504.2

Computer and Network Hacker Exploits Part 1

SANS

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

SEC504.2 | Hacker Tools, Techniques, Exploits and Incident Handling

SANS

Computer and Network Hacker Exploits - Part 1

Hello and welcome to Computer and Network Hacker Exploits.

Day 1 covered policy and procedures regarding incident handling. Today, we discuss how the technical attacks work, and how you can prepare your defenses to handle them. Our exploration of computer attacks and defenses encompass the remainder of this course.

For each type of computer attack, we address how we can apply our six-step incident-handling process. We see how to prepare, identify, contain, eradicate, recover, and conduct lessons learned for each type of attack.

Let's start our journey.

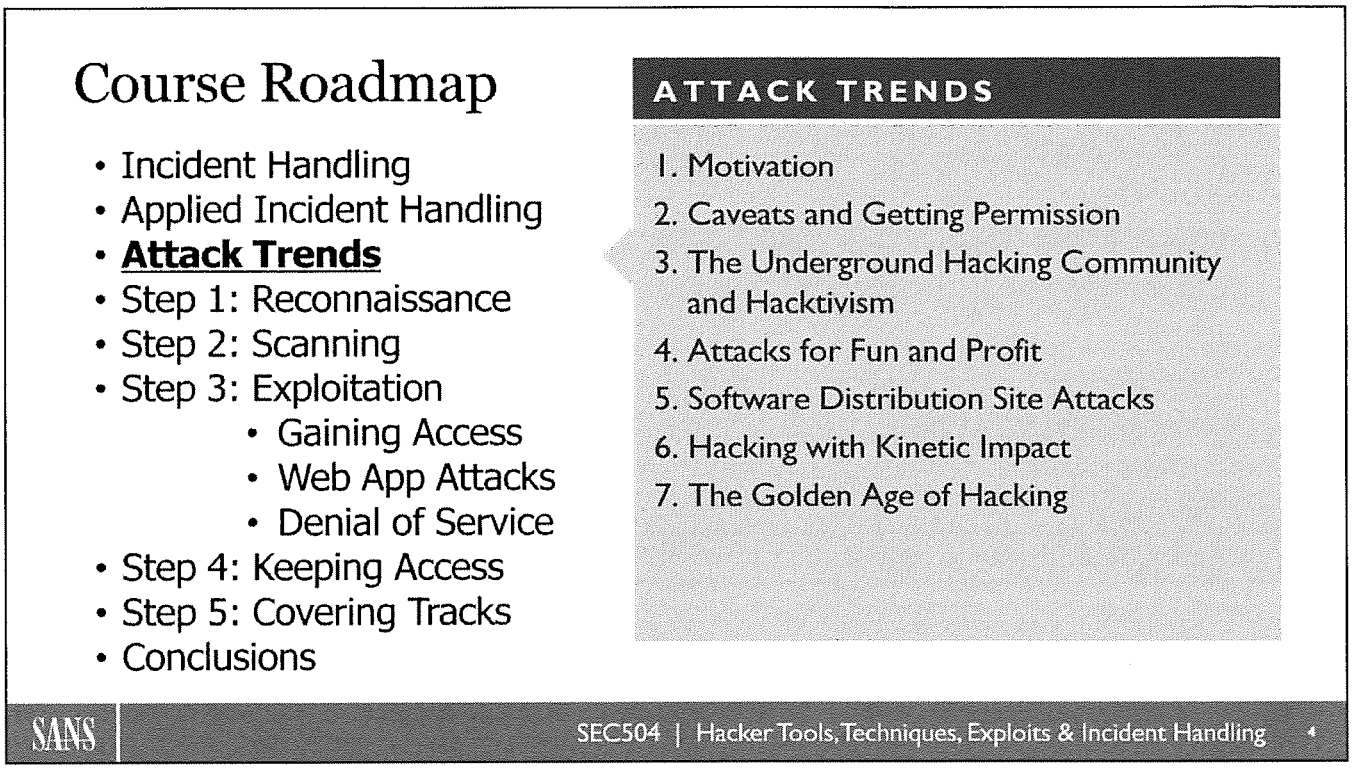
TABLE OF CONTENTS

This table of contents can be used for future reference.

Note that we put the labs in boldface so you can more easily find them if you need to refer to them during the Hacker Workshop on Day 6.

TABLE OF CONTENTS

We finish up today by concluding the Scanning phase.



These five steps represent the flow of an attack, from initial information gathering to "you are owned" to covering tracks.

Although particular scenarios may deviate slightly from this routine or expand upon it, these essential steps are used in many attacks.

An attack starts with reconnaissance (step 1), whereby an attacker conducts an open-source investigation to gain information about a target.

Step 2 is scanning. An attacker uses a variety of mechanisms to survey a target to find holes in the target's defenses.

Step 3 involves exploiting systems. In this phase, an attacker tries to gain access, undermine an application, or deny access to other users.

In step 4, the attacker maintains access by manipulating the software installed on the system to achieve backdoor access.

Finally, in step 5, the attackers maintain their hard-fought access by covering their tracks. They use a variety of techniques to hide from users and system administrators.

After we discuss this overall attack process, we address how attackers put these pieces together in a variety of sample attacks. We end with conclusions.

So, what is the purpose of this course? Why are we here? I'm glad you asked....

Purpose of This Course

- The purpose of this course segment is to understand attack methods...
- ...so we can implement effective defense strategies
- Designed for incident handlers, security personnel, and system administrators
- What do the attacks look like, and how can we apply the incident-handling process we discussed in 504.1
  - How we can create effective defenses
  - That's the reason we're here

This course is not designed to teach you how to hack. Still, to create an effective defense, we must understand the offensive tools attackers use. That's what this segment of the course is all about: Learning what the attackers do so we can defend ourselves. This course is designed for incident handlers, security personnel, and system administrators.

We cover attack concepts, which includes scanning, gaining access, modifying systems, and hiding.

Also, note the difference that the underground community applies to the terms hacker and cracker. According to the hacker community, a hacker is a highly intelligent individual who wants to explore technology to learn. A cracker is someone who maliciously breaks into a system. I try to follow this "correct" usage in this course. I refer to intruders as attackers or crackers. I actually prefer the term "attacker," because it is neutral with regards to the motivation of the perpetrator.

Unfortunately, the major media do not observe this distinction and label crackers as "hackers."

Why We're Covering What We're Covering

- Why we chose these tools and techniques
  - They are in widespread use right now
    - Based on the collective experiences at the SANS Internet Storm Center, incident-response activities in large and medium enterprises and input from hundreds of incident handlers
  - They provide us fundamental information about the principles the attackers employ
  - They illustrate what we need to do to defend ourselves
  - Some of them are pretty darn nifty (although nasty)

Over the next several days, we cover approximately 100 tools and techniques. We select these particular attacks because they are the most damaging and widely used today. In addition, each attack illustrates what we need to do to defend our systems. For example, there are dozens of sniffing tools in widespread use. However, by covering tcpdump, Wireshark, and Dsniff, we can get an excellent feel for the nastiest of attacks and how to stop them.

In addition, never underestimate your adversaries! You need to understand their tactics and be ready to defend your network.

Caveat

- To the extent possible, we are platform-independent
  - Individual tools may run on UNIX or Windows
  - But we will cover attack concepts that can be applied against Windows, UNIX, or other platforms (Novell Netware, VAX, MVS, Cray, and so on)
- We include links to tools; use at your own risk
  - We neither recommend nor endorse any tools
  - They could harm your network in unexpected ways
  - Review the source code for legitimacy
  - Experiment on a test network, separate from production systems
  - Use of tools may be illegal in your area; check with your lawyer We are not liable if you cause damage
- Always get permission before running these tools, even on your own network

Many of the tools run on the platforms of choice of the attacker communities: UNIX and Windows. Although these tools run on these platforms, many of them are used to target any type of platform. For example, an attacker may use a session-hijack tool on a Linux machine to take over a session between a VMS machine and your mainframe. Alternatively, an attacker may launch a Denial of Service attack against your old Novell Netware network or IP Toaster using a Windows 2000 system. So, although we may discuss particular platforms, the attacks are applicable against all types of systems.

Also, we extensively deal with both Windows and UNIX. Don't ignore the UNIX stuff, saying that you are an all-Windows environment! To be a solid incident handler, maximizing your value to yourself and your employer, make sure you understand both Windows and UNIX. Don't confine yourself to just one environment. Work effectively in both, and you have the opportunity to go further in your career!

Note the legal restrictions your particular geographic location may impose on the use of these tools. In some countries, use of these tools across a public network is illegal, even if you target your own computing systems. Be sure to check with your legal folks before running these attacks. Also, if you plan to use the tools, make sure you have authority and/or permission to run these tools against your organization's computer systems.

Translation of "use at your own risk:" Don't call me whining if you use these tools and they hurt you... unless you want to pay me to listen to the whining.

Always Get Permission

- Full and documented permission is essential before your run any of these tools on a network
- When getting permission it needs to be in writing
  - Verbal agreements don't hold up to well in court
- The documented permission should also state that the giver of permission understands there may be "adverse" side-effects of the scanning or testing activity
- This is also known as a "Get Out of Jail Free" document
- Get written permission! Sample form at: http://www.counterhack.net/permission_memo.html

It is often the case where testing is done without proper permission. Often, people believe that if they are testing a "friends" network or a network for a company they work for they will be fine. However, we would caution anyone attempting to do any testing at all to always get documented permission. A verbal is never good enough.

Remember, if something goes wrong, often people try to find someone to blame. Unfortunately, it is very easy to target the person who ran the test.

Always get permission from the appropriate authorities in your company before using any computer attack tools to locate vulnerabilities. You can find a sample form for getting such permission at http://www.counterhack.net/permission_memo.html. Have your lawyers look it over and tweak it to fit your needs. Please note that this form is suitable for an employee doing a test of his or her employer. It is NOT suitable for a third-party penetration testing company, as it does not include limitation of liability language required for such contractual relationships.

General Trends – The Underground Community

- What are we seeing in the wild?
- Attack tools are getting easier to use and more easily distributed
- High-quality, extremely functional attack tools
  - Better quality than from some major software houses
- Rise of the antidisclosure movement
  - Script kiddies are abusing tools
  - Vendors don't want vulnerabilities to be publicly released
  - Some groups are no longer releasing exploits publicly
    - The "no-free bugs" movement started by some researchers
  - Other hacker groups are targeting proponents of full disclosure
  - The pendulum swings to and fro
  - Significant implications on disclosure with respect to the DMCA

With the rise of various groups writing and releasing computer-attack tools, a lot more information about security vulnerabilities is available to the general public. The less-informed attackers (often called "script kiddies" or "ankle biters") use this information in attacks. We must also use this information to defend ourselves. I included several references at the end of the handouts to help you stay informed.

In addition, we see major debates on whether information about security vulnerabilities should be widely disclosed in public (full disclosure) or should be hidden until adequate defenses are released (antidisclosure). There are significant legal issues here, including copyright protection and prohibitions against reverse engineering copy-protection schemes manifested in the Digital Millennium Copyrights Act (DMCA). Some security researchers have expressed frustration at the fact that vendors do not want to pay for information about the vulnerabilities the researchers discover in their products. These researchers have discussed a "no free bugs" policy, in which they try to get paid for the vulnerabilities they discover. But, vendors often lament that they feel that such researchers are engaging in extortion, holding the vendor's business hostage with their results. There is no clean or easy answer, as the dilemmas regarding disclosure continue and intensify.

General Trends

- Excellent communication through the computer underground
  - Chat, web, informal grouping, and hacker conferences
- Rise of hacktivism
  - Hacking to make a political point
  - Not just web-site tampering
  - Manipulating the computer and financial infrastructure of a target for political reasons is also a form of hacktivism
  - Allowing political dissidents to communicate without interference from oppressive governments

The computer underground has a highly effective means of sharing information. We need to keep up by sharing information.

Hacktivism, which is launching computer attacks to make a political point, has been increasing. The most obvious form of hacktivism is web-site tampering. However, don't think that web-site tampering is the only form of hacktivism. Other elements of hacktivism include setting up anonymous remailers so people can communicate without being observed by oppressive governments. In addition, manipulating the financial infrastructure of a target organization for political reasons is an extreme form of hacktivism.

General Trends – Attack for Fun and ***PROFIT***

- Attackers are figuring out how to make money from their malicious code
- Ask law enforcement; if there's money in a given crime, we'll see much more of it
- How to make money on malicious code
  - Sell the code for backdoors/bots
  - Spam and web-based advertising
  - Pump and dump stock schemes
  - Phishing: e-mail, phone, and targeted (spear) phishing
  - Denial of Service extortion
    - Not just porn and gambling sites as targets any more
  - Keystroke loggers stealing financial information
  - Rent out armies of infected systems for all of the above
  - RAM scrapers pulling CC numbers of POS terminals

One of the big infosec stories of recent times involves attackers learning to make money from their activities, ranging from exploiting browser holes for grabbing financial data to utilizing worms as vehicles for Denial of Service extortion. Indeed, we have seen attackers directly selling to the highest bidder customized malicious code to control victim machines or even renting out armies of infected systems useful for spam delivery, phishing schemes, Denial of Service attacks, or identity theft. As the bad guys hone their business models, look for more of the attack types we've seen this year, but cranked up several notches.

General Trends – Software Distro Site Attacks

- Hack into web and FTP sites and alter software to include backdoor
- Everyone who downloads and uses the tool is impacted
  - Numerous historical examples
- Another approach is embodied in ISR-Evilgrade tool
  - Listens for software to request update
  - Sends response with malware
  - Currently includes modules for Java browser plug-ins, Winzip, WinAmp, MacOS X, OpenOffice, iTunes, LinkedIn toolbar, and more
  - More than 60 software packages in total whose Internet updates can be subverted this way

A disturbing attack vector involves an attacker placing a Trojan Horse backdoor version of a commonly used software tool on the web or FTP server used to distribute that tool. There are numerous examples over the past decade of such attacks.

A related attack vector involves undermining the automatic update process used by some software. The ISR Evilgrade tool implements this attack by listening for requests for software upgrades and injecting malware as the response. Currently, Evilgrade can undermine the update process for the Java plug-in for various browsers, WinZip, WinAmp, Mac OS X, OpenOffice, iTunes, and more.

Unfortunately, this tools has not been updated since 2010. Even more disappointing is the fact that it still works on many of these products.

Software Distro Site Defenses

- Check hashes across multiple mirrors
  - Check both MD5 and SHA-1 at least
  - Md5sum and sha1sum are built into Linux
  - Md5summer is available for free for Windows (md5summer.org)
  - Md5deep is another good project at http://md5deep.sourceforge.net/
    - Calculates MD5, SHA-1, SHA-256, Tiger, and Whirlpool hashes
    - Available for Win and Linux/UNIX
  - RIPEMD-160
- Check PGP signatures if available
  - Make sure you check against a trustworthy key
- Don't put new software directly into production; test first

To defend against this attack vector, make sure you check the integrity of the packages you download. Whenever I upgrade a software tool across the Internet, I always download copies from at least three different mirrors. I then verify the checksum of the programs from each mirror to make sure that they all match. Because I'm the cautious type, I always check both the MD5 and SHA-1 hashes to make sure that they match, using tools such as md5sum, md5summer, and md5deep. The latter (md5deep) can calculate MD5, SHA-1, SHA-256, Tiger, and Whirlpool hashes. Now, there are some possible attacks against MD5 and maybe even SHA-1. Checking both gives extra protection, because an attack that would fool both simultaneously would be difficult to pull off. Md5sum is built-in to most variants of Linux, and the free MD5summer program works great on Windows. For SHA-1, a variety of implementations are available, including the sha1sum program built into many Linux distros. If you are paranoid, you could also use RIPEMD-160 in addition to MD5 and SHA-1, which is available in source-code form at several locations. Also, if there is a PGP signature, I check it, using the author's PGP key that I retrieve from a keyserver or another website on the Internet.

The idea is that an attacker would have a more difficult time compromising all mirrors of the code, and therefore, I'll be able to catch his treachery by observing different versions on the mirrors. Of course, if the attacker alters all mirrors, or changes the author's PGP key to match his/her own, we're still sunk. But, we have raised the bar nonetheless.

In addition, you should always test new tools before rolling them into production. Such a test process not only gives you a chance to detect the malicious software in advance, but it also gives you some precious time for others to discover the problem before you blindly put code into production. I had a client whose bacon was saved simply because they spend at least one month reviewing any new release of Sendmail before putting it into production. Now, if it's a screamingly urgent security issue, you have to rapidly push fixes into production. But, for run-of-the-mill upgrades with minimal security implications, you can safely wait a few weeks for testing in a lab before deploying new software into production.

General Trends – The Golden Age

- The marriage of general attack tools and worms, viruses, and bots is resulting in powerful techniques
  - Worms are increasingly being used to carry bots, backdoors, password crackers, and scanners
  - Botnets are growing large with self-replicating code
  - Several active botnets with more than 1 million hosts
- Attacks from multiple sources simultaneously
  - Distributed, cooperative attacks are all the rage
  - Using groups of coordinating attackers or a single attacker with a botnet
- Bottom line: It's a good time to be an attacker (or a security practitioner)

As computer systems and networks advance, the analogies to biological systems greatly increase. As we move forward, the linking of worms and virus-distribution methods to attack tools is growing. Distributed Denial of Service tools, sniffers, scanners, and backdoors, are spread using worm techniques. All of these are being bundled into bots, which are remote-control tools that are used by attackers to control a vast number of systems. We've seen some botnets grow to over a million compromised hosts!

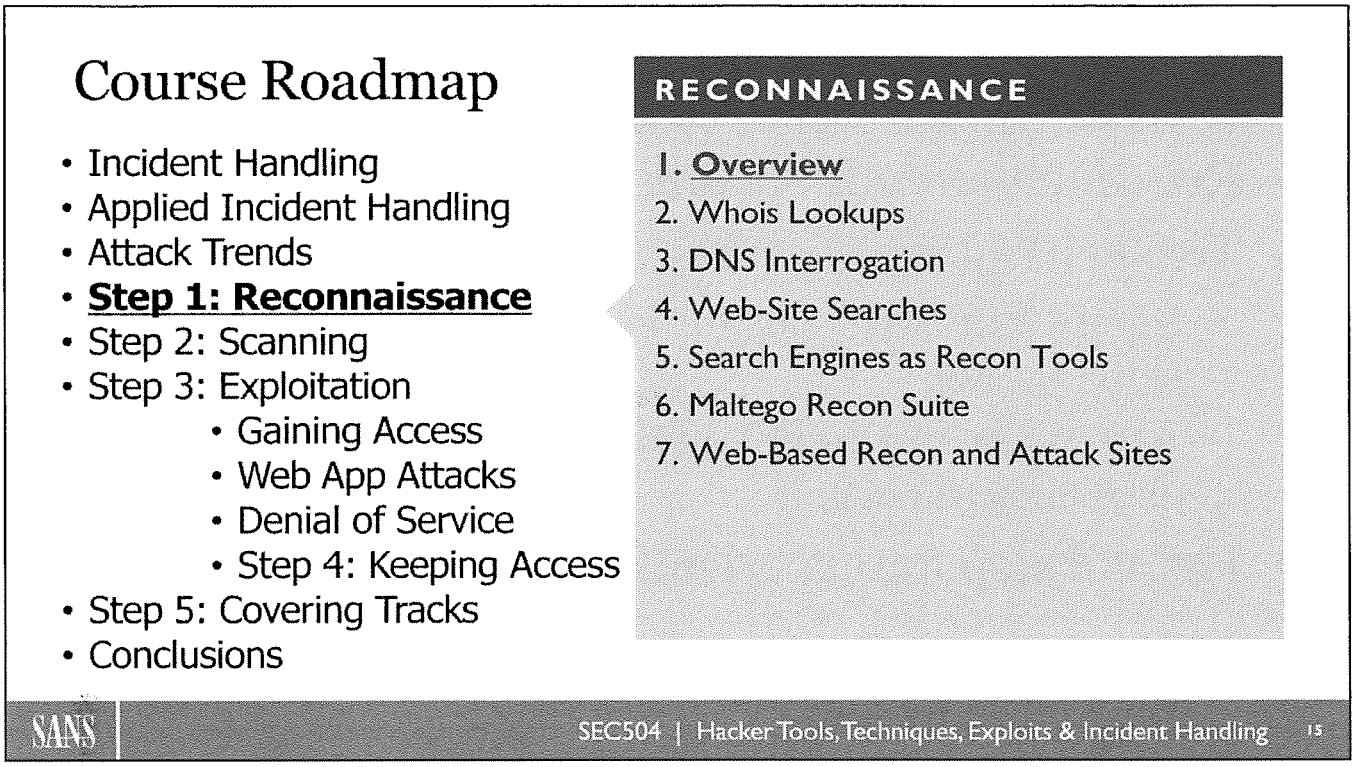
From the *Hacker Manifesto* (written by "The Mentor" on 1/8/86, available at http://en.wikipedia.org/wiki/Hacker_Manifesto):

"But did you, in your three-piece psychology and 1950s technobrain, ever take a look behind the eyes of the hacker? Did you ever wonder what made him tick, what forces shaped him, what may have molded him? I am a hacker, enter my world...

...This is our world now... the world of the electron and the switch, the beauty of the baud. We make use of a service already existing without paying for what could be dirt-cheap if it wasn't run by profiteering gluttons, and you call us criminals. We explore... and you call us criminals. We seek after knowledge... and you call us criminals. We exist without skin color, without nationality, without religious bias... and you call us criminals. You build atomic bombs, you wage wars, you murder, cheat, and lie to us and try to make us believe it's for our own good, yet we're the criminals.

Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for. I am a hacker, and this is my manifesto. You may stop this individual, but you can't stop us all... ."

The bottom line here is that we live in the Golden Age of Hacking. But, it's also the Golden Age of Information Security. The two go hand-in-hand.



The first step in most attacks is to gain as much information about the target as possible. With the wide variety of information sources available today, a great deal of useful data can be gathered.

Reconnaissance

- Reconnaissance is "casing the joint"
- Two general types of attackers
  - **Script kiddies**: Look for low-hanging fruit, and may skip this step
  - **Attackers out to get a particular site**: This step is extremely important
- Helpful step for experienced attackers

Detailed reconnaissance helps an attacker get a feel for your network before ever firing a packet in anger. The Internet itself is a treasure trove of information for a curious attacker.

To begin an attack, your adversaries gather as much information as possible from open sources. Think about attacks in the plain-old real world for a minute. (I know it's hard to think about non-virtual things… but occasionally, we must.) Before bandits rob a bank, they visit the particular branch, look at the times that the security guards enter and leave, and observe the location of security cameras. In addition, they may even use white pages to find the address of the bank and a map of the city to plan their get-away path. This is the same first step in cyberattacks.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- **Step 1: Reconnaissance**
- Step 2: Scanning
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

RECONNAISSANCE

1. Overview
2. **Whois Lookups**
3. DNS Interrogation
4. Web-Site Searches
5. Search Engines as Recon Tools
6. Maltego Recon Suite
7. Web-Based Recon and Attack Sites

One of the best sets of tools to use for reconnaissance is the various whois databases on the Internet.

Domain Name Registration — Whois Lookups

- When registering a domain name, the registrar requests
  - Postal addresses
  - Phone numbers
  - Names of points of contact
  - Authoritative domain name servers
- This information can be used in an attack
  - Social engineering, war dialing, war driving, scanning

Attackers rifle through domain name information for useful tidbits in attacking a target. Every time you register a domain name (such as sans.org), you provide detailed contact information about your organization, including the name, postal address, phone number, and e-mail address for your technical, billing, and administrative contact. In addition, your registration information includes the IP addresses of your authoritative domain name servers.

All this information can be used to attack your organization. A bad guy could use this registration information as follows:

- **Contact names**: Social engineering, duping users via the telephone into giving up useful information
- **Telephone numbers**: War dialing, finding unsecure modems to infiltrate an internal network
- **Postal addresses**: War driving, finding unsecure wireless access points to attack
- **IP addresses**: Scanning, looking for openings in the target

Whois Research — Whois Lookups

- Whois databases contain a treasure trove of information
  - Many can be accessed via the web
  - Alternatively, use the "whois" command built into many UNIX implementations
  - Used to gather contact names, DNS information, and other data
- First, look up the target at InterNIC to determine the registrar
  - www.internic.net/whois.html
  - Operated by Internet Corporation for Assigned Names and Numbers (ICANN)
- Then, go to registrar's whois database to get detailed records
  - For example, http://www.networksolutions.com/whois/index.jsp

Whois databases are distributed throughout the Internet. When conducting reconnaissance using whois lookups, the first stop is often InterNIC, the Internet Network Information Center, a website that is currently operated and maintained by the Internet Corporation for Assigned Names and Numbers (ICANN). This high-level database doesn't have many detailed registration records. It does, however, contain information about the particular registrar used for .aero, .arpa, .biz, .com, .coop, .edu, .info, .int, .museum, .net, and .org.

A researcher first looks up a target in InterNIC to determine which registrar the target used. Then, the researcher goes to the whois database of that particular registrar to get detailed registration information. Network Solutions is still a dominant player, but there are hundreds of other registrars, such as GoDaddy.com.

Whois Research for IP Blocks

Whois Lookups

- Attackers look for IP address assignments in these geographic whois databases:
  - ARIN (American Registry for Internet Numbers)
    - www.arin.net
  - RIPE NCC (Reseaux IP Europeens Network Coordination Centre)
    - www.ripe.net
  - APNIC (Asia Pacific Network Information Centre)
    - www.apnic.net
  - LACNIC (Latin American and Caribbean NIC)
    - http://lacnic.net
  - AFRINIC (Africa's NIC)
    - www.afrinic.net
  - DoDNIC (Department of Defense NIC)
    - http://www.nic.mil (requires account and certificate)
- Another useful site to check out for Whois information
  - www.uwhois.com with over 246 countries

In addition to getting the registration information (name, phone number, e-mail address for various contacts and DNS information), an attacker may also want to see if any IP address blocks are assigned to the target. If the target organization has been allocated a block of addresses, the attacker can discover the assignment by using the whois databases maintained by ARIN, RIPE NCC, APNIC, LACNIC, and DoDNIC. The last one of these, used by the U.S. Military, requires an account and a client-side certificate to access.

Many organizations don't have their own IP address allocation. Instead, they get addresses from their ISPs. In such cases, ARIN, RIPE NCC, APNIC, LACNIC, AFRINIC, and DoDNIC reveal little information about the target.

Another useful source of whois information is uwhois.com. It supports domain name registration lookups for more than 246 countries.

Sample Whois

- The final element of the whois record includes DNS information

Registrant:
Name/Address/Phone number

Administrative Contact:
Name/Address/Phone number

Technical Contact:
Name/Address/Phone number

NameRecord expires on 22-Jun-2015.
Record created on 22-Jun-2013.
Database last updated on 22-Jun-2013 12:04:21 EDT.

Domain servers in listed order:

Name.server.1 10.10.10.45
Name.server.2 10.10.10.99

Attacker can now query those DNS servers to get more target info

This slide shows a detailed registration record. Note that it includes the IP addresses of the authoritative DNS servers associated with the target at the end of the record. This DNS information is useful in the next phase of the attack.

Whois Recon Defenses

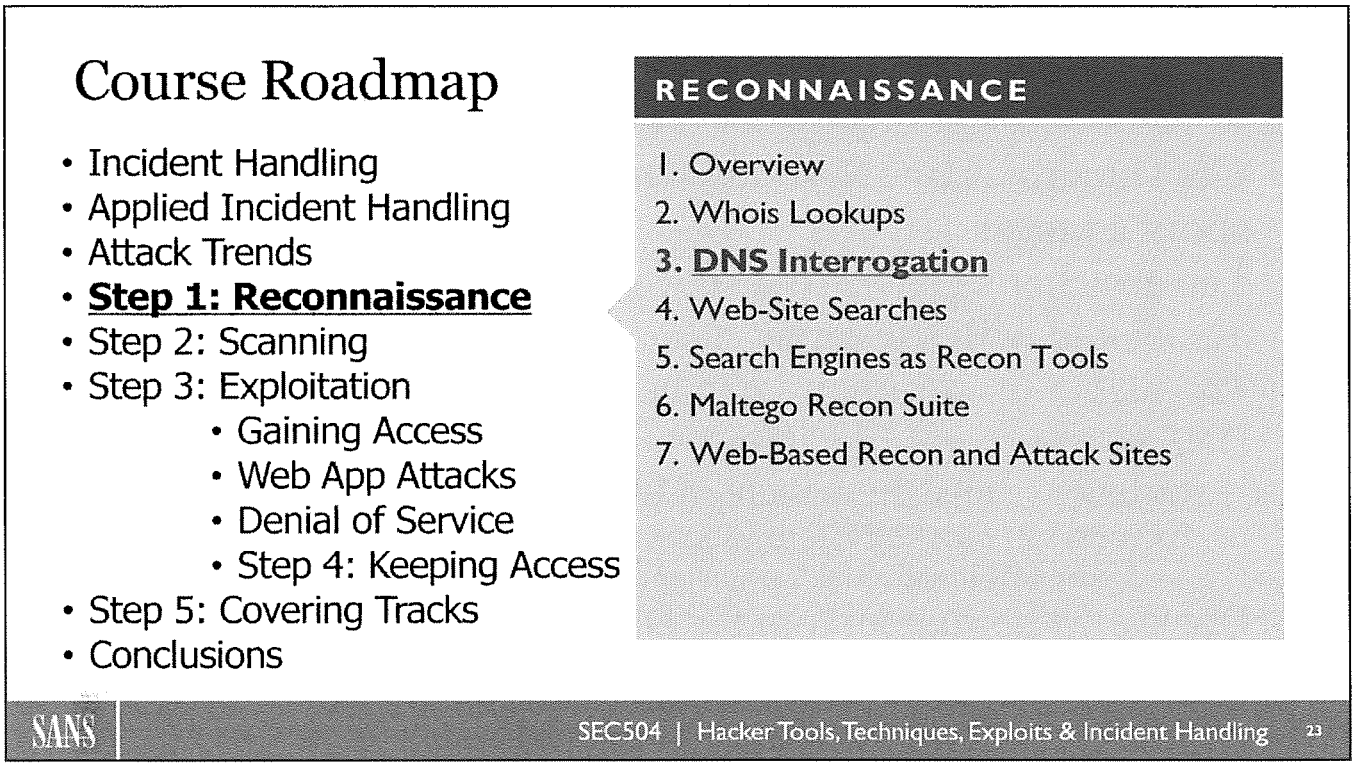
Whois Lookups

- Preparation
  - Just live with it: That's just the way the Internet is
  - Use organization name or title with real e-mail and phone number
    - Be CAREFUL with this! Make sure a real, live, responsive human answers
  - Keep records up to date
  - Be wary of anonymous registration agents
    - Network Solutions and Go Daddy offer private registrations for an extra U.S. $9.99 per year
      - https://www.networksolutions.com/domain-name-registration/private.jsp
      - http://www.godaddy.com/domainaddon/private-registration.aspx
    - Incident handlers depend on being able to use whois info to contact each other
- Identification:
  - You can't actually tell someone has looked you up
- Cont, Erad, Recov: N/A

How do you defend against this style of reconnaissance? Most organizations just live with it, accepting that detailed information loaded in whois databases is part of Internet life. Others choose to replace particular people's names with an organization name in their registration records. This procedure may limit the value of the information to a social engineer. However, make sure that you use a real organization name where someone responds to any requests that come in from the people using whois database information.

If you use a fake name or an unresponsive organization number in your registration data, no one can contact you in the event of an emergency. If your site starts attacking mine, I contact you through your whois entry information. If that data is out of date or bogus, we cannot coordinate and stop the attackers.

Also, be wary of anonymous registration agents. These companies allow you to register through them, and enter their contact info into a whois database instead of your own contact data. Network Solutions and Go Daddy offer such a service for an additional US$9.99 per year plus the registration fee itself. Incident handlers depend on being able to use whois info to contact each other quickly. If you go through a registration agent that doesn't reveal your information, you slow down how quickly other incident handlers can reach you. That gives bad guys more time to attack us, sadly.



The attacker now has some useful information about the target. The last element obtained by the attacker included DNS server IP addresses for the target. The attacker can now try to harvest information from these DNS servers.

DNS and nslookup — DNS Interrogation

- The Domain Name System is full of useful information about a target
- The attacker's goal is to discover as many IP addresses associated with the target domain as possible
- The nslookup command can be used to interact with a DNS server to get this data
  - Included in modern versions of Windows
  - Included in most UNIX implementations
    - Deprecated in some UNIXes and limited on some Linux variants
- Dig is another useful tool for DNS recon

Nslookup is a program that can be used to interrogate DNS servers.

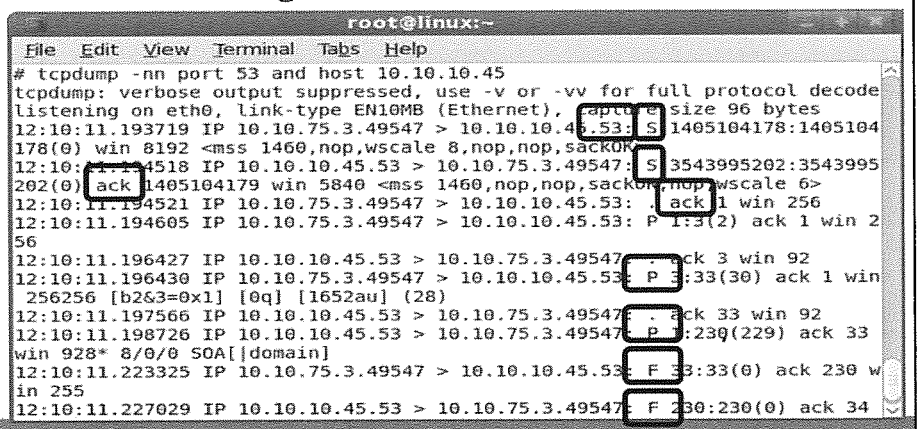
The nslookup command works in both modern Windows systems and UNIX. However, in UNIX, nslookup is being deprecated. If you run it, it may bark at you, telling you to use dig or host. In the latest versions of Linux nslookup, the command has been stripped so that it cannot perform zone transfers, which is a useful technique for getting a lot of information about a target domain. If your Linux nslookup gives you an error message saying that zone transfers aren't supported, use dig on Linux.

DNS Zone Transfer in Windows

DNS Interrogation

- By dumping records from your DNS servers, attackers can determine which machines are accessible on Internet
- Using nslookup, information can be gathered
- Type

```
C:\> nslookup
> server [DNSServer]
> set type=any
> ls -d [domain]
```



A zone transfer allows an attacker to connect with your DNS server and grab all records associated with a particular domain. Essentially, zone transfers let an attacker grab a dump of your DNS server's brain.

By dumping all records from your DNS servers using zone transfers, an attacker can determine which machines are accessible on the Internet. Using the UNIX `nslookup` command, a great deal of information can be gathered. On Windows, simply type the following commands to perform a zone transfer:

```
C:\> nslookup
> server [authoritative_server_IP_or_name]
> set type=any
> ls -d [target_domain]
```

The "set type=any" directive means that we want any type of DNS record, including Address (A) records, Mail eXchanger (MX) records, Host Info (HINFO) records, and nameserver (NS) records. Remember to run these commands against the primary, secondary, and any other domain name servers associated with the target organization.

In the slide's screenshot, we show a packet capture of a zone transfer using tcpdump. We invoked tcpdump with the –nn option, with the first n indicating that we don't want to resolve names (we want IP addresses), and the second n indicating that we want port numbers (instead of looking up related service names in /etc/services). We use a filter of "port 53 and host 10.10.10.45" to indicate that we want packets associated with port 53 that are going to or from the IP address of the DNS server, 10.10.10.45. In the output of tcpdump, we see a series of packets going from 10.10.75.3, a client machine running nslookup, to 10.10.10.45, on TCP port 53. The machines complete the TCP three-way handshake, with the first packet being a SYN (the S on the screen shows this) from 10.10.75.3 to 10.10.10.45. The second packet is a SYN-ACK response (note the S and the ack indication). The machines complete the handshake with another ACK, also circled above. Then, through a series of PUSH (indicated by P), the zone information is exchanged. Finally, the FINs (shown with F) illustrate the tear-down of the connection.

DNS Zone Transfer in UNIX

- On some UNIX variations, nslookup can be used for zone transfers
  - Using the same technique used for Windows on previous slide
- Other nslookup variations (including the one for recent versions of Linux) do not support zone transfer
- Use dig instead

```
$ dig @[DNS_server_IP] [target_domain] -t AXFR
```


On some versions of UNIX, you can use nslookup with the same syntax as Windows to do a zone transfer, just like we saw on the previous slide.

However, on recent versions of Linux, nslookup cannot do a zone transfer. Instead, we can use dig to achieve the same goal. To run a zone transfer using dig, type the following at the command prompt:

```
$ dig @[DNS_server_IP] [target_domain] -t AXFR
```

DNS Recon Defenses | DNS Interrogation

- Preparation
  - Do not allow zone transfers from just any system
    - Limit zone transfers so primary DNS server accepts zone requests to be initiated only by secondary and tertiary DNS servers... no one else
    - Secondary and tertiary accept zone transfers initiated by no one
  - Use split DNS
    - External name information in external server
    - Internal name information in internal servers
  - Make sure your DNS servers are hardened
    - All internal and external DNS servers
- Identification
  - Look for zone transfers (in DNS server logs or data transferred to/from TCP port 53)
- Cont, Erad, Recov: N/A

To defend against DNS-style reconnaissance, make sure you limit zone transfers. Your primary DNS server should allow zone transfers to be initiated by your secondary and tertiary DNS servers only. These servers, in turn, should be configured to deny all zone transfer requests.

In addition, use split DNS. With such an implementation, you have two components of your DNS infrastructure: external DNS servers and internal DNS servers. Publicly available DNS information is loaded on your external DNS servers. Internal names are loaded only on internal DNS servers.

Also, make sure your DNS servers are hardened. They are among the most sensitive components of your infrastructure from a security perspective. As we see over the next two days, an attacker who undermines DNS can redirect traffic throughout the Internet, completely compromising your network.

To identify zone transfers, look for packets going to and from TCP port 53 on your DNS servers. Normal DNS queries and responses use UDP port 53. Zone transfers use TCP port 53, a tell-tale sign.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- **Step 1: Reconnaissance**
- Step 2: Scanning
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

RECONNAISSANCE

1. Overview
2. Whois Lookups
3. DNS Interrogation
4. **Website Searches**
5. Search Engines as Recon Tools
6. Maltego Recon Suite
7. Web-Based Recon and Attack Sites

After getting information from whois databases and DNS servers, attackers also interrogate your own web servers.

Website Searches

- Search the target's own websites
  - Press releases
  - White papers
  - Design documents
  - Sample deliverables
  - Open positions
  - Key people
  - Contacts

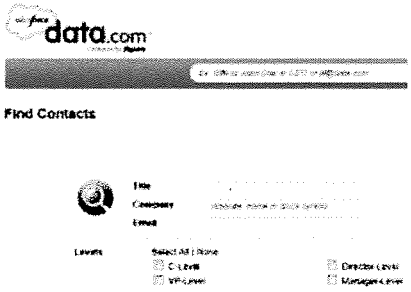
  } **Especially useful for attackers**
- Search related sites
  - Business partners, ISP, suppliers

Corporate websites often contain contact information with phone numbers, which are useful for war dialing and social engineering. Some sites even include a description of their computing platforms and/or architecture. Attackers grab a copy of your entire website to look for juicy tidbits about your organization.

Search engines are also useful. By searching for information about a target, an attacker can often learn about their platforms and architecture through UseNet postings of employees. Also, websites can indicate business partners and other potential links useful in spoofing and other attacks. Modern search engines (such as Google and Bing) include the ability to search for sites linking to the target. Simply search "link:www.[target_company].com" for all sites that link to the target.

Other Open-Source Information | Website Searches

- Public databases
  - SEC's Edgar database for publicly traded U.S. companies
    - http://www.sec.gov/edgar.shtml
  - Job sites (such as monster.com)
  - www.pipl.com and
  - https://connect.data.com
  - www.namechk.com
  - Hacker sites
- Other open-source information
  - Newspapers blogs, and magazines
  - Social networking sites (what expertise, which friends/associates)
    - Facebook, Twitter, Orkut, LinkedIn
    - What are people sharing about your organization
  - Newsgroups with postings from employees



Job-site databases can be particularly interesting. If a company is looking to hire a Checkpoint FireWall-1 administrator, what type of firewall do you think it has?

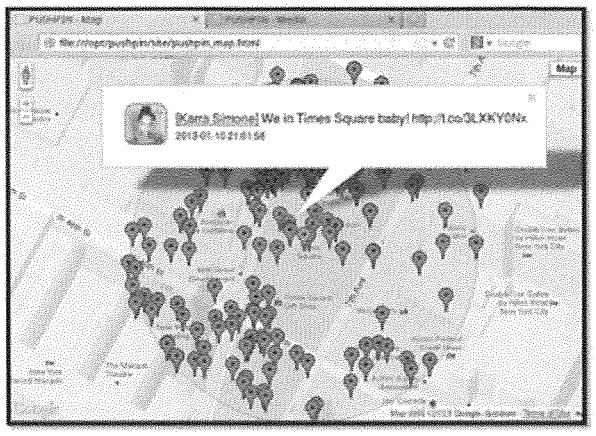
In addition, various other open-source information is available. Newspapers, magazines, blogs, social-networking sites, newsgroups, and other sites could provide just the information required by an attacker to launch a more focused attack against your organization.

We can also use sites such as namechk to identify which social-networking sites a target user account may be using. Currently, namechk checks more than 100 social-networking sites to see if a given account is in use. This can be used by an attacker to develop Social Engineering (SE) pretexts. For example, if a target user receives an e-mail stating his account with last.fm is about to expire, he is far more likely to click the link if he actually has an account with that site or service.

This can also be helpful for an attacker to identify which users are more susceptible to SE attacks. Generally speaking, users who are more active online are easier targets because they have a greater predisposition toward clicking links and interacting with strangers.

Pushpin — Website Searches

- Pushpin by Tim Tomes
  - Part of Recon-ng
- Social-media geolocation
  - Flicker
  - Twitter
  - Picasa
- Simply provide a latitude, longitude, and radius (in kilometers) and pushpin pulls all available social-media posts from that area
- Can map targets to behavior patterns
  - When and where they have lunch
  - Their religious and political leanings
- It can even be used to gather internal pictures of secured locations
  - People love to take pictures of their office and badges



One of the bigger struggles for many attackers is trying to tie together physical locations with cyber profiles. Pushpin, by Tim Tomes, addresses this issue by pulling all Flicker, Twitter and Picasa posts from a specific location and a radius.

This can be used in a number of different attack situations. For example, many users tend to take their work computer with them to get coffee or to lunch. When they do this they tend to use whatever free wireless is available and pushpin can find their location. With this data, an attacker can use a number of wireless attacks when the user is not protected by their organization's security support structure.

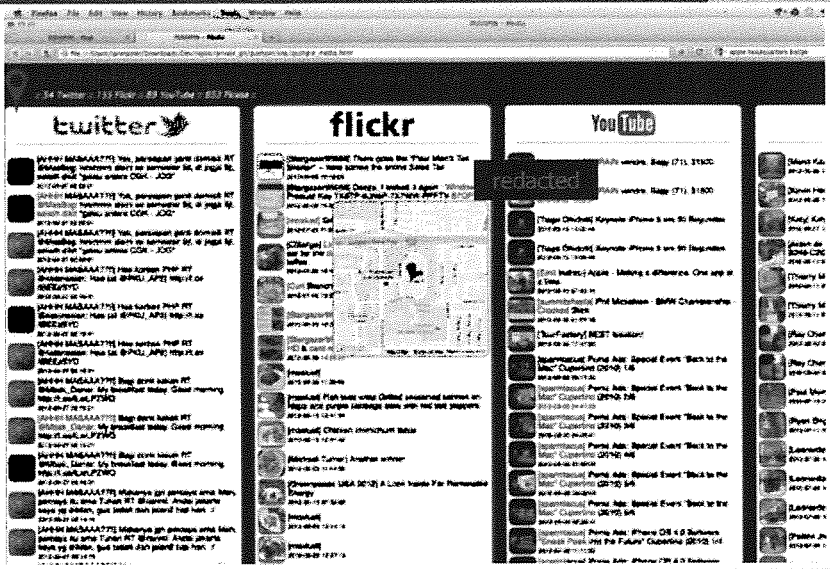
Another oddity is how often people take geotagged pictures with their phones at work. We have found pictures of Network Operations Centers (NOCs), offices, and egress/ingress points. Finally, pushpin is a great way to get pictures of people's badges so an attacker can clone them for access.

When pushpin runs, it provides two sets of data. The first (shown above) is a map of a location with all of the posts located on it . When you click any of the "pins," it shows you the social media post, picture, or video.

Note that the timeframe in which the data is pulled varies wildly from provider to provider and location to location.

Pushpin Output

Website Searches



The second pushpin output is a full listing of each social-media post. If you hover over any one of the posts, it shows you where on the map that particular post was made.

This has been used in numerous assessments to discover access points and even people bringing their phones into secure (and sometimes even classified) areas, such as military bases.

Note the one post that was redacted above. This post originated from Apple headquarters and was a full Windows 7 product key.

The best way to use this tool is to regularly run it for your sensitive locations. Always be on the lookout for users who frequently post social-media posts and pictures from work. These people tend to be easy targets, because they regularly interact with numerous people online (many of whom they don't know personally).

Website Searches Defenses

Website Searches

- Preparation
  - Limit and control information
  - Know what information a company is giving away and perform risk analysis
  - Make employment ads more general, if HR lets you
  - Limit information on a website
  - Determine what other sites are linked to your company
- Identification
  - Look for web spider/crawler activity
    - Logs show systematic access of entire website, page by page
    - That could simply be the Google bot or another search engine
  - Someone just sucked down the entire contents of our site
- Cont, Erad, Recov: N/A

You should periodically check various open sources of information to see what your company is leaking. This analysis can be done by the security organization, legal department, and public relations, because all have a vested stake in protecting your corporate information.

For identification, have your web administrators look through their logs for an indication that someone has used a web spider (also known as a web crawler) to access every page on your site in a short period of time (say, within 5 minutes). Most likely, this activity is just the crawler of a search engine (like the Google bot). But, from another source, it could be a sign of pre-attack recon.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- **Step 1: Reconnaissance**
- Step 2: Scanning
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

RECONNAISSANCE

1. Overview
2. Whois Lookups
3. DNS Interrogation
4. Web-Site Searches
5. **Search Engines as Recon Tools**
6. Maltego Recon Suite
7. Web-Based Recon and Attack Sites

Beyond trawling the web generally, attackers are increasingly utilizing the single most popular source of info on and about the web: Google. Let's see how.

Reconnaissance with Search Engines

- The easiest way to get information is to ask for it
  - Ask someone (or something) that has a lot of information
  - Like Google, Bing, Biadu, and Yahoo!
- Great resources on this topic
  - The Exploit Database GHDB page, the current home of the GHDB http://www.exploit-db.com/google-dorks
  - Many of the listed search directives work on other search engines
  - Based on original work by Johnny Long

Increasingly, search engines are the resources of choice for detailed recon activities. Let's explore some search-engine features that are especially useful in conducting reconnaissance during a computer attack.

There are great resources on this topic, including the Google Hacking Database (GHDB) with more than 1,000 different useful searches to locate many problems on target domains. The current home of the GHDB is at the Exploit Database website (http://www.exploit-db.com/google-dorks).

Interestingly enough, many of the search directives used by Google also work with other search engines. Another cool fact is that the results you may get may differ from one search engine to another. This is because many search engines index and store data differently. Because of these discrepancies, it is important to perform search-engine recon through multiple different search engines.

Google Maps for Planning Physical Access for Computer Attacks

- Check out maps.google.com for satellite imagery
  - Sometimes used by attackers planning physical attacks, or coordinating physical and computer attacks
- Google Street View shows even more detail
  - Available for some geographic locations, but not everywhere
- This map API can be used with multiple third-party applications

Google added satellite imagery maps, accessible via maps.google.com. Look up an address, and click the Satellite link. You have a scrollable and zoomable photographic image of the area you just mapped. It's useful and a bit scary. Some attackers use these features when planning physical attacks against a target organization, scoping out roads, walkways, and doors associated with target buildings.

For even more detail, Google supports its Street View option for some geographic locations, showing a view of a person on a street, with the ability to pivot, turning around to see detail in a 360-degree radius.

Later, we see how the Google Maps API can be used to tie together data from multiple sources to give a great physical view of a specific location.

Useful Search Directives

Search Engine Recon

- "site:" directive
  - Searches only within the given domain
  - site:www.counterhack.net
- "link:" directive
  - Shows all sites linked to a given site
  - link:www.counterhack.net
- "intitle:" directive
  - Shows pages whose title matches the search criteria
- "inurl:" directive
  - Shows pages whose URL matches the search criteria
- "related:" directive
  - Shows similar pages (sometimes useful, sometimes not)
- "info:" directive
  - Finds cached page, related pages, pages that link to it, pages that contain the term (NOT USEFUL)

Many search engines offer some useful directives for the Reconnaissance phase.

The "site:" directive allows an attacker to search for pages on just a single site or domain, narrowing down and focusing the search. For example, if you only want to search pages in the counterhack.net domain for the occurrence of the string "wireless," you could search "wireless site:counterhack.net." In essence, this type of search lets you target the recon of only specific sites.

The "link:" directive is also useful. It shows you everyone that links to a given website. During recon, this directive can be used to find business partners, suppliers, and customers. To look for everything linking to www.counterhack.net, search "link:www.counterhack.net."

The "intitle:" directive is helpful because it shows pages whose title matches the search criteria. I also use the "inurl:" directive a lot, which shows pages whose URL matches the search criteria.

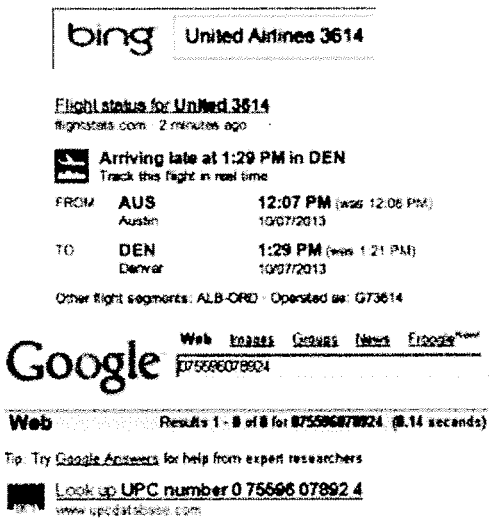
The "related:" directive shows pages that have similar content and links to the searched page. It's not extremely useful, because it often returns fairly unrelated items.

The "info:" directive isn't very useful at all, as it returns a bunch of data, including results from "link:" and "related:" searches, as well as cached pages. I prefer to perform each of these different searches by themselves, to get maximum value for my search results.

Additional Search Tips and Types

Search Engine Recon

- Surround literals with " ", as in "Soc Sec Num"
- Add minus (-) to a search term to maximize effectiveness of resulting hits
  - Excludes pages with a given word
  - For example, search on: site:sans.org (www.sans.org)
- Search for airline status
  - Type in airline and flight number
  - Front end for Travelocity search for VIN for vehicle information
- Search for UPC number for product info



In addition to the search directives listed on the previous page, keep these other useful features in mind.

First, you can search for a literal string using double quotes. For example, to search for the phrase social-security number, with those exact words in that exact order, search for "social security number." However, remember that Google is always case insensitive, even with the double quotes.

In addition, you can add "-" before a word in a search to find results that do not include that given word. This is immensely helpful in narrowing down a search and maximizing the value of the 1,000 results that Google will give you. For example, you could search for site:sans.org –www.sans.org to find domain names of systems associated with sans.org other than www.sans.org. It shows you some other machines associated with SANS.

Finally, some cute and useful search possibilities include looking up an airline and flight number. Google responds with the current status of the flight and its location in the air! Also, searching for VIN shows motor-vehicle information, and a UPC number shows product information. That can be helpful research! In essence, Google is acting as a front-end search engine for Travelocity (and fboweb) for flight tracking, CarFax (for VINs), and UPCDatabase.com (for product info).

Google's Cache and Wayback

Search Engine Recon

- Search on: "cache:www.counterhack.net"
- Brings up the cached version of the page
  - Can be useful for attackers to pull information that was removed from a website (perhaps by an incident response team)
  - Useful for bad guys if IR containment isn't thorough
- Browse the Google cache
  - HTML is loaded from Google
  - Any images on site are loaded from original site (NOT Google's cache)
  - Also, any links browsed take you to the real site
  - Not a good approach for anonymous surfing
  - Still, it's useful for finding recently removed pages
- The Wayback Machine (www.archive.org) is a thorough view, with multiple images over time
  - Lets you interactively surf the cached pages
  - Images still come from the original site (if they are still there)

The attackers don't even have to go to the target site itself. Instead, by using the "cache:" directive, they can simply browse Google's cached pages. As it crawls a website, Google grabs the first 101k of HTML from each web page and stores it in its cache. (Note that the Google cache only stores HTML.) Any images referred to in the web page are loaded from the original site itself. In addition, if you click any links on the Google cache page, the linked-to pages are loaded from the original site. Because of this, the Google cache is not a useful way to anonymously surf the Internet. Instead, the Google cache is useful for finding recently removed pages and limiting the target site's knowledge of what you are doing.

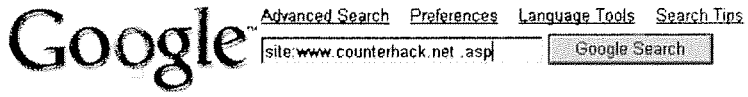
Pulling information from Google's cache (and other caches on the Internet) is particularly useful for retrieving pages recently removed from a website. For example, an incident-response team may discover some sensitive information leakage through a website. If it removes that page from the site itself, but fails to remove it from the Google cache, attackers can still retrieve the page from the cache. If the incident-response team is not careful about performing containment of the information leakage, the results could be damaging to the organization.

Beyond Google, the Wayback Machine located at www.archive.org has more thorough archives, which also include old views of various websites. This site features cached pages from billions of web pages for the last several years, including multiple views over time of each site. More popular sites have more frequent snapshots in the archive, with some sites' views featured once per quarter, some once per month, a few once per week, and some even on a daily basis. Images not located on the current site are loaded from the archive cache. However, if the images are still on the original site, they are loaded from that site.

Searching for File Types

Search Engine Recon

- Search for specific file types on a target domain
- Look for active content: .asp, .jsp, .php, or .cgi
- Excel spreadsheets: Search for .xls and view it as HTML...
  - Spreadsheet image comes from Google cache
- .ppt can also be useful
- For example, search for
  - site:www.[target].com asp
- "filetype:" is useful, but also try just the suffix
- "filetype:" is the same as "ext:"



We've seen how to search for links, info, and cached information, all associated with a specific site. But, for what will the attackers actually search? Here's where search engine recon shines.

Whenever I do search-engine reconnaissance, I use the "site:" directive and look for the following file types:

- asp, jsp, php, cgi, and others: These types of files indicate active web content and may be vulnerable
- xls and ppt: Organizations sometimes don't even realize that they've left an Excel spreadsheet or PowerPoint presentation on their websites
- Other miscellaneous file types suited to that target

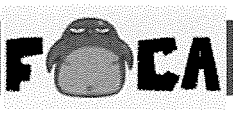
For example, to search the site www.counterhack.net for Active Server Pages, an attacker could perform a Google search on site:www.counterhack.net asp.

Note that I do not use the "filetype:" directive in my searches, although Google supports it in finding doc, pdf, ps, xls, rtf, txt, ppt files, and countless other file types. I find that, sometimes, Google doesn't properly categorize a file. That's why I use the suffix (doc, pdf, ps, xls, rtf, txt, ppt, and more) as a search directive. Then, after searching for the suffix as a search term, I usually also do a "filetype:" search. It works far better for me to do both kinds of searches (for the suffix as a search term and for the suffix as part of the filetype: directive). Note that the filetype directive is synonymous with the ext directive. They do exactly the same thing.

This is also why it is important to use different search engines. For example, some search engines won't allow you to search for the @ as part of an e-mail search. Rather, they simply replace the @ with a wildcard. However, other search engines, such as Biadu, allow it.

Note that this is more of an art than a science. Search-engine providers are constantly tweaking the way they handle searches and provide results.

FOCA



Search Engine Recon

- Many files (.doc, .xls, .pdf) have metadata that can be useful for attackers
  - Usernames, vulnerable versions of software, directory paths
- Searching for different file types by hand is time consuming
- FOCA automates this process by searching for various files, downloading them, and extracting their metadata
- In addition to metadata extraction, it has basic Google Hacking Database and basic web-vulnerability scanning
- Integrates with Shodan and Robtext to identify network ranges and additional targets
- Can perform subdirectory brute-forcing to identify additional hosts
- tps://www.elevenpaths.com/labstools/foca/index.html

One of the best tools to easily identify which files are being hosted on your site is FOCA by ElevenPaths. As we mentioned on the previous slide, searching for different file types can provide a tremendous amount of useful data for attackers. Usernames, versions of vulnerable software, and directory paths are all great pieces of recon that an attacker can use in later targeted attacks.

FOCA automates the process of discovering these files, downloading them, and extracting the metadata from the files. In addition to its excellent automated metadata support, it also has some helpful vulnerability discovery modules. For example, it can incorporate the Google Hacking Database for Google searches, it can search for basic web vulnerabilities (for example, directory indexing and basic SQLi), and it can interface with Shodan and Robtext to identify network ranges and additional systems.

Finally, it has a basic (yet effective) subdomain directory brute-forcing module that can be used to enumerate additional exposed servers and services on the Internet.

It can be found for free at http://www.informatica64.com/foca.aspx.

Searching for Vulnerable Systems

Search Engine Recon

- We can perform various searches associated with commonly exploited systems
  - Available remote desktop systems: ext:rdp rdp
  - Default web material (Apache, IIS, ColdFusion, and others)
    - "Test Page for the Apache Web Server"
    - "Welcome to Windows 2000 Internet Services"
  - Web-based FileMaker Pro databases: "Select a database to view"
    - Make sure to use quotes
  - Indexable directories: intitle:index.of "parent directory"
  - User IDs and passwords (look for "password" and "userid")
  - Shell history (look for common shell names and commands)
  - Video cameras (example: search for inurl:"ViewerFrame?Mode=")
- FOCA has the ability to identify many of these vulnerabilities

An attacker can also do searches for potentially vulnerable systems directly. One of the most startling is doing this search: ext:rdp rdp. This turns up systems that can be remotely managed via Windows Remote Desktop Protocol. Also, if I search for text from the default Apache install and your site appears, I know that you are likely running the Apache web server. The same logic applies to IIS. Likewise, I might be able to determine that you built your website using ColdFusion or another development platform by looking for text associated with default material on those platforms.

Searches can help find HTTP-accessible File Maker Pro databases. By searching for "Select a database to view," an attacker gets a list of Internet-accessible databases. Although some are password protected, many aren't. Indexable directories that someone has left on a website are also useful and can be discovered by searching for: intitle:index.of "parent directory."

Attackers can also look for command shell history, and even hidden hyperlinks and indexes that aren't easily accessible by humans. The attackers just let Google bot work its magic. This technique for finding vulnerable systems has becomes so widely used that, starting in December 2004 with Santy, worms use Google to locate vulnerable systems and spread. Santy searched Google for a vulnerable version of the phpBB script and then attacked systems running it. Because of this, Google is now filtering some of the common PHP and related searches conducted by worms. Still, new searches for vulnerable systems are discovered all the time.

Another set of fascinating searches involves finding web-accessible video cameras. The search inurl:"ViewerFrame?Mode=" shows numerous Panasonic cameras around the world, some of which allow you to control zoom, tilt, and pan. Likewise, numerous other searches find other types of cameras and their associated web controls.

Automated Search Engine Recon

- Bishop Fox's SearchDiggity is a fantastic suite that includes Google Diggity, Bing Diggity, and other search capabilities
  - Malware Diggity, Data Loss Prevention Diggity, Flash Diggity
  - Many of these "diggity" components require an API for the respective service
  - Sometimes, free APIs provide fewer results than the web interface
- Recon-ng, by Tim Tomes, is another powerful recon tool
  - Ties together numerous different recon sources into one framework
  - Currently more than 60 different recon modules
  - Most modules are free, some require a third-party API key
  - Workspace and reporting capabilities to keep projects separate
  - Some modules can tell if any target organization has been compromised via third-party sites
  - Uses the web interface for many sites to scrape results (be careful, doing this may violate terms of service)
- Determined attackers use these tools to gain access to target environments without even using an exploit

Bishop Fox's SearchDiggity is one of the finest tools available for tying together all the different search engine techniques we have discussed thus far. A wide number of different modules are available for performing searches with Google, Bing, and Shodan in one framework.

In addition to the modules for pulling data from search engines, there are also modules available for searching a site to see if it is hosting malware. Another, called DLP Diggity, can check for data leakage from an environment. Finally, there is a module that can decompile flash objects to see if any sensitive data (such as passwords) exists in the action script.

A number of modules require a free API key from the data provider in order for the queries to function properly. As an extra note of precaution, keep in mind that many of the API providers actually provide less data than the human or web-interface counterpart.

Another great web-search recon tool is Recon-ng by Tim Tomes. Although this framework has a number of excellent search engine components, it also has a number of additional modules that can query data from third-party data services. For example, it has the capability to hook into sites, such as pwnedlist.com and https://breachalarm.com/, to see if any target accounts have been compromised.

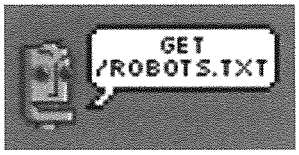
Recon-ng also has a number of workspace and reporting modules to keep the data separate and accessible from project to project.

A final word of caution: Recon-ng uses the human interface from many services and web search engines. Although this can provide better results, it also may violate the terms of service for the various data providers. Review the TOS of your data providers before using.

Search Engine Recon Defenses

Search Engine Recon

- Look for information leakage using Google yourself
- Instructions at http://www.google.com/remove.html
  - Remove the website (robots.txt file)
    - robots.txt is NOT a security feature; it must be world readable for the search engine crawlers to find it
    - It draws attention to files, and careful attackers are wise to plunder it for possibly interesting directories and files on a target website
    - Interesting place to refer to a honeypot web page, only referred to in robots.txt
    - Monitor all IP addresses that try to access page
  - Remove individual pages ("NOINDEX, NOFOLLOW" meta tag)
  - Remove snippets ("NOSNIPPET" meta tag)
  - Remove cached pages ("NOARCHIVE" meta tag)
  - Remove an image from Google's Image Search
- Remove unwanted items from Google
  - URL re-crawl request form (www.google.com/addurl.html)
- See http://www.robotstxt.org for info about non-Google crawlers



To defend against these cyber reconnaissance raids, check your own environment to see what information you are leaking. Check out what information you have publicly available on your own websites and think about what an attacker could do with that data. Use the Google tips we discussed on the previous slides for searches.

Also, if you find that Google has indexed a URL or cached a page that you didn't want it to, you can use the URL re-crawl request submission form at www.google.com/addurl.html. This removes the page the next time the Google bot crawls your website, which likely occurs within the next 24 hours.

You cannot just submit a URL to remove. If that were the case, someone could actually have you removed from Google without your knowing it! To get Google to remove you, you have to not only fill out the form, supplying it with your URL. You also have to alter the page on your own website, using a robots.txt file or a meta tag, to indicate that you want it removed. That way, when you fill out the form saying you want a page removed, Google automatically goes to that page to see if it has been altered to include the robot.txt file or removal meta tag. So, you have to coordinate with your web-site administrator to have pages removed from Google.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- **Step 1: Reconnaissance**
- Step 2: Scanning
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

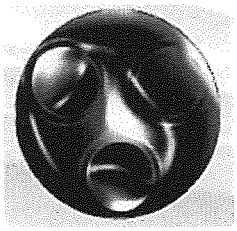
RECONNAISSANCE

1. Overview
2. Whois Lookups
3. DNS Interrogation
4. Web-Site Searches
5. Search Engines as Recon Tools
6. **Maltego Recon Suite**
7. Web-Based Recon and Attack Sites

Beyond whois, nslookup, and web search engines, a variety of tools offer convenient mechanisms for reconnaissance. One of the most powerful is Maltego, which is the next topic.

Maltego — Maltego Recon Suite

- Paterva's Maltego is an intelligence-gathering tool that searches through various public information sources
  - Gathers information about relationships between
    - People, social networks, companies, websites, domains, IP addresses, and more
  - Based on the concept of "transforms"
    - Converts one piece of information into another
  - Graphically displays relationships of information
    - Cascading hierarchies of data points mapping to other data points
  - Runs on Linux, Windows, and Mac OS X
- Available in two forms at http://www.paterva.com/web6/products/maltego.php
  - Commercial edition: Approximately US$760 per year
  - Community edition: Free, but with some limitations:
    - Nag screen for 15 seconds
    - Cannot save or export results
    - Zoom levels are limited
    - Only 75 transforms per day
    - Run transforms on only one entity at a time



The Maltego tool was released by a company named Paterva. This intelligence-gathering tool provides a wealth of information for researchers, investigators, and computer attackers conducting detailed reconnaissance. Maltego allows a user to start with one or more pieces of information, such as a person's name, phone number, domain name, e-mail address, web-site URL, IP address, and so on. Given that piece of information, Maltego applies the concepts of transforms, a series of lookups into public sources of information to find related pieces of information. Transforms convert one piece of information (such as a domain name) to another piece of information (such as an IP address). Many dozen transforms are included in Maltego.

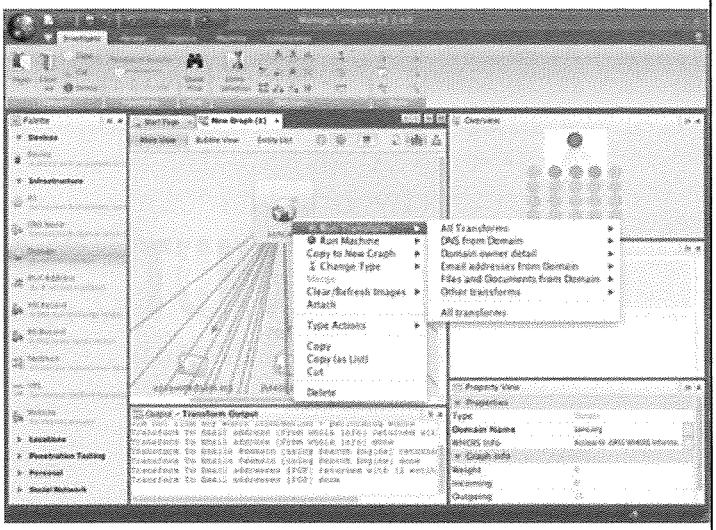
The result, when many transforms are applied repeatedly, is a cascading hierarchy of related information all associated in some way to the original data.

Maltego runs on Linux, Windows, and Mac OS X, and is available in two forms: a commercial edition for approximately US$430 per year, and a free Community Edition that has some limitations. The Community Edition allows for some powerful reconnaissance activities, but it includes a nag screen for 15 seconds asking you to purchase it every time it is launched. It also prevents saving or exporting results, limits the level of depth you can zoom to in the hierarchical display, and only allows 75 transforms to be applied per day. Also, you can only run transforms on one entity at a time, instead of being able to launch simultaneous look-ups on multiple pieces of data on the display.

Maltego Transforms

Maltego Recon Suite

- Some example transforms
  - DomainToPhone_Whois
  - DomainToMXrecord_DNS
  - DomainToPerson_PGP
  - IPAddrToPhone_Whois
  - PersonToPerson_PGP
  - EmailAddressToEmailAddr SignedPGP
- Commercial edition supports specialized transform servers and creating custom transforms



The slide's screenshot shows a starting point of a domain name. To create this, we simply drag and drop from the Palette on the right a domain name onto the main view. We enter a domain name of sans.org. We then right-click to apply a transform of domain name to e-mail address based on PGP keys available on public PGP keyservers. We receive over a dozen different results of e-mail addresses in the sans.org domain. We could then apply additional transforms of sans.org to get other information, or we could right-click on any of the e-mail address entities returned by our original transform, and then apply other transforms to it. Some example transforms that are available in Maltego include

DomainToPhone_Whois
DomainToMXrecord_DNS
DomainToPerson_PGP
IPAddrToPhone_Whois
PersonToPerson_PGP
EmailAddressToEmailAddr_SignedPGP

Note that the transform name includes the piece of information the transform must start with (such as Domain), the information it will look up up to transform it to (for example, Phone), and the mechanism it uses to make the transform work (such as Whois database lookups).

The commercial version of Maltego includes a subscription to various transform databases that Paterva operates, plus the ability to create your own transforms that go beyond those baked into the tool.

Defenses Against Maltego

Maltego Recon Suite

- Preparation
  - Ensure that publicly available information about your organization is accurate
    - Keep records up-to-date
  - Conduct your own recon
    - Check to see what is available about your organization and your important personnel
    - Request that inaccurate or damaging information be removed from sources
      - May be politically difficult or impossible to compel removal of some information
- Ident, Cont, Erad, Recov: N/A

To defend against Maltego, make sure that information about your organization in various public sources is accurate by keeping your whois and domain information up to date. You should also conduct reconnaissance about your own organization, with appropriate permission, to see whether the information available is accurate. If you find information that is inaccurate or damaging, you may want to work with your lawyers to formulate requests to have it updated or removed. Depending on the nature of the information, and the particular publicly available database in which it is located, you may or may not be able to compel someone to purge the data.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- **Step 1: Reconnaissance**
- Step 2: Scanning
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

RECONNAISSANCE

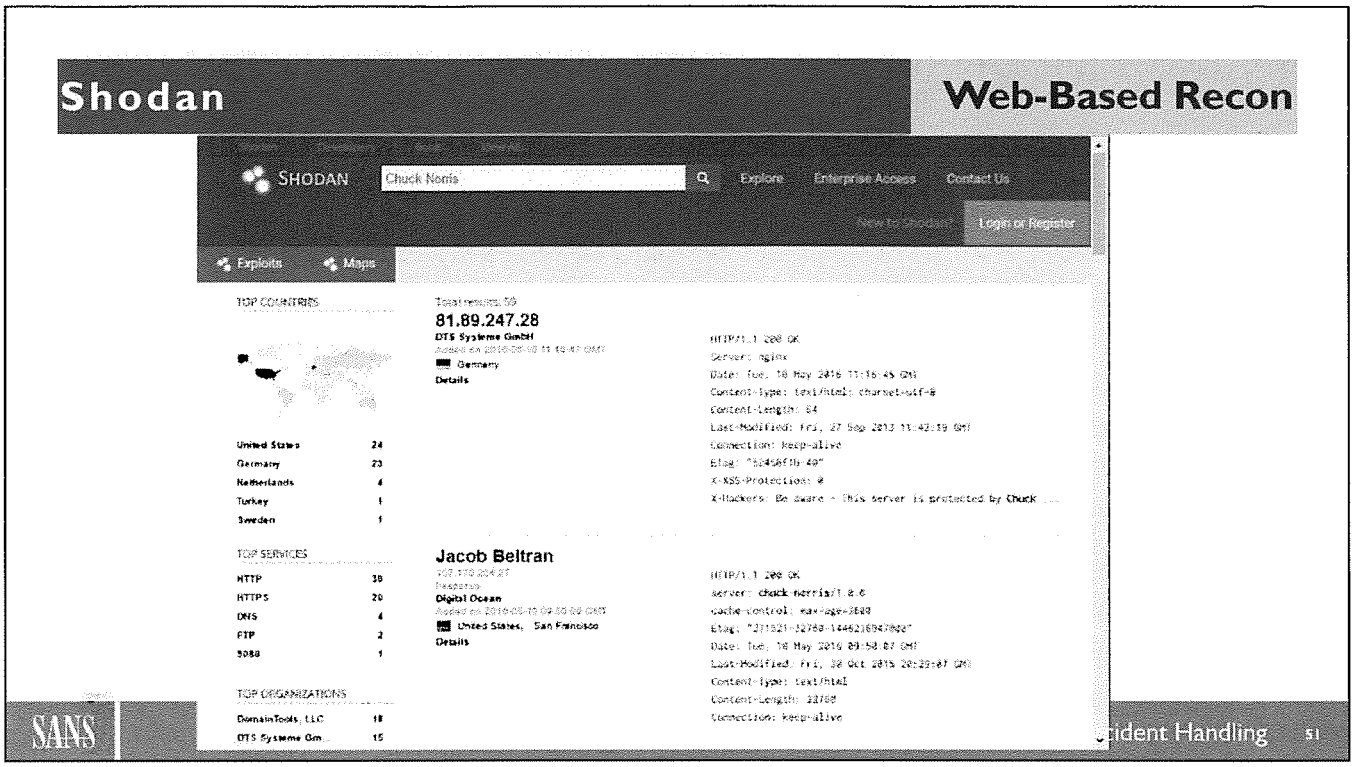
1. Overview
2. Whois Lookups
3. DNS Interrogation
4. Web-Site Searches
5. Search Engines as Recon Tools
6. Maltego Recon Suite
7. **Web-Based Recon and Attack Sites**

Reconnaissance tools are not limited to Maltego, however. There are numerous other web-based reconnaissance tools.

Web-Based Recon/Attack Tools

- Numerous websites offer the capability to research or even attack other sites
- Links to Internet Scanning Web Pages (Traceroute, ping, port scans, Denial of Service tests)
  - Shodan at www.shodanhq.com
  - www.dnsstuff.com
  - www.tracert.com
  - www.traceroute.org
  - www.network-tools.com
  - www.securityspace.com (commercial with free trial)

These websites include forms that allow you to enter a target site and do research, or, in some cases, even attacks. These websites can perform DNS lookups, reverse lookups, traceroutes, and a variety of other valuable services. From a test machine on a separate network from your production system (such as a dial-up account), these tools can be interesting to experiment with.



One of the main tenants of being a malicious attacker is not getting caught. But, how exactly is an attacker to find vulnerabilities and not touch a network or target system? One of the answers is Shodan. Shodan is an online service which crawls the Internet in much the same way Google crawls web pages. Instead of reading and indexing web page text, like Google, Shodan indexes service banners. Banners for services like FTP, and Telnet will often have a unique signature to identify that service, vendor and version number. All an attacker, or enterprising defender, needs to do is search for a string associated with a service and vendor and Shodan will display its cached results. This service can also be restricted to specific network ranges so a defender can see what Shodan has stored in relation to your organization.

Check it out at:

https://www.shodan.io/

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- **Step 1: Reconnaissance**
- Step 2: Scanning
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. **War Dialing**
2. War Driving
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. Evading IDS/IPS
6. Vuln Scanning with Nessus
   - Lab: Nessus
7. SMB Sessions
   - Lab: SMB Sessions

After completing thorough reconnaissance of the target, attackers begin scans to find openings in the target system.

Let's start by discussing war dialing: an older technique, but still amazingly successful.

Even today, an unprotected modem provides the easiest method for penetrating a network. Whenever we perform a war dialing assessment, more often than not, we get into the target network.

War Dialers

- War dialers dial a sequence of telephone numbers, attempting to locate modem carriers or a secondary dial tone
- Demon dialers dial a single number to conduct a brute-force attack against passwords
- Often, an unprotected modem provides the easiest method for penetrating a network
- Many recent news stories about hacking voicemail

War dialers dial a sequence of telephone numbers attempting to locate modem carriers or a secondary dial tone.

Where does an attacker get the numbers for conducting war dialing?

- The Internet is a treasure-trove of this information. Your users' queries to mailing lists and news groups are helpful.
- The Whois database at InterNIC has telephone numbers for your network contacts.
- Your organizations' website may include numbers.
- Social engineering: "I'm from the phone company, and I need to verify what extensions you folks are using."

WarVOX

War Dialing

- HD Moore released a war-dialing tool called WarVOX
- Free at https://github.com/rapid7/warvox
- Conducts war dialing using one or more VoIP accounts
  - No telephony hardware required... just an Internet connection and VoIP account
    - Provider must support IAX protocol
    - Several compatible VoIP providers that do not prohibit war dialing are listed on the WarVOX website
  - Traditional modem-based war dialing: 1,000 numbers in approx. 8 hours
  - WarVOX war dialing: 1,000 numbers per hour
- A significant increase in speed
- Supports caller ID spoofing
  - Enter a single number for all calls dialed
  - Enter variable number of Xs for pseudo-random source number
  - Enter "SELF" to make caller ID same as dialed number; may bypass PIN authentication in some voicemail systems

HD Moore released a tool called WarVOX that focuses on conducting war dialing assessments of target telephone number ranges. Unlike traditional war dialers that use a modem to dial phone numbers, WarVOX relies on VoIP communications. A user configures WarVOX with account information from a VoIP service provider, and WarVOX uses that account to dial a list or range of provided phone numbers. No telephone line or modem is required by WarVOX. Instead, just a broadband Internet connection and one or more VoIP accounts suffices. The VoIP service provider must support the Inter-Asterisk eXchange (IAX) protocol for VoIP. The WarVOX website includes a handy list of different VoIP service providers that are compatible with WarVOX and that do not explicitly prohibit war dialing using their service.

The real benefits of WarVOX are increased speed and flexibility. A traditional modem-based war dialer, such as THC-Scan, can complete about 1,000 phone calls in an 8-hour span. WarVOX, on the other hand, can typically dial more than 1,000 numbers per hour, provided that it is connected with a typical residential broadband Internet connection, and the VoIP provider supports multiple calls simultaneously from one account, which most do.

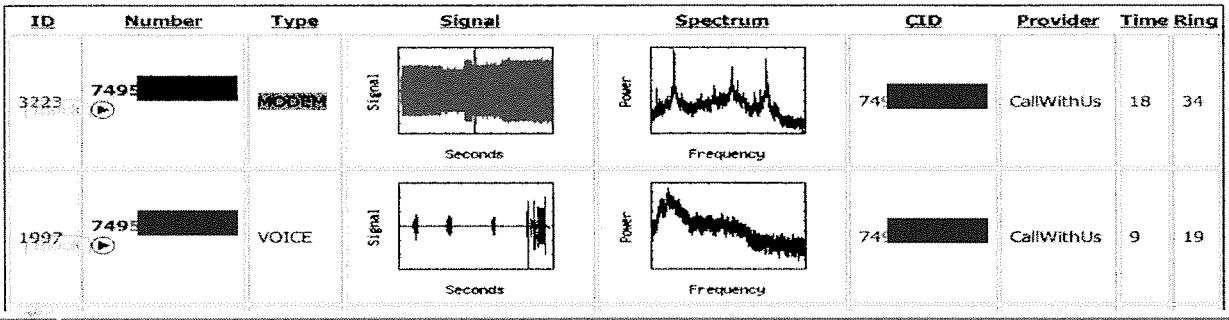
Besides the increase in speed, the flexibility provided by WarVOX includes caller ID spoofing, where it provides three options: The user can configure it with a single number to use as the caller ID value for all numbers dialed; or, the user can specify a number, such as 1 555 555 XXXX, where each X will be replaced with a pseudo-random digit between 0 and 9 for each number called; finally, WarVOX can be configured with "SELF" as the caller ID value, which makes it set the caller ID value to the same number that it is dialing. This option can be used to bypass PIN authentication in some voicemail systems.

Get it at https://github.com/rapid7/warvox.

WarVOX Results

War Dialing

- WarVOX records an MP3 audio file associated with each number dialed and answered, with results stored in a PostgreSQL database
- You can apply a series of signatures to determine what answered... entirely new signatures to match individual human voices
  - Modem, fax, voicemail box, and more



For each number dialed, WarVOX listens for an answer and records an MP3 audio file of the communications. Thus, at the end of the war dialing exercise, numerous MP3 files are available in the WarVOX database for analysis to determine what answered each dialed number.

WarVOX provides a series of signatures to apply against the captured audio to determine whether a modem, fax machine, voicemail box, or human voice answered the call. This signature set will likely be expanded in the future to help better characterize particular types of systems that are discovered. And, because WarVOX records all the audio for later analysis, new signatures can be applied against already gathered results, making WarVOX flexible.

The results of WarVOX are displayed on the screen in a browser window, showing the number dialed, the type of system that answered (based on its audio characteristics matching a signature), the signal over time captured in the audio file, and a spectrum analysis of the resulting audio.

War Dialers – Now What?

War Dialing

- So I've found a bunch of modems... what do I do now
- Review the war dialer logs and look for familiar login prompts or warning banners
- Connect to each discovered modem
  - Oftentimes, you find a system without a password
    - Old, neglected machine still on the network
    - Router
  - If there is a user ID/password prompt, guess
    - Make it an educated guess, based on the system
    - What are default accounts/passwords
    - What common things are associated with the target

Many systems tell you what platform they are (for example, "Hi, I'm BSD!"). For others, you can determine this information from the nature of the prompt. UNIX boxes and Cisco router prompts are particularly easy to identify.

Although guessing passwords is a time-consuming process, keep in mind that time is the single greatest resource your adversaries have.

For password guessing, a complete list would take up numerous pages, indexed by system type. This partial list can get you started (try each for userID and password, and all combinations):

- root
- sync
- bin
- nobody
- operator
- manager
- admin
- administrator
- system
- days of the week
- COMPANY_NAME
- COMPANY_PRODUCT

Defenses – Preparation (1)

War Dialing

- An effective dial-up line and modem policy is crucial
  - Inventory all dial-up lines with a business need
- Conduct war dialing exercises against your own network
  - Reconcile your findings to the inventory
  - Utilize a commercial war dialer
    - NIKSUN's Phonesweep (http://www.niksun.com/product.php?id=17)
  - Or utilize WarVOX
  - Get list of phone numbers from the phone company *based on the bills*; they make sure they get paid
- Conduct periodic desk-to-desk checks in the evenings
  - Use the two-person rule

When war dialing against your own network, how do you determine which numbers to dial?

At a minimum, get a list of all analog lines at your PBX. You may also want to consider dialing digital lines, because inexpensive digital-line modem adapters are readily available.

A major concern involves numbers not accessible through your PBX (such as direct lines from the telco). The best, although not ideal, approach for finding these is to follow the money: Get the telephone bills from the telco. Ask your telco to give you a copy of all bills being mailed to a given address or, if possible, all bills for lines at a certain address.

When you do desk-to-desk checks, always employ the two-person rule (a.k.a., "the buddy system"). With an explicit two-person team checking for unwanted/unregistered modems, you will not be subject to claims of unfairness or, worse yet, theft from people's desks. If a single person checks for modems late at night, and something turns up missing from someone's desk, you may have significant problems.

Defenses – Preparation (2)

War Dialing

- Identification
  - Activate scanning-detection functionality in your PBX, if available
  - Consider "PBX Firewall/IPS," such as SecureLogix Voice IPS
    - Monitors trunk connecting PBX to phone network, looking for fax tones
- Containment
  - Shut off modems when they are discovered
  - Know whom to call in your own telecom group and at the phone company to geographically isolate a modem
- Erad, Recov
  - Remove renegade modems from network
  - If modem is absolutely required, change phone number and secure it with strong authentication (token, crypto, or others)

For containment, eradication, and recovery, removing the victim modem is a reasonable idea. If it is absolutely required, move it to another phone number and add stronger authentication, such as a time-based token, smart card, or another technology.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. War Dialing
2. **War Driving**
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. Evading IDS/IPS
6. Vuln Scanning with Nessus
   - Lab: Nessus
7. SMB Sessions
   - Lab: SMB Sessions

A more popular recent attack vector is targeting unsecured wireless LANs through war driving and sniffing.

Wireless Misconfigurations — War Driving

- Many wireless access points (base stations) are configured with no security
  - Blank or default SSIDs are common
- By default, most access points broadcast beacon packets with their SSIDs 10 times per second
- Even for those APs configured not to include the SSID in beacons (SSID cloaking), SSIDs are still sent in clear text whenever anyone uses the wireless LAN
  - Therefore, the SSID is, in no way, a security feature
- Various wireless security protocols (WEP and LEAP specifically) have significant flaws
  - Just "turning on" security is often not enough protection for sensitive traffic and systems

The SSID is the services set identifier. It acts as a name of a wireless LAN. Despite what some people think, SSIDs offer no real security. It is NOT a password.

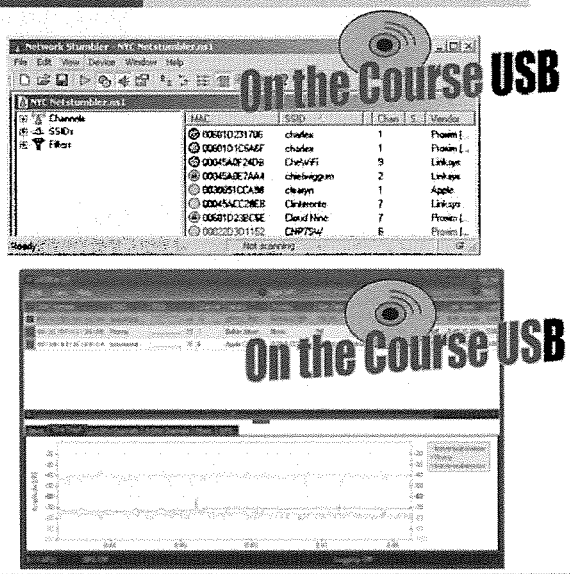
Most access points include their SSID in beacon packets sent approximately every 100 milliseconds. You can configure most access points to omit their SSIDs from their beacons, which is a feature known as SSID cloaking. This is a small boost to security, however, because access points still send all SSIDs in clear text when authenticated users use the access point. We cover some tools shortly that can gather SSIDs from sniffed traffic even when broadcast beacons are disabled on an AP.

Compounding the problem, various wireless security protocols (WEP and LEAP specifically) also have significant flaws, as we see later. Thus, just "turning on" security is often not enough protection for sensitive traffic and systems.

Tools for Wireless LAN Discovery

War Driving

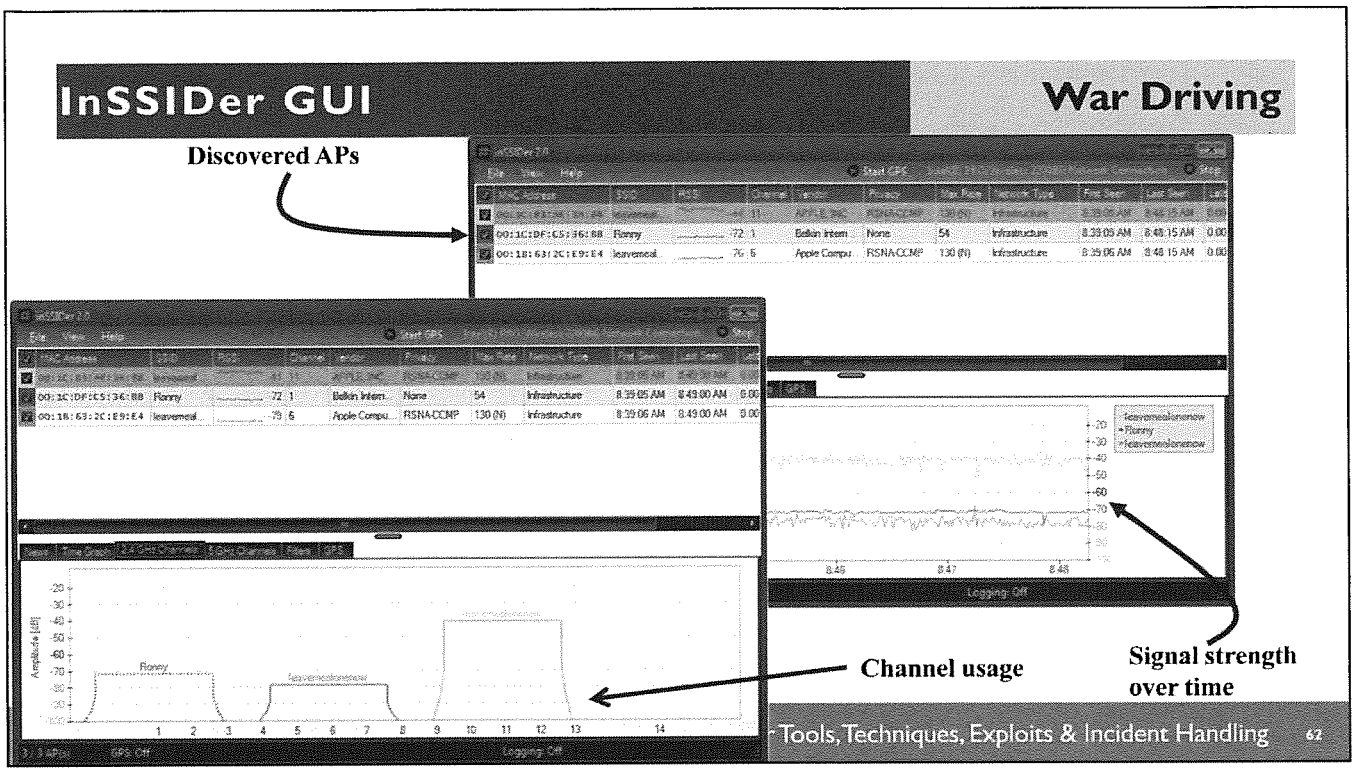
- NetStumbler by Marius Milner, free but no source
  - www.netstumbler.com
  - Detects 802.11 a/b/g
  - Windows-based, but has problems with WinVista, 7, and 8
- InSSIDER by MetaGeek
  - http://www.metageek.net/products/inssider
  - Detects 802.11 a/b/g/n
  - Works on XP, Vista, 7, and 8
  - Linux version also available
- Both tools are noisy; they send SSID-less probe requests and look for probe responses
  - Therefore, cannot detect APs that don't respond to such requests!



NetStumbler is a free war-driving tool for Windows. It can be used to detect 802.11a/b/g interfaces and can tie in Global Positioning System (GPS) data. The tool works fine on Windows XP, but has problems on some Windows Vista, 7, and 8 machines.

InSSIDer is another free war-driving tool for Windows, which functions properly on XP, Vista, Windows 7, and 8. A Linux version of InSSIDer has also been released. It can detect 802.11 a, b, g, and n access points and provide interesting visualization options for signal strength and channel usage.

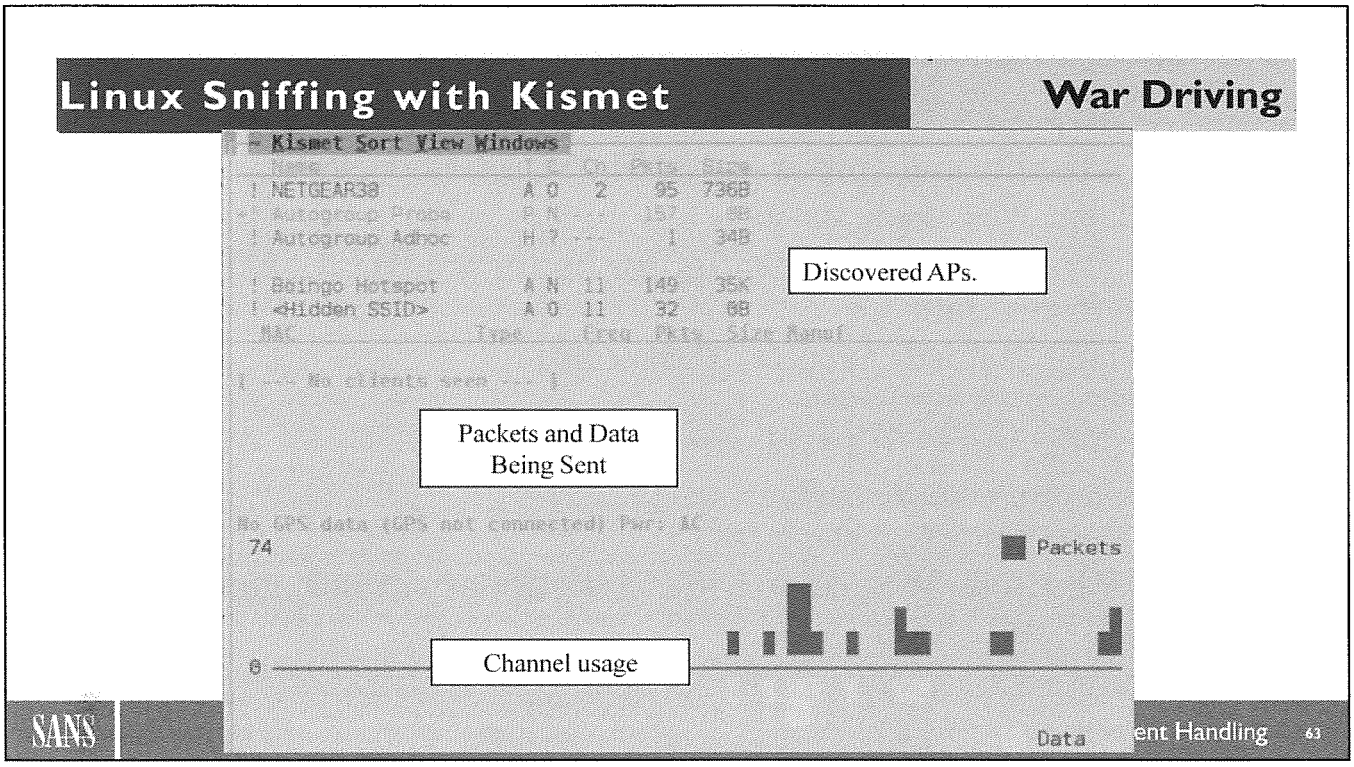
Both NetStumbler and InSSIDer work solely by sending out a constant stream of probe requests without an SSID, hoping that an access point responds with a probe response that includes its SSID. If an AP is configured to not respond to a probe that omits an SSID (another feature of SSID cloaking), NetStumbler and InSSIDer cannot determine the SSID. However, the AP still might be detectable and vulnerable; the attacker just has to use a different tool, such as Kismet, which is discussed next.



The InSSIDer GUI includes three sections. The upper-most section (a thin grey line at the top of the screen) is associated with control of the wireless interface, with its File, View, and Help menus, along with the Start and Stop toggle button at the top right, which turns on the gathering of wireless information. There is also a button in this bar to start or stop GPS recording. A drop-down menu lets the user choose a wireless interface.

Below the configuration line, we have an inventory of all wireless devices InSSIDer has spotted, including the MAC address, SSID, channel, and type of encryption in use, such as WEP, WPA2 (which InSSIDer displays by the use of two protocols associated with WPA2: RSNA-CCMP), none, and so on.

At the bottom of the screen, our third section shows various tabs we can use to look at how the signal varies over time (Time Graph), a graphical display of channel usage (2.4 GHz or 5 GHz Channels), filters (which can be used to focus on only certain devices given their SSID, MAC address, channel, encryption type, or other properties), and GPS settings.



Kismet is one of my favorites, with its wide support for various wireless cards and easy installation on Linux. It can also passively discover access points without ever sending a beacon message. It just sniffs, looking for SSIDs in the messages sent across a network. So, even if you disable beacon responses on the access point, Kismet can still detect the AP's presence, as long as someone is sending traffic over the wireless LAN.

The main difference between many Linux-based tools, such as Kismet, and many Windows-based tools is that many Linux tools have the capability to passively sniff wireless networks.

With the proper adapters, Kismet also has the ability to detect other wireless protocols, such as Zigbee.

Additional Tools for Wireless Sniffing and Crypto Attacks

War Driving

- Utilize a traditional sniffer, gathering wireless packets
  - Tcpdump, Wire Shark, and more
- Or use a wireless-specific sniffer for better analysis of wireless-specific frame data
  - Omnipeek (formerly Airopeek), Commercial
    - http://www.wildpackets.com
- Aircrack-ng and WEPCrack crack WEP keys
  - http://www.aircrack-ng.org/
  - http://wepcrack.sourceforge.net
- ASLEAP by Josh Wright provides a dictionary attack against LEAP authentication

These tools allow an attacker to sniff a wireless network. They grab traffic from the wireless LAN so an attacker can look at the data.

To sniff data from a wireless LAN, you could use Tcpdump or Wire Shark, bound to your wireless card. However, they may not be able to decode all the wireless-specific packets as thoroughly as some of the wireless-specific sniffing tools.

Aircrack-ng and WEPCrack add an incredibly useful function for attackers. They can crack the keys used in the 802.11 security scheme called Wired Equivalent Privacy (WEP). WEP is a flawed algorithm, in that an attacker can sniff encrypted packets and look for "interesting packets." The crypto algorithm is broken, and if an attacker can grab enough interesting packets, he or she can determine the WEP key. With tools like Aircrack-ng, the attacker needs to sniff about 50 to 100 Megs of data, which can often be done in 10 to 30 minutes. After grabbing this data, the tool cracks the WEP key, and the attacker can view all data on the LAN recorded earlier and sent later, as long as the WEP key remains constant.

Josh Wright's tool called ASLEAP provides a dictionary-based attack against the LEAP protocol used in some wireless environments. By attacking the user's Windows password hashes based on sniffed LEAP challenge and response messages, ASLEAP can determine a user's password to gain access through a LEAP-protected WLAN.

Additional Wireless Crypto Attack Tools — War Driving

- Josh Wright has released a tool called CoWPAtty
  - A dictionary-based cracking tool for pre-shared keys with WPA1 and WPA2
  - Must sniff four-way handshake
  - Cryptographically, WPA is a complex protocol
    - On a modern laptop, crypto routines can try between 10 and 50 guess/encrypt/compares per second
  - Thus, pre-computed encrypted dictionaries are a big help... But, WPA folds SSID into its cryptographic exchange
  - Pre-computed dictionaries are available for
    - The 1,000 most common SSIDs (linksys, tsunami, for example) with 172,000 passwords for > 7 Gigs
    - The 1,000 most common SSIDs with 1 million words for > 33 Gigs

Just as Aircrack-ng attacks WEP, other tools attack other forms of wireless encryption, specifically the newer Wifi Protected Access (WPA). In particular, the CoWPAtty tool by Josh Wright allows an attacker to sniff a WPA1 or WPA2 four-way handshake used for authentication, and then mount a cryptographic cracking attack against pre-shared keys used for that exchange. This tool assumes that pre-shared keys were hard-coded into both the client and the access point for WPA instead of using public key techniques, such as those provided by the Protected Extensible Authentication Protocol (PEAP).

Because WPA (either WPA1 or WPA2) are cryptographically complex protocols, WPA cracking with CoWPAtty is not a fast operation. A modern laptop can perform approximately 10 to 50 guess, encrypt, compare cycles per second with CoWPAtty. To help improve this speed by about a thousand-fold, some people have released pre-computed encrypted dictionaries for CoWPAtty. However, WPA folds in the SSID of the wireless network in its cryptographic exchange. Thus, these pre-computed dictionaries are tied into a fixed set of SSIDs, able to crack only those pre-shared keys that are in their dictionaries for networks that use the given SSIDs. Available for free, one set of such dictionaries uses 170,000 words for 1,000 of the most common SSIDs in use today, clocking in at just over 7 Gigs of information. A larger set includes more than a million words, again for those most popular SSIDs, consuming 33 Gigabytes of space! If a non-standard SSID is in use, however, raw cryptographic attacks are required, considerably slowing the process (10 to 50 attempts per second).

Surreptitious Wireless LAN Usage — War Driving

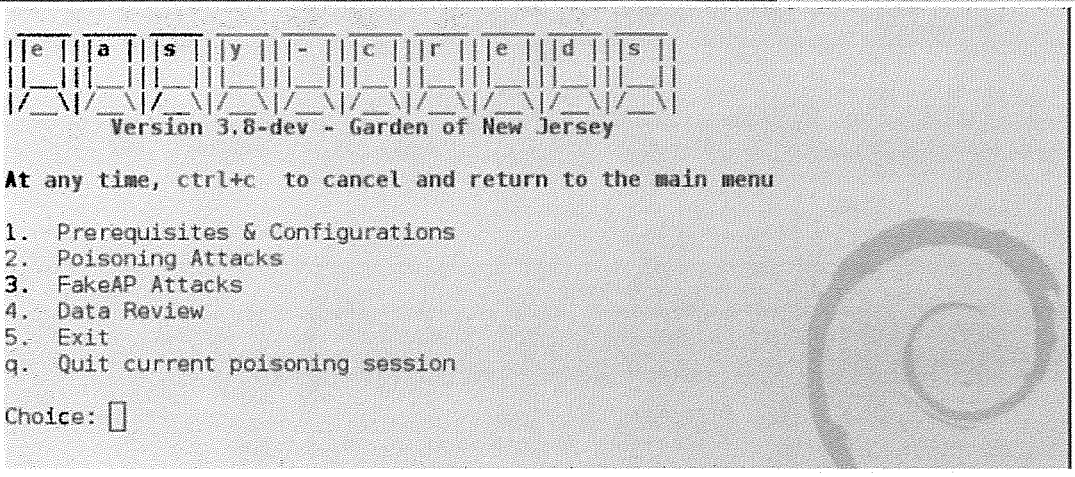
- Many people occasionally connect to untrusted WLANSI
  - To surf Internet for free
  - "But I just wanted to check my e-mail," he or she may say
- If their machines are hardened, they are safe, right?

A lot of users surreptitiously use wireless LANs. They need Internet access, so they fire up their wireless card and see what's available. They find an open access point and use it to surf the net and synch their e-mail. The legality of such access is determined on a state-by-state basis in the United States.

Given the legal concerns, I'm sure that no one in this class would ever use a wireless access point for Internet access without permission. But, as you know, large numbers of users do.

What's the problem from a security perspective? If their machines are hardened, they are safe, right?

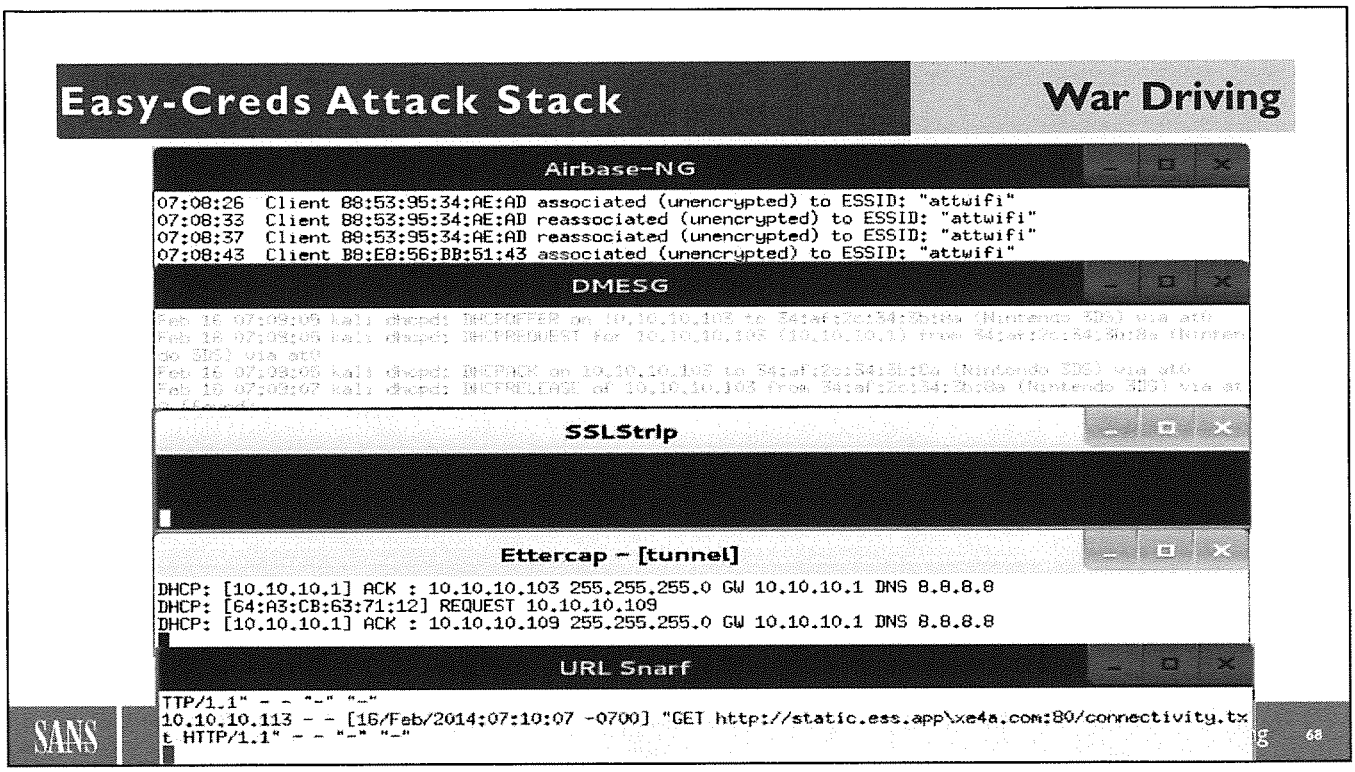
Linux Attack with Easy-Creds



Easy-Creds greatly simplifies the process of creating malicious access points

One of the easiest ways to attack wireless communications is to attack the client. Although many people think this entails launching direct attacks against a client, a far easier (and more effective) way to attack a client is to have the client system connect to a wireless access point that she believes is safe. Once she has connected, the attacker can launch a number of different passive attacks to sniff sensitive data, such as HTTP session IDs, URLs, and even intercept SSL traffic.

Easy-Creds does all of this and more. It allows an attacker to quickly configure an evil wireless access point that the attacker has full control over.



When Easy-Creds starts, it automatically configures the attacker system to bridge communication between its wireless interface and the Ethernet interface. Remember, the goal is not to just have the victim system connect to the attacker, it is to allow the traffic pass through so it can be intercepted. To create the access point, Easy-Creds automatically starts Airbase-NG. In the slide's example, an access point with the ESSID "attwifi" was created. This is one of the more effective ESSIDs, because many devices (such as phones) by AT&T wireless automatically joins the nearest access point with this name.

It then logs DMESG messages to track all DHCP leases. In the slide, you can see a Nintendo DS connecting to the evil access point.

If any SSL redirects are encountered, SSL strip intercepts and brokers the SSL/HTTP session so clear-text passwords can be harvested.

Finally, it launches Ettercap and URLSnarf to monitor URLs and launch session hijacking/manipulation attacks.

On Day 3, we go into detail about tools like SSLStrip and Ettercap.

Karmetasploit

War Driving

- HD Moore, author of the Metasploit exploitation framework, has integrated features of a tool called Karma into Metasploit
- With this feature set, Metasploit listens on a wireless interface for probe requests
  - Then pretends to be sought-after wireless access points
- Metasploit serves up a series of exploits for various vulnerable clients when they try to connect
- We discuss Metasploit in 504.3

HD Moore has integrated a series of features from a tool called Karma into Metasploit, a flexible exploitation framework containing exploits for hundreds of different vulnerabilities. With this Karma feature set in Metasploit (together called Karmetasploit), an attacker can configure and activate Karma from within a Metasploit user interface. These features cause Metasploit to listen on a wireless interface for client probe requests. When it gets such a request with an SSID, Karmetasploit sends a response, pretending to be an access point. Then, when a wireless user associates with the Karma elements of Metasploit, the rest of the Metasploit framework serves up exploits for client-side software running on the victim machine. This combination of Karma-style wireless attack and Metasploit-style exploitation is powerful.

Really Bad Wireless Karma with Karmetasploit

War Driving

- Once a client joins imposter network, Karmetasploit includes various services
  - **DHCP** (of course)
  - **DNS**: All DNS requests are intercepted, and the attacker's own IP address is returned
  - **POP3**: "I'm your mail server. Authenticate to me."
  - **HTTP**: "I'm also every server on the entire Internet. Want to talk?"
  - **Samba**: All Windows file sharing points back to the attacker's machine

Once the wireless client associates with Karmetasploit, the tool includes a variety of fake services to provide access to the client. First off, it has a DHCP service to provide the client with an IP address. It also includes a DNS server that intercepts all DNS requests (not just those going to the DNS server configured by DHCP) and sends response of the attacker's choosing. By default, all IP addresses are resolved to the attacker's IP address of the box running Karma.

It includes a POP3 server that pretends to be all POP3 servers on the Internet, accepting and logging passwords sent by the user.

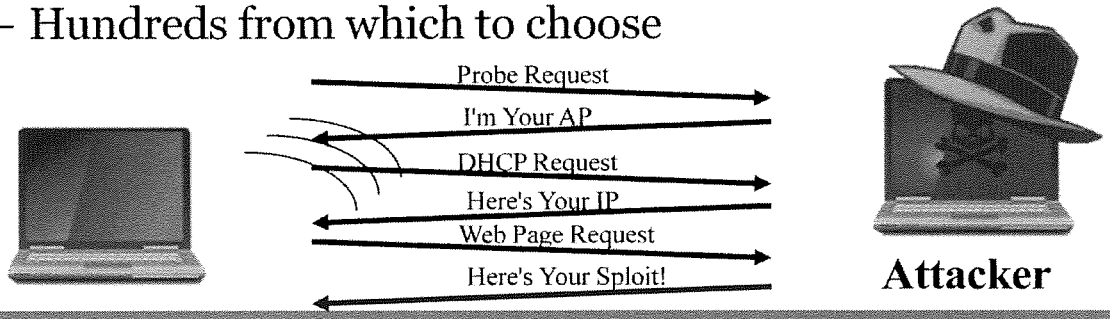
It includes a web server that pretends to be any server on the Internet. Gee, the White House website looks an awful lot like the Gnu website today. Microsoft and Google also look the same. The attacker, of course, can configure specific sites to pretend to be something else, such as a bank or wireless access provider.

Finally, the tool includes a SAMBA server, which pretends to be any file share for which the user is looking.

Karmetasploit Exploitation

War Driving

- Karemetasploit exploits various client software, of course
  - All Metasploit client-side exploits can be configured and launched
  - Hundreds from which to choose



Once the victim connects to Karmetasploit, which is impersonating the entire Internet, what happens? The tool tries to exploit any client-side technology that tries to access the outside world, of course.

Karemetasploit can exploit vulnerable browsers, mail readers, Windows file-sharing clients (such as svchost), and other technologies.

Defenses – Preparation (1)

War Driving

- MAC address filtering at an access point isn't secure
  - Doesn't scale well, and MAC addresses can be spoofed
  - On Linux, you can use the ifconfig command to change MAC addresses
    `# ifconfig wlan0 hw ether [MACaddr]`
  - On Windows, PowerShell's Set-NetAdapter commandlet can do this, with a process described at http://blog.commandlinekungfu.com/2013/06/episode-167-big-mac.html
- Set SSID so that it doesn't attract attention
- Improve authentication beyond standard WEP
  - WPA and WPA2 are far more secure than WEP
  - 802.11i (most commonly advertised as WPA2) appears to have better security capabilities
    - Avoid using TKIP because of security concerns from November 2008 research
    - Instead, use AES for crypto in WPA2

So, how can you defend your network against such attacks?

Do not rely on MAC address filtering at your access point to implement security. Although you could allow only traffic from registered MAC addresses, such security is deeply flawed. MAC addresses can be easily spoofed (either by using the ifconfig command in Linux/UNIX or with PowerShell's "Set-NetAdapter" commandlet in Windows). In addition, MAC address filtering doesn't scale well. If you have 100 users constantly buying new wireless cards, it is hard to keep your filters up to date.

To defend your wireless systems, first, set the SSID to a value that doesn't attract attention to your network. Establish a standard for naming wireless LANs in a way that doesn't include your organization's title in the SSID. It's better to name a wireless LAN something like "1234" rather than "bigbank1234." One solution to this dilemma of naming access points is to simply set the SSID to the device's serial number.

A better solution involves improving authentication and encryption beyond standard WEP. Consider using WPA, which is a security capability compatible with most current access points. Or, if your systems support it, the newer and stronger 802.11i is even better. Some implementations of 802.11i are referred to as WPA2. Both WPA and WPA2 (that is, 802.11i) include stronger cryptographic protections. Both WPA and WPA2 support a protocol called the Temporal Key Integrity Protocol (TKIP) as an option. Avoid TKIP, because researchers have found some vulnerabilities with it, publishing their work in November 2008. Instead, we recommend that you use AES encryption with WPA2 for a more secure wireless environment.

Defenses Preparation (2)

War Driving

- Furthermore, use Layer 3 encryption to bolster or even supplant Layer 2 encryption
  - Use Virtual Private Network
  - All data from end system to VPN gateway inside of wireless device encrypted and authenticated
  - Use commercial or free VPN
- Be careful with your wireless VPN config! With wireless, an attacker might be able to grab all data and try to crack it!
  - It's crucial to disable Aggressive Mode IKE, because IKE Crack and Cain can break pre-shared keys used with IKE/IPsec
  - IKE Crack is at http://ikecrack.sourceforge.net
  - A dictionary attack against pre-shared IKE keys

Organizations should consider securing their wireless LANs using a Virtual Private Network (VPN).

You can use commercial VPNs or one of several free, open-source VPNs.

When setting up your VPN for wireless use, be careful with your wireless VPN configuration. Remember, with wireless, attackers might be able to grab all encrypted data from the wireless LAN and try to crack it, an option they often don't have when going after an Internet-based VPN infrastructure. With wireless, all the data can be easily sniffed by attackers. In configuring your wireless VPN, it's crucial to disable Aggressive Mode IKE, because a tool called "IKE Crack" can break pre-shared keys sent via that mode. IKE Crack, available at http://ikecrack.sourceforge.net/, performs a dictionary attack against pre-shared IKE keys used with IPsec. Although IKE Crack is not wireless-specific, it can be used very well against wireless LANs, provided that they are implemented with pre-shared IKE keys and use aggressive mode IKE. Aggressive mode IKE is far weaker cryptographically and should be disabled in your VPN gateways. The Cain tool, which we cover on Day 4, can also crack pre-shared IKE keys exchanged with aggressive mode IKE.

Defenses Preparation (3) — War Driving

- Identification
  - Wireless IDS tools are starting to get some traction
  - Aruba Networks, Motorola AirDefense, AirMagnet, and others offer products
  - IBM also offers such services on a subscription basis, using Linux-based sensors
  - Cisco (and others) offer options to use existing APs to detect unregistered APs inserted into the network; they can generate an alert or a Denial of Service
- Cont, Erad, Recov
  - Remove renegade access points

For identifying wireless intruders, you could look for the appearance of renegade access points or strange messages sent by intruding wireless clients (including repeated probes and frequent deauthenticate messages). Aruba Networks, Motorola AirDefense, and AirMagnet all have wireless IDS offerings. IBM offers a managed wireless IDS service, using Linux-based sensors distributed throughout your environment.

Finally, for detecting renegade access points, Cisco (and some other AP vendors) offer built-in capabilities in existing access points to detect unregistered (renegade) access points that appear in your environment. When one of your Cisco APs detects an unregistered AP in your environment, it can alert you. Alternatively, Cisco provides features that attempt to jam the renegade access point by launching a Denial of Service flood against it. I strongly recommend that you avoid this DoS feature, because its legal implications could be dire!

For containment, eradication, and recovery, make sure that you remove renegade access points before an attacker causes significant damage through them.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. War Dialing
2. War Driving
   - **Lab: Wireless LAN Discovery with InSSIDer**
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. Evading IDS/IPS
6. Vuln Scanning with Nessus
   - Lab: Nessus
7. SMB Sessions
   - Lab: SMB Sessions

This page intentionally left blank.

Lab: Install InSSIDer

- Install InSSIDer from the Windows directory on the Course USB
  - Extract either the x64 or x86 InSSIDer Installer zip file to your desktop, depending on whether you have 64-bit or 32-bit Windows
  - Run setup.exe from the extracted contents
  - You may be prompted to install Microsoft .NET Framework, which is also included on the Course USB (dotnetfx[version].exe)

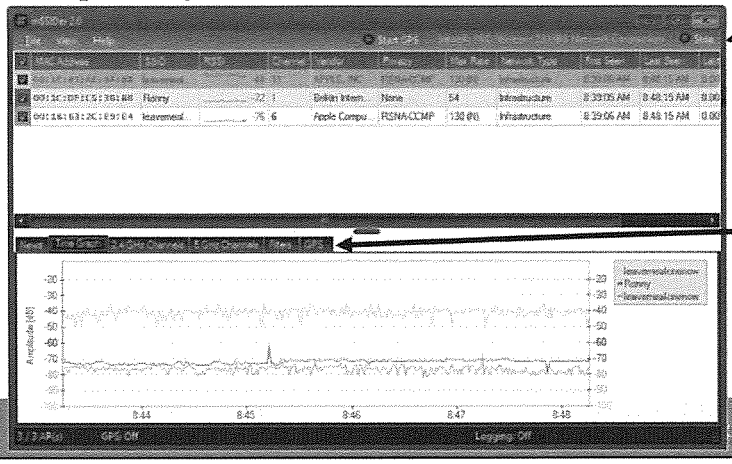
Let's run an active scan of wireless devices using InSSIDer. Install the tool from the Course USB in the Windows directory.

There are two versions of InSSIDer on the Course USB in the Windows directory: one for 64-bit versions of Windows (with an x64 in its file name) and one for 32-bit versions of Windows (with an x86 in its file name). Choose the appropriate file for your version of Windows and extract the contents of the ZIP file (a setup.exe file and .msi file) to your desktop. Run the setup.exe file to install InSSIDer. On Windows Vista, Windows 7, or Windows 8, right-click setup.exe and select "Run as administrator."

You may be prompted to install a compatible version of Microsoft's .NET Framework, which is included on the Course USB in the Windows directory in the file called "dotnetfx[version].exe."

Lab: Run InSSIDer

- Click Start in the upper-right corner. Which wireless devices do you see?
  - On most virtualized Windows machines, InSSIDER cannot access the virtualized network adapter, so you can see the interface, but not discover access points



After you install InSSIDer, run it by going to Start-->All Programs-->MetaGeek-->InSSIDer. Click the Start button in the upper-right corner of the screen. Next, look through the different tabs in the bottom portion of the screen, especially the Time Graph and the 2.4 GHz channels tab. How many access points does InSSIDer spot? What are their names? What channels are they using?

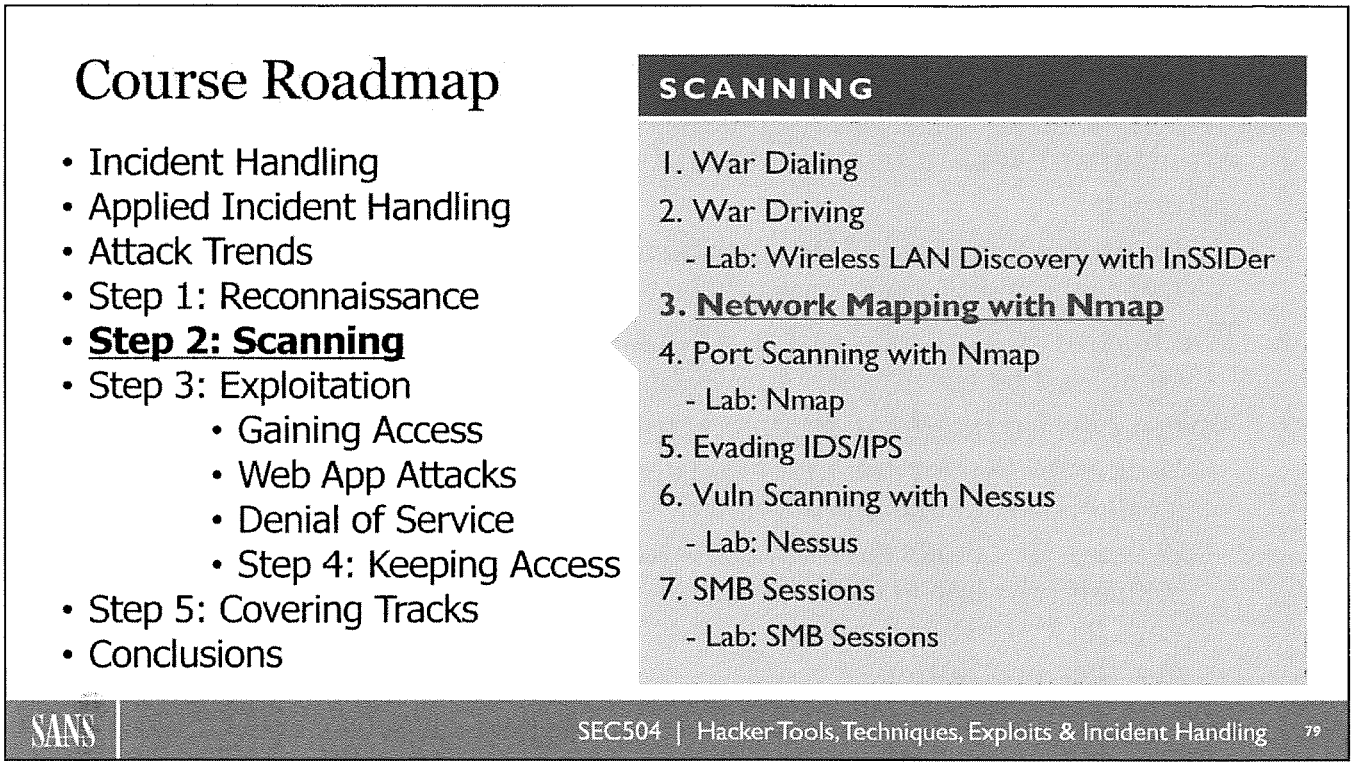
Lab Conclusion: Go Forth and Find Interesting Access Points

- Let's go out and find interesting access points
  - This can be done around the hotel, your neighborhood, or office
  - This is something we should do on a regular basis at work
- When you return to class, your instructor asks what you found
- Points for access points with fun names and access points without security enabled
- ***Please do not connect to any access points without permission***

Let's take some time to go out into the conference, offices, and/or our neighborhoods to find interesting and unsecured access points. The goal of this lab is two-fold.

First, we want you to get familiar with doing this, because it is something you should do at work. Second, it can be eye-opening to see just how many unsecured access points there are.

Remember that we are just passively recording the access points around us. At no time are you to connect to any of these access points without permission.



An attacker now sits in one of three places:

- Across the Internet, staring down your DMZ using the IP addresses gathered during reconnaissance
- Connected to your internal network via a modem discovered during war dialing
- Connected to your wireless LAN infrastructure via an access point discovered during war driving

Now what? Well, attackers want to get a feel for your network topology, so they'll turn to network mapping tools.

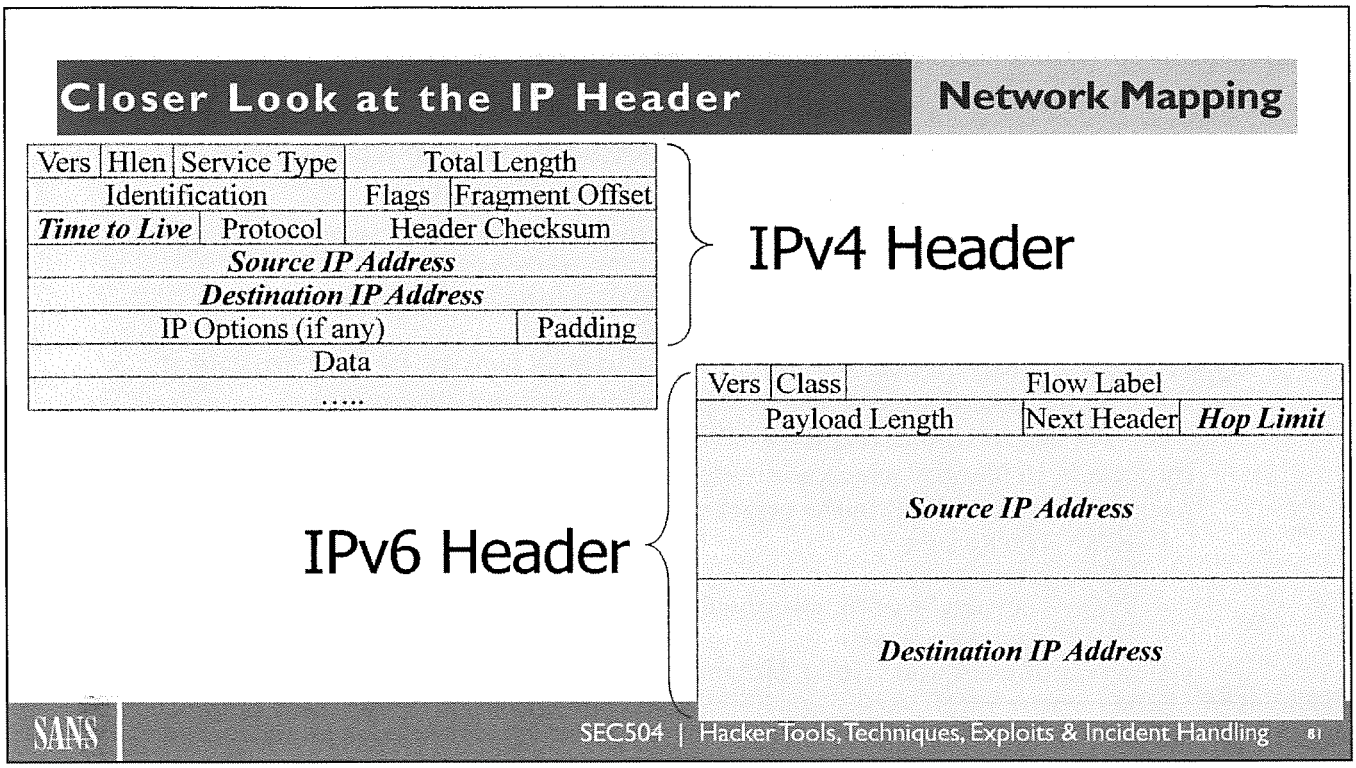
Network Mapping with Nmap

- An attacker wants to understand the topology of the target network
  - Internet connectivity: DMZ, perimeter networks
  - Internal network (with access from modem or wireless access point)
- The layout of routers and hosts can show vulnerabilities
  - Or at least let the attacker know where things are
- Nmap can be used for network mapping and port scanning
  - Written by Fyodor and the Nmap development team
  - Available for Linux and Windows
  - Zenmap GUI lends itself to network mapping and visualization
  - Available for free at www.nmap.org
  - Let's look at network mapping first, and then port scanning

An attacker wants to understand the topology of your network, mainly Internet connectivity: DMZ, perimeter networks, and your intranet. The layout of routers and hosts can show vulnerabilities or at least let the attacker know where things are.

Nmap, the popular network-analysis tool by Fyodor and the Nmap development team, can be used for network mapping and port scanning, although most people associate it with the latter. We look at using Nmap for network mapping first, and then for port scanning.

Nmap includes a GUI-based front end called Zenmap, which provides excellent network mapping and visualization features.



This slide shows the IP packet header (for IPv4 and IPv6), highlighting the key areas associated with mapping a network. The source and destination IP address are of interest to us at this point, as well as the Time to Live field for IPv4 and the Hop Limit field for IPv6. The source IP address indicates from where the packet comes. The destination identifies to where it's going.

The Time to Live field in IPv4 and the Hop Limit field in IPv6 both indicate how many hops the packet can go across the network before it's discarded. Let's explore TTL in more detail to see how traditional tracerouting works for network mapping, as manifested in the Linux and UNIX traceroute command and the Windows tracert command. In this discussion, the IPv6 Hop Limit field behaves exactly like the IPv4 Time to Live field.

Sweeping for Network Mapping

Network Mapping

- A common first component of network mapping is to identify the addresses in use by sweeping through address space
  - For example, send an ICMP echo request to a range of IP addresses
    - If something replies, that address is in use by some target system
    - If nothing replies, we assume system not on the network at that address
      - It could be that the firewall is blocking ICMP
- By default, Nmap sweeps each target address before port scanning it
  - This can be reconfigured to use TCP packets or ignored all together (the –PN flag in Nmap, formerly –P0)
  - By default, to identify which addresses are in use, Nmap sends the following four packets to each address in the target range:
    - ICMP Echo Request
    - TCP SYN to port 443
    - TCP ACK to port 80 (if Nmap is running with UID 0)
    - ICMP Timestamp request
  - When running without UID 0, Nmap sends SYN to port 80 instead of ACK

A common initial step in network mapping is to sweep through the target network addresses, sending one or more packets to each address trying to solicit a response. The response indicates that the given address is in use by some target machine.

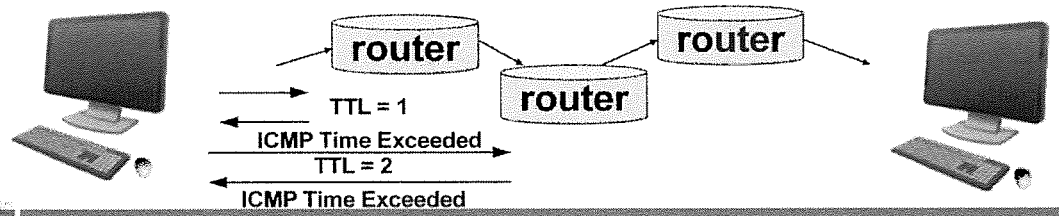
By default, Nmap sweeps through each target address before it launches a port scan of the address. However, this scanner behavior can be reconfigured so that all addresses are scanned, regardless of whether they are pingable or not. In Nmap, the –PN command flag tells Nmap not to ping the target ("no ping"), but to just start the port scan. In older versions of Nmap, this option was –P0, but it was changed in more recent versions.

By default, to identify which addresses are in use, Nmap sends the following four packets to each address in the target range: ICMP Echo Request, TCP SYN to port 443, TCP ACK to port 80 (if Nmap is running with UID 0), and an ICMP Timestamp request. If Nmap is not running with UID 0 on a Linux machine, it runs through the same set of four packets, but uses a TCP SYN to port 80 instead of an ACK, because it cannot craft the ACK packet without UID 0. If any of those packets receives a response, Nmap assumes the address is in use by a valid target machine and proceeds to conduct its port scan.

How Traditional Traceroute Works

Network Mapping

- Traceroute sends packets with small Time to Live (TTL) values
  - The Linux traceroute and Windows tracert commands support a -6 option to force IPv6 tracerouting using Hop Limit fields
- IPv4 TTL and IPv6 Hop Limit is the number of hops the packet should go before being discarded
  - An ICMP Time Exceeded message comes back
- Based on the source address of the TTL-exceeded message, you can determine the router for a given hop
- The scanning system increments TTL for each packet to determine each router hop



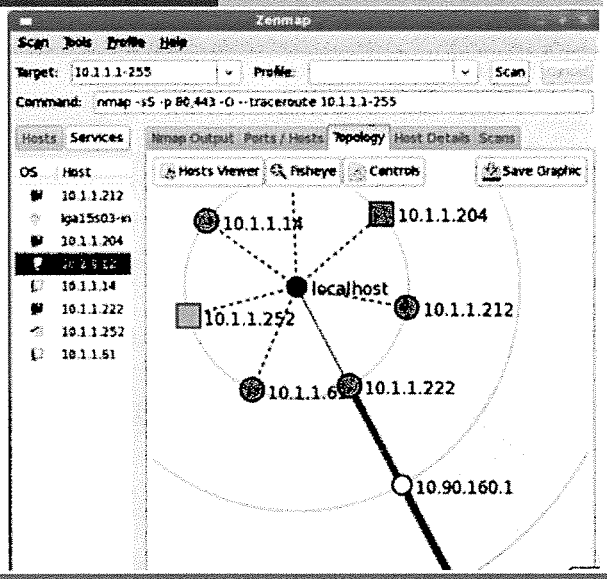
How do the IPv4 TTL field and IPv6 Hop Limit field work? When a router receives an incoming IP packet, it first decrements the value in the TTL (or Hop Limit) field by 1. For example, if the incoming packet has a TTL value of 29, the router sets it to 28. Then, before sending the packet on toward its destination, the router inspects the TTL field to determine if it is zero. If the TTL is zero, the router sends back a Time Exceeded message to the originator of the incoming packet, saying, "Sorry, but the TTL wasn't large enough for this packet to get to its destination." TTL was created so that packets would have a finite lifetime (up to 255 hops), and we wouldn't have phantom packets circling the Internet for eternity.

This TTL field is especially useful in determining how the various components of a network are interconnected. The Linux and UNIX traceroute command, the Windows tracert command, and Nmap all rely on making variations in this field during network mapping to measure the paths that packets take across the network.

By sending a series of packets with various TTL values, the traceroute and tracert tools measure all routers from a given source to any destination. As shown in the slide's graphic, they start out by sending a packet from the source machine with a TTL of one. The first router receives the packet, decrements the TTL to zero, and sends back a Time Exceeded message. What is the source address of the Time Exceeded message? It's the IP address of the first router on the path to my destination. Bingo! I know the address of the first router on the way to my destination. Next, I send out a packet with a TTL of 2. The first router decrements the TTL to 1 and forwards the packet. The second router in the path decrements the TTL to zero and sends a Time Exceeded message. I now have the address of the second hop. This process continues as I send packets with higher TTLs until I reach my destination. At that point, I know every hop between me and my target. The traceroute and tracert commands determine whether to use IPv4 or IPv6 based on the format of the IP address they are provided. However, you can force traceroute or tracert to use IPv4 by using -4 in your command, and force IPv6 using -6.

Zenmap Network Map Output

- Output shows topology
  - Cumulative view of recent scans
  - Supports changing focus, zooming, and fisheye view



Once Nmap finishes conducting a network sweep and its tracerouting activities, the Zenmap GUI can provide an interactive graphical portrayal of the network. This output is a cumulative view of recent scans conducted by Nmap, showing each system identified during ping sweeping, along with the series of connections between the systems.

The Zenmap graphical view allows for zooming in to targets or specific portions of the network. The user can simply double-click a host to move it to the center of the map. There is also a fisheye view that lets the user have one portion of the network map zoomed in, while making other portions of the graph smaller.

On the left-hand side of the display, we can see each host with a graphical depiction of its operating-system type, discovered using the active OS fingerprinting features of Nmap that we discuss later.

Defending Against Network Mapping (1)

- Preparation
  - You could disable incoming ICMP echo request messages
    - But your users couldn't ping you
    - That may be OK
  - You could disable outgoing ICMP Time Exceeded messages
    - But your users couldn't traceroute all the way to you
    - Is that actually all that bad
- Identification
  - IDS signatures looking for ping sweep or traceroutes
  - Many false positives possible

How do you defend against this type of network mapping?

I'm a fan of filtering incoming ICMP messages to anything on my network, except perhaps a web or FTP server. All other ICMP coming from a hostile network (such as the Internet) can be dropped.

Of course, if your ISP wants to ping you as a keepalive signal, you could set up a filter that allows ICMP from its source address/network.

You could disable outgoing ICMP Time Exceeded messages, but your users couldn't traceroute all the way to you. That might not be a bad thing. Many sites are starting to block all incoming and outgoing ICMP messages.

Defending Against Network Mapping (2)

- Containment
  - If you notice a particularly frequent ping sweep, you could temporarily block source address
  - Mark such rules as temporary in a comment field, and then purge them on a regular basis (such as monthly)
- Erad, Recov: N/A

If you notice a particularly frequent ping sweep or traceroute coming from a single IP address or network, you could filter that address in your border router or firewall.

If you start blocking source addresses, make sure you regularly examine your filters so they don't become too large and unwieldy. I usually block something for a few weeks or months, and the attackers go away. I can then remove the filter fairly safely.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

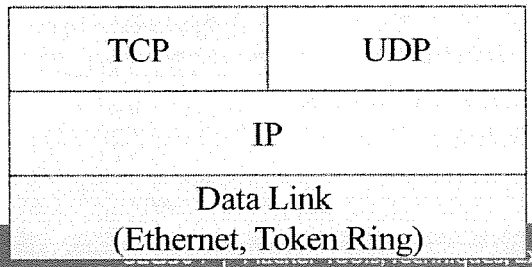
SCANNING

1. War Dialing
2. War Driving
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. **Port Scanning with Nmap**
   - Lab: Nmap
5. Evading IDS/IPS
6. Vuln Scanning with Nessus
   - Lab: Nessus
7. SMB Sessions
   - Lab: SMB Sessions

So, the attackers now have a feel for the target environment's network topology. They now turn to their port-scanning tools to find open ports on the target systems.

Port Scanners and the TCP/IP Family — Port Scanning

- Port scanners are a must for any attacker's toolbox
- They help identify openings on a system and the type of system
  - Allow an attacker to focus an attack
- Most Internet applications use TCP or UDP
  - **TCP**: Connection-oriented (for example, sequence preserved and retransmission)
  - **UDP**: Sessionless (for example, datagram; get it there if you can)



To understand port scanners, we first need to do a brief protocol review.

The Internet Protocol (IP) is the common protocol used to move information around the Internet. IP includes the source and destination address of each packet. The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ride on top of IP. That is, TCP and UDP messages are plopped inside of an IP message. TCP is session-oriented, in that it applies sequence numbers to messages and tries to deliver them in appropriate order. It also resends dropped messages. UDP makes best-effort delivery, but messages may be dropped or delivered out of order.

If someone is going to access your system, ports are the entry points or the doors and windows into your system. Therefore, a list of open ports gives an attacker various possible avenues for compromising the system.

Port scanners are a must for any attacker's toolbox. They help identify openings on a system and the type of system and therefore allow an attacker to focus on an attack.

TCP and UDP Ports

Port Scanning

- TCP and UDP have ports: a field in the TCP and UDP headers
  - Total of 65,536 (times 2) ports
  - Yes, you sometimes see packets going to or from port 0
- Port scanners send packets to various ports to determine what's listening
  - Find tcp 80, web server
  - Find tcp 445, Windows Server Message Block
  - Find udp 53, DNS server
  - And more
- Current official port numbers can be found at IANA (http://www.iana.org/assignments/service-names-port-numbers/)

There are $2^{16} = 65,536$ different TCP ports and 65,536 different UDP ports. Common services listen at well-known port numbers. The latest port listing is maintained by the IANA. For example

- tcp 80 usually indicates a web server
- tcp 445 usually indicates Windows Server Message Block (SMB)
- udp 53 usually indicates a DNS server
- tcp 6000 usually indicates a X Window server

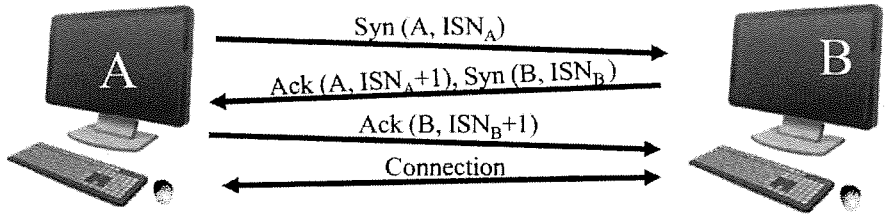
Each open port (for example, listening) offers a potential way into a system.

With a list of open ports, the attacker can get an idea of which services are in use by consulting IANA.

TCP Three-Way Handshake

Port Scanning

- Initial SYN establishes sequence number for A to B
  - Usually, B must remember this, allocating state in its connection queue
- Response SYN-ACK establishes sequence number for B to A



All *legitimate* Transmission Control Protocol (TCP) connections (for example, HTTP, telnet, FTP, and so on) are established through this three-way handshake.

The handshake allows for the establishment of sequence numbers (ISN = Initial Sequence Number) between the two systems. These sequence numbers are used so that TCP can provide for reliable packet delivery in sequential order. Sequence numbers are used for sequencing and retransmissions.

Six control bits describe the packet's role in the connection:

- **SYN**: Synchronize
- **ACK**: Acknowledgment
- **FIN**: End a connection
- **RESET**: Tear down a connection
- **URG**: Urgent data is included
- **PUSH**: Data should be pushed through the TCP stack

Note that the control bits can be set independently of one another. For example, you could have a SYN-ACK packet with both bits set.

Closer Look at the TCP Header

Source Port			*Destination Port*
Sequence Number			
Acknowledgment Number			
Hlen	Rsvd	*Cont Bits*	Window
Checksum			Urgent Pointer
TCP Options (if any)			Padding
Data			
.....			

The TCP header includes the source and destination ports, as well as other elements that a port scanner manipulates as it generates packets, such as the TCP control bits.

Also, the sequence number for this packet, and the ack number for previous packets, are included in the header.

Closer Look at the UDP Header

Source Port	*Destination Port*
UDP Message Length	UDP Checksum
Data	
.....	

The User Datagram Protocol (UDP) does not have a three-way handshake or sequence numbers and is, therefore, a "stateless" protocol. Retransmissions are handled by the application or are not done at all. UDP is also known as the Unreliable Damn Protocol. It is useful for applications that value speed over reliable delivery, such as voice or video transmissions. Packet sequencing takes up processing overhead, while an occasional dropped packet has minimal impact on a user's perception of voice or video.

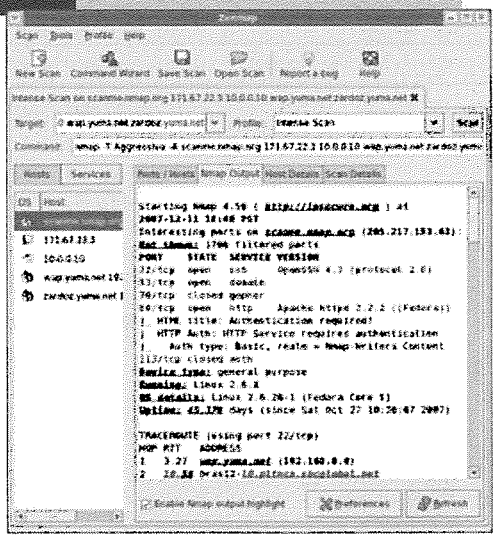
UDP is also used for simple query/response type applications, such as databases or DNS. For such services, I send in one packet and get one response. Therefore, there's no need for sequence numbers for a series of packets.

The UDP packet header is simple. It includes the source port and destination port. No sequence numbers are included.

Nmap – Scan Types

All these are TCP scans

- Ping sweeps and ARP scans
- Connect TCP scans
  - Uses three-way handshake
- SYN scans (aka "half-open" scans)
  - Only do initial SYN
  - Harder to detect and much quicker
- ACK scans
  - Stealthy and bypass some filters
- FIN scans
  - Stealthy and bypass some filters
- FTP Proxy "Bounce" scanning
- "Idle" scanning
- UDP scanning
  - Send empty payload to most ports
  - Send protocol-appropriate payload to about a dozen ports (53, 111, 161, etc.)
- Version scanning
- IPv6 scanning (-6) now supported for all scan types
  - Used to be just for ping sweeps (-sP), TCP connect scans (-sT), and version scans (-sV)



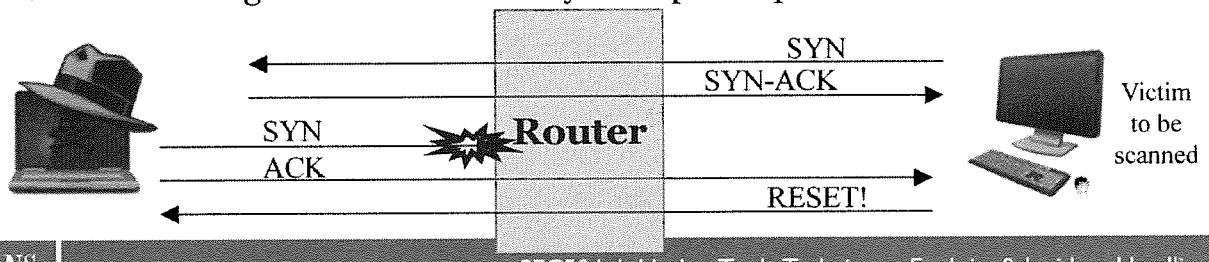
Nmap allows for conducting numerous types of scans:

- **Ping sweeps**: Send a variety of packet types (including ICMP Echo Requests, but many others as well).
- **ARP scans**: Identify which hosts are on the same LAN as the machine running Nmap. The ARP scan does not work through a router, because ARP traffic just goes on a single LAN.
- **Connect scans**: Complete the three-way handshake; are slow and easily detected. Because the entire handshake is completed for each port in the scan, the activities are often logged on the target system.
- **SYN scans**: Only send the initial SYN and await the SYN-ACK response to determine if a port is open. The final ACK packet from the attacker is never sent. The result is an increase in performance and a much more stealthy scan. Because most host systems do not log a connection unless it completes the three-way handshake, the scan is less likely to be detected.
- **ACK scans**: Particularly useful in getting through simple router-based firewalls. If a router allows "established" connections in (and is not using any stateful inspection), an attacker can use ACK scans to send packets into the network.
- **FIN scans**: Send packets with the FIN control bit set in an effort to be stealthy and get through firewalls.
- **FTP Proxy "Bounce Attack" scans**: Bounce an attack off a poorly configured FTP server.
- **"Idle" scans**: This scan type can be used to divert attention, obscuring the attacker's location on the network.
- **UDP scanning**: Helps locate vulnerable UDP services. For most UDP ports, Nmap sends packets with an empty payload. But, for about a dozen specific ports, Nmap includes an application-appropriate payload for the given port, including UDP port 53 (DNS), 111 (portmapper), 161 (SNMP), etc.
- **Version scanning**: Tries to determine the version number of the program listening on a discovered port for both TCP and UDP.
- **IPv6 scanning**: Iterates through a series of IPv6 addresses, scanning for target systems and ports, invoked with the "-6" syntax. Today, all Nmap scan types support a -6 option. In older versions of Nmap, IPv6 scans were limited to ping sweeps to identify target host addresses in use, TCP connect scans, and version scans only.
- **RPC scanning**: Identifies which Remote Procedure Call services are offered by the target machine.
- **TCP sequence prediction**: Useful in spoofing attacks, as we shall see in a short while.

Nmap – ACK Scanning

Port Scanning

- Suppose you want to allow outgoing connections, but not incoming (network diode)
- You may configure a router to allow in only established connections (for example, connections with ACK control bit set)
  - Allow outgoing SYNs
  - Allow incoming connections only if ACK control bit is set
- This blocks session initiations from the outside
- But an attacker can conduct ACK scan to get past some filters
- ACK scans are useful for mapping, but not for port scanning
- Great for finding sensitive ***internal*** systems post exploitation



A router may allow outgoing SYNs and their incoming responses (for example, established connections with the ACK control bit set), but not incoming SYNs. The reason is that incoming SYNs indicate the beginning of a new session. In this case, the attacker can scan with ACK packets, which the router allows into the network.

A stateful packet filter remembers the outgoing SYNs, so it only allows the incoming packet if it is tied to an earlier outgoing packet. Therefore, an ACK scan does not work through a properly configured stateful packet-filtering device.

ACK scans cannot reliably determine which ports are open or closed, unfortunately for the attackers. Different systems respond in different ways to an unsolicited ACK packet to an open or closed port. In other words, the returned RESET doesn't necessarily indicate that the port is open or closed. However, it DOES indicate that there is a system at the address. So, the ACK scan result can be used to do network mapping instead of a ping sweep. It just cannot be used for a port scan.

This is a great approach for finding internal sensitive systems, such as network management servers, SIM servers, and remote access servers. This technique works because many organizations use simple IP address filtering for segmentation of sensitive LAN segments.

Nmap Active OS Fingerprinting

Port Scanning

- Attempts to determine the operating system of target by sending various packet types and measuring the response
  - Concept originated with a tool called QueSO
- Original ("first generation") Nmap OS fingerprint techniques
  - TCP sequence prediction
  - SYN packet to open port
  - NULL packet to open port
  - SYN|FIN|URG|PSH packet to open port
  - ACK packet to open port
  - SYN packet to closed port
  - ACK packet to closed port
  - FIN|PSH|URG packet to closed port
  - UDP packet to closed port

Different operating systems exhibit different behaviors under these conditions

In addition to finding out what ports are open on a system, an attacker also wants to determine on which platform the system is based. By determining the platform, an attacker can further research the system to determine the particular vulnerabilities it is subject to. For example, if the system is a Windows XP box, the attacker can utilize www.packetstormsecurity.org to hone the attack.

The RFCs defining TCP specify how a system should respond during connection initiation (the three-way handshake). The RFCs do not define, however, how the system should respond to the various illegal combinations of TCP flags. Because of this lack of a coherent standard in the face of illegal combinations, different implementations of TCP stacks respond differently to illegal flags.

Nmap has a database of how various systems respond to these flags. By sending out various packets to both open and closed ports, Nmap can determine what type of platform the system is running.

This technique is called active OS fingerprinting, because it is sending packets out to measure the response of the machine in an effort to identify the OS type. It is active because it sends packets.

Nmap OS Fingerprinting

- More than 30 different methods are included in second-generation fingerprinting, including
  - TCP ISN greatest common denominator (GCD)
  - TCP ISN counter rate (ISR)
  - TCP IP ID sequence generation algorithm (TI)
  - ICMP IP ID sequence generation algorithm (II)
  - Shared IP ID sequence boolean (SS)
  - TCP timestamp option algorithm (TS)
  - TCP initial window size (W, W1 - W6)
  - IP don't fragment bit (DF)
  - IP initial Time to Live guess (TG)
  - Explicit congestion notification (CC)

The second-generation active OS fingerprinting of Nmap includes more than 30 different tests to determine the operating system type of a target. Included in these tests are measures of the TCP sequence numbers of responses, such as their greatest common denominator and how quickly they change over time. Also, Nmap measures the changes in IP ID values for responses to TCP and ICMP packets. Some operating-system types have different sets of IP ID numbers for TCP versus ICMP, while others do not. (Windows uses the same incremental number for both sets of protocols.)

It also looks at TCP timestamp behavior and TCP window sizes the target system negotiates. Also, Nmap evaluates the behavior of the system to a message with the Don't Fragment bit set in its IP header. It attempts to guess the initial Time To Live for the packet by rounding it up to the next nearest power of 2, because many system types have a TTL of 2**n or (2**n)-1. Finally, Nmap analyzes the explicit congestion notification behavior of the target machine to see how it handles the extended control bits associated with congestion control.

Other Scanners: Masscan

- Traditional port scanning can be "slow"
  - When scanning thousands of hosts
  - Keeping track of all SYN and SYN/ACKs
- However, Masscan and other tools like it separate out the SYNs and the SYN/ACKs
  - One part of the program sends SYN packets very quickly
  - Another separate part waits for SYN/ACKs
- By decoupling the two halves of the Three-way handshake speed is greatly improved
- Concept was originally released by Dan Kaminsky in Scanrand
- Created by Robert David Graham
- https://github.com/robertdavidgraham/masscan

One of the problems with scanning a large number of systems is the inherent restrictions of TCP/IP and the three-way handshake. Good and proper scanners will gracefully attempt the three-way handshake when trying to determine which ports are open on a target system. While this works fine for scanning a handful of systems, it can be very time consuming when scanning thousands of computers. Years ago, Dan Kaminsky developed a tool called Scanrand which separated the process sending the SYN packets from the process receiving the SYN/ACK responses. By doing this, it allows the scanner to run in a massively parallel process thereby scanning thousands of systems in very short order. This process has carried forth to a tool called masscan by Robert David Graham.

Other Scanners: EyeWitness

- Takes screenshots of websites, VNC and RDP servers
- Can be very effective to sort through hundreds of different websites
- Attackers and testers look for default pages, out-of-date servers, RDP servers which show domains, index-able directories, etc.
- Many vulnerabilities are not necessary vulnerabilities which have a Metasploit module
  - Finding backup files and install scripts on web servers can lead to easy access to external systems
- Developed by Chris Truncer
  - https://github.com/ChrisTruncer/EyeWitness

When attacking websites one of the larger tasks for the bad guys, and good guys testing there systems, is trying to identify what is running on hundreds, if not thousands of webservers. To get around this issue an attacker can run an excellent tool called EyeWitness. This tool will take a screenshot of every webserver it detects. It is based on the initial PeepNtom research of Tim Tomes with Black Hills Information Security. This can be used to very quickly identify the purpose of multiple websites simply by reviewing the pictures.

What an attacker is looking for are things like default pages, management pages and pages which may be serving up index-able files to the Internet. Remember, there is more to attacking a network than simply discovering ports and looking for exploit.

Below is a nice usage guide:

https://www.christophertruncer.com/eyewitness-2-0-release-and-user-guide/

Get it here:

https://github.com/ChrisTruncer/EyeWitness

Proxying Scans: Remux

- Proof of concept tool to demonstrate scanning through multiple open proxies online
- Revers multiplexes connections
- Browser connects to remux.py
- Remux.py federates connections through the proxies
- The list of proxies are automatically downloaded at runtime
  - You can also specify a list of known good proxies at runtime
- Makes identifying the scanning system very difficult
- When remux.py starts, it is very slow and buggy
  - It slowly learns which proxies are alive and which are no longer active
  - Gets more stable and faster over time

When scanning one of the things which is unacceptable to an attacker is getting caught. To get around this, some attackers may bounce their scanning activities through multiple different bots or other compromised systems. Another approach is using a reverse multiplexor. Something like remux.py. What remux.py does is set up a proxy listener. The attacker then configures their browser or scanning tool to go through remux. Next, it pulls down a list of known proxy servers online. When remux.py first runs the data transfer is quite bad. This is because many of the proxies it pulls down are either very slow or offline completely. So, remux.py learns which ones are active and slowly builds a list of known good proxies. After a while, it will federate scanning or browsing activity through dozens of different proxies. This makes trying to determine the actual source of the attack or scanning activity very difficult to do.

Remux.py is provided by Black Hills Information Security and is on the class USB.

Port Scanners – Defenses

Port Scanning

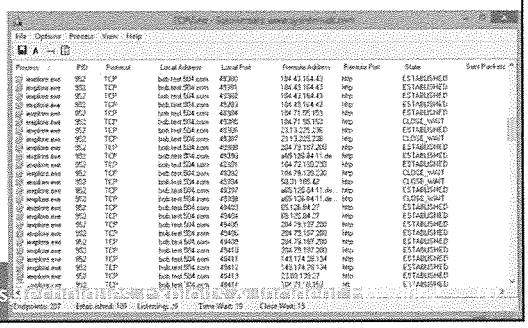
- Preparation
  - Close all unused ports by shutting off services and applying filters
  - Utilize stateful packet filters and/or proxy firewalls
  - Utilize an intrusion detection system
- Identification
  - Several IDS signatures for port scans
  - Log analysis shows pesky connection attempts
- Cont, Erad, Recov: N/A

When you bring a new system online, you should be familiar with the ports that are open on the box and why they are required.

The only way to be secure is by using a Defense-in-Depth posture. In addition to the information in the slide, utilize an intrusion detection system.

Locally Checking for Listening Ports on Windows

- In Windows, run C:\> netstat –na → Shows listening TCP/UDP ports
- We can go further
  - `C:\> netstat -nao` → Shows pid
  - `C:\> netstat -nab` → Shows EXE and all DLLs used
- As a separate download, Microsoft has the Port Reporter tool
  - It periodically generates logs showing port activity
  - Free at http://support.microsoft.com/kb/837243
- For a GUI view of port usage, use TCPView



The procedure for checking locally listening ports varies between Windows and Linux/UNIX.

For Windows, you can run "netstat –na" from a command prompt to see which ports are in use.

To be even more specific and look for just listening ports, you can type

C:\> **netstat –na | find "LISTENING"**

We can go further for more info. The "-o" flag of netstat, as in "netstat –nao," shows the listening ports and the process ID of the listening process.

Finally, Microsoft added the –b flag to netstat. The –b flag indicates the EXE and all of its associated DLLs for each listening port.

Microsoft also has a tool called TCPView available as a separate download. It runs as a application, and periodically generates logs showing port activity. You can also use the command line version called TCPVcon.

Disabling Windows Services Listening on Ports

- Kill running process using Taskmgr (be careful)
- Or use **`wmic process [pid] delete`**
- Disable service in Service Control panel
  - Start→Run and type "services.msc"
  - Hit Stop
  - Set Startup type to "Disabled"
- Or, use the "sc" command
  - For a list of services, type `C:\> sc query`
  - To shut off a service, type `C:\> sc stop [service]`
  - To disable a service, type
    `C:\> sc config [service] start= disabled`
- Be careful

Don't forget that space after "start="

Once you find listening ports, you need to evaluate whether the given network service is required on the box.

If the service is not needed, you can disable it temporarily, abruptly, and unsmoothly by killing the associated process in Task Manager (if there is a single associated process). Be careful with this maneuver, because it could make your system highly unstable. Also, the process may return when you reboot the system.

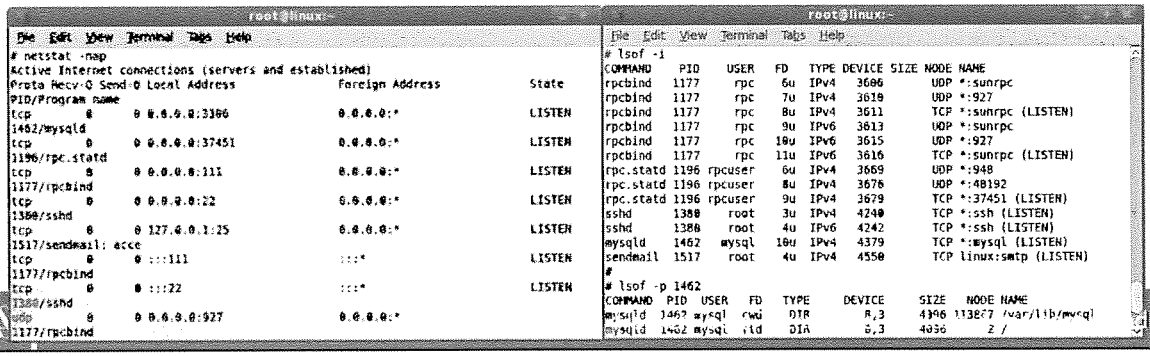
A cleaner way to disable a listening port, if the listening process was started as a Windows service, involves disabling the service itself. You can do this by running the Services control panel, which is easily invoked from Start→Run and typing services.msc. Then, double-click the offending service, hit Stop and set its Startup Type to "Disabled."

If you are more of a command-line person, you can do the same thing using the Service Controller command, "sc." To get a list of services and their status, type "sc query." To stop a service, type "sc stop [service]" (this works only temporarily; the service returns at reboot). To permanently disable a service, type "sc config [service] start= disabled." Remember, you must have a space after the "start=" and before the "disabled" or the syntax doesn't work. Also, you cannot have a space between the start and the =.

Finally, please, please, please be careful with this. If you disable a crucial service, you could make your system highly unstable.

Locally Checking for Listening Ports on Linux/UNIX

- On Linux/UNIX, you could run
  - `# netstat -nap`
  - Shows listening ports, PID, and program name
  - To get more detail, run



By default, Linux/UNIX gives us far more detail about listening TCP and UDP ports using built-in tools.

We could use the netstat command, with the –p flag to show PIDs and program names, as in "`netstat -nap`." Note that for full information from the "-p" flag, you have to run the command as root.

We can get even more detail about processes listening on ports by using the lsof command, which I find absolutely essential in analyzing my own UNIX boxes. I run the lsof command with the "-i" flag to list all TCP and UDP port usage. (It also includes X.25 network usage, if you are interested.) Then, using the PID of the listening process that I get from "`lsof -I,`" I get a lot more detail from the "-p" flag by typing "`lsof -p [pid].`" That command shows all files associated with the listening process, including the program file that it ran out of, any libraries it uses, all configuration files that it has opened, and numerous other juicy tidbits.

Disabling Linux/UNIX Services Listening on Ports

Port Scanning

- Kill running process using kill or killall (be careful)
- Disable service by reconfiguring inetd or xinetd
  - **inetd**: Comment out lines in /etc/inetd.conf
  - **/etc/xinetd.d**: Delete file or make sure it contains "disable=yes"
- Disable service by altering /etc/rc.d files or running chkconfig (which alters rc.d automatically)

```
# chkconfig --list
# chkconfig [svc_name] off
```

- Takes effect on next boot
- Chkconfig is built into RedHat, and available for other Linuxes and Solaris
- Be careful not to kill critical processes

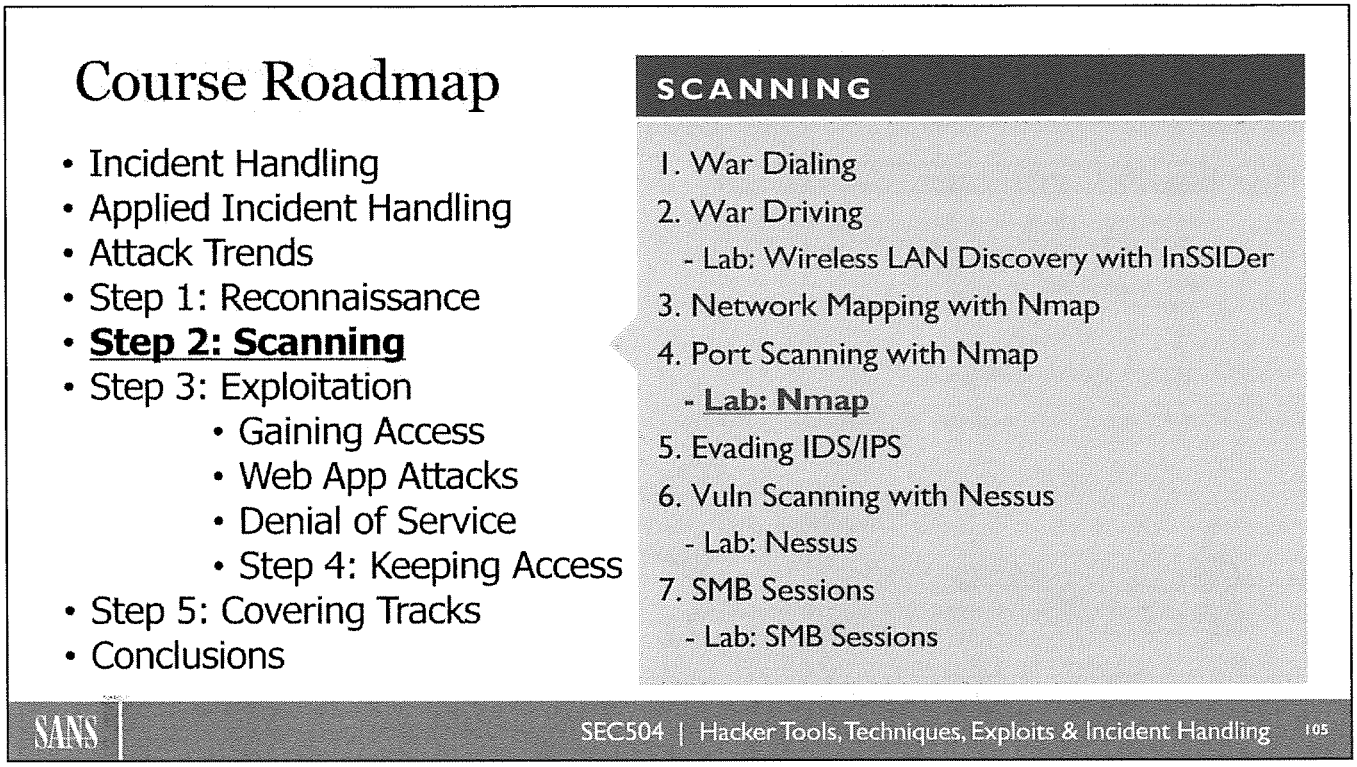
To stop a process on Linux or UNIX, you can use the "kill [pid]" or "killall [process_name]" commands. Be careful with these, because they could make your system unstable. Also, this only temporarily disables the process. It may restart automatically or during the next boot.

The process for disabling a service listening on a port permanently depends on whether the service is invoked by inetd, xinetd, or one of the service initialization scripts.

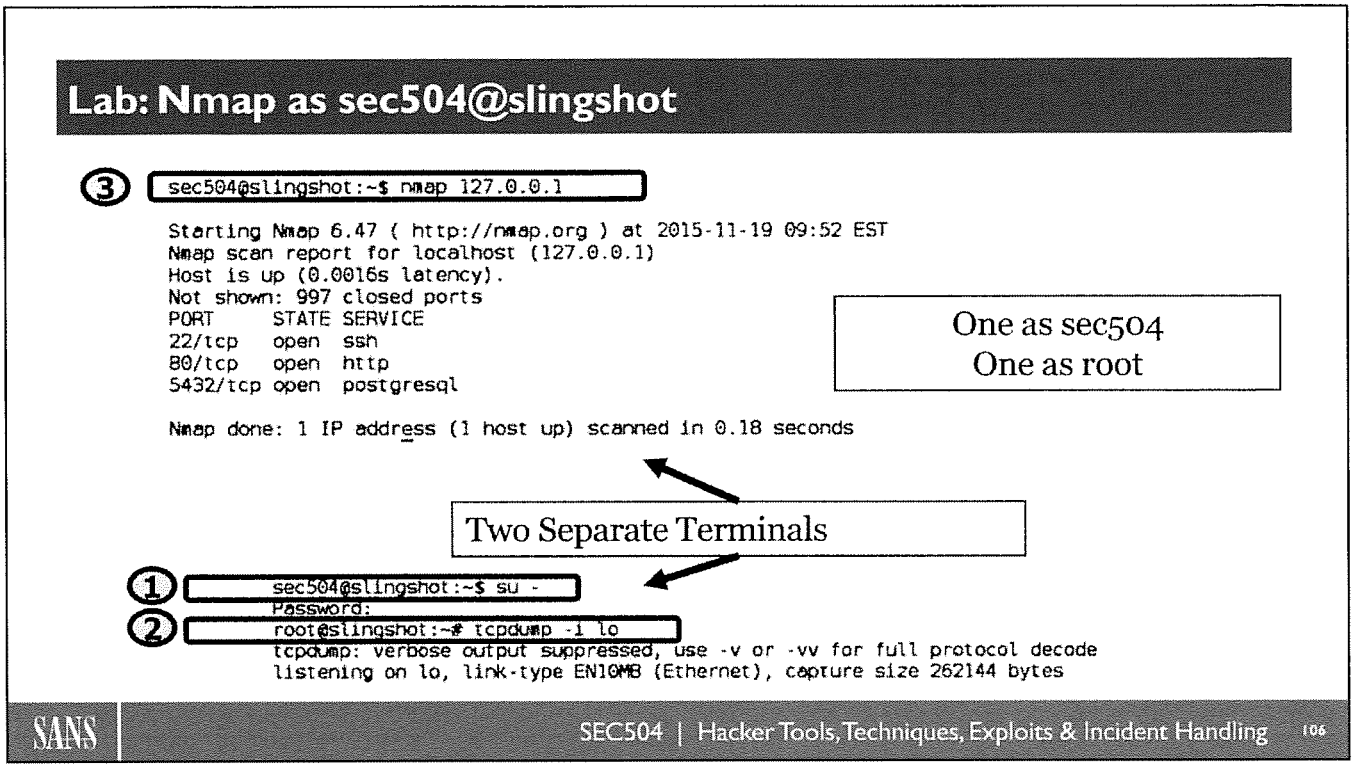
If the service is started by inetd, you can comment out its line in /etc/inetd.conf by placing a "#" at the beginning of the line.

If the service is started by xinetd, you can delete the file /etc/xinetd.d/[service] or edit that file so that it contains a line that says "disable=yes."

If the service is started by one of the service initialization scripts, it will have a link called S[Number][Service] in the directory /etc/init.d. You can shut off such services by editing the rc.d file for each runlevel on your system, removing the S links for that runlevel. More easily, you can use the chkconfig command, which is built into RedHat and Mandrake Linuxes. It's also available as a separate download for Debian Linux, and there is a port for Solaris. To get a list of services installed on the machine, as well as their configuration for startup, run the command "chkconfig --list." To stop a service, you can type "chkconfig [svc_name] off." That service automatically is disabled in all the appropriate rc.d directories.



Let's conduct a hands-on lab that looks at Nmap's features and capabilities.



When you first log in to your 504 VM, you are logged in as the sec504 account.

First, we open a simple terminal by select the terminal icon on the desktop. Note that, when you first log in, your command prompt is a dollar sign. Now, let's

open a second terminal (by once again clicking the terminal icon on your desktop) and becoming root with the following command:

1.$ **su -**

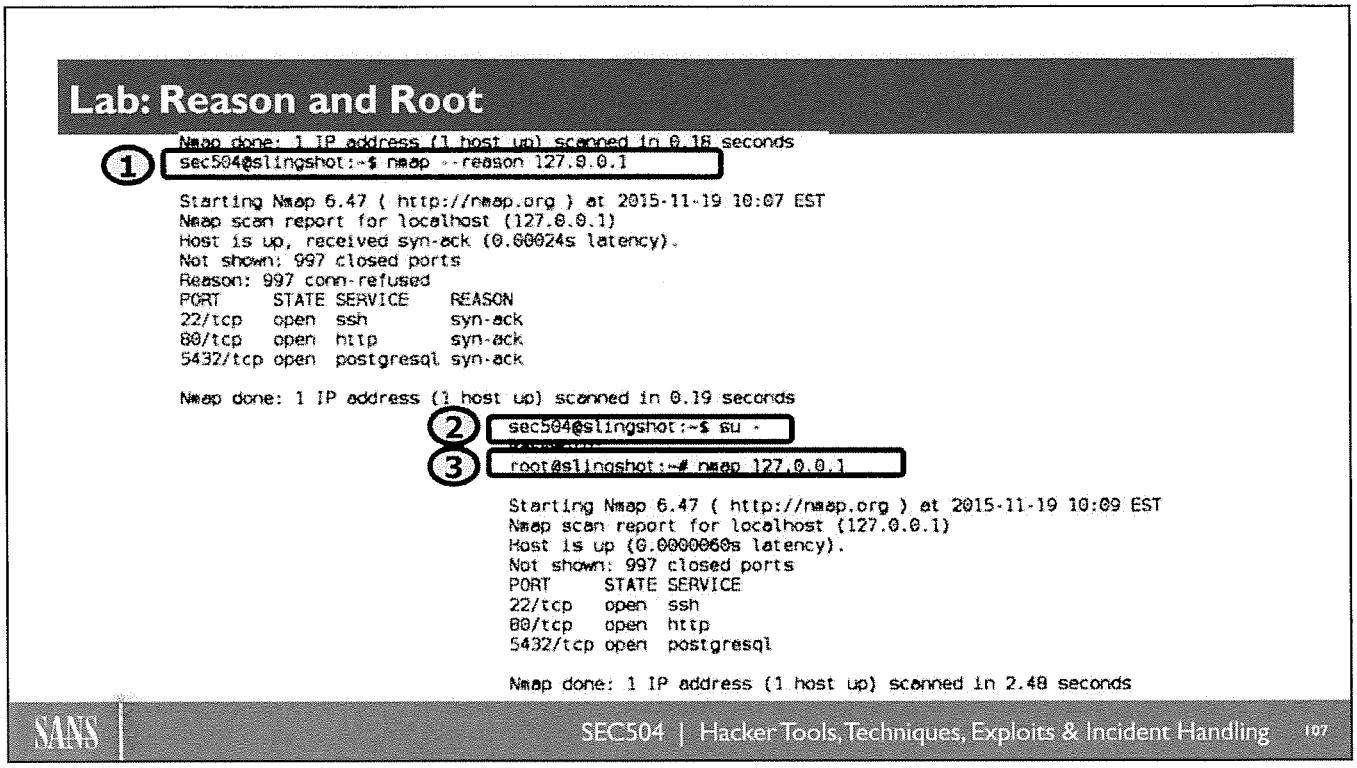
When asked for a password, enter the root password of **root**. Now, note we have two separate terminals with two separate levels of access open at the same time. The terminal on the left has you logged in as student, and the terminal on the right is logged in as root. Note that root is a powerful account and one should always be careful when running your system with this level of access.

Next, we start a sniffer in the second (root@linux) terminal. This allows us to see the different Nmap scans and how the packets are different from each other based on options and privilege level.

2. # **tcpdump -i lo**

Now, let's start a basic Nmap scan in the first (student@linux) terminal. The goal of having both of these windows open at the same time is to see the stimulus (the Nmap scan) and the response (the tcpdump output).

3. $ **nmap 127.0.0.1**



By default, when not running as root, Nmap does a full TCP connect scan, so it completes the TCP/IP three-way handshake we discussed earlier. This type of scan runs a bit slower than the Nmap scan we run in a few moments. We are showing you how Nmap runs as a non-root user, because it is important as an incident handler to understand that you do not need to be root to run Nmap. Attackers who gain access to a system can still discover systems and ports even with limited privileges if Nmap is installed on a compromised host. For many Linux systems Nmap is installed by default.

Another great option Nmap has is the capability to tell you why it believes a port is open. For example, for some scans (such as UDP), Nmap lists a port as Open|Filtered. This means it did not receive a response, so the UDP port in question may be either open or filtered. Another example is if a firewall drops a packet, Nmap responds with Filtered as the status of a port.

Let's try this with the following command:

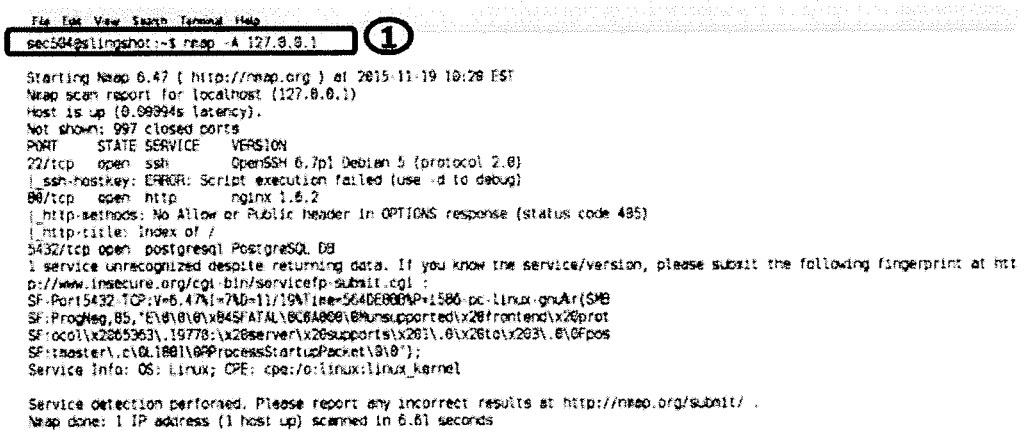
1. $ **nmap --reason 127.0.0.1**

Note that there are two dashes before reason.

Now, in the first window (sec504@slingshot), let's become root and scan again. Leave the second root window open so you can see the scan when it runs.

2. $ **su -**
3. # **nmap 127.0.0.1**

Lab: Getting More Information



Although identification of ports is a good thing, today's modern attacker (and incident handler) needs more data. For example, in book one, we discussed the importance of being able to identify ports and services across entire environments. We do this for two reasons: 1) to possibly identify vulnerable services, and 2) we may want to identify possible backdoors the bad guys are using.

Nmap has a powerful option called –A. This option enables OS detection, version detection, script scanning, and traceroute. It gives you far more information than a simple syn or TCP connect scan.

Let's run it now:

1. # **nmap -A 127.0.0.1**

Note: This takes a while to run.

As you can see, it provides us with not only the ports, but also now identifies the full service version (such as nginx 1.6.2). It also queries the services to identify what commands it supports (for example, smtp-commands). This is critical because there will be times when Nmap does not have a proper fingerprint for a service. In these situations, it provides you with banner information for the service. With this information, an incident handler can then do a Google search of the banner information Nmap provides and, in some situations, discover a new backdoor on your environment.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. War Dialing
2. War Driving
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. **Evading IDS/IPS**
6. Vuln Scanning with Nessus
   - Lab: Nessus
7. SMB Sessions
   - Lab: SMB Sessions

Now, we talk about evading intrusion detection systems by using packet fragmentation techniques.

IP Fragmentation Attacks

Evading IDS/IPS

- Useful in avoiding detection by some network-based intrusion detection systems (IDS) and network-based Intrusion Prevention Systems (IPS)
  - Useful in getting around older packet filters in routers and firewalls
- Attackers carefully use these techniques while they scan (Step 2) and attempt to gain access (Step 3) and cover their tracks (Step 5)

When IP packets are sent across a network, at certain points in the network (usually at slower links), the packets can be broken up into smaller pieces. This process is known as fragmentation. Once a packet is fragmented, it is not reassembled until it gets to the destination. Attackers carefully utilize the techniques to hide what they are doing.

Packet fragmentation can be used to confuse network-based intrusion detection systems (IDS) and Intrusion Prevention Systems (IPS). Also, IP fragmentation attacks exploit holes in the way some older packet filters are implemented. It's useful in getting around older packet filters in routers and firewalls (vintage Cisco IOS 10.3, approximately 1996–1997).

Refresher of the IPv4 Header

Vers	Hlen	Service Type	Total Length			
Identification			Res	***DF***	***MF***	***Fragment Offset***
Time to Live		Protocol	Header Checksum			
Source IP Address						
Destination IP Address						
IP Options (if any)						Padding
Data						
.....						

The Fragment Offset identifies where in the reassembled packet this particular fragment belongs. Each fragment includes the relative offset from the start of the payload in this field, in increments of 8-byte slots. Also, all fragments that are the part of the same bigger packet have an identical IP Identification field. That's how the IP stack knows that all these fragments belong together. The MF bit tells the system that more fragments are on the way. The DF bit tells systems "don't fragment." Note that the IPv4 header includes these options for fragmentation. IPv6 does not support fragmentation at Layer 3.

Analysis of Fragmentation

Evading IDS/IPS

- The following is tcpdump of fragmentation:

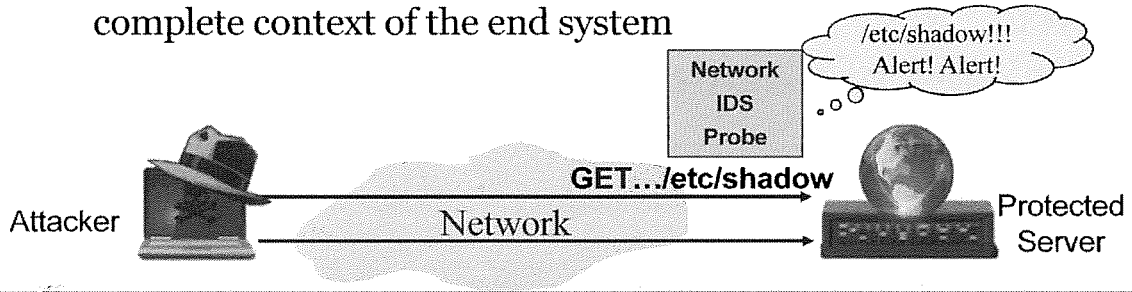
```
10.10.75.1 > 10.10.10.45: icmp: echo request (frag
  21223:1480@0+)
10.10.75.1 > 10.10.10.45 : (frag 21223:1480@1480+)
10.10.75.1 > 10.10.10.45 : (frag 21223:1048@2960)
```

- tcpdump calls the IP ID field the "frag" number
- Only first fragment has embedded protocol header
- A single fragment does not include the context for how the information is treated by the receiver

This ping packet has been broken into three fragments. The first two are 1480 octets, and the final is 1048 octets in length. They all have the same IP Identification field value (21223). The "+" in the first two fragments indicates that the "MF" (More Fragments) bit is on.

IDS Signature Matching

- Recall how IDS signature matching works
  - Look for anything matching the string "/etc/shadow" and warn me
  - I don't want anyone grabbing a list of user accounts
  - How can we slip things by the network IDS
  - Take advantage of the fact that the IDS doesn't have complete context of the end system



Intrusion detection systems listen in on networks looking for signatures of known attacks. Similar to virus-detection software, they perform pattern matching. They look for a certain pattern and, if they find it, they set off an alarm.

These fragmentation attacks exploit the fact that a network-based IDS and packet filters do not have the complete context of how a packet will be treated and reassembled at the end system. The network-based IDS must do virtual packet reassembly, and do it in the same manner as the protected hosts, which may use multiple different operating systems.

For example, let's look at how an IDS might do signature matching. Suppose I have a signature that looks for anything containing the text "/etc/shadow" and then warns me when it sees this pattern. I don't want anyone to grab my password file with its valuable hashes that he might try to crack.

How can we slip things by this signature and network IDS?

We could take advantage of the fact that the IDS doesn't have complete context of the end system when it reassembles packet fragments.

Let's Create Some Interesting Frags

- IP allows packets to be broken down into fragments for more efficient transport across various media
- TCP packet (and its header) are carried in the IP packet
- Many types of fragmentation attacks possible
- Two examples
  - Tiny fragment attack
  - Fragment overlap attack

To support different transmission media, IP allows for the breaking up of single large packets into smaller packets, called fragments. The higher-level protocol carried in IP (usually TCP or UDP) is split among the various fragments.

We use this fragmentation capability built into IP to bypass some network-based IDS systems. Also, older packet-filtering devices do not handle packet fragments properly, allowing an attacker to avoid detection.

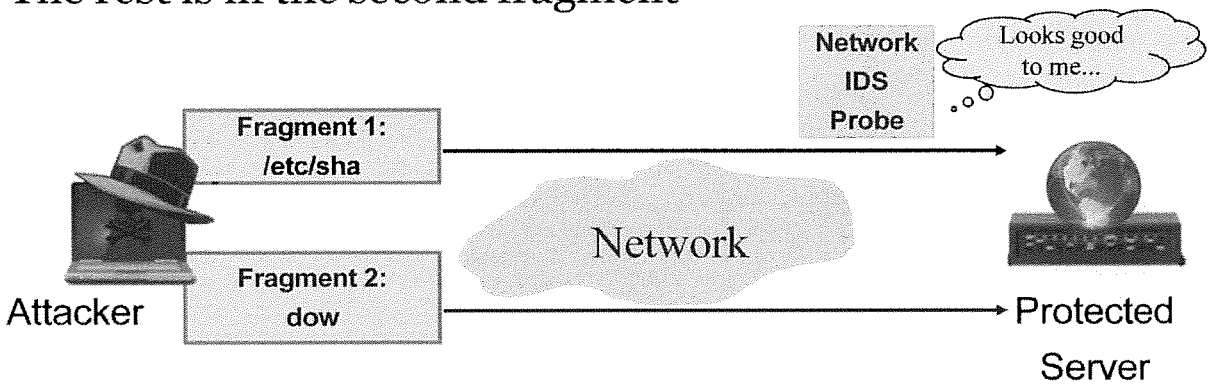
A wide range of fragmentation attacks are possible. We look at two examples:

- Tiny fragment attack
- Fragment overlap attack

Tiny Fragment Attack

Evading IDS/IPS

- Make a fragment small enough so that part of the offending string is in the first fragment
- The rest is in the second fragment

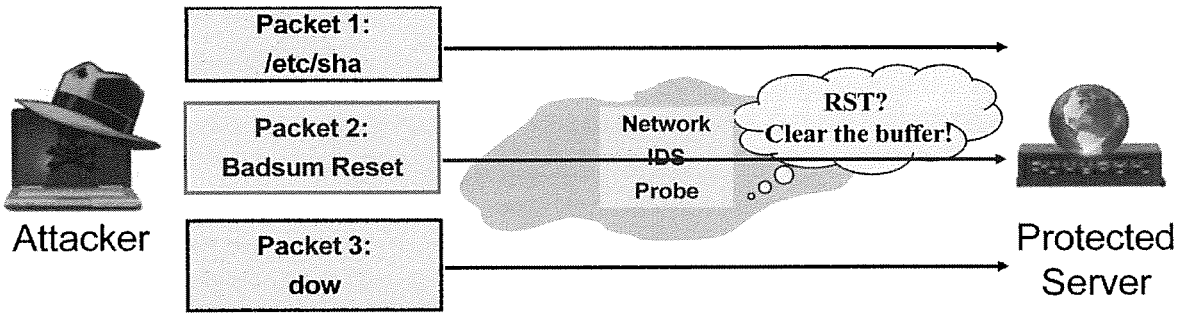


The tiny fragment attack is designed to fool the IDS by creating an initial fragment that is small. It's so small that no single fragment has everything necessary to match a signature. A very stupid IDS sensor may allow this type of attack to pass by unnoticed, because it doesn't reassemble the packets. Today, most IDS tools are capable of detecting this type of attack. Still, a large number of tiny fragments is a burden for the IDS, which has to reassemble the fragments for analysis.

Invalid TCP Checksum Bypass

Evading IDS/IPS

- Many IDS/IPS systems do not validate the TCP checksum
  - Too much overhead
- An attacker can insert a TCP Reset with an invalid checksum to clear the IDS/IPS buffer
- Target systems drop any packet with an invalid TCP checksum



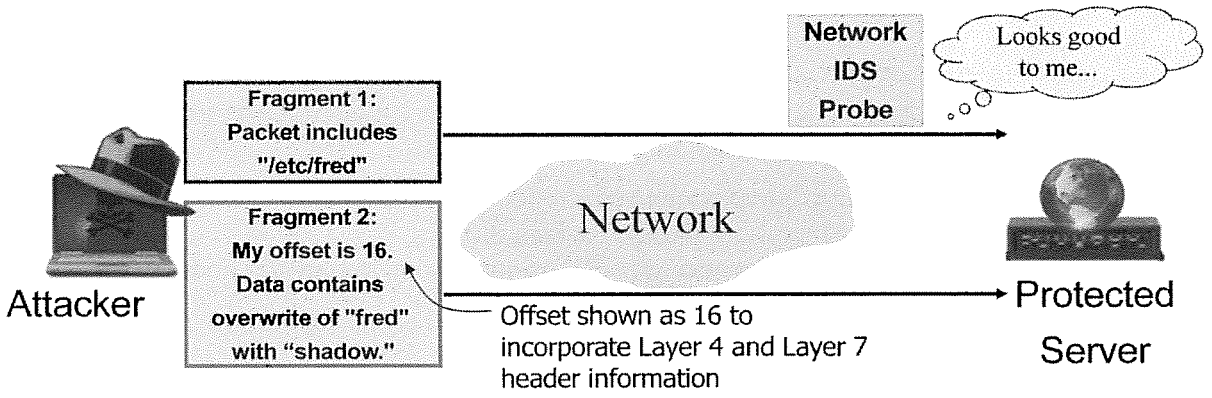
Another outstanding bypass discovered by Judy Novak involves sending a Badsum TCP Reset packet in the middle of an attack. This works because of the difference in the way a target operating system handles TCP checksums and the way many firewall/IDS/IPS systems handle the TCP checksum. According to the checksum RFCs, if a packet is received with an invalid TCP checksum, it is to be dropped. This is because nothing in the packet can be trusted.

However, many IDS/IPS/firewalls do not calculate the TCP checksum because of the processing overhead involved with checking this for every packet. So, if they receive a TCP Reset, they believe the session will be closed. So, they flush the buffer relating to that specific stream. If attackers can have their attacks signature bridge two packets and have a TCP Reset packet with an invalid TCP checksum split the two halves of the attack, they stand a good chance that their attack will bypass the IDS/IPS system.

Fragment Overlap Attack

Evading IDS/IPS

- In the second fragment, lie about the offset from the first fragment. When the packet is reconstructed at the protected server, parts of the previous fragment are overwritten.



A more insidious fragment attack is the fragment overlap attack. In this scenario, the attacker creates two fragments for each IP packet. One fragment has the TCP header, including a string that is not being looked for. The second fragment has an offset value that is a lie. The offset is too small, so that when the fragments are reassembled, the second fragment overwrites part of the first, particularly the part of the first fragment that includes some data that is sensitive.

The IDS ignores the first fragment (after all, it doesn't match the signature). The device then might ignore the second fragment (after all, it's just a fragment of the previous packet that it allowed, which doesn't include a complete match for the signature).

When the two fragments arrive at the targeted protected server, they are reassembled. The reassembly overwrites the sensitive data.

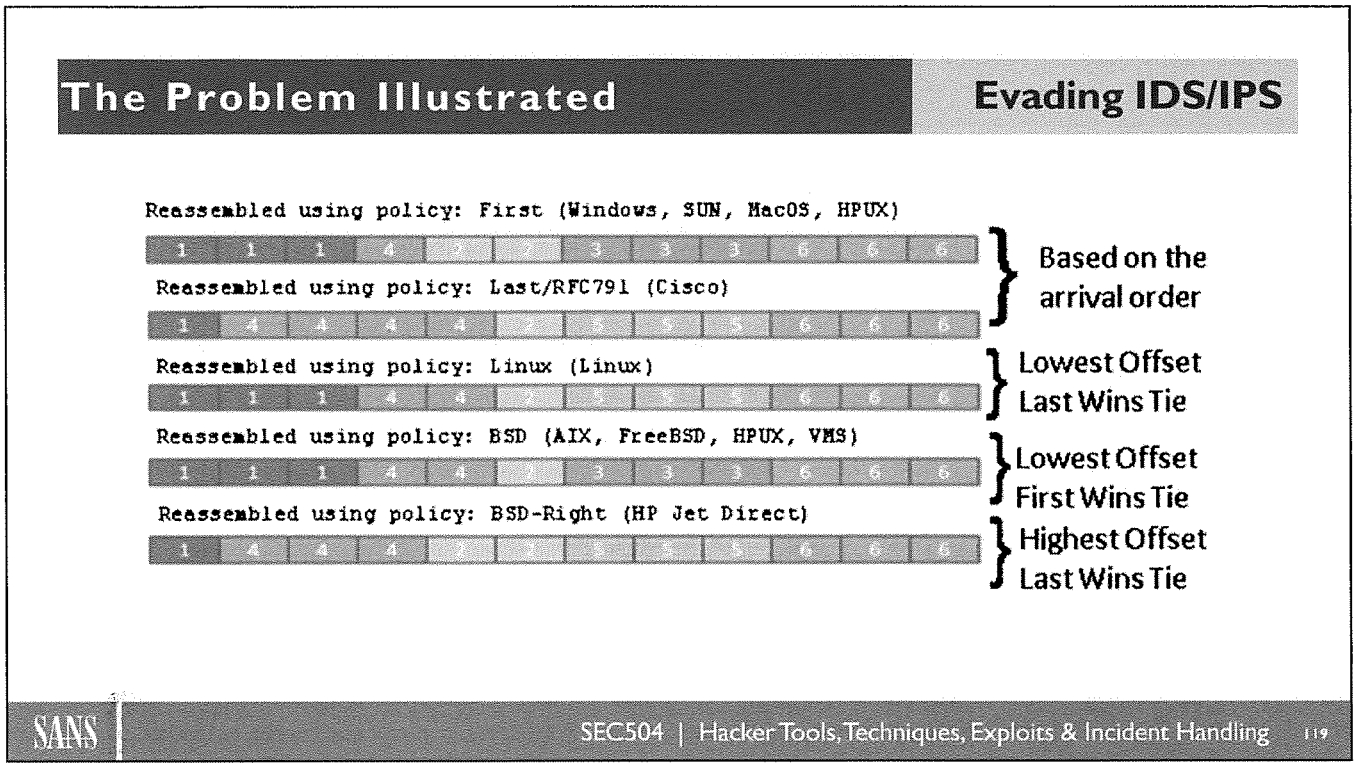
Certainly, you are wondering why the virtual fragment reassembly buffer of the IDS cannot just reassemble the packet in the proper way. The problem with this is described on the next slide.

The Problem – How to Reassemble Fragments

- There are many ways to slice packets into fragments
- Different operating systems reassemble packets differently
  - On some OSs, earliest fragment in can't be overlapped
  - On others, the fragment with the lowest offset overwrites others, regardless of arrival time
  - Complete overlap or partial overlap are handled differently in different OSs
- IDS doesn't necessarily know which method the end system will use, so it could get confused
  - Snort with Frag3 has multiple parallel virtual defrag buffers
  - Many other vendors also support parallel defrag buffers
  - It is essential your IDS/IPS has enough CPU and memory to handle the increased overhead required for defrag processing

There are many, many different fragmentation scenarios. New ones are frequently discovered.

So, why can't the IDS sensor just reassemble all the packets before it makes filtering decisions? Unfortunately, different operating systems reassemble packets differently. On some OSs, the earliest fragment (the one received first) can't be overlapped. It sticks. On other OSs, the fragment with the lowest offset overwrites others, even if it arrives later. Also, in various OSs, complete overlap or partial overlap are handled differently, thereby confusing the IDS. Today's IDS tools have a single method of reassembling the fragment. However, the IDS doesn't necessarily know which method the end system will use, so it could get confused. The Frag3 preprocessor includes multiple virtual defragmentation buffers, making Snort better at handling these attack. When implementing a defrag processor in an IDS or IPS, it is essential that you give your IDS/IPS enough CPU and memory to handle the increased processing requirements.



Here you can see the different final reassembled buffers for various operating systems. Windows, Apple, OS, Sun and some HPUX distributions all use what is called the "FIRST" policy. They accept the packet that arrived first in time and allow it to overwrite anything that arrive later in time. Cisco uses the "LAST" policy and does the exact opposite. Linux gives preference to the packet with the lowest offset in the buffer (furthest to the left) and if there is a tie it prefers the one that arrived last. AIX, FreeBSD, VMS and some HPUX distributions use a policy called BSD. The BSD policy prefers packets with the lowest offset but prefer the first to arrive when there is a tie. The BSE-Right policy which is used by HP-JetDirect cards prefers the packets that arrive with the highest offset in the buffer and in the case of a tie it prefers the packet that arrived last.

There is another possibility here. A TCP stack could prefer the packets that arrive with the highest offset in the buffer and then honor the packets that arrive first if a tie occurs. The result would be the same as BSD-RIGHT but we would have threes instead of fives. We can call this "BSD_Left". Why doesn't anyone talk about it? Well, it may be irrelevant. I am not aware of any OS that uses that technique.

Reference:Shankar, U., & Paxson, V. (2003). Active mapping: Resisting nids evasion without altering traffic. Retrieved April 29, 2012 from http://www.icir.org/vern/papers/activemap-oak03.pdf

Reference: Novak, J. (2005, April). Target-based fragmentation reassembly. Retrieved April 29, 2012 from http://www.snort.org/assets/165/target_based_frag.pdf

.

Special Thanks to SANS Instructor Mark Baggett for this slide. It is out of the SANS Python class. You should check it out.

FragRoute

Evading IDS/IPS

- Dug Song released a tool called FragRoute
  - Similar to FragRouter, but more flexible
  - Doesn't route
    - You must install it on the same machine that the attack tool itself runs on
  - Includes a language for defining specific twisted fragment attacks
    - That's from where the flexibility comes

Another tool by Dug Song, FragRoute, makes creating mystifying fragmentation schemes even more flexible for the attacker.

FragRoute differs from the older FragRouter tool in that it doesn't route. The attack tool has to sit on the same machine as FragRoute itself. That's not a huge change, but it's worth noting.

The biggest difference in FragRoute is the inclusion of a new language for creating brand-new fragmentation schemes. The old FragRouter tool had a limited number of pre-defined recipes for creating frags. The newer tool can have an arbitrary number of fragmentation recipes, limited only by the imagination and creativity of the attacker. Therefore, it's harder for IDS vendors to keep up with this.

IDS/IPS Evasion – Defense

Evading IDS/IPS

- Preparation
  - Reassemble packets before making filtering or intrusion-detection decisions
  - A firewall can do this, imposing its impression of the reassembly before the IDS and end system get the packet
  - Keep your IDS and IPS up to date
  - Supply IDS and IPS with recommended resources (network performance, processor, RAM, and hard drive)
  - For sensitive systems, use host-based IDS in addition to network-based IDS and IPS
- Identification
  - IDS signatures indicate heavy fragmentation or overlapping fragments
  - IPS can block overlapped fragments
- Cont, Erad, Rec: N/A

To avoid these problems, make sure that your systems reassemble packets before making filtering or intrusion-detection decisions. A firewall can do this, imposing its impression of the reassembly before the IDS, IPS, and end system get the packet. Everything after the firewall has the same interpretation of the packet, because the reassembly by the firewall forces the fragments into a single packet.

Also, make sure that you carefully follow your vendor's specifications for the processing power, RAM, network, and other performance characteristics for your IDS sensors and IPS tools. Those analytic capabilities require resources to keep up with the attacker's tricks.

Furthermore, a host-based IDS has the local TCP/IP stack and the complete context of the end system. Therefore, it can avoid the problems associated with fragmentation attacks.

Of course, a minimal set of absolutely required ports should be open in the first place. This minimizes the chance that an attacker will find a sensitive port through scanning with or without fragments.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. War Dialing
2. War Driving
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. Evading IDS/IPS
6. **Vuln Scanning with Nessus**
   - Lab: Nessus
7. SMB Sessions
   - Lab: SMB Sessions

The attacker knows which ports are open. Next, he tries to determine which vulnerabilities are present on the target system.

Vulnerability Scanners

- Vulnerability scanning tools help map a network, scan for open ports, and find various vulnerabilities
- Test against a list of known exploits
  - What about the unknown
  - That's why we want to have security in-depth
  - Multi-layered, sound architecture needed
- Generate pretty reports
  - Information overload
  - What do you do with a 2,000-page report

Vulnerability scanning tools are extremely useful because they automate security checks across a large number of systems over the network. However, understand their limitations:

- The tools only check for vulnerabilities that they know. They cannot find vulnerabilities that they don't understand.
- The tools tend to be flat: They look for vulnerabilities, but most cannot exploit them and pivot beyond an initial surface target to find other targets and vulnerabilities. A real attacker applies a great deal of intelligence to try to reverse engineer your network. Instead of just looking at outside interfaces, intelligent attackers try to understand what's going on behind them.
- The tools often don't perform detailed correlation among many vulnerabilities to ascertain overall risk. You may have low-risk vulnerability A, low-risk vulnerability B, and low-risk vulnerability C, each of which, by itself, is low risk. But, because you have all three present in a given way in your environment, you may face a high risk. Most vulnerability assessment tools cannot perform that kind of analysis, although a real-world computer attacker may.

Bunch of Vuln Scanners

Vuln Scanning

- Many commercial scanners are available
  - Rapid7 NeXpose (www.rapid7.com)
  - SAINT, by SAINT Corporation (www.saintcorporation.com)
  - BeyondTrust Retina Network Security Scanner (www.eeye.com)
  - Nessus, by Tenable Network Security (www.tenablesecurity.com)
  - OpenVAS, a fork of the previous free, open-source version of Nessus 2
- Some commercial services offer these features (as web-based application service providers)
  - Qualys (www.qualys.com)
  - McAfee's Foundscan (www.foundstone.com)

A large number of scanning tools are available today, as indicated on the slide. NeXpose, SAINT, Retina, and Nessus are all available on a commercial basis, while OpenVAS is a free fork of an older, open-source version of Nessus 2.

If you don't want to buy or maintain software, you can even subscribe to web-based scanning services, such as Qualys and Foundscan, which you can configure to run across the Internet (or even from an intranet appliance) to scan on a regular basis.

Vulnerability Scanners – Nessus

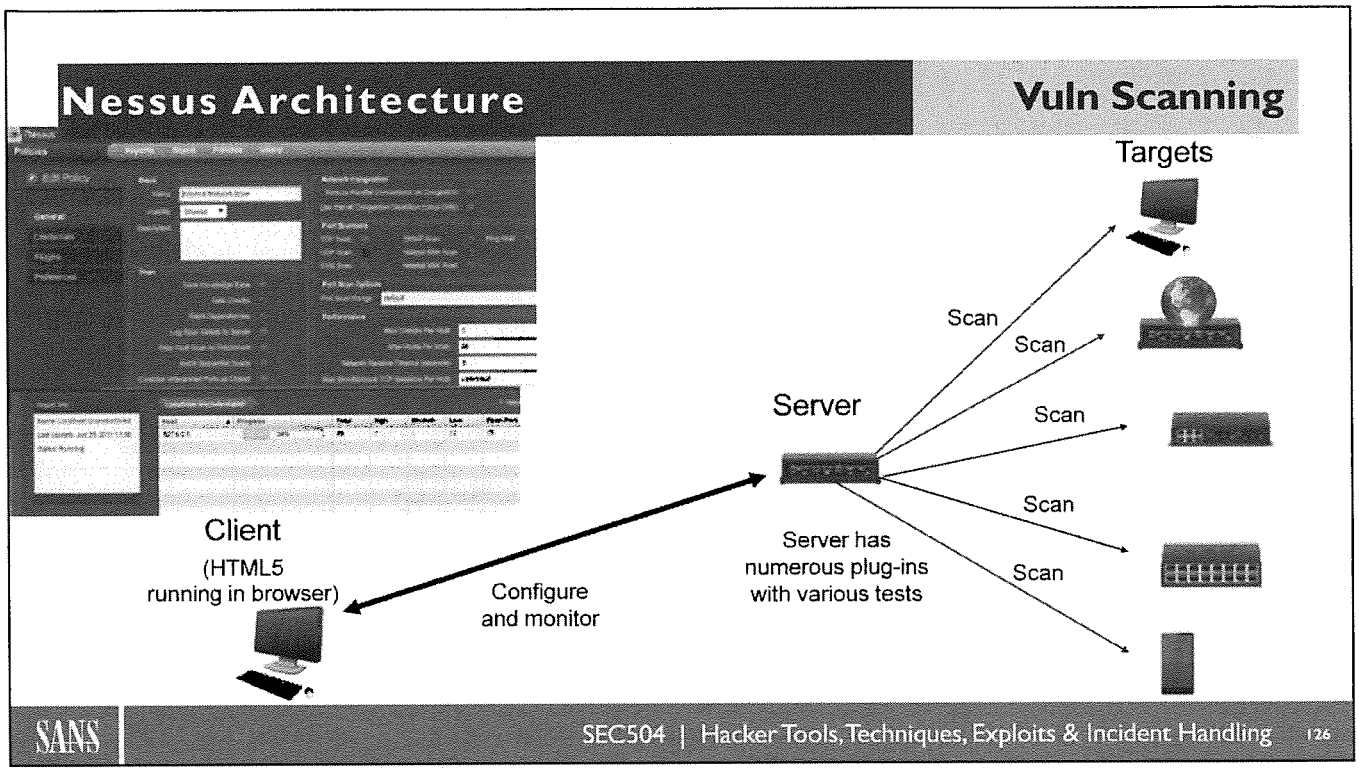
Vuln Scanning

- We focus on Nessus, the most popular vulnerability scanner
  - Commercial license for all commercial use
  - Free home-use license
- Project originally started by Renaud Deraison, now run by Tenable Network Security
- Available at www.tenable.com/products/nessus
- Nessus consists of a client and server with modular plug-ins for individual tests

Nessus is a very useful tool. For this course, we look at it in detail, because it remains one of the most popular vulnerability scanners. Nessus is offered via two different license mechanisms. The commercial Nessus license is for all commercial use of the Nessus product and costs US$1,200 per year per Nessus server. Tenable also makes available a free home-use license, but this license prohibits its use in commercial environments; instead, it is to be used for non-commercial home scanning.

The Nessus project was originally started by Renaud Deraison, who went on to co-found Tenable Network Security with other information-security industry luminaries.

Nessus is a client-server architecture with a large number of plug-ins that measure targets for individual vulnerabilities.



The Nessus client-server architecture is shown in this slide's illustration. The Nessus server includes the various plug-ins, each of which performs a single test against the chosen target systems. The server is configured using the Nessus client. For recent versions of Nessus, this client is actually an HTML5-based GUI that runs in a browser on the client machine.

The Nessus user invokes a browser and surfs to the Nessus server machine using HTTPS to TCP port 8834. After logging in to the Nessus GUI, the user configures a scan policy and invokes a scan. The GUI provides status information about the scan in progress. The Nessus server conducts the scan and stores the results. These results can then be displayed or exported in a variety of formats in the client.

Of course, the Nessus client and server can run on the same computer system, which is a common approach to using Nessus.

Nessus Platform Support

Vuln Scanning

- The Nessus server is available for
  - Linux, FreeBSD, MacOS X, and Windows
- The server is accessed and configured using a browser, which runs an HTML5-based client
- Of course, client and server can run on the same machine, which is the most common use case
- Some plug-ins are characterized as "dangerous"
  - They may impact targets with crashes or locked-out accounts
  - Some of the plug-ins in the Denial of Service family of plug-ins are dangerous; others are not, because they merely check version number
  - "Safe Checks" is the GUI option that turns off dangerous plug-ins
  - These dangerous plug-ins are disabled by default

The Nessus server is available on many variations of Linux, FreeBSD, Mac OS X, and Windows. The Nessus client runs inside of any HTML5-capable browser, such as Firefox, IE, or Chrome.

Most people run the Nessus client and server on the same machine, as we do in this class for the next lab.

Nessus includes the concept of "dangerous" plug-ins, which could impair a target system, making it crash or otherwise unstable. Some dangerous plug-ins could likewise lock out accounts, resulting in a Denial of Service condition for legitimate users.

Some of the plug-ins in the Denial of Service family of Nessus plug-ins are dangerous, while other plug-ins in that category are not. The non-dangerous Denial of Service plug-ins typically just check the version number of a target service and indicate whether there is a known Denial of Service attack against that version, without actually launching the attack. That's why they aren't dangerous. But, the dangerous plug-ins in the Denial of Service family actually launch the attack, potentially causing problems for the target system.

In the Nessus GUI, the "Safe Checks" option ensures that dangerous plug-ins are not run in a given scan. This Safe Checks option is activated by default, which means that dangerous plug-ins are turned off in the default configuration of Nessus.

Nessus Plug-Ins

Vuln Scanning

- There is a defined API for writing Nessus plug-ins
  - Some plug-ins written in C
  - Or plug-ins can be written in the Nessus Attack Scripting Language (NASL)
  - One plug-in is in charge of doing one attack and reporting the result to the Nessus server (nessusd)
  - Each plug-in can use some functions of the Nessus library and store information in a shared knowledge base
- There are approximately 50,000 plug-ins, updated frequently
  - Automatically updated every 24 hours (be careful to check what you have configured to run to make sure it won't impair targets)
  - Or invoke manual update by running nessus-update-plugins script

A nice capability of Nessus is the ability to write your own plug-ins, which is a capability not supported in some other commercial scanners. When plug-ins are written, one plug-in is in charge of doing one attack and reporting the result to the Nessus server daemon (nessusd). So, the number of plug-ins equates roughly to the number of tests conducted by the tool. Each plug-in can use some functions of the Nessus library and store information in a shared knowledge base.

Currently, there are about 50,000 plug-ins in Nessus.

Once the Nessus server is registered with Tenable, it automatically updates plug-ins every 24 hours, downloading the latest. This could be a problem, because a vulnerability assessor or penetration tester may have a copy of Nessus automatically update itself one evening without noticing, and then run a test the next day with an unknown and untested new set of plug-ins. You may want to disable this auto-update of plug-ins and instead only update them manually when you want to evaluate the newest plug-ins in a test environment. Plug-in updates can be downloaded manually by running a script that comes with Nessus called "nessus-update-plugins."

Vulnerability Scanners – Defense

Vuln Scanning

- Preparation
  - Close all unused ports
  - Shut off all unneeded services
    - In Windows, stop or delete services in Services control panel, as discussed earlier
    - In UNIX, edit /etc/inetd.conf or /etc/xinetd.d files, as well as rc.d files (remember chkconfig from earlier)
  - Apply all system patches
    - Keep up-to-date
  - Run credentialed scans of your environment
- Identification
  - Utilize intrusion detection system signatures
  - Most vulnerability scanners trip hundreds of signatures
- Cont, Erad, Rec: N/A

Again, this is not rocket science. Still, it is difficult to keep up with all the patches on numerous types of systems. However, we must keep our critical systems patched and up to date.

The following are some key defense measures one can take.

Close all unused ports by shutting off all unneeded services.

- In Windows, stop or delete services in Services control panel by using the techniques we discussed during the port scanning section.
- In UNIX, edit /etc/inetd.conf or /etc/xinetd.d, as well as your rc.d files (as covered with chkconfig earlier).

Make sure that you keep your systems up to date by applying all system patches. You can easily test this by running scanning tools using something called credentialed scans. These scans use a valid user ID and password to access a server and validate configuration and patches.

Finally, utilize an intrusion detection system, which specializes in detecting this type of scanning tool.

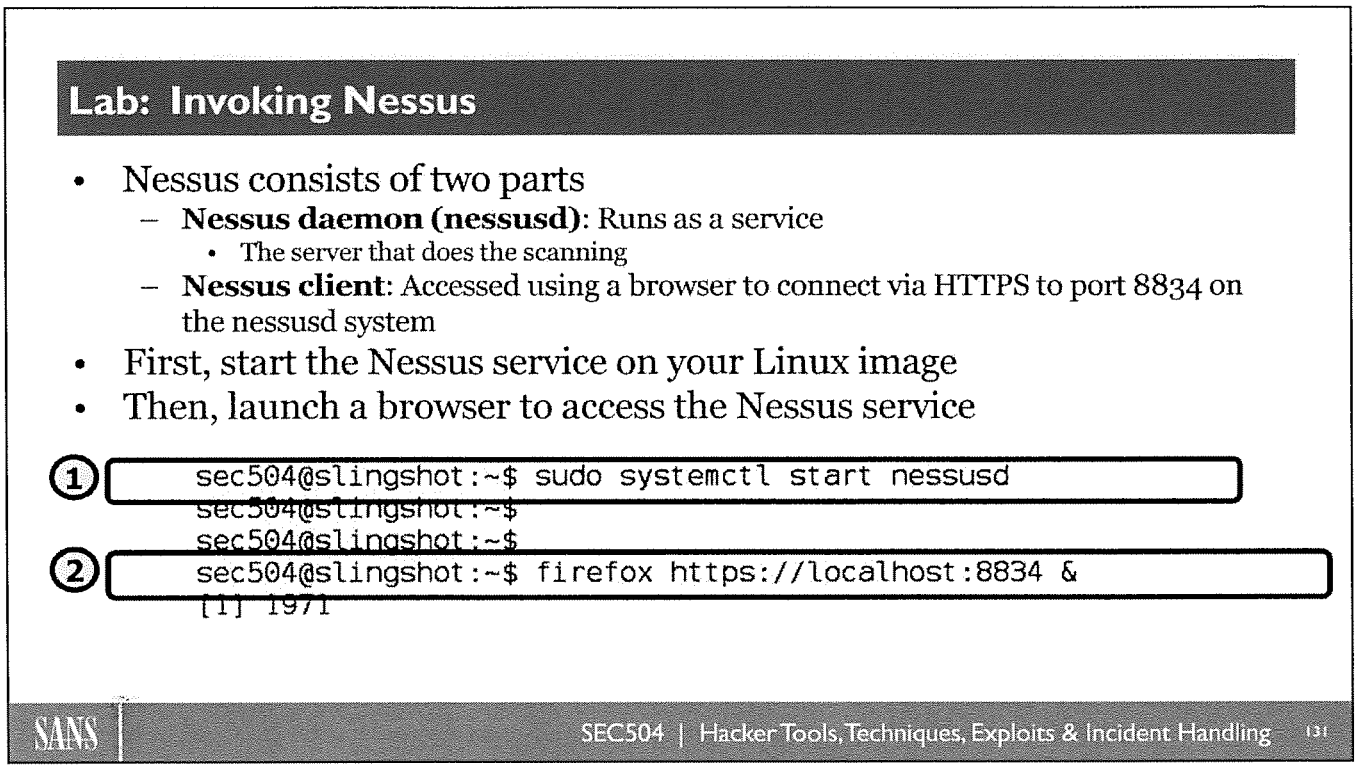
Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. War Dialing
2. War Driving
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. Evading IDS/IPS
6. Vuln Scanning with Nessus
   - **Lab: Nessus**
7. SMB Sessions
   - Lab: SMB Sessions

Let's perform a lab with Nessus, in which we configure it to perform a scan using an Internal Network Scan profile. We launch the scan against our Linux hosts and, if you have extra time, you can scan your Windows system. We sniff the packets generated by the scan while the scan is underway. We also review the reports generated by Nessus in the lab.



Next, let's conduct a lab using Nessus. As a reminder, we are starting a service, so this requires you to be root. To become root, enter `sudo` and then sec504's password of sec504

First, in the course VMware Linux image, start the Nessus service:

```
# sudo systemctl start nessusd
```

Although an "OK" message will NOT appear, the service starts when you get a command prompt back.

We use the web-based interface as a Nessus client, so launch the Firefox browser to connect to your localhost on port 8834 using HTTPS:

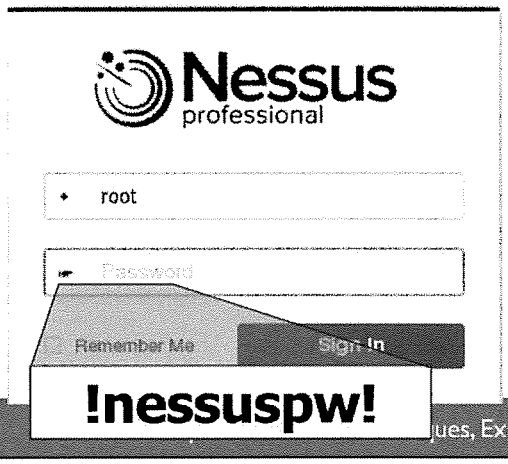
```
# firefox https://localhost:8834 &
```

Note the site is HTTPS. The S is important.

The & at the end of this command will kick the browser process into the background so we can get our terminal command prompt back. If you receive some Glib errors, that is OK. As long as the Firefox window opens you are fine.

Lab: Logging in to Nessus

- Log in using the Nessus username of root and password of !nessuspw!
  - DO NOT use the password of the OS root account



You now may see a screen that says "Nessus is initializing. Please wait..." with a status bar. That screen appears while the Nessus server loads plug-ins and prepares for a connection. Depending on your system speed, it may be on the screen for two minutes or so.

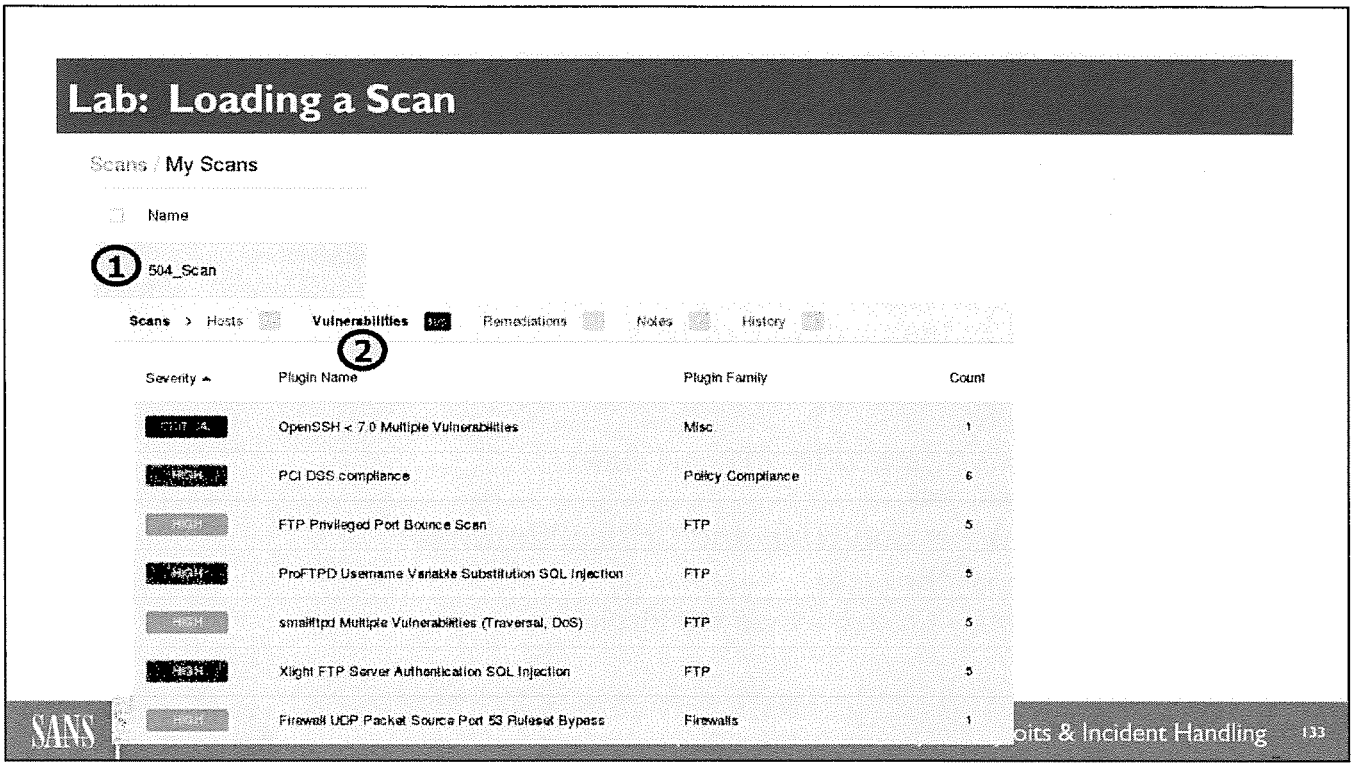
Once you see the screen asking for a username and password, you can log in to Nessus. Enter the following Nessus credentials:

Username: **root**

Password: **!nessuspw!**

It is important to note that you must enter the password for the *Nessus* account named "root," not the password for the root account of the *operating system*. Enter !nessuspw!.

Once you enter the proper username and password, click "Sign In."



To get started, we first select the 504_Scan already loaded for you. If you have time, start a scan of the local system after the core steps of the lab are done.

Next, select "Vulnerabilities." This loads the results in such a way as to sort the output by vulnerability ID rather than by IP address. This is important, because far too many organizations try to sort their vulnerabilities by IP address. This seems like a good idea at first, but is often impossible to address at scale. Why? Well, as you see later, it is far quicker and easier to see a large number of vulnerabilities when they are stacked. Service banners are good this way. However, if you were to sort by IP address, then you would have to review the same vulnerability ID or category multiple times for each IP address. Sorting by Vulnerability allows us to quickly and efficiently review large number of vulnerabilities.

Lab: More Than Just Red

- Let's spend time looking through the Low, Medium, and Informational findings
  - Many of the most dangerous vulnerabilities are in these categories
  - Ignore High and Critical for now
- Work together in teams of five or less
- Remember to sort by vulnerabilities
- Things to look for
  - Service banners (telnet, TFTP, FTP, and more)
  - Files (.doc, .pdf, .txt)
- The goal: Think beyond High and Critical findings
- It helps to export the scan
  - Export > HTML > Custom > Group By Plugin

The goal of this lab is to break the thought process of only looking to Critical and High vulnerabilities. In our testing at BHIS, we discovered that many organizations ignore anything less than High because of the bone-crushing volume of vulnerabilities. However, it is possible to review the lows and even informational findings if you know how to properly address them.

For this lab, ignore High and Critical. Instead, focus on Medium and below. Take a few moments and review the vulnerably output and look for vulnerabilities that could lead to an immediate breach of the organization. Some things to look for are service banners and files. Mainly, we want you to see how easy it is to review the *output* of the tools when the data is properly sorted.

Record Your Results Here

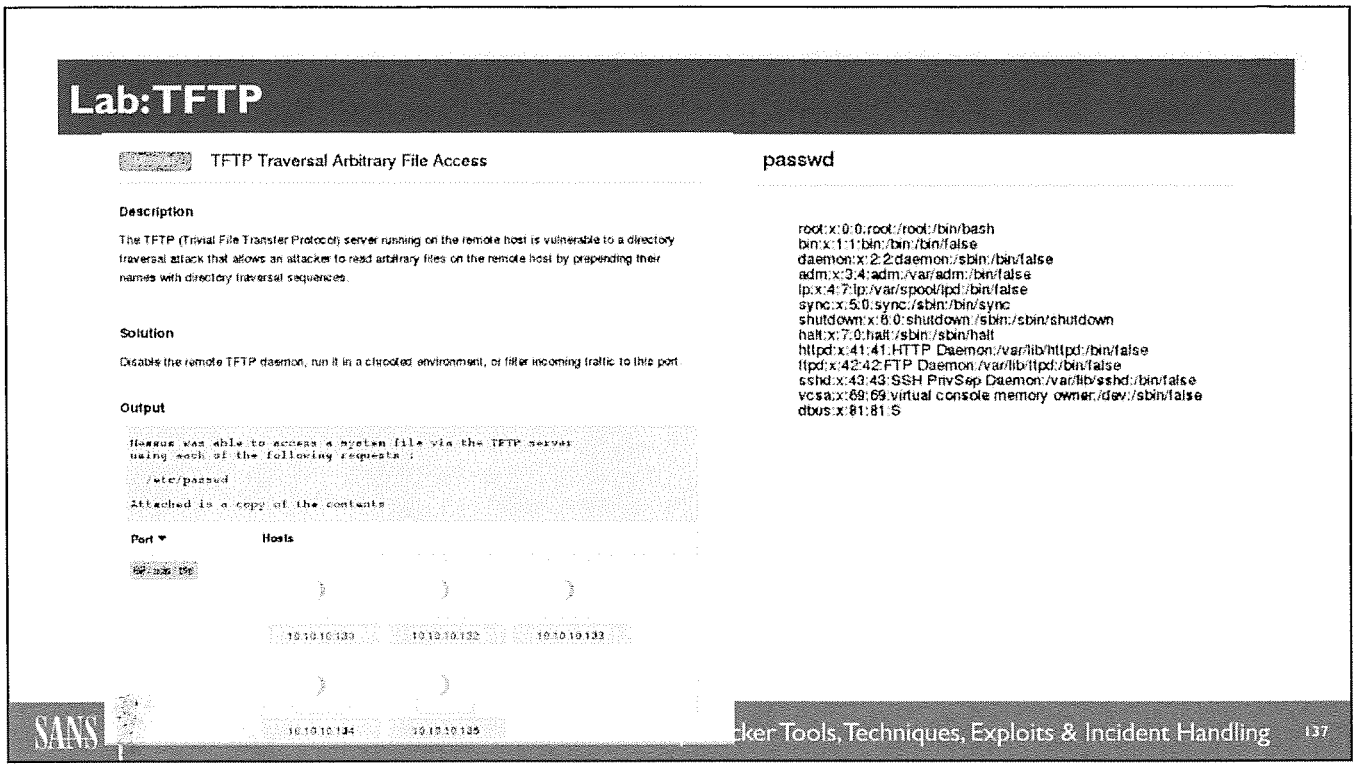
Vulnerability	Host(s)	Original Risk Rating	Actual Risk Rating

Feel free to use this for your notes.

Lab: Suggestions to Follow

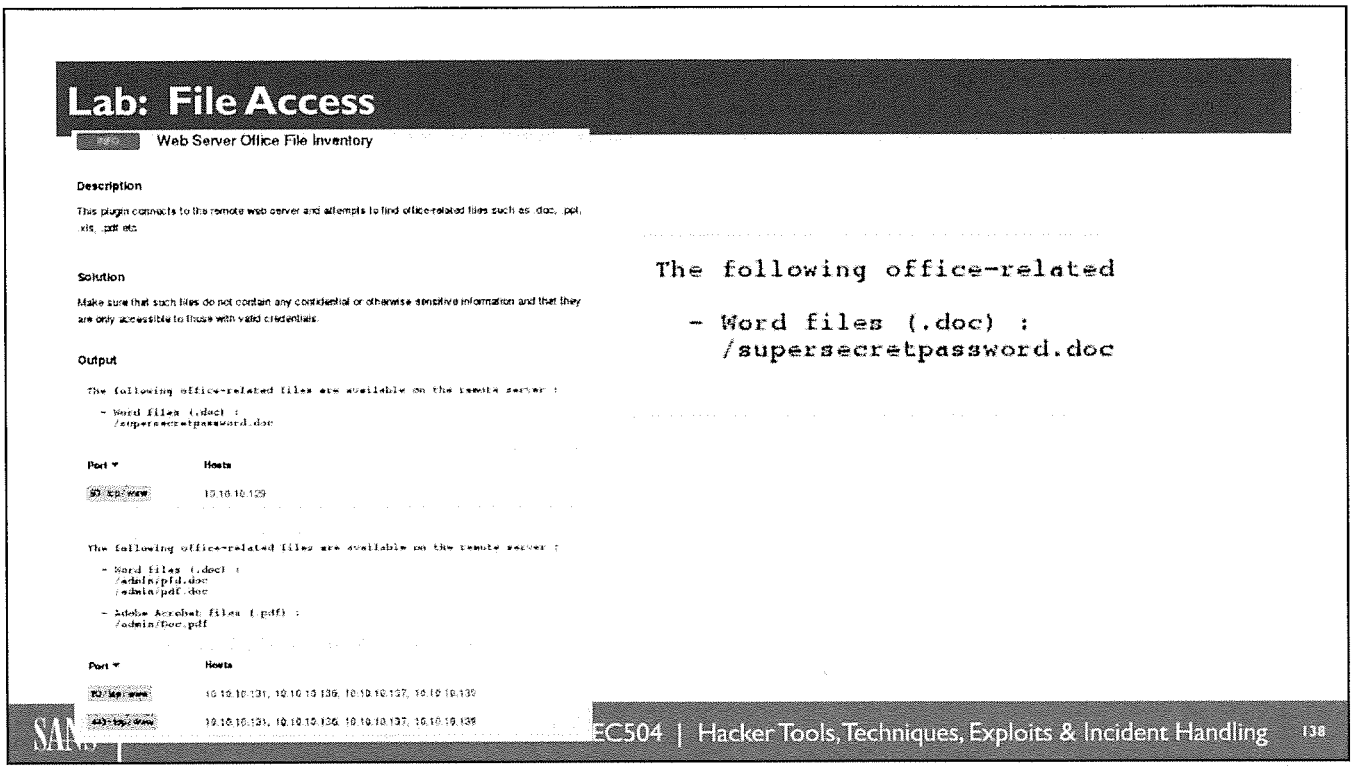
- Do not go further unless you want all super secrets revealed....

Go beyond this slide for the answers.



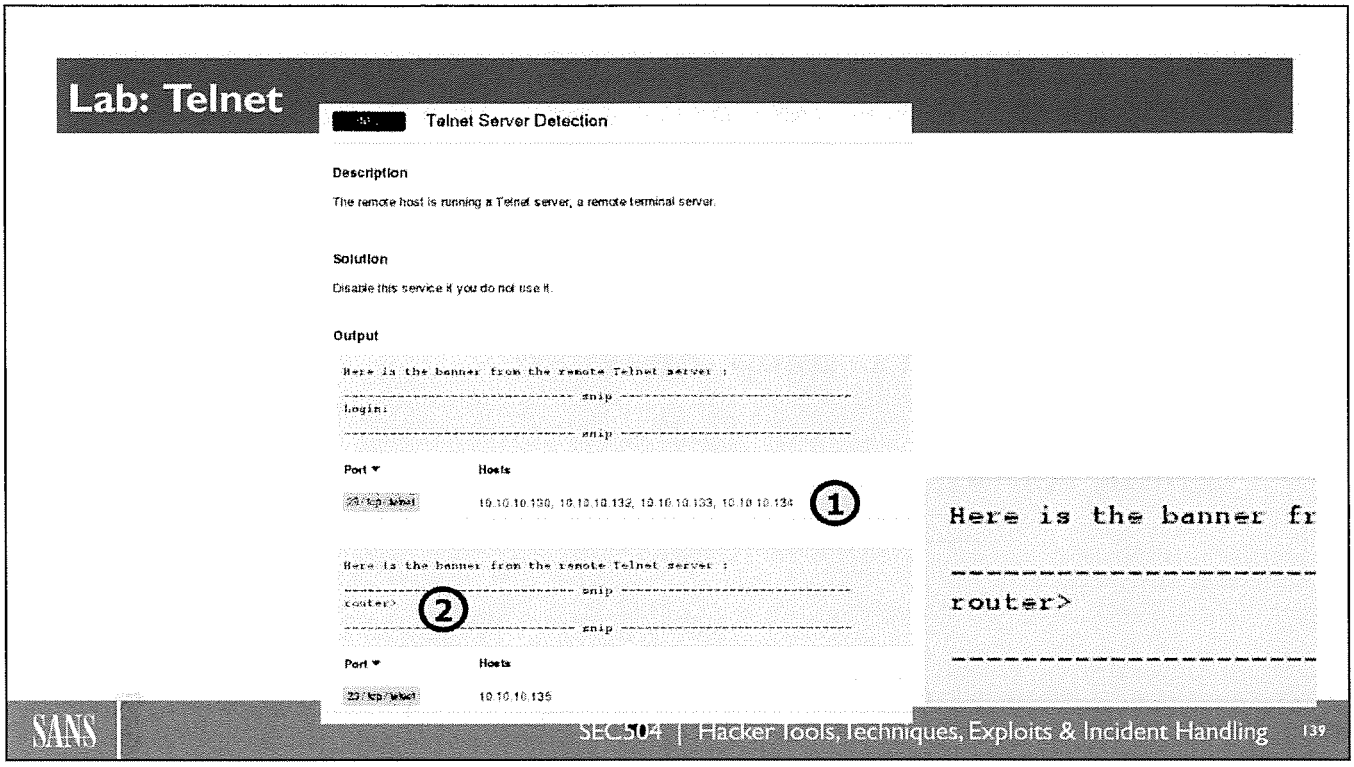
Start looking at scanner results as less of a roadmap of what exactly needs to be done, but as your eyes and ears on a target network. Trivial File Transfer Protocol (TFTP) is a good example of this. When Nessus runs, it attempts to pull the /etc/passwd file via TFTP. In this situation, it succeeded. Arbitrary file access is bad. In addition to grabbing /etc/passwd, It could also grab config files from the web server or backend databases.

Often in our testing, simple file access is one step away from a total server takeover.



Moving up the chain, lets look at direct file access via the web server. Many web servers support directory indexing. This allows us to see the files hosted on the server. Sometimes, this this intentional, as in a file server. However, we often find backup and config files. From time to time, we even find build scripts and other files with passwords in them.

Something like supersecretpassword.doc has actually been found in our testing. Yes, it happened, and it was magnificent.



It is often the case where dozens, if not hundreds, of systems will be running the exact same service. For example, telnet is still often enabled on Internet-facing systems to this day. It may even be intentional. For example, some ISPs use Telnet for remote management of their routers. If configured properly, they ask for a user ID and a password. However, if viewed properly, you can see how systems are clustered. In section 1 of the slide, you see a large number of Telnet servers with the prompt "Login:." This would be expected. Not ideal, but expected. However, in section 2, you see there was one with a prompt of "router>," which means there is no authentication necessary for this router. Again, this is a informational vulnerability, which is actually quite critical.

Lab: When You Finish, Stop Nessus Service

- When you finish the lab, kill your terminal windows and shut down your Nessus service to free up resources for later labs

```
# systemctl stop nessusd
```

- Then, reboot your VM

When you finish the Nessus lab, disable the Nessus service to free up resources on your Linux virtual machine for later labs:

```
# systemctl stop nessusd
```

You could leave the Nessus daemon running, waiting for connections from a browser, without impairing later labs. However, you get better performance if you stop the Nessus service using this command.

Furthermore, you have a Netcat backdoor running on your system. To kill it, simply close all terminal windows. A reboot does not hurt, either.

Course Roadmap

- Incident Handling
- Applied Incident Handling
- Attack Trends
- Step 1: Reconnaissance
- **Step 2: Scanning**
- Step 3: Exploitation
  - Gaining Access
  - Web App Attacks
  - Denial of Service
  - Step 4: Keeping Access
- Step 5: Covering Tracks
- Conclusions

SCANNING

1. War Dialing
2. War Driving
   - Lab: Wireless LAN Discovery with InSSIDer
3. Network Mapping with Nmap
4. Port Scanning with Nmap
   - Lab: Nmap
5. Evading IDS/IPS
6. Vuln Scanning with Nessus
   - Lab: Nessus
7. **SMB Sessions**
   - Lab: SMB Sessions

An additional scanning technique lets an attacker grab data from a Windows environment across Server Message Block (SMB) sessions. This powerful technique allows an attacker to grab information about available users, groups, shares, and more, all by using a non-admin username and password. We look at a variety of tools to pull this information, especially enum for Windows and rpcclient for Linux. We also perform a hands-on lab to illustrate the technique.

The Server Message Block (SMB) Protocol — SMB Sessions

- SMB is a Layer 7 protocol that implements file and printer sharing, domain auth, remote admin, and other features
  - Used throughout Windows environments
    - Workstation Service implements much of the client code
    - Client tools include File Explorer, net use command, reg command, sc command, Sysinternals psexec tool, and more
    - Server Service implements much of the server-side code (running on both servers and workstation machines)
  - Supported in Linux and UNIX via Samba client tools (smbclient, smbmount, rpcclient, and more) and the smb daemon
  - Heavily used post exploitation to avoid detection
- Accessed via TCP port 445 on modern systems
  - On older (WinNT and 2K) systems, SMB is carried over NetBIOS, which uses TCP and UDP ports 135–139

Microsoft created the Server Message Block (SMB) protocol to support a variety of network-accessible features of Windows machines, including file sharing, printer sharing, domain authentication, remote administration (through commands such as reg, sc, and many others, as well as via enterprise admin GUI tools), and countless other capabilities.

These features are used throughout most Windows deployments. The client-side implementation of most of the SMB code is in the Workstation Service. Various client tools that use SMB for remote access include the Windows File Explorer, the net use command, the reg command (for remote registry access), the sc command (for remote service control), and even the psexec command from Microsoft Sysinternals that lets you cause a target Windows machine to run a program.

The Server Service on Windows implements the service side of SMB, making file shares and remote registry access available, again with many other features. This service is running by default on Windows workstations and servers alike. Given all the features supported by SMB, it is often subject to attack. Years ago, there were numerous buffer overflow and related flaws in Microsoft's SMB implementation. But, even without those flaws (many of which are patched today), an attacker can still enumerate a target system across SMB sessions, as we'll explore in this section. Furthermore, even with a fully patched Windows system, SMB offers an attacker with admin credentials the ability to run code on a target, a technique we'll explore with our Metasploit discussion and lab in 504.3.

Via the SAMBA project, Linux and UNIX machines also have SMB implementations, including SMB client tools, such as smbclient, smbmount, and rpcclient. To act as a file server for Windows machines, Linux also can take advantage of the samba daemon (smbd).

On modern Windows machines, SMB is typically accessed using TCP port 445. On older Windows NT and 2000 systems, SMB was carried over the NetBIOS protocol, which used TCP and UDP ports 135 through 139.

Establishing a SMB Session from Windows

- On Windows machines, the net use command establishes a session

```
C:\> net use \\[TargetIP]
```

  - The currently logged-on user's credentials are sent via pass-through authentication
  - The default administrative share is selected (typically ipc$, but other shares such as admin$, c$, or others may be connected)
- To connect as another user or to a specific share, use

```
C:\> net use \\[TargetIP]\[ShareName] [password]
/u:[UserName]
```

  - That user does not need to be in the admin group to connect to ipc$ or other open shares (although c$ and admin$ require admin privs)
  - If you leave off the [password], Windows prompts you for it
- To connect as no user (an anonymous or NULL SMB session), use

```
C:\> net use \\[TargetIP] "" /u:""
```

  - A NULL SMB session has a blank username and password

To establish an SMB session from one Windows machine to another at a given target IP address, type the following at the command line:

```
C:\> net use \\[TargetIPaddr]
```

Because we have not provided a username in this command, the current user running the "net use" command has his or her authentication credentials passed through to the target machine. So, for example, if user "bob" runs this command on Windows machine A, that Windows machine attempts to authenticate as bob to the TargetIPaddr.

Also, because we have not put in a share to connect to after the TargetIPaddr, Windows tries to connect to the next available admin share, which is typically IPC$, a share used for remote access of system information.

If we want to connect as a different user or to a specific share (such as IPC$, ADMIN$, C$, or a user-specified file share on the target), run this more general command:

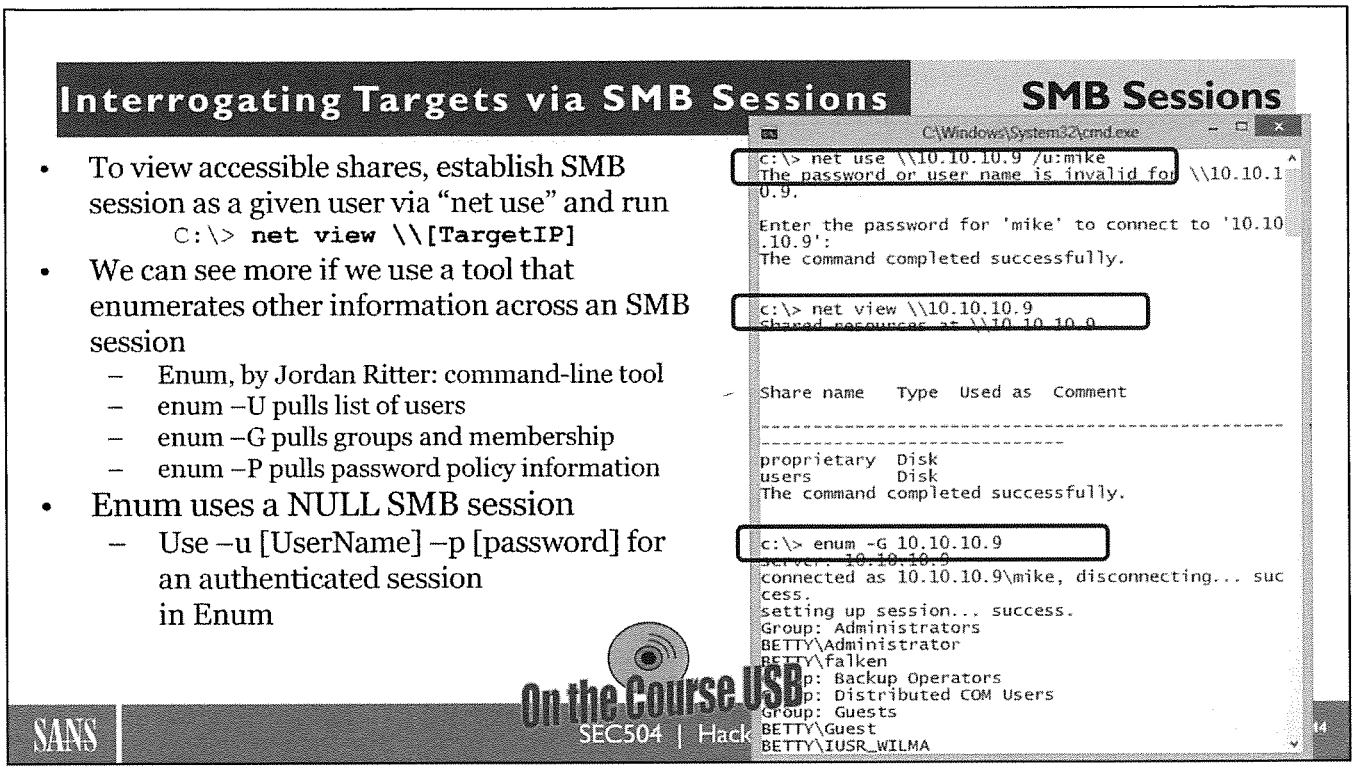
```
C:\> net use \\[TargetIPaddr]\[ShareName] [password] /u:[UserName]
```

Note that the user does not have to be in the admin group to connect to most available shares, such as IPC$. That is, you can establish an SMB session to most Windows targets as long as you have a non-admin username and password. Some shares (notably C$ and ADMIN$) require admin privileges to which to connect. If you leave off the password in this command, you are prompted for it.

Some Windows machines support the concept of a NULL SMB session. With a NULL session, the username and password fields are blank, essentially making them anonymous. A NULL SMB session can be established with the following command (note the blank password and username):

```
C:\> net use \\[TargetIPaddr] "" /u:""
```

Older versions of Windows (such as Windows NT and 2000) allow for detailed interrogation of a target machine across NULL SMB sessions with their default configurations. More recent versions of Windows may still allow NULL SMB sessions to connect, but they block the enumeration of configuration information across the NULL session.



After you establish an SMB session with a target machine using the "net use" command, you can get a list of shares by running the "net view" command, as follows:

```
C:\> net view \\[TargetIPaddr]
```

Important note: Windows machines hide the default administrative shares (IPC$, ADMIN$, and C$) from the net view command. Those shares are still there, but "net view" omits them from its output.

To get more detailed information via SMB sessions with a target machine, you could run the enum tool. Written by Jordan Ritter, this free command-line tool interrogates target Windows machines across an SMB session with several configuration options:

- `C:\> enum -S [TargetIPaddr]` pulls a list of shares, including showing the default administrative shares (IPC$, ADMIN$, C$) that "net view" does not show
- `C:\> enum -U [TargetIPaddr]` pulls a list of users
- `C:\> enum -G [TargetIPaddr]` pulls a list of groups and member accounts in each group
- `C:\> enum -P [TargetIPaddr]` pulls password policy information, including minimum password length, maximum password age, and account lockout settings

By default, enum tries to connect with a NULL SMB session. But, if you invoke it with –u [UserName] and –p [password], it provides an authenticated SMB session from which to extract information from a target machine. For example, in the next lab, we run enum the –G option as follows:

```
C:\> enum -u [UserName] -p [password] -G [TargetIPaddr]
```

First Three Commands

SMB Sessions

- Show shares and systems
  - `C:\> net view`
- Users
  - `C:\> net user /domain` Put users from above into users.txt
- Now, we crack passwords
  - `C:\> @FOR /F %n in (users.txt) DO @FOR /F %p in (pass.txt) DO @net use \\DOMAINCONTROLLER\IPC$ /user:DOMAIN\%n %p 1>NUL 2>&1 && @echo [*] %n:%p && @net use /delete \\DOMAINCONTROLLER\IPC$ > NUL`

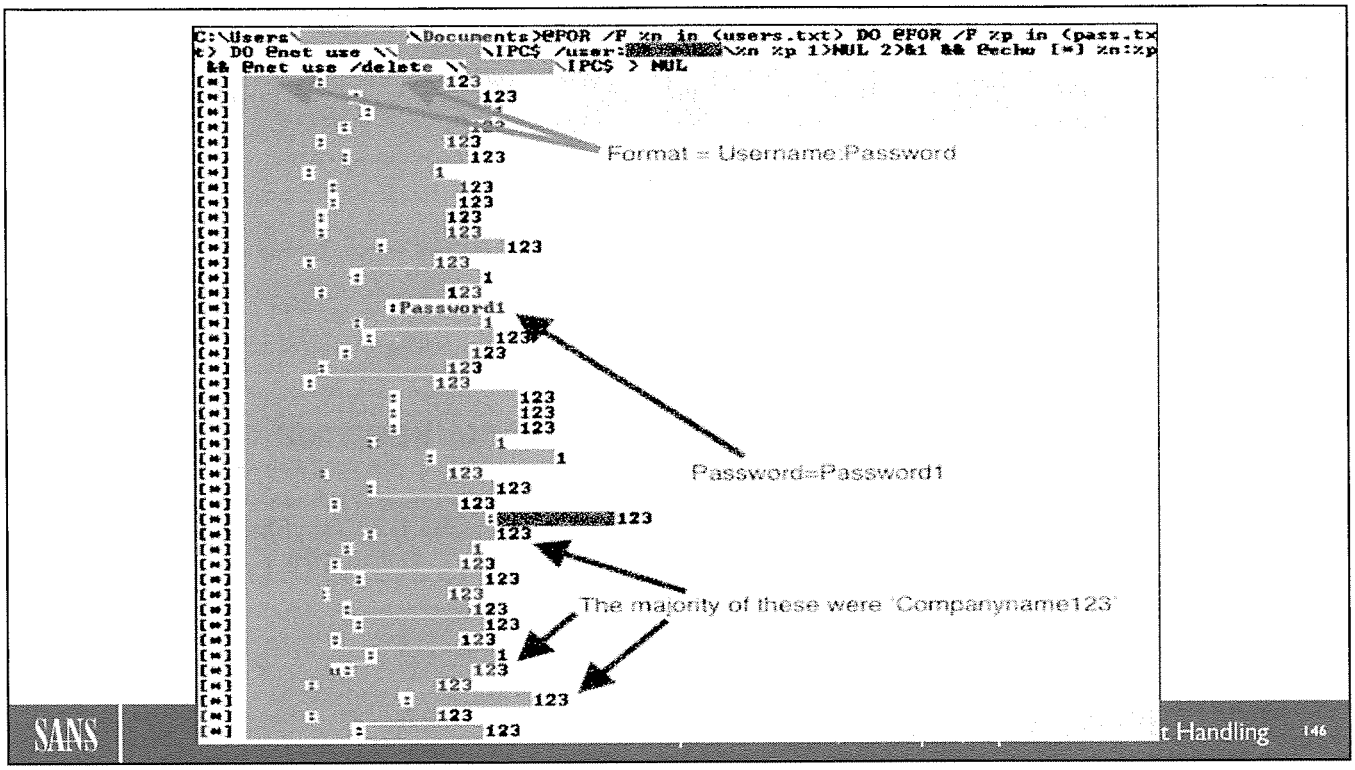
When we test, we often use these first three net commands to gain access to a large number of systems.

The first command is used to identify sensitive systems and shares. We are particularly interested in file shares.

Next, we look for users. The net user command (shown in slide) pulls all the domain users.

Finally, we take the users and load them into a file called users.txt and create a limited password file called pass.txt. The number of passwords should be less than the account lockout threshold of the environment.

Finally, we have a FOR loop that passed the users (%n) and the passwords (%p) into the net use command where we are trying each combination to authenticate against a domain controller.

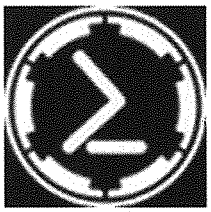


This slide shows a screenshot from a run of this against an enterprise environment. In a matter of about an hour, dozens of accounts were compromised. Three of them were domain administrators.

Yes, attackers do this. No, it is not hard. Worse, it often flies under the radar of many IDS/IPS systems.

PowerShell Empire — SMB Sessions

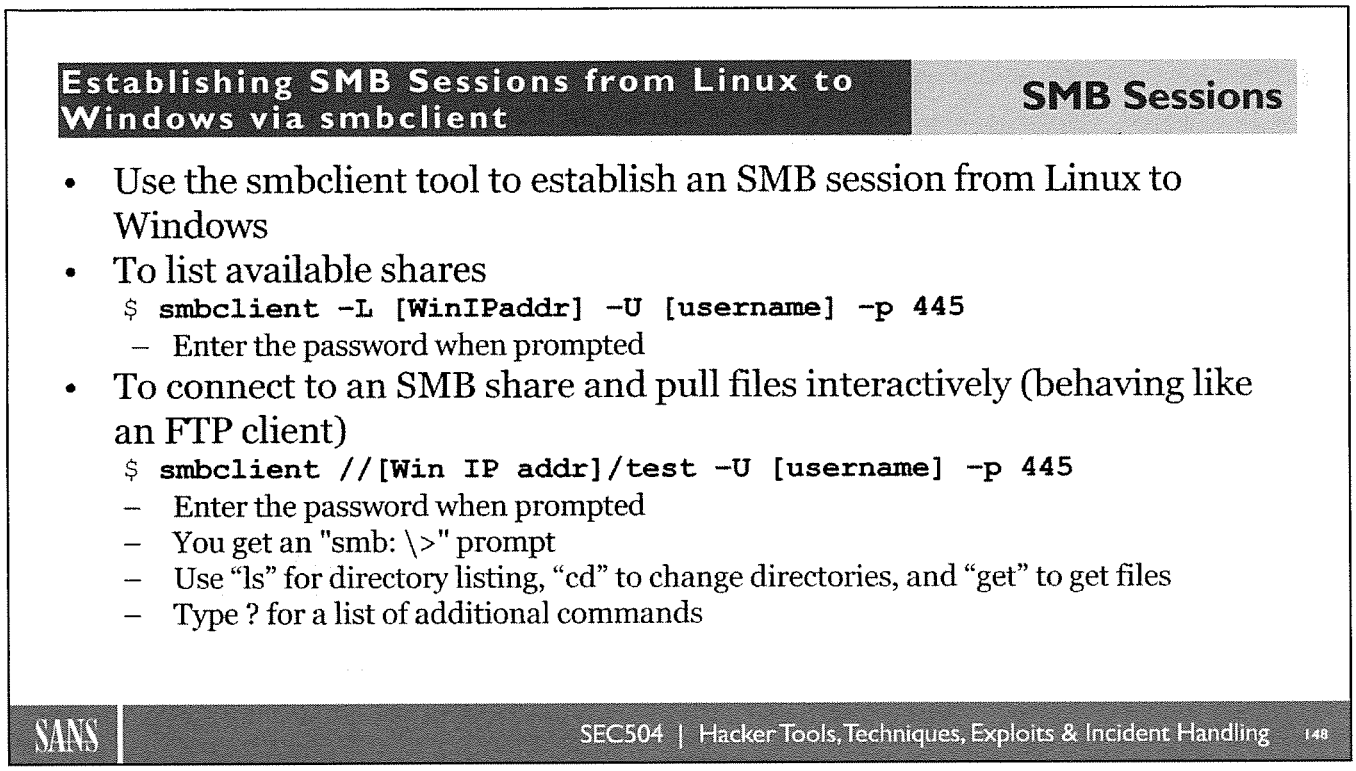
- Backdoor built in PowerShell
- Fantastic post-exploitation scanning abilities
- Family of modules under Situational Awareness
  - situational_awareness/network/sharefinder: Find accessible shares
  - situational_awareness/network/arpscan: ARP Scan the local IPv4 systems
- Also has the ability to map domain trusts, group membership, portscan and conduct reverse DNS lookups
- Uses built in Microsoft Protocols like SMB
- Get it here:
  - http://www.powershellempire.com/



PowerShell Empire is easily one of the more disruptive tools to hit the security industry in quite some time. The reason for this is because it is built purely in PowerShell, which is installed on most Windows systems by default. By using the built-in functionality of Windows there is tremendous power in what you can do post-exploitation to scan for additional vulnerabilities and systems to compromise. For example, situational_awareness/network modules will allow an attacker to easily enumerate shares and users across a domain. Dramatically reducing the amount of time needed to find sensitive shares and documents on the inside of a network.

It can be found here:

http://www.powershellempire.com/



Besides Enum running on Windows to attack Windows, we can also pull information from target Windows machines using a Linux system. In particular, the smbclient program that is part of the Samba suite can pull a list of shares from a target Windows box by running

```
$ smbclient -L [WinIPaddr] -U [username] -p 445
```

Alternatively, you can use smbclient to make an *interactive* SMB connection to a target Windows machine and then push files to or pull them from the target. When connecting to an SMB share, this smbclient tool provides an interactive command prompt that is reminiscent of an FTP client, letting you navigate the directory structure with cd, get a directory listing with ls, and pull files with the get command. To make an interactive smbclient connection, you'd run the following command (which includes ls and get afterwards of examples of things you might do):

```
$ smbclient //[Win IP addr]/test -U [username] -p 445
Enter [username]'s password:
smb: \> ls
smb: \> get [filename]
```

Using Samba's rpcclient from Linux for More Info

- The Linux rpcclient tool can pull even more information
- Establish a session with
  `$ rpcclient -U [username] [WinIPaddr]`
- Enter the password when prompted
- You have an rpcclient prompt with many commands available
  - **enumdomusers**: List users
  - **enumalsgroups [domain]|[builtin]**: List groups (stands for "enum alias groups")
  - **lsaenumsid**: Show all users SIDs defined on the box
  - **lookupnames [name]**: Show SID associated with user- or group name
  - **lookupsids [sid]**: Show username associated with SID
  - **srvinfo**: Show OS type and version
- The rpcclient man page lists hundreds of other commands
  - Those listed here are the most useful and a lab covers them shortly

The biggest treasure trove of information you can get via SMB sessions is available via a Linux tool called rpcclient. Originally created as a troubleshooting and debugging tool for the Samba suite, rpcclient is super flexible and includes hundreds of features. To establish an SMB session with rpcclient, you first run

```
$ rpcclient -U [username] [WinIPaddr]
```

After you provide a password, you receive the rpcclient prompt

```
rpcclient $>
```

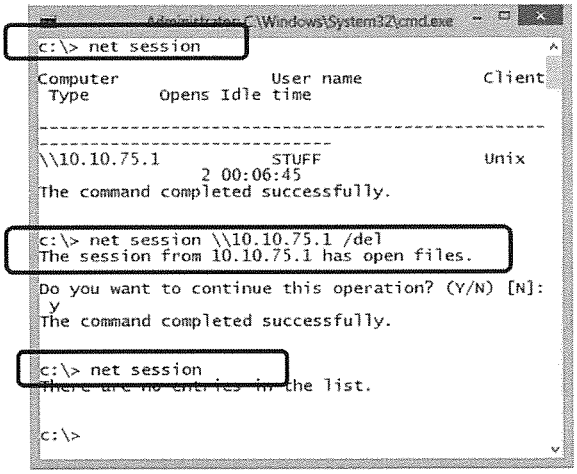
At this prompt, you can type any one of more than 100 commands. Some of the most useful are

- **enumdomusers**: This command shows users defined locally on the machine and any domain users the system knows about
- **enumalsgroups**: This command, followed by the word "domain" or "builtin," shows groups defined on the box. The "als" in the middle of enum and groups in this command's name refers to the word "alias"
- **lsaenumsid**: This command shows the Security Identifier (SID) of all users defined locally on the target Windows machine
- **lookupnames**: This feature lets you see the SID for a username that you provide
- **lookupsids**: This feature converts a username you provide into the SID on the target machine
- **srvinfo**: Shows the version of the target Windows machine

We use each of these commands in the next lab.

Seeing and Dropping SMB Sessions

- On Windows, to see where you have established outbound SMB sessions (you are acting as an SMB client), run
  `C:\> net use`
- To drop an outbound SMB session, run
  `C:\> net use \\[IPaddr] /del`
- On Windows, to see who has established inbound SMB sessions (you are acting as an SMB server), run
  `C:\> net session`
- To drop an inbound SMB session, run
  `C:\> net session \\[IPaddr] /del`



On a Windows machine, to see which SMB sessions you've made outbound to other systems (for example, when you are acting as a Windows client for SMB), you can run the "net use" command by itself, as follows:

```
C:\> net use
```

The output displays the target machine and the share to which you are connected. To drop an outbound SMB session to the given IPaddr, you could run

```
C:\> net use \\[IPaddr] /del
```

If you want to drop all outbound SMB sessions (instead of just one associated with a given IP address), you could run

```
C:\> net use * /del
```

When prompted, if you want to delete all sessions, type y and press Enter. Or you could append /y to your command, and you won't be prompted.

Flipping things around, let's discuss how you can list and drop SMB sessions that are opened inbound *to* your system (for example, you are acting as a Windows SMB server). To list the inbound sessions (as shown in the slide's screenshot), you could run

```
C:\> net session
```

Then, to drop an inbound SMB session, you could run

```
C:\> net session \\[IPaddr] /del
```

The ability to drop individual SMB sessions (inbound or outbound) can be useful for incident handlers, because doing this can temporarily stop an attacker from using the SMB session. This introduces a small pause in the attacker's progress, perhaps buying you some time.

Preparation: Defenses Against Evil SMB Sessions (I)

SMB Sessions

- Verify that you're blocking data exfiltration via NULL SMB sessions
  - HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymous
    - 0 = Null sessions can enumerate shares (default on many Windows systems)
    - 1 = Null sessions cannot enumerate shares
  - HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymousSAM
    - 0 = Null sessions can enumerate usernames
    - 1 = Null sessions cannot enumerate usernames (default on all modern Windows)
  - HKLM\System\CurrentControlSet\Control\Lsa\EveryoneIncludesAnonymous
    - 0 = Null sessions have no special rights (default on all modern Windows)
    - 1 = Null sessions are part of the everyone group (lots of config enumeration)
- These only block information for NULL SMB sessions
  - Good idea, but even with these settings, an attacker can enumerate all info with one valid username and password

To defend against SMB-based enumeration attacks, first, make sure you aren't leaking information via NULL SMB sessions. The default registry key settings on modern Windows machines are decent, but check them periodically via audits, vulnerability assessments, and penetration testing to ensure that their settings are strong.

Keep in mind that these settings only control NULL SMB sessions. An attacker with a valid username and password can still fully interrogate a target machine regardless of these settings, even with a non-admin account. That's why we need further control of SMB sessions, as described next.

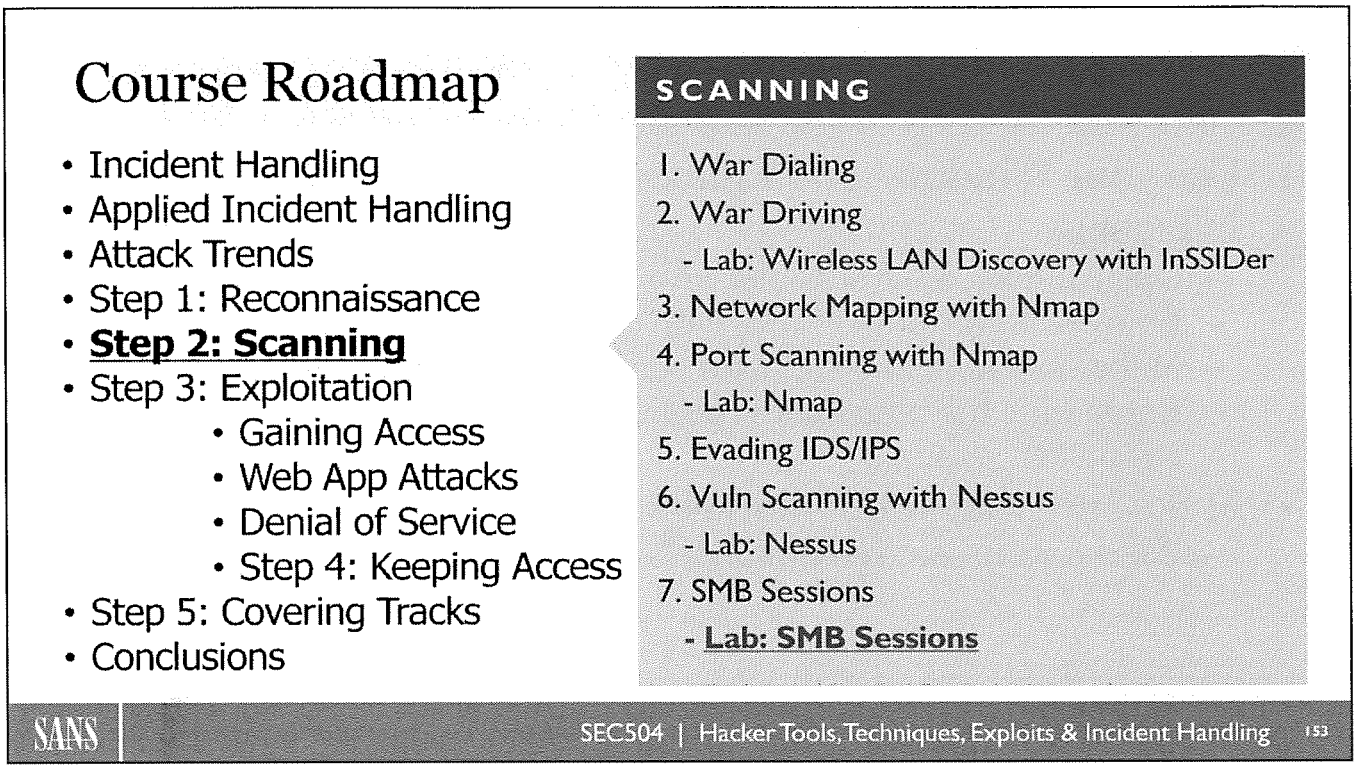
Preparation: Defenses Against Evil SMB Sessions (2)

SMB Sessions

- Preparation (cont.)
  - Block access to the following ports across network boundaries where SMB sessions are not required for admin or file share usage:
    - **TCP/UDP 445**: MS Server Message Block
    - **TCP 135**: RPC/DCE Endpoint mapper
    - **UDP 137**: NetBIOS Name Service
    - **UDP 138**: NetBIOS Datagram Service
    - **TCP 139**: NetBIOS Session Service
    - Of course, block all ports except those required
  - Alternatively, allow access to these ports only from systems or networks that absolutely require SMB access to a given destination (such as file servers and domain controllers)
    - Private VLANs (PVLANs) are a switch feature that can help implement this
- Identification
  - Check for access to the ports listed above in logs and IDS alerts
- Cont, Erad, Rec: N/A

Typically, you need to allow SMB sessions only from clients to a specific set of servers (such as file servers and domain controllers). Usually, you don't need clients to establish SMB sessions to other clients. Thus, you can implement some solid defenses by configuring routers and firewalls to block SMB sessions with TCP port 445, as well as the NetBIOS ports TCP and UDP 135 through 139. Allow such traffic only to specific systems where there is a business need for SMB.

Some organizations deploy client systems on Private VLANs (PVLANs), a switch feature that provides them a major degree of control of what goes into and out of the network interface connecting to each individual host. With PVLANs, you could block all inbound SMB to client machines and allow outbound SMB only to specific servers.



Let's finish SEC 504.2 by conducting a lab on SMB sessions. In this lab, we establish SMB sessions to and from Windows, as well as from Linux to Windows. We interrogate a target Windows machine across those sessions, and we analyze the sessions themselves. We also look at ways to drop inbound and outbound SMB sessions.

Lab: SMB Sessions with net use, smbclient, and rpcclient

- Use your Windows and Linux systems for this lab
- Lab goals
  - Open and list SMB sessions with "net use" and "net session"
  - Enumerate various settings with enum on Windows
  - Make smbclient and rpcclient connections from Linux to Windows
  - Enumerate the target with rpcclient on Linux
  - Drop SMB sessions
- Make sure that your Windows machine is ready
  - We return to some common defaults

We now do a lab associated with SMB sessions, establishing them to and from our Windows machines.

Our goals are to open and list SMB sessions with "net use" and "net session," enumerate all kinds of information on our target Windows machines using enum on Windows, use the Linux smbclient and rpcclient tools to make SMB sessions, enumerate detailed data with rpcclient, and then drop SMB sessions.

These skills are helpful for incident handlers and are often used by computer attackers. You also rely on them on the Day 6 Workshop.

Script for Configuring the Registry and Services

- A few other registry and service settings need to be configured in your Windows machine for this lab to properly function
  - Configuring them manually can be difficult because of typos
  - To help you automatically make those settings, we created a script called ***504msf_exercise.bat*** in the Windows directory of the Course USB
  - Run the script by right-clicking it and selecting "Run as administrator"
  - In addition to making changes to the registry and starting services, the script also creates an additional restore script on your desktop
    - This restore script is called 504msf_restore.bat
- Run the 504msf_exercise.bat script

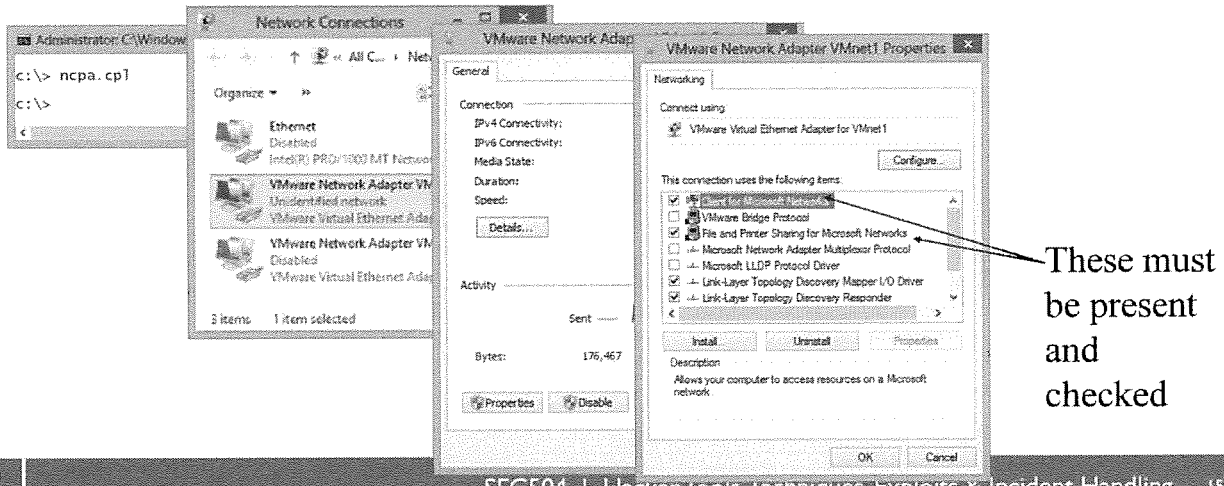
Your Windows machine needs a few registry key settings and service settings. You could make these settings manually, as described on the next few slides. Alternatively, we provide a script on the Course USB that automatically makes all these changes so that you do not have to worry about typos in your reg command syntax or starting services.

Run the script called 504msf_exercise.bat from the Course USB Windows directory. Right-click the script and select "Run as administrator."

This 504msf_exercise.bat script changes the value of various registry keys to enable the Metasploit/psexec exercise to work properly. It also starts the "lanmanserver" service if it is not already running. When you run the script (by right-clicking and selecting "Run as administrator"), it also creates a restore script called "504msf_restore.bat" that restores all the registry settings that were changed back to their original values (and stop the "lanmanserver" service if it had not previously been running). By default, the restore script is created on the current user's desktop, but if you pass the "." parameter when calling the 504msf_exercise.bat script, the restore script is instead created in the current working directory.

Making Sure You Have Client for Microsoft Networks

- For this lab to work, have the Client for Microsoft Networks and File and Printer Sharing bound to your network adapter; run ncpa.cpl to start



For this lab, you should have the "Client for Microsoft Networks" and "File and Printer Sharing for Microsoft Networks" both bound to your network adapter. To check this, run ncpa.cpl to bring up your network interfaces

```
C:\> ncpa.cpl
```

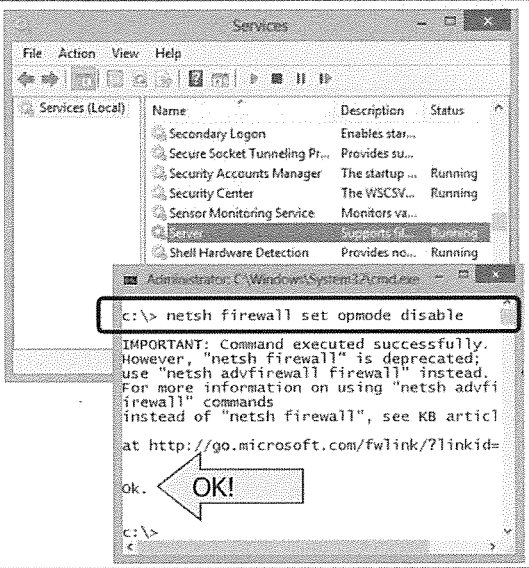
Then, choose your VMnet1 interface by double-clicking it. Go to Properties. Verify that "Client for Microsoft Networks" and "File and Printer Sharing for Microsoft Networks" are present and checked, as shown in the slide's screenshot.

If the "Client for Microsoft Networks" is not present, click "Install" and add it. You may or may not require your Windows Install media to do this, depending on how your system was originally built.

Preparing Your Box for the Lab – Services and Firewall

- Make sure the Workstation and Server services are running
- At command prompt, type "services.msc"
- Verify that "Server" and "Workstation" services are "Started"
- Run the
- **DISABLE A PERSONAL FIREWALL THAT BLOCKS SESSIONS TO LOCALHOST**

```
C:\> netsh firewall set opmode disable
C:\> netsh advfirewall set allprofiles
     state off (For Windows 8+ systems)
```



To prepare your machine for this lab, also make sure the Workstation and Server services are running. At a command prompt, type "services.msc" to bring up the Services control panel:

```
C:\> services.msc
```

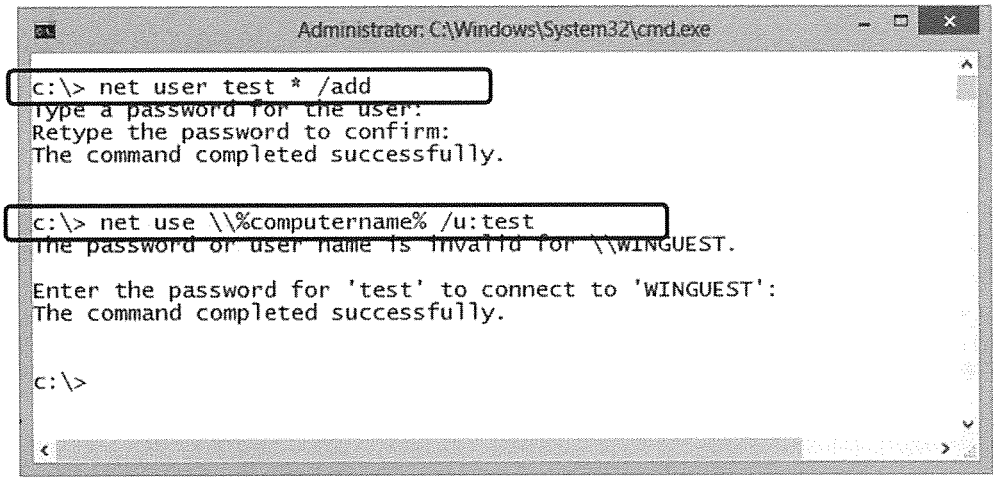
Verify that the "Server" and "Workstation" services are "Started." If they aren't started, click them and start them.

Finally, disable your Windows firewall, which blocks inbound sessions. Do this by running the following command:

```
C:\> netsh firewall set opmode disable
C:\> netsh advfirewall set allprofiles state off (For Windows 8+ systems)
```

On Windows 8, 7, and Vista, you see a message displayed on the screen saying that the firewall context of netsh will be deprecated in the future. However, this command still takes effect and is compatible with all versions of Windows from Windows 7, 8, and 2012, all the way back to Windows XP and 2003. If you see "OK" in the output of this command at the end, it worked.

Create an Account and Open an SMB Session



Let's start by creating an account on Windows that we can use to establish authenticated SMB sessions. To help illustrate the power of authenticated (but non-admin) SMB sessions, this account WILL NOT be in the admin group. On your Windows machine, run an elevated command prompt (by searching for cmd.exe and right-clicking and selecting "Run as administrator").

Then, run the following command to create a user called "test":
`C:\> net user test * /add`

The * means that you want to set the password right now for the user. When prompted, type a password you can remember. You need to type it a second time to set the password for this account.

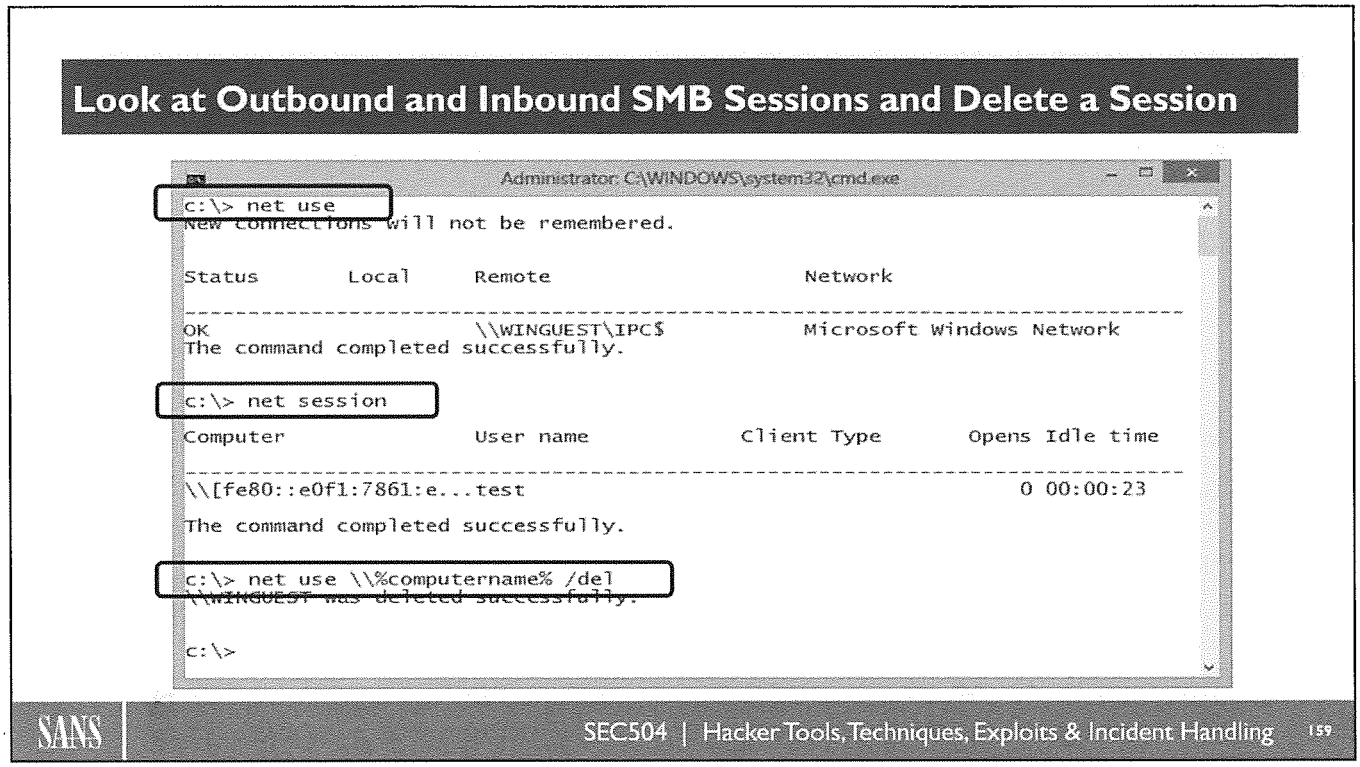
You should see "The command completed successfully." If not, verify that you have an elevated command prompt.

Now, with our non-admin account created, let's establish an SMB session from our Windows machine to our Windows machine. Do this by running the following command:
`C:\> net use \\%computername% /u:test`

This command establishes an SMB session as user test to the machine that is represented in the environment variable computername. This variable contains your system's hostname. The % at the front and the back causes the Windows shell to expand that variable into its value, your localhost name. That's easier than typing the hostname set on some computers that students bring to the class.

When prompted for a password, type in the test account's password. If you see "The command completed successfully," you have established an SMB session with your local machine.

If it fails, try it again, but prepend %computername% before your backslash and username of test (in other words, try: **net use \\%computername% /u:%computername%\test**).



Now, let's look at our outbound SMB session (where we're acting as a Windows SMB client):

```
C:\> net use
```

You should see a session that has a "Remote" name of your Windows system's hostname followed by \IPC$. You attached to the default administrative share.

Now, let's look at the inbound SMB session (where we're acting as a Windows SMB server):

```
C:\> net session
```

Here, you see the connection you've got coming in (on many versions of Windows, it shows your IPv6 address as the connected computer when you establish the SMB session to your local machine this way).

Now, let's drop our SMB session from the client-side:

```
C:\> net use \\%computername% /del
```

The session should go away (which means that both the client and server session are dropped). You can verify that it is gone (from the client side) by running

```
C:\> net use
```

You can verify that it is gone from a server side by running

```
C:\> net session
```

Both should show you no active sessions, because you deleted the SMB session we established.

Networking Windows and Linux

- Configure Windows and Linux to communicate using host-only networking
  - If you use a Windows *host* machine (most students in the course):
    - Use ncpa.cpl to set Windows VMnet1 IP address to 10.10.0.1 with netmask 255.255.0.0
    - Set VMware Linux guest to use Host-only networking (VM→Settings, or Player→Manage→Virtual Machine Settings, or press CTRL-D in VMware)
    - Linux IP address is 10.10.75.1
  - If you use a Windows *guest* machine
    - Use ncpa.cpl to set Windows Local Area Connection IP address to 10.10.0.1 with netmask 255.255.0.0
    - Set VMware guests (both Windows and Linux) to use Host-only networking
    - Linux IP address is 10.10.75.1
- Make sure that your Linux machine can ping Windows

```
C:\> ping 10.10.75.1
# ping 10.10.0.1
```

- If you can't ping, double-check that you disabled your firewall (from an elevated command prompt on Windows)

```
C:\> netsh firewall set opmode disable or C:\> netsh advfirewall set allprofiles state
off (For Windows 8+ systems)
$ sudo ifconfig eth0 10.10.75.1 netmask 255.255.0.0
$ sudo iptables -F
```


Let's make sure we have communication enabled between our Windows and Linux machines so that we can use our Linux VMware guest to establish SMB sessions to our Windows machine.

If you are using a Windows *host* machine for this exercise, run ncpa.cpl and select your VMnet1 interface. Make sure that you set its IP address to 10.10.0.1 and netmask to 255.255.0.0. Then, in VMware, bring up the virtual machine settings (by going to VM→Settings or Player→Manage→Virtual Machine Settings, or pressing CTRL-D in VMware). Select network adapter and click the "Host-only" radio button.

If you have a Windows *guest* machine inside VMware (along with our Linux guest machine in VMware), run ncpa.cpl inside your Windows guest. For the Local Area Connection, set an IP address of 10.10.0.1 and netmask of 255.255.0.0. Then, for both the Windows guest and the Linux guest, in VMware, bring up the virtual machine settings (by going to VM→Settings or Player→Manage→Virtual Machine Settings, or pressing CTRL-D in VMware). Select network adapter and click the "Host-only" radio button.

Your Linux guest machine that we provided has the IP address of 10.10.75.1 by default.

Make sure that you can ping from Windows to Linux and vice versa:

Windows: `C:\> ping 10.10.75.1`

Linux: `# ping 10.10.0.1`

If you can't ping, make sure that your firewall is off on Windows and Linux by running

Windows: `C:\> netsh firewall set opmode disable`

`C:\> netsh advfirewall set allprofiles state off (For Windows 8+ systems)`

Linux:

`$ sudo ifconfig eth0 10.10.75.1 netmask 255.255.0.0`

`$ sudo iptables -F`

Using smbclient to Get a List of Shares

```
sec504@slingshot:~$ smbclient -L 10.10.0.1 -U test
Enter test's password:
Domain=[8415] OS=[Windows 7 Ultimate 7601 Service Pack 1] Server=[Windows 7 Ultimate 6.1]

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Connection to 10.10.0.1 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
NetBIOS over TCP disabled -- no workgroup available
```


We start by using smbclient on Linux to pull a list of shares from Windows. Type the following command:

```
# smbclient -L 10.10.0.1 -U test
```

When prompted, type your test account's password.

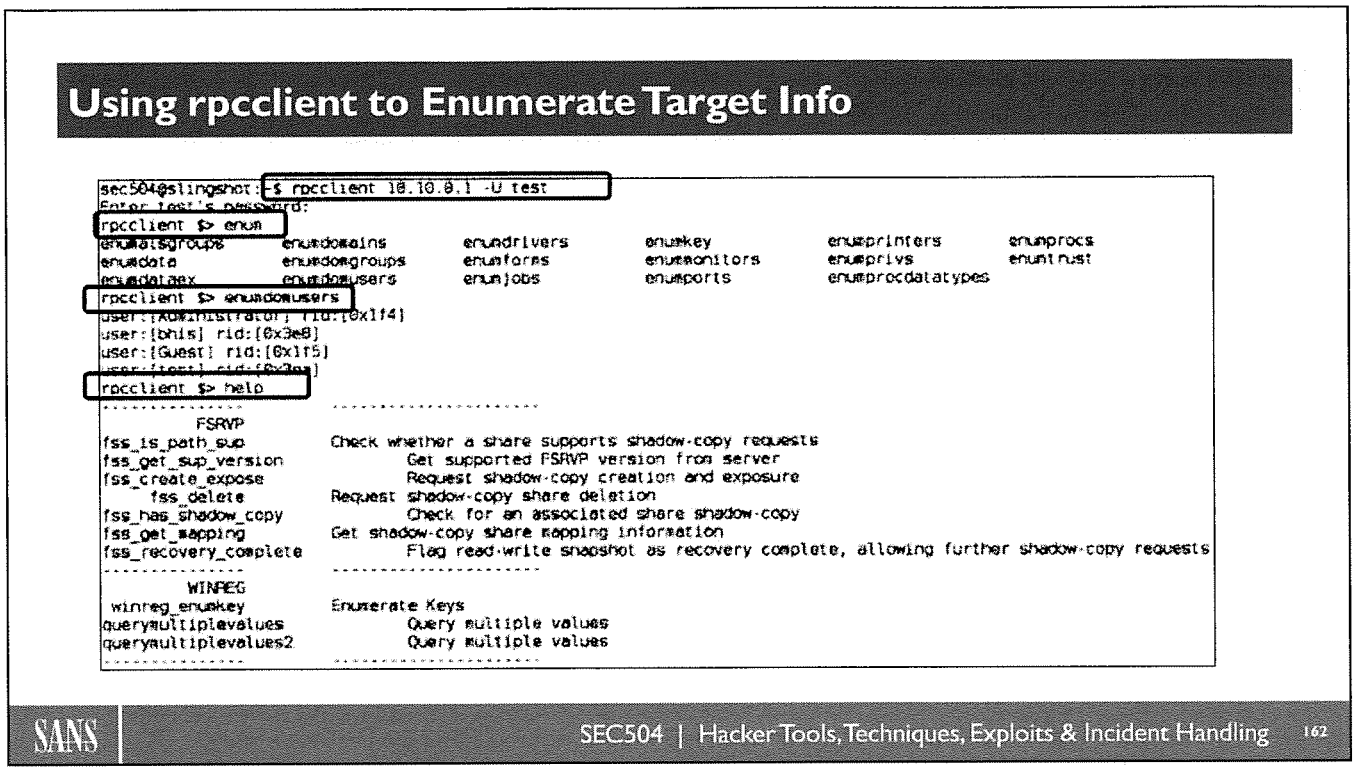
You should see a list of shares on the Windows box, including ADMIN$, IPC$, and C$. These are the default admin shares that the Windows "net view" command hides by default. But, smbclient can see them. You may see additional shares if you created any.

You may also see some warning messages at the bottom of the output (as shown on the slide's screenshot), including "Called name not present," "NetBIOS over TCP disabled," and more. You can safely ignore these. The share list is the focus of this lab.

If it fails on your system, double-check your password. If your password is correct, you may have some system hardening that blocks the connection. You can try to run the following command on Windows to set your SMB authentication to "Accept NTLMv2" (the default on most Windows systems including Vista, Windows 7, and Windows 8) instead of the more hardened "Accept NTLMv2 Refuse LM & NTLM:"

Only if it fails, type the following and try to connect again with smbclient:

```
c:\> reg add hklm\system\currentcontrolset\control\lsa /v
lmcompatibilitylevel /t REG_DWORD /d 3
```



Let's dig into this target by using the Linux rpcclient program. Run it as follows:

```
# rpcclient -U test 10.10.0.1
```

You are prompted for the test account's password. Enter that. If you have access to the target, you should see the rpcclient prompt:

```
rpcclient $>
```

If you don't see this prompt, the next slide has some troubleshooting suggestions for your configuration.

If you see the rpcclient prompt, let's experiment with some commands to extract information from the target. First, note that the rpcclient prompt has Tab autocomplete. Type "enum" at the prompt WITH NO SPACE AFTER IT, and then hit the Tab key twice (<TAB><TAB>):

```
rpcclient $> enum<TAB><TAB>
```

You now see all the commands that rpcclient has that match the string "enum." We can enumerate many things. Let's try enumerating users:

```
rpcclient $> enumdomusers
```

This command shows all users on the box (local users and any domain users the system knows about). We can see the users' names and their Relative Identifiers (RIDs), which are the suffix of the SID number for each account in hexadecimal form. (The admin account has a RID of 0x1f4, which is decimal 500.)

To get an idea of all the commands available within rpcclient, run the following:

```
rpcclient $> help
```

A huge number of commands are available. Let's explore some of the most useful ones.

If It Can't Connect, Add Windows Hostname to /etc/hosts

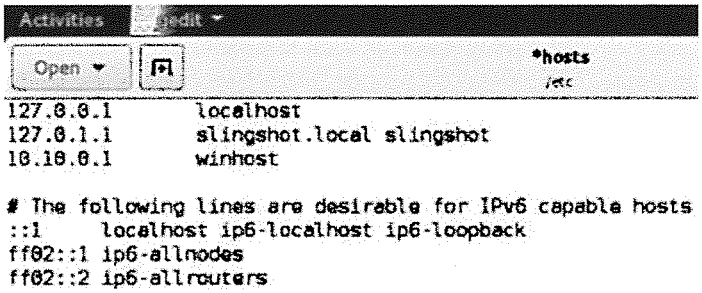
- If rpcclient cannot connect, on your Windows box, determine your hostname

```
C:\> hostname
```

- Take that hostname and append a line to the /etc/hosts file on Linux that maps the IP address of Windows (10.10.0.1) to its hostname

```
# gedit /etc/hosts
```

Append a line that contains
10.10.0.1 [WinHostname]→



If the rpcclient program connected to your Windows machine and you were able to run enumdomusers, move to the next slide.

If the rpcclient program could NOT connect to your Windows machine, you may need to enter a line in your /etc/hosts file on Linux associated with your Windows machine. Some versions and configurations of Windows require this for rpcclient to work properly. Start by determining your Windows hostname by running the hostname command on Windows:

```
C:\> hostname
```

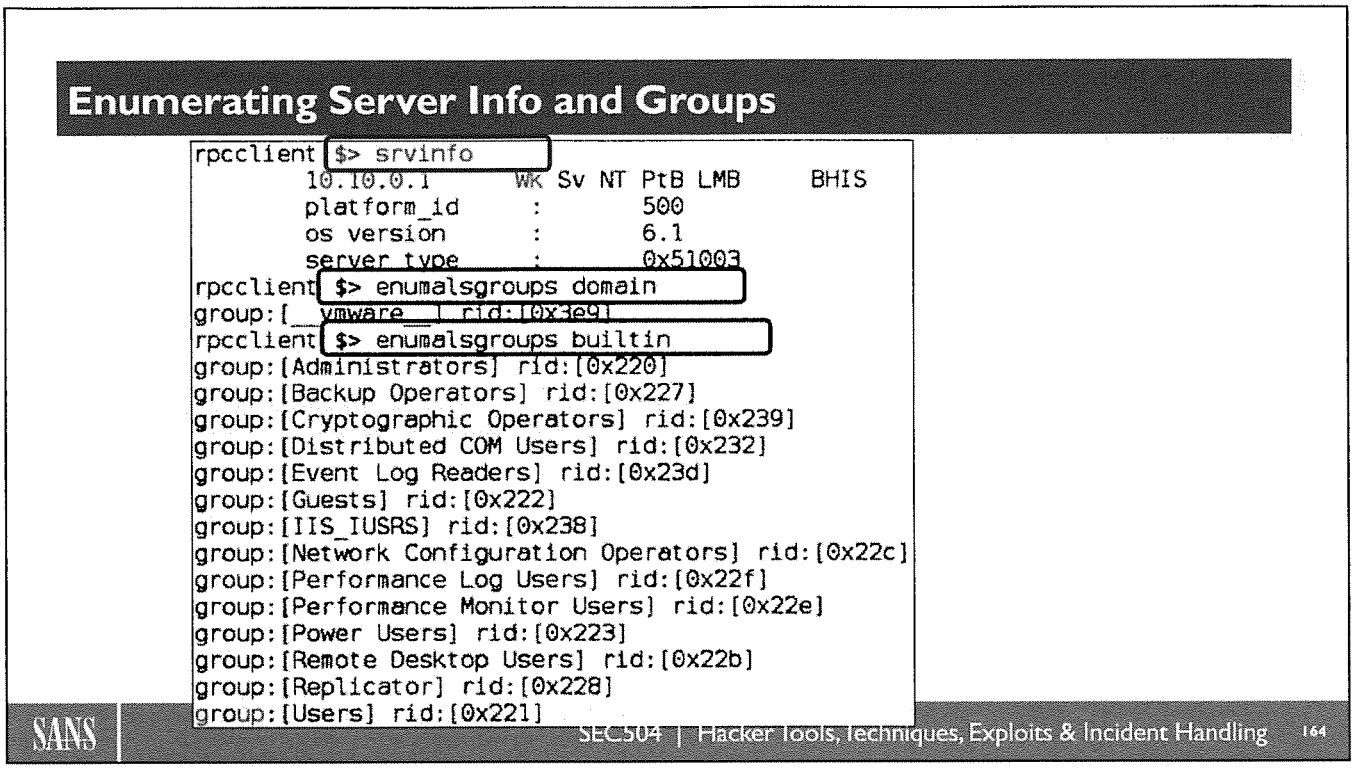
Then, on Linux, open the /etc/hosts file in your favorite editor (such as gedit):

```
# gedit /etc/hosts
```

Now, append a line at the bottom of /etc/hosts that maps your Windows IP address of 10.10.0.1 to your Windows hostname (separate the IP address and hostname by a tab):

```
10.10.0.1    [WinHostname]
```

Save the /etc/hosts file and return to the previous slide to see if rpcclient can connect.



Let's use rpcclient to enumerate server info:

```
rpcclient $> srvinfo
```

Here, you see the IP address and the OS version. The following list shows common Windows versions and their associated OS version numbers:

Windows 8	6.2	Windows Server 2003 R2	5.2
Windows Server 2012	6.2	Windows Server 2003	5.2
Windows 7	6.1	Windows XP 64-Bit Edition	5.2
Windows Server 2008 R2	6.1	Windows XP	5.1
Windows Server 2008	6.0	Windows 2000	5.0
Windows Vista	6.0		

Now, let's get a list of groups. First, we pull domain-related groups (typically groups created on the local machine either by an admin there or within the domain):

```
rpcclient $> enumalsgroups domain
```

Remember, the "als" in the middle of enum and groups stands for "alias."

Next, we pull internal groups (typically the default groups defined by Microsoft):

```
rpcclient $> enumalsgroups builtin
```

Together, these are all the groups defined on the machine. Note that we have the group names and their RIDs in hexadecimal form.

Enumerating Admin Group Membership

```
rpcclient $> lookupnames administrators
administrators S-1-5-32-544 (Local Group: 4)
rpcclient $> queryaliasmem builtin 544
        sid:[S-1-5-21-901181562-3235664568-2803082179-500]
        sid:[S-1-5-21-901181562-3235664568-2803082179-1000]
rpcclient $> lookupsids S-1-5-21-901181562-3235664568-2803082179-1000
S-1-5-21-901181562-3235664568-2803082179-1000 8415\bhis (1)
```


Now, we see how to get a list of users included in a group, specifically the administrators group. First, we need to look up the SID for the administrators group, using the lookupnames command in rpcclient:

```
rpcclient $> lookupnames administrators
```

Make sure you have an "s" at the end of administrators, or else you are looking at the administrator account, not the administrators group. In your output, you see the SID for the administrators group. Note the last three digits of that SID number (which is likely 544). That is the RID (in decimal form) for the administrators group.

If your RID is a number other than 544, write the result here: _____

To see which accounts are in the administrators group, we use the "queryaliasmem" command, which shows membership of groups, as follows:

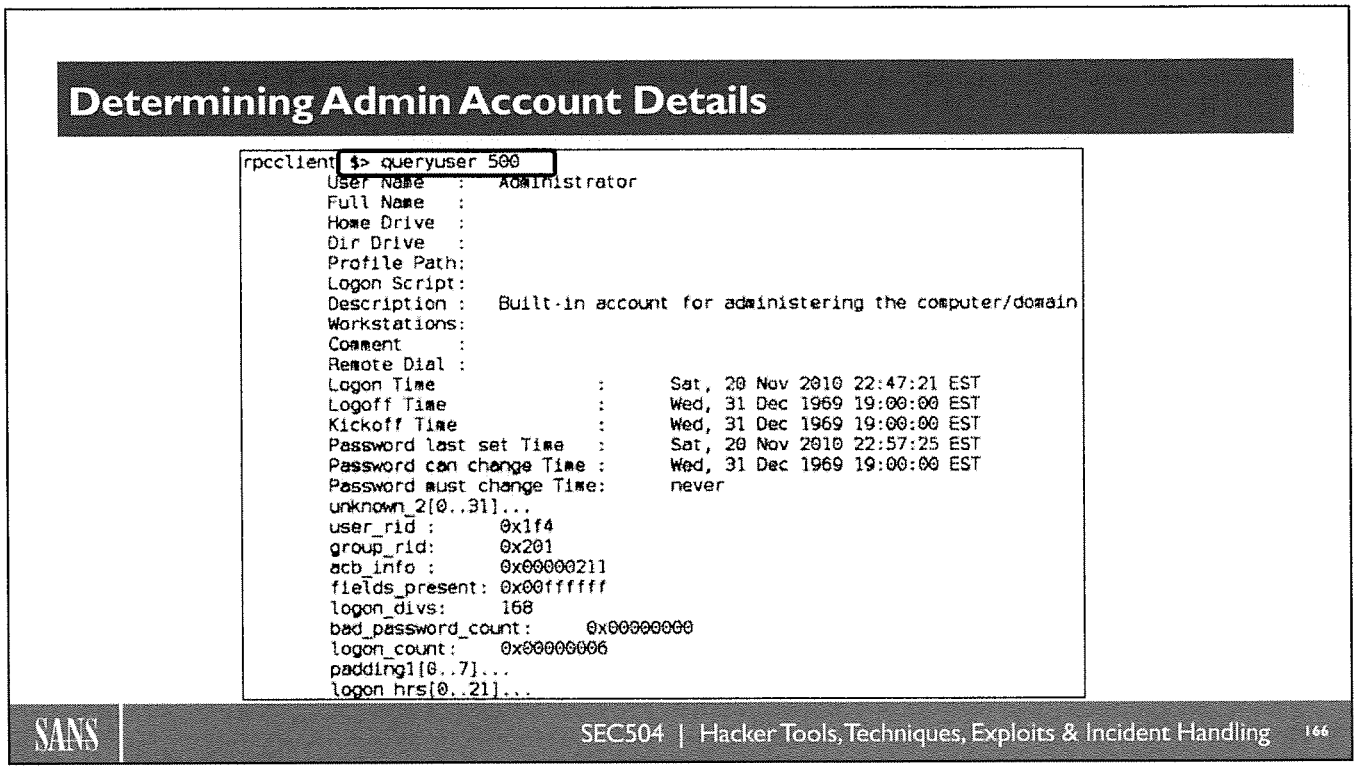
```
rpcclient $> queryaliasmem builtin 544     (or substitute your RID if it
                                            is something other than 544)
```

We now have the SIDs for all members of the administrators group. To determine the associated username, highlight one of these SIDs (from the S-1- all the way to the end, where there should be a number like 500 or 1001 and up). Then, go to your terminal's Edit menu and select Copy.

Now, at your rpcclient prompt, type the following command, pasting in the SID you just copied:

```
rpcclient $> lookupsids [PastedSID]
```

You should see the name of the account in the administrators group. By using this technique, you can determine the names of all the administrators on the box.



We can get even more info about a given user account using the rpcclient queryuser command. Let's run it with an RID of 500, which is the RID of the original administrator account in Windows. Even if the administrator account is renamed, it still has this RID:

```
rpcclient $> queryuser 500
```

In our output, we can see the account's name and other details, such as the last time the user set the password for this account! We can also see the bad_password_count for logon failures and more.

Looking Up SIDS and Dropping SMB Sessions

```
rpcclient $> lookupnames test
test S-1-5-21-901181562-3235664568-2803082179-1002 (User: 1)
rpcclient $> lookupnames administrator
administrator S-1-5-21-901181562-3235664568-2803082179-500 (User: 1)
rpcclient $> lookupnames administrators
administrators S-1-5-32-544 (Local Group: 4)
rpcclient $> srvinfo
        10.10.0.1      Wk Sv NT PtB LMB     BHIS
        platform_id     :       500
        os version      :       6.1
        server type     :       0x51003
```

```
C:\Windows\system32>net session
Computer                User name            Client Type        Opens Idle time
-------------------------------------------------------------------------------
\\10.10.75.1            TEST                 Unix                    5 00:01:32
The command completed successfully.

C:\Windows\system32>net session \\10.10.75.1 /del
The session from 10.10.75.1 has open files.
Do you want to continue this operation? (Y/N) [N]: Y
```


Let's look at a couple of other accounts with the rpcclient lookupnames feature, starting with our test account:
`rpcclient $> lookupnames test`
We see the SID of our test account.

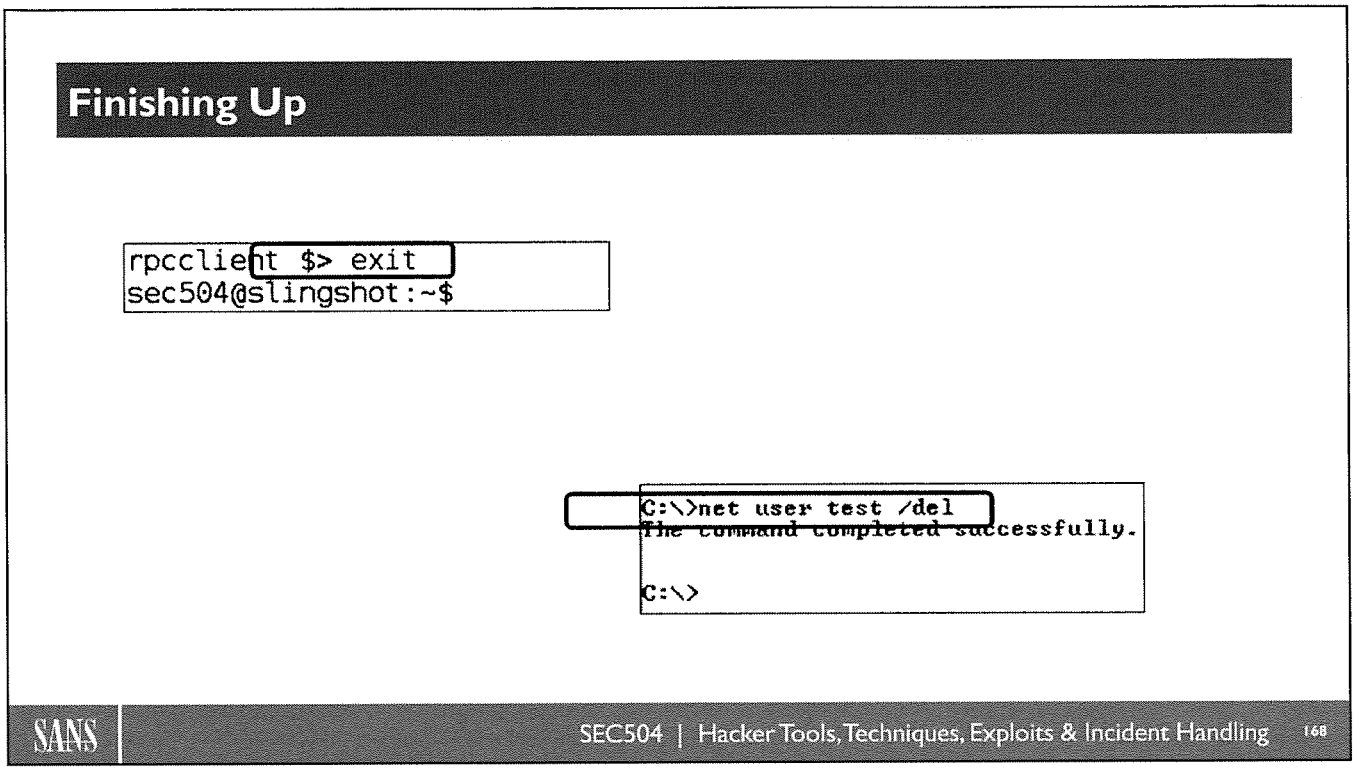
Next, let's look up an account called "administrator:"
`rpcclient $> lookupnames administrator`
We see the administrator SID (with its RID of 500).

Now, let's look up a name that is a group, not a user, by adding an s at the end of administrator:
`rpcclient $> lookupnames administrators`
Here, we see the SID of the administrators group (group SIDs are usually shorter than user SIDs).

Finally, let's see what happens if we disconnect the rpcclient's SMB session on our Windows box. At a Windows command prompt, run
`C:\> net session`

You should see an inbound session from a client type of "UNIX." rpcclient made this session from Linux to Windows. Let's drop it in Windows:
`C:\> net session \\10.10.75.1 /del`
You are prompted to make sure that you want to do this. Press Y and then Enter. The session should close.

Now, try to run srvinfo in rpcclient on Linux:
`rpcclient $> lookupnames administrators`
It won't work, because the Windows machine shut down that SMB session. Nice! On Windows, we can kill individual SMB sessions at will by using this technique.



To finish the exercise, exit our rpcclient program:

```
rpcclient $> exit
```

Then, remove the test account we created on Windows:

```
C:\> net user test /del
```

Lab Conclusions

- In this lab, we covered how to
  - Create Windows accounts at the command line
  - Make SMB sessions with the Windows "net use" command
  - Analyze and drop SMB sessions with "net use" and "net session"
  - Use Linux rpcclient to enumerate users, groups, group membership, and other detailed account information
  - These are immensely useful capabilities for attackers... and incident handlers

In conclusion, in this lab, we've seen some of the power of SMB sessions, particularly in harvesting information from target Windows machines. We covered how to make and drop SMB sessions at the Windows command line. We looked at interrogating Windows machines using enum from Windows. We also looked at how to use smbclient and rpcclient on Linux to gain detailed information from target Windows machines.

The best part of this is that it can all be done with just a single username and password on the target system, even if that user does not have admin privileges on the target box.

Attackers use these capabilities on a regular basis, and incident handlers can also benefit from them as they analyze the security settings of systems in their environments.