560.1

Comprehensive Pen Test Planning, Scoping, and Recon

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE “USER”) AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With this CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, USER AGREES TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, USER AGREES THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If User does not agree, User may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There’s an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP® and PMBOK® are registered trademarks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

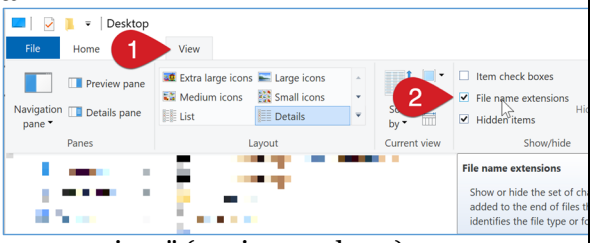
SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

Welcome to SEC560: Enterprise Penetration Testing

See the Workbook for detailed instructions on setting up your VMs

- Download the ISO file from your SANS Portal
- Open the ISO file
- Copy the two 7z files to your desktop
- Use 7zip (*Utilities* directory in ISO) to extract the files to your desktop
- Double click on the .vmx file in each of the two newly created directories
  - Windows users: you may need to enable "File name extensions" (see image above)



- Credentials for both systems are: `sec560 / sec560`
- Follow the instructions in the workbook to connect to the lab infrastructure

Please complete the steps in Lab 1.0 (Setting Up the Virtual Machine Images) before class begins. If you need assistance, ask your TA or Instructor.

Welcome to SEC560: Network Penetration Testing and Ethical Hacking! Please complete the steps described in the workbook and wiki "Lab 0".

You will be using two virtual machines (VMs) during the lab exercises throughout this class. To allow for more class and lab time, you will need to configure your VMs before class begins. If you need help, please contact the instructor, teaching assistant (TA), or the support contact identified in the email you have received.

Throughout the class we will be using two VMs customized for this course, a Windows 10 VM and a Slingshot Linux VM based on Ubuntu. The VMs have been customized and tested with the course content. Your personal copy of Windows 10, Kali, or another VM will likely not work in this course as they will be missing tools and artifacts necessary to complete this class, and the tool versions may be incompatible with the target infrastructure. The authors of this course updated and patched software in the VMs to ensure it works in this course. Outside tools may not function as expected.

SEC560.1 | Enterprise Penetration Testing

SANS

Comprehensive Pen Test Planning, Scoping, and Recon

Hello and welcome! The purpose of this course is to prepare you to perform ethical hacking and penetration testing projects in a professional, safe, and repeatable manner for your organization. Also, by covering some powerful attack techniques, this course is designed to help all security professionals (not just penetration testers) improve the security stances of their organizations. Throughout this course, we cover hundreds of tools and techniques, detailing how you can use them to find vulnerabilities in your organization to help improve your organization's security. We include hands-on labs throughout, culminating in a full-day, capture-the-flag penetration test lab for the entirety of 560.6.

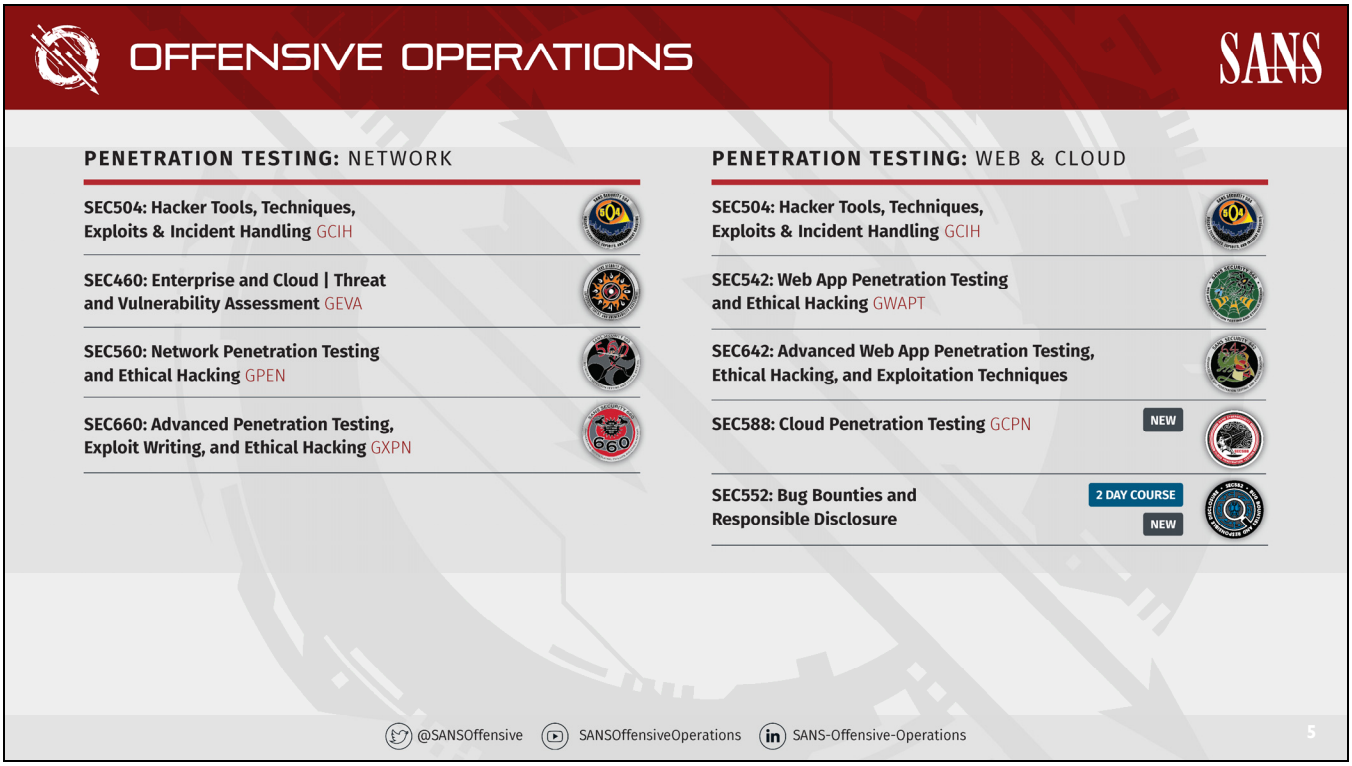
Let's keep this session interactive. If you have a question, please let the instructor know. Discussions about relevant topics are incredibly important in a class like this because we have numerous attendees with various levels of skill attending the class. Share your insights and ask questions. The instructor does reserve the right, however, to take a conversation offline during a break or outside of class in the interest of time and the applicability of the topic.

TABLE OF CONTENTS (1)	SLIDE

This slide is a table of contents (1) and, also, acts as an overview of what we will discuss throughout 560.1.

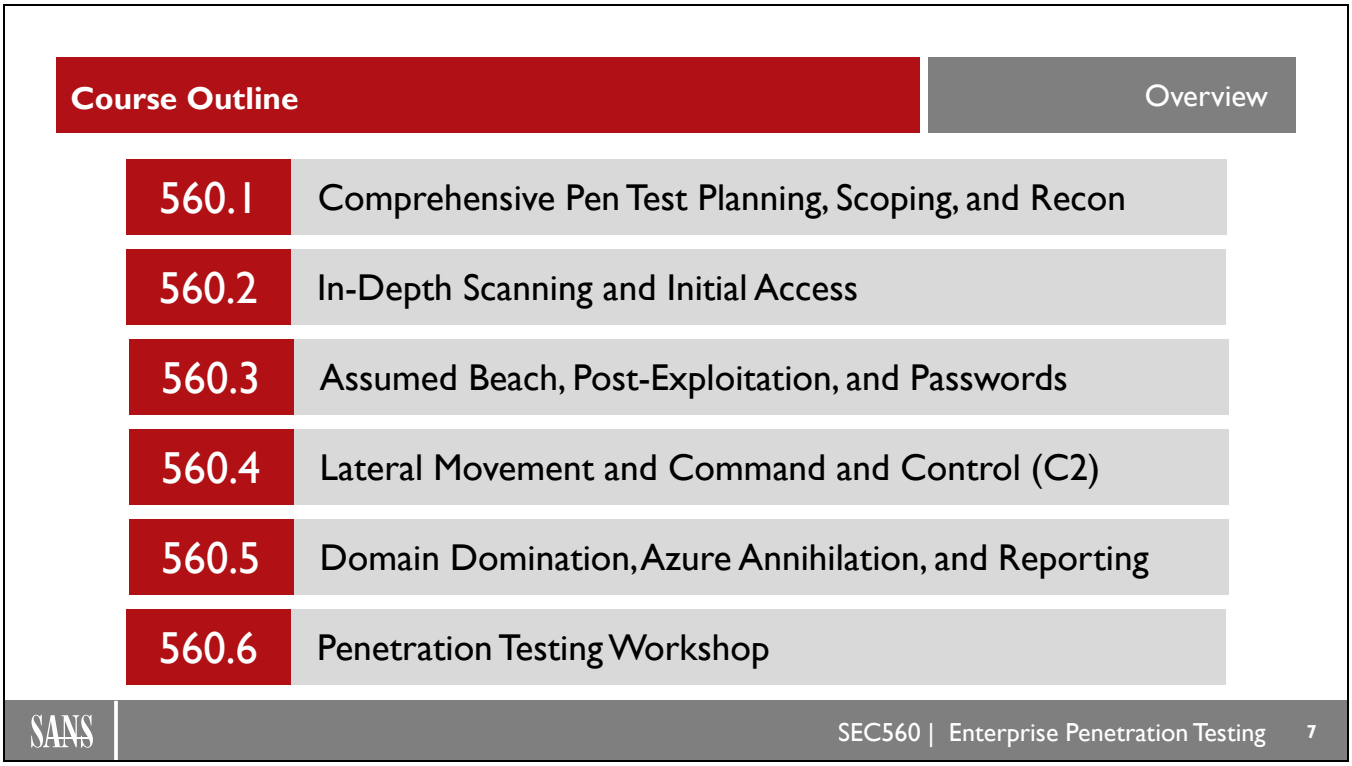
TABLE OF CONTENTS (2)	SLIDE

This slide is a table of contents (2) and, also, acts as an overview of what we will discuss throughout 560.1.



This page intentionally left blank.

This page intentionally left blank.



This course is divided into several sections, each designed to prepare you in a vital aspect of Network Penetration Testing and Ethical Hacking:

- 560.1 sets the stage, defining terms and providing a detailed discussion of the planning process, including building a penetration testing and ethical hacking infrastructure, establishing ground rules for testing, and scoping projects. This section also covers the reconnaissance phase of a test in detail.
- 560.2 zooms into scanning, covering the tools and techniques that professional penetration testers and ethical hackers need to master to find target machines, openings on those targets, and vulnerabilities. We'll also cover ways to initially gain access, such as through password guessing attacks and exploitation.
- 560.3 deals which post-exploitation, what we do after our initial compromise or access. Initial access can be difficult, so modern testing may be more efficient if access is ceded to the penetration testers in a test known as Assumed Breach (or Assumed Compromise). We'll discuss methods to conduct an Assumed Breach test to get more value out of a penetration test.
- 560.4 deals with lateral movement or moving from one compromised system to another. We'll also discuss Command & Control (C2 or C&C). Maintaining and managing access to multiple systems can be difficult, so we can use C2 frameworks to manage the systems and credentials.
- 560.5 deals with the Microsoft Windows domain, the brain and nervous system of most organizations. Modern attackers not only look at the computers on the network, but also key on their interconnectivity. This section concludes with a review of Azure and its integration with the Windows Domain.

This all leads to 560.6, in which we apply the concepts from throughout the course in a full-day, end-to-end penetration test lab, which includes a Capture the Flag (CtF) contest. The skills you master from the labs throughout the class will be applied in this lively contest in the last section.

