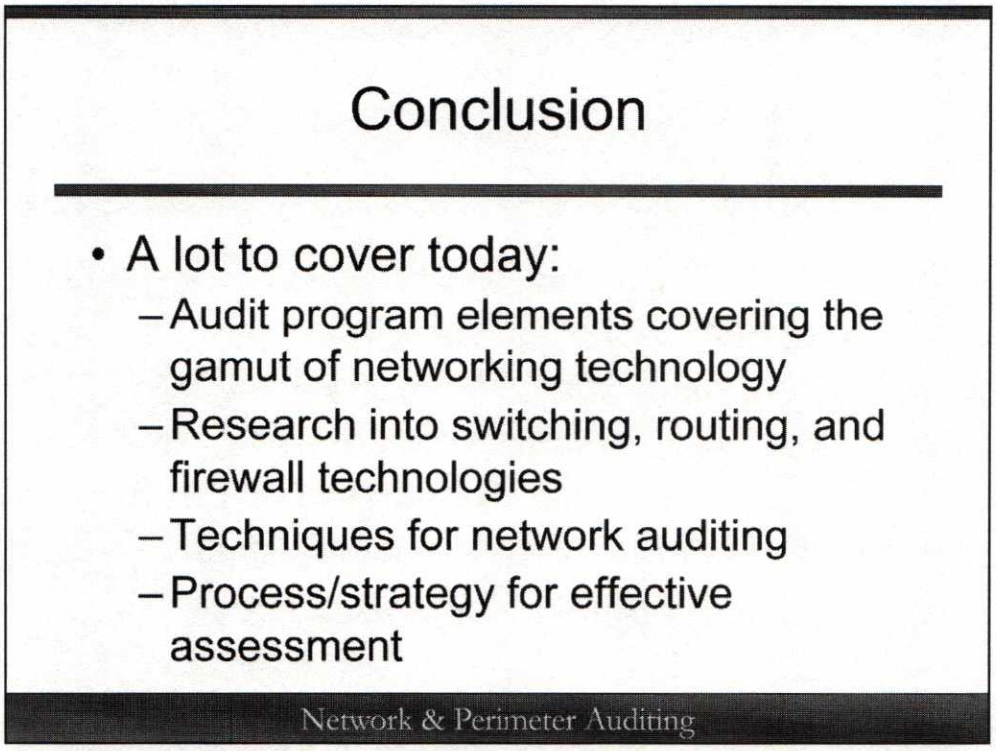
If you run Tenable's vulnerability scanner against my host, it finds absolutely no vulnerabilities. In fact, there are no known vulnerabilities in the version that we are running. However, it also generates a purple colored alert, the most serious it can generate, telling you that the operating system is out of date. Is the system actually vulnerable? Absolutely not. This is an excellent example of why we need to review and reclassify findings based on systems.



One last suggestion before we wrap up today's material. If you are ever in a position to run a vulnerability scan to perform some measure of validation, we strongly recommend that you run a sniffer side by side with it.

Running a sniffer to monitor the vulnerability scanner provides you with some extremely important data. Not only do we have a record of absolutely everything that was actually done, but we now have a means to research poorly documented plugins and signatures! If we need to figure out how a particular issue was assessed, we simply need to open up our sniffer, find that probe, and extract the content!

This allows us to perform manual validation in addition to better research what the alert is about.



Wow! What a day! We've covered a lot of ground, covering topics from Layer 2 networking fundamentals up through firewalls, routers, and general service configuration! We hope that along the way you perceived the process that we outlined as an approach to network security auditing. We also hope that you've seen and heard a lot of things that you have identified as important things to investigate when you return to your office.

As always, if you have questions or comments, we are always open to them! If you have suggestions on how to improve the course or come across any grammar or typographical errors, please feel free to get in touch with me at dhoelzer@enclaveforensics.com.