About the Course

Overview

- Our focus is helping you to master the skills needed for hands-on Network Penetration Testing and Ethical Hacking
- Organized around the workflow of professional testers
- Numerous hands-on labs, culminating in a full-day, end-to-end penetration test
- Tips for avoiding common pitfalls and for saving time, making the tester more efficient

The overall objective for this course is to help prepare you with the skills needed to perform Network Penetration Testing and Ethical Hacking. Some people who take this class are professional penetration testers looking for some extra tips and tools for their arsenals. Others have never hacked a box before and want to get started. Others are cyber defenders who want to learn more about the offensive skills attackers use. Some course attendees are forensics experts looking to better understand the attacks they will analyze. We welcome attendees from across the spectrum of information security professionals. We have strived to develop the materials to help you master the skills of a network penetration tester and ethical hacker regardless of where you sit on that spectrum.

This course is organized around the workflow of a professional penetration tester and ethical hacker, describing the various steps and options a tester takes at each step. Note that the general flow of work, however, isn't set in stone. Good testers are pragmatic, often improvising based on the particulars of a given project when the opportunity arises. The class includes numerous hands-on labs, each of which is designed to impart an important skill that network penetration testers and ethical hackers require.

The course is also chock-full of tips for avoiding common pitfalls that network penetration testers and ethical hackers face. Based on input from numerous professional penetration testers who have learned these lessons the hard way, these tips throughout the course are designed to help you maximize the effectiveness of your own penetration practices. Also, many of these tips are designed to save you a lot of time, making you more efficient. Often, when testing, you need to achieve some goal. One way of going about that goal may take three hours and work only 10% of the time, whereas another method might take three minutes and have a 90% success rate. Following the tips of this class can help you focus your valuable time on the latter.

Two, often contradictory-sounding concepts

Do things differently
- Think outside of the box
- Unconventional
- Pragmatic
- Follow promising leads

Methodical
- Thorough
- Careful
- Repeatable process
- Documentation

Balance between these two is crucial for success

At the outset of this class, let's briefly explore the mindset of penetration testers and ethical hackers. A noted penetration tester, someone whose name you would likely recognize but who has requested anonymity, said, "We break computers, making them do stuff that their designers, implementers, and system administrators didn't plan on them doing".

That's what our job is: find flaws that enable attackers to do evil on target machines so that an organization can better understand its business risks and resolve vulnerabilities before mayhem ensues. However, to successfully achieve that goal, penetration testers and ethical hackers must maintain a mindset that involves two often contradictory-sounding concepts.

First, a penetration tester or ethical hacker must be flexible and pragmatic, thinking outside of the box. To be successful, you need to think differently than most traditional system administrators or network architects, trying to solve problems in often untraditional ways.

But at the same time as you wield your pragmatic style, you have to be thorough, methodical, and careful. Your work, to be valuable, must be understandable and reproducible so that the target organization can understand its vulnerabilities and risks and take action to mitigate the flaws. You need to take good notes and produce a high-quality report that presents your findings in a digestible form for people who don't perform penetration testing or ethical hacking professionally—people who may not share your pragmatic, "think-differently" mindset.

Some people struggle with this mindset, erring by allowing one side to dominate over the other. However, many people can resolve this conflict between these two mindsets, thus balancing them. To be a successful penetration tester, you need to strive for this balance.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

To start the session, we need to define some terms so that the terminology is consistently used throughout the rest of the class. What is ethical hacking? How is it associated with penetration testing? How do vulnerability scans and penetration tests differ? We address each of these questions next.

It is important to note that different people use the various terms we define in different ways. We present a set of definitions that are common but not universal. In other words, we introduce this common terminology so that we can be consistent throughout this course and with the most common use of these terms. However, keep in mind that usage can vary for your organization or with some of the enterprises that you test.

Vulnerability

Defining Terms

A flaw or weakness that can be exploited by a threat actor

- MITRE's Common Weakness Enumeration (CWE) lists a number of software and hardware weakness types https://cwe.mitre.org/
  - Memory corruption flaws
  - Misconfiguration
  - Design and architecture flaws
  - Missing audit features
  - Weak/guessable/default passwords

RFC 4949 defines vulnerability as, "A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy"

As penetration testers, our goal is to exploit security vulnerabilities, but what is a vulnerability? RFC 4949 defines a vulnerability as:

- A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

Simply put, we can define it as a flaw or weakness that can be exploited by a threat actor. Most people associate vulnerabilities with memory corruption flaws (e.g., buffer overflows), but that is only a portion of all the security vulnerabilities. Vulnerabilities include:

- Misconfiguration
- Design and architecture flaws
- Missing audit features
- Weak/guessable passwords
- Default credentials

The above list is by no means exhaustive. There are many other categories of vulnerabilities that are not listed above. Miter created the Common Weaknesses Enumeration, which is a list of hardware and software weakness types. This isn't to be confused with CVE (Common Vulnerabilities and Exposures), which lists specific flaws in a specific piece of software.

Threat

Defining Terms

An **agent or actor** that can **cause harm**

- A threat could be human
- A malicious actor on the keyboard with intent to cause harm
- A user who accidently clicks the wrong button
- A threat could be code
- A worm that the author no longer controls
- A threat doesn't have to be intentional, it could be accidental
- What if a web app scanner "clicked" on a delete button?

A threat is an agent or actor that can cause harm. More broadly, it is anything that can cause harm. A threat could be the organized crime attacker attempting to stake credit card numbers, but it also could be the user who accidentally clicks the wrong button. A threat doesn't even have to be human. A worm is a threat, since the code is what is propagating and causing damage. Similarly, our own security tools can be considered a threat in certain contexts. A web application vulnerability scanner will crawl the website and submit data. If the scanner is pointed at a production website with production data, what would happen if the scanner "clicked" the delete button? Scanners will attempt to avoid sensitive actions, but things like these do happen. This is one of the reasons it is important to have backups and to have test systems.

Exploit

Defining Terms

Code or technique that **takes advantage** of a vulnerability

- Publicly available exploit code to take advantage of a buffer overflow
- Uploading a web shell to a web server
- Abusing input filtering to execute code
- Using an improperly secured executable to escalate permissions

We'll spend more time discussing exploit categories later in this course

An exploit is a code or technique that takes advantage of a vulnerability. Here are a number of examples:

- An exploit could take advantage of a vulnerably in memory management and allow the attacker to gain remote access to the system
- An attacker could use an exploit and could use existing functionality to upload code and execute it, such as uploading a web shell to a web server using a feature that allows users to upload a resume or avatar
- An attacker changes their name to JavaScript, exploiting the weak input filtering on the website
- An attacker could run an executable that executed privileged operations on the system, thereby escalating access

There are many different ways exploits can take advantage of vulnerabilities. In fact, the same vulnerability may be exploited differently by different threats depending on their end goal. We'll discuss exploits and exploit classes in more depth later in this course.

Potential for loss or damage

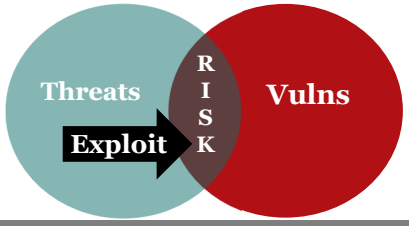
- Often calculated as: Risk = Likelihood * Impact
  - Potential impact is a number we can easily calculate with reasonable certainty
  - Probability of occurrence (likelihood) is often much more difficult to calculate
- Many times, the "risk" number comes down to expert opinion based on the individual's or team's knowledge and experience
- Balancing risk is one of the fundamental goals of the security team
- As a penetration tester, often the most important (and sometimes most difficult) task is estimating the risk of issues found within the organization

Risk is defined as the potential for loss or damage. Risk is often calculated as (Probability of Occurrence) * (Potential Impact). The potential impact is a number we can calculate with a reasonable degree of certainty; however, the likelihood is a much more contentious number. Often, the likelihood (and ultimately the risk value itself) comes down to the expert opinion of the security team. The team uses its collective knowledge and experience to determine the risk and ultimately provide guidance to the business. Balancing the risk with organizational needs is one of the fundamental goals of the security team. Unfortunately, many security teams are viewed as roadblocks—hindering the organization instead of enabling it develop and progress securely.

As a penetration tester, part of your role is to estimate the risk related to the vulnerabilities you uncover. This is a powerful and important role. Junior testers will often overestimate the risk of vulnerabilities. Their reports will often be top heavy, where most of the risks are categorized as high or critical. As penetration testers mature, they learn to understand the risk and how it plays into the organizational processes. Understanding the risks, and identifying them, accordingly, helps the organization prioritize remediation efforts.

Risk is where a threat, vulnerability, and exploit overlap

- Risk reduction or elimination examples:
  - Preventing the threat from accessing the system
  - Removing sensitive data from the system
  - Using a web application firewall, IPS, etc. to prevent an exploit from working (be careful!)
  - Patching the vulnerability



We can simplify the risk by thinking about it as the intersection of the threat, vulnerability, and the risk. If we can reduce or eliminate one of these pieces, the risk is reduced or eliminated, respectively. We can limit access to the system, preventing a threat from accessing the service. Similarly, if the system no longer has sensitive data (PII, passwords, etc.), then there is no risk since it provides the attacker no value and no data can be lost, stolen, or destroyed.

We can implement filtering tools to prevent an exploit from working. For example, we could use a WAF (web application firewall) to filter the characters of what would trigger the vulnerability. Similarly, an IPS could prevent specific actions against the target system that would trigger an exploit. In both cases, the vulnerability still exists, but it can't be exploited. Note: be careful with this approach as attackers will often change signatures or encoding to bypass these tools.

Finally, we could remove the risk by patching the vulnerability.

We need consistent terms

- Penetration Test
- Red Team
- Purple Team
- Vulnerability Assessment

We differentiate the terms, recognizing that others do not

There is another set of terms that many information security practitioners use interchangeably, which results in a lot of confusion. The following terms are associated with what an ethical hacker or penetration tester actually does on a day-to-day basis:

- Penetration testing
- Red Teaming
- Vulnerability assessment (and security assessment)
- Security audit

Although these terms are often used interchangeably, they do have subtle distinctions that we should observe.

What Is Penetration Testing?

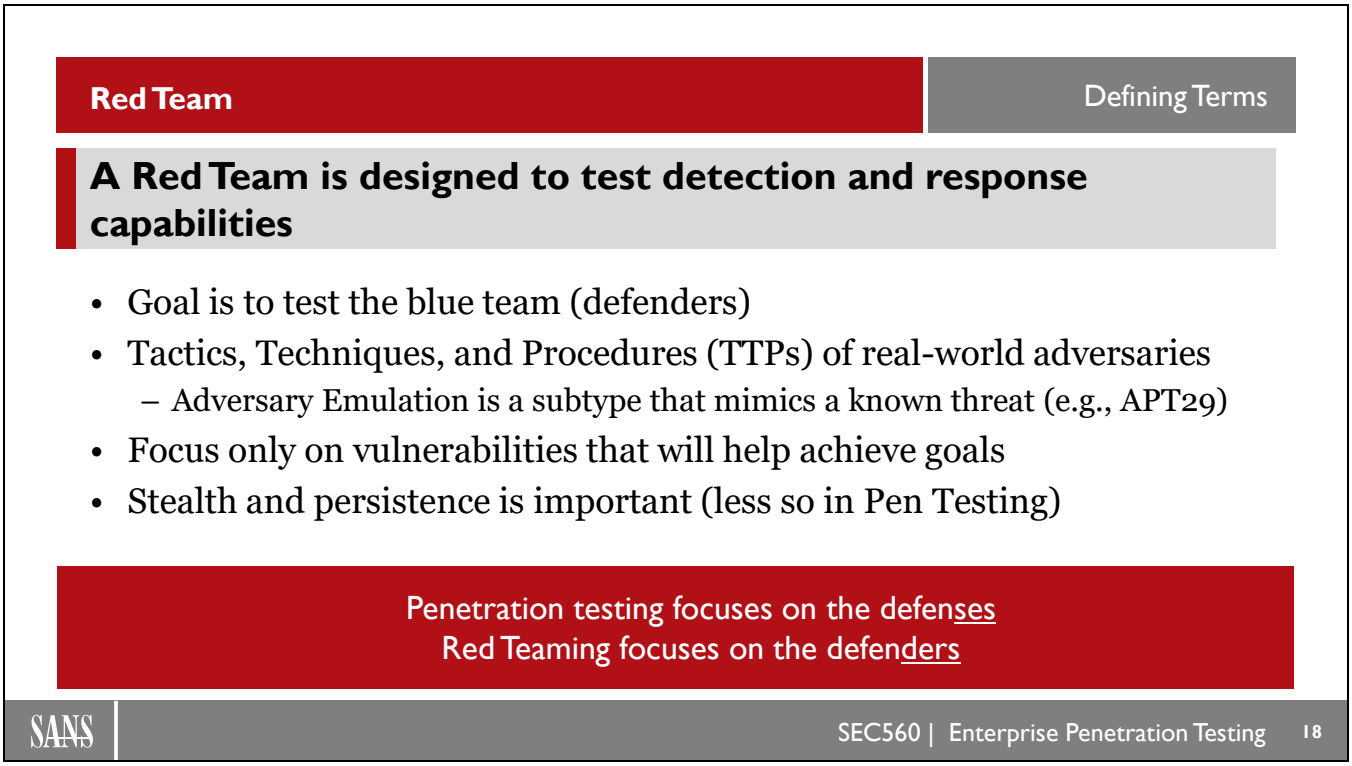
Defining Terms

Identify security vulnerabilities that could let an attacker either penetrate the network or computer systems or steal information

- Use tools and techniques similar to those used by criminals
- To prevent a thief, you need to think like a thief
- The goal is actual penetration
  - Compromising target systems and getting access to information to determine the business impact

A penetration test allows for penetration, or it isn't a penetration test!

Penetration testing focuses on identifying vulnerabilities in a target that could allow an attacker to penetrate the computer or network and steal, damage, or corrupt data. Penetration testers (pen testers) will use tools similar to the malicious attackers. Sometimes, the malicious actors even take tools and techniques from penetration testers. As the name implies, penetration testing requires penetration (if possible). A test that forbids penetration is not a penetration test. Sometimes targets ask for a penetration test, but then disallow exploitation, password guessing, and other activities that would allow the testers to verify flaws and find second order vulnerabilities. These handcuffed tests are more akin to vulnerability assessments.

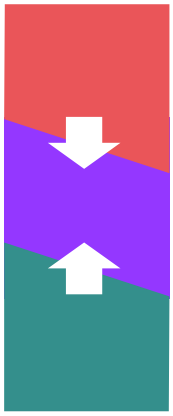


Red Teams are focused on testing the effectiveness of the entire security program. They emulate the Tactics, Techniques, and Procedures (TTPs) of real-world adversaries with the goal of measuring weaknesses in defense, detection, and response. Red Teams use many of the same tools as penetration testers; however, they have a different focus. The Red Team is focused on identifying deficiencies in the Blue Team's (defenders) ability to detect, respond, and eradicate a threat. A Red Team only uses vulnerabilities that help them accomplish their goal, while penetration testers are most often tasked with finding many vulnerabilities.

Penetration testing is focused on finding flaws, understanding their business risks, and helping the organization to improve its security stance. The Red Team's primary job is to help make the Blue Team better equipped to detect and respond to attacks. To put it simply, penetration testing focuses on the defenses (technology) and Red Team focuses on the defenders (people).

Cross-functional team consisting of Red and Blue Teamers

- Slow, intentional steps by Red and then measurement of Blue's ability to prevent, detect, and respond
- Summarized as "ACE"
  - Automation: How automated is the detection and notification workflow?
  - Coverage: Scope of the environment in which this will detect
  - Effectiveness: How effective is the detection? High false positive rate? Low detection rate? What is the signal/noise ratio?
- Includes metrics to find gaps and improve defenses
- Additional reading: redsiege.com/purple



Purple Teams are cross-functional teams where the Red and Blue Teams work together. The closer the interaction, the better. This type of assessment includes metrics on Blue's ability to identify, detect, and respond to an attack. The key components can be summarized with "ACE":

- Automation: How automated and timely is the detection and notification workflow?
- Coverage: Across how much of the environment will this detection work? Is it limited to specific systems or subnets?
- Effectiveness: How effective is the detection? Is there a high false positive rate? Does the detection only work in specific circumstances?

The testing should be documented to show advances in detection and response capabilities. These metrics can be used to justify investments in tools and training.

Additional reading on the Purple Team: redsiege.com/purple

Audit implies testing against a rigorous set of standards

- Almost always done with detailed checklists
- Though checklists are created for penetration testing and security assessments, they tend not to have the depth and rigor of an audit
- The focus in this class is not on audits
  - The concepts and techniques we cover will be helpful for auditors

Finally, we have the phrase security audit. An audit implies that we are measuring things against a fixed, predetermined, rigorous set of standards. These audits are almost always done with detailed checklists.

Some penetration testing and ethical hacking organizations have created their own internal checklists of items that need to be covered in a test, but these checklists aren't as detailed as those for comprehensive audits.

Our focus in this class is not on auditing. SANS has numerous other classes that address security audits in detail. Our focus is on ethical hacking and penetration testing.

Vulnerability Assessments

Defining Terms

Identify, quantify, and rank vulnerabilities (no exploitation)

	Vulnerability Assessment	Penetration Test	Red Team
Focus	All vulnerabilities	All exploitable vulnerabilities	Only vulns to get to goal
Depth	Surface issues (no 2nd tier issues)	Multiple layers (pivoting)	Multiple layers (pivoting)
Risk	Estimate Risk	Demonstratable Risk	Demonstratable Risk
Stealth	None	Varies (usually minimal)	High

Penetration Testing requires exploitation and pivoting
If an assessment does not include exploitation, it is not a penetration test!

Many people use the phrases vulnerability assessments and security assessments to describe the work done by penetration testers and ethical hackers, but there is a subtle distinction between a penetration test and a security assessment.

A penetration test is focused on getting in or stealing data. The emphasis is on penetrating the target environment by exploiting discovered vulnerabilities.

Vulnerability assessments and security assessments are focused on finding vulnerabilities, often without regard to actually exploiting them and getting in.

Thus, penetration testing often goes deeper, with its goal of taking over systems and stealing data, whereas security and vulnerability assessments are broader, involving the process of looking for security flaws. These assessments also often include policy and procedure reviews, which are usually not included in penetration testing.

The table above shows the differences between the different types of assessments. Of course, each can vary to some degree with penetration testing having the greatest variability. A penetration tester is often tasked with finding all vulnerabilities, even those that aren't exploitable.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

There is a large number of different types of ethical hacking and penetration tests. Let's now explore the different types, realizing that many of the tests we'll engage in for our jobs will be a mixture of subsets of these various types.

Penetration Testing is demonstrating risk by offering real-world proof

- **Model** the techniques used by real-world attackers
- **Find** vulnerabilities (before the adversary does)
- **Exploit** those flaws under controlled circumstances
  - In a professional, safe manner according to a carefully designed scope and Rules of Engagement
- To determine **organizational risk and potential impact**,
- Help the organization **improve security practices**

A formal definition of penetration testing is as follows:

Penetration testing involves modeling the techniques used by real-world computer attackers to find vulnerabilities, and, under controlled circumstances, to exploit those flaws in a professional, safe manner according to a carefully designed scope and Rules of Engagement to determine business risk and potential impact, all with the goal of helping the organization improve security practices.

The key is that the vulnerability identification and exploitation should always be business focused. The penetration test findings should always focus on the organizational risk, not just the technical risk.

Many organizations use ethical hacking and penetration testing to find security flaws before the bad guys do. After applying their security policies, procedures, and technology, organizations can use thorough penetration tests to see how effective their security is in light of an actual attack, albeit by friendly attackers.

An added benefit of ethical hacking and penetration testing is that because they show real vulnerabilities and indicate what a malicious attacker might be capable of achieving, they can get management's attention. Decision makers, when presented with the carefully formulated results of a test in business terms, are more likely to provide resources and attention to improve the security stance of an organization.

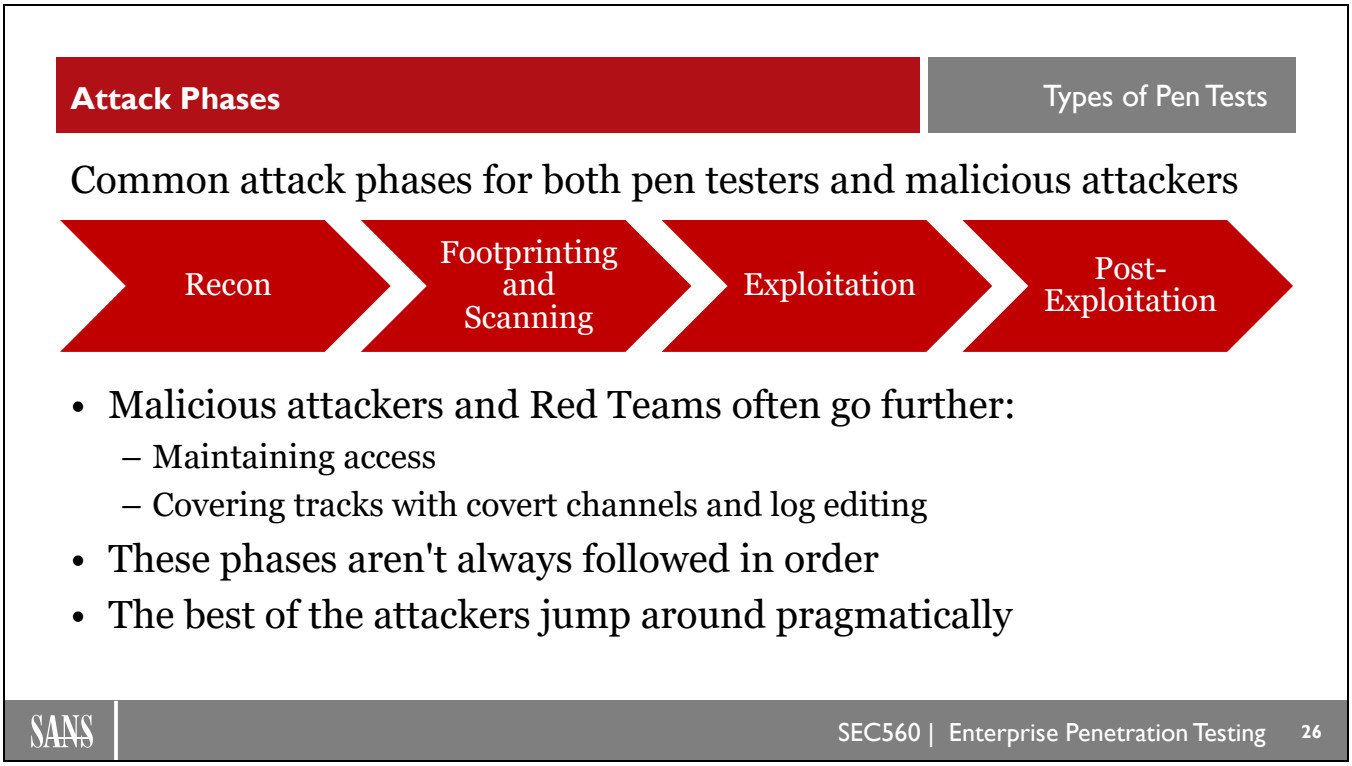
Types of Penetration Tests

- Network services test
  - One of the most common
- Assumed breach test
- Web application test
- Social engineering test
  - Email-based or phone-based
- Wireless security test
  - Not just Wi-Fi
- Physical security test
- Product test
  - Could be software package or hardware (e.g., IOT)
  - Breaking or bypassing encryption on local data or intercepted traffic

There are numerous kinds of penetration tests:

- **Network services test:** This is one of the most common types of tests and involves finding target systems on the network, looking for openings in their underlying operating systems and available network services, and then exploiting them remotely. Some of these network service tests happen remotely across the internet, targeting the organization's perimeter networks. Others are launched locally from the target's own facilities to evaluate the security of the internal network or the DMZ from within, seeing what kinds of vulnerabilities an internal user could discover.
- **Assumed breach test:** This kind of test is designed to find vulnerabilities in the network once an attacker has gained access to a system in the network. This is a fantastic way to find issues in Active Directory permissions, file permissions (excessive sharing on file-shares), and client-side software.
- **Web application test:** These tests look for security vulnerabilities in the web-based applications deployed in the target environment.
- **Social engineering test:** This type of test involves attempting to dupe a user into revealing sensitive information, such as a password, or possibly convincing a user to click a link in an email. These tests are often conducted via email or over the phone, targeting selected help desks or users and evaluating processes, procedures, and user awareness.
- **Wireless security test:** These tests involve exploring a target's physical environment to find unauthorized wireless access points or authorized wireless access points with security weaknesses.
- **Physical security test:** This test looks for flaws in the physical security practices of a target organization. Testers might attempt to gain access to buildings and rooms or to take laptops, desktops, or recycling bins out of target facilities. A dumpster diving test is a variation of a physical security analysis. Physical testing must be conducted carefully to ensure that the testers do not get hurt or arrested during their work.

- **Product security test:** This test focuses on one specific product. It could be a hardware or software. In this kind of test, you look for security flaws in products that can be used in the tester's laboratory systems. Such tests look for flaws in the software, such as exploitable buffer overflow conditions, privilege escalation flaws, and the exposure of unencrypted sensitive data. These tests could focus on bypassing or breaking the encryption of data stored on a local system or across the network. Some of these tests also evaluate the strength of digital rights management (DRM) solutions. Due to legal restrictions regarding reverse engineering copyright protections (such as those imposed by the Digital Millennium Copyright Act in the United States), any contract regarding the analysis of DRM software should be inspected by a lawyer to ensure that the proper permission has been derived from the owners of the given DRM solution.



Both malicious attackers and professional penetration testers/ethical hackers apply various phases in their attacks. Attacks are often separated into these phases:

- *Reconnaissance* is the process of investigating the target organization to gather information about it from publicly available sources, such as domain registration services, websites, and so on. Some people include techniques such as social engineering and dumpster diving in the recon phase.
- *Scanning* is the process of finding openings in the target organization, such as internet gateways, available systems, listening ports, and vulnerability lists.
- In the *Exploitation* phase, attackers exploit target systems to compromise them, possibly getting control of them, or causing a denial-of-service attack.
- *Post-exploitation* is what happens after the initial compromise. Both penetration testers and malicious attackers use their access to pivot and move throughout the target environment.

Although legitimate tests often include the previously listed phases, malicious attackers often go further than the Rules of Engagement allow for a professional penetration test. The next phase, often used by a malicious attacker to maintain access to and control of a target machine, involves setting up the compromised machine so that the attacker can keep control over it, with techniques such as installing backdoors and planting rootkits. Malicious attackers also often use a final phase, Covering the Tracks, in which they employ log editing, file hiding, and covert channels to hide their activities on a system.

Please note that the best of the attackers (both the good guys, and the evil ones) are pragmatists. They don't always proceed from reconnaissance to scanning to gaining access and so on. Sure, they use these steps, but they are likely to jump around among them as events and discoveries warrant. For example, during the recon phase, attackers may discover an exploitable flaw that they will use to gain access directly, temporarily bypassing scanning. Then, after they gain access to one machine, they may go back and start scanning.

From a professional testing perspective, though, be careful when jumping out of order among these steps, making sure that you return to the earlier phases to conduct a comprehensive test.

Course Roadmap

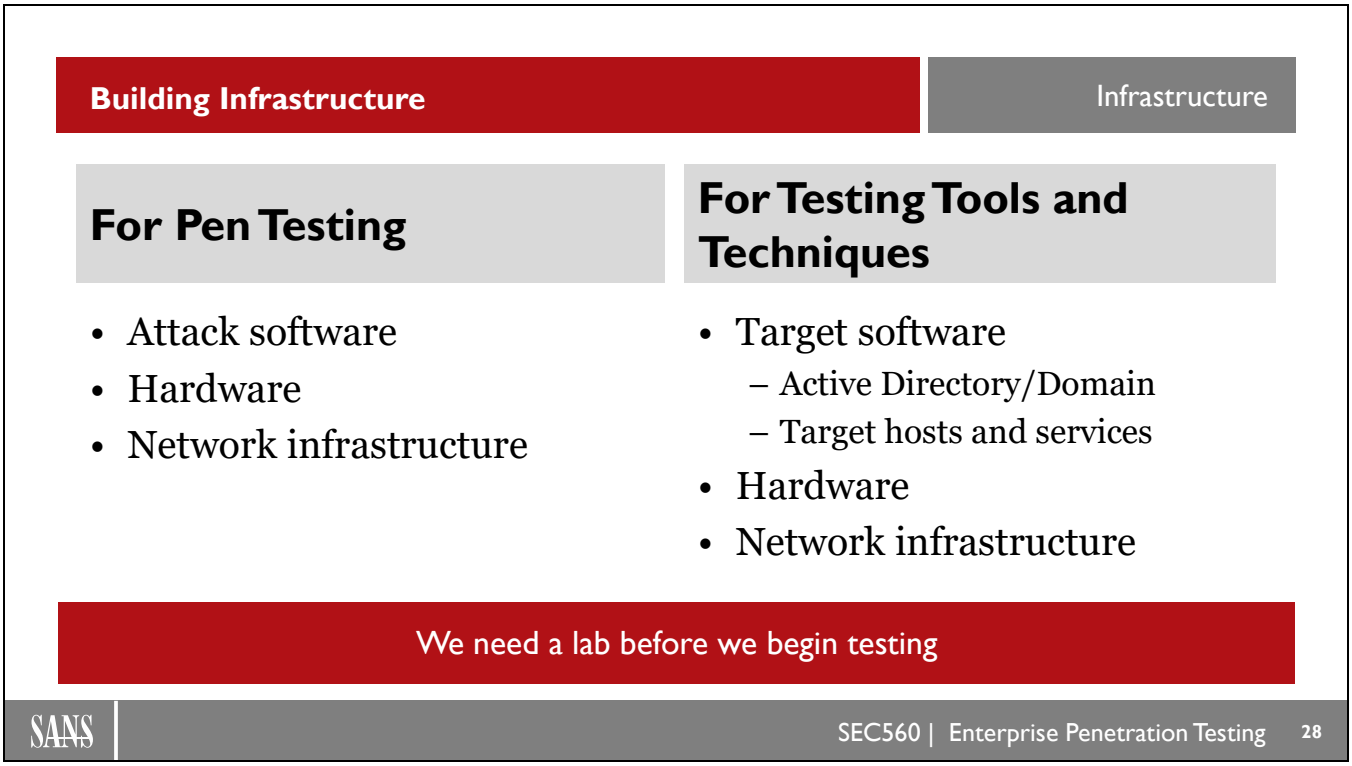
- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

A well-stocked lab and an arsenal of testing tools are crucial to the success of an ethical hacker and penetration tester. Let's now discuss the hardware, software, and network connectivity used by testers in their work.

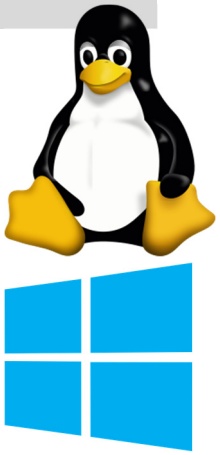
Keep in mind that these infrastructure items we discuss are not a one-size-fits-all proposition. Instead, we cover the areas of tools that you need, with some notable examples. Then based on your budget, expertise, and test types, you can construct an appropriate arsenal to match your test regimen.



Before we begin pen testing, we need systems from which we can test and systems which we can use as a test lab, such as for testing new tools and techniques. These two pieces will likely be separate, but we need the systems, nonetheless. Your needs may differ from what is presented here. This is designed to be a baseline infrastructure, which you can expand or modify based on your needs.

Windows and Linux are simply tools

- Many attack tools work better/only on Linux
- Domain Authentication is easier on Windows
- Switch between them as needed (VMs)
- MacOS is acceptable, but use VMs
  - MacOS is required for some iOS mobile testing



A common question among penetration testers and ethical hackers is "Should I focus my skills and toolbox on Linux or Windows?" When confronted with this question, we recommend that your pen test toolset include both operating systems side by side, working together to maximize your efficiency and capabilities. The truth is that some tools work better on Linux, whereas others work better on Windows. Some tools work just fine on both, whereas other tools have been released for only one of those platforms. Thus, if you choose to work in only one OS or at least just focus on that OS, you'll be missing out on a lot of useful tools and techniques. To improve productivity and streamline workflow, we recommend virtualizing one of these two OSs, perhaps using VMware, and running the two simultaneously on the same hardware so that you can quickly switch between them.

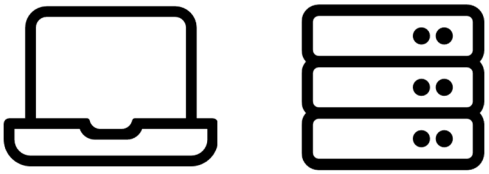
The entire question posed at the start of the preceding paragraph illustrates a mindset that should be transcended. Don't think of them as two different operating systems. Think of them as one set of tools that you use in your penetration testing and ethical hacking job. As a carpenter or plumber would use the best tool that is available and convenient for a given job, so should you. To continue with that analogy, don't think of Windows and Linux as two different toolboxes. Instead, they are two different compartments in your single toolbox.

Some of you are no doubt wondering whether macOS is an acceptable platform for penetration testing and ethical hacking. It is—with remarkable stability and ease of use. However, there are some tools for Linux and Windows that simply will not run on macOS, no matter how hard you try to get them installed. Thus, if you plan to use macOS, make sure you get a virtualization solution for it (such as VMware Fusion or Parallels) so that you can also run both Windows and Linux on top of macOS.

Testing System / Attack System

Testing machines: Systems used by the penetration tester or ethical hacker to evaluate the security of other machines. We will also call them *attack machines*.

Target ,Victim, User, and Server



Target machines: Systems whose security stance is being evaluated. We will also call them *victim machines*.

For the remainder of this class, we need to carefully differentiate between the machines used by the penetration tester or ethical hacker and the machines whose security is being evaluated.

We use the terminology testing machines and attack machines to refer to the systems that the tester uses to evaluate the security of other systems. These testing machines often run the attacker's scanning and exploitation tools. In figures, these machines will be represented with a red screen and with the attacker inside. We use these pictorial clues to help you rapidly identify where the attacker's machines are in a diagram. However, please do not think that the black hat on this computer implies that the attacker is somehow evil. The attackers we refer to here are professional ethical hackers and penetration testers. The black hat just makes this system easier to quickly locate in the figure.

The machines whose security is being evaluated are referred to as target machines and, occasionally, as victim machines. They are represented pictorially as a standard machine, with no hat.

Test new tools and techniques in your lab

- If issues occur in your lab, no problem
- If issues occur in the live environment, that is messy
- Red Teams use their lab often to make sure commands work as expected and reduce the number of artifacts in the target environment
- Use VMs: Easily revert, store, clone, and build systems

See Jeff McJunkin's Webcast "Building Your Own Super Duper Home Lab"
YouTube: youtube.com/watch?v=KogdkuEbfwc Short link: redsiege.com/560/lab

As we've seen, your tests might rely on free, commercial, and/or in-house tools, depending on the policies of your organization. Whichever tools you use, you should test them in a laboratory environment to make sure you understand how they work and their potential impacts on a target machine. Such laboratory testing and analysis is especially vital for free tools downloaded from the internet because of concerns about quality, the potential to crash a target, and hidden functionality that could compromise the test systems.

It is important to have a good lab that consists of many of the same systems and software you will see in the target environment. Obviously, the lab can't contain all possible combinations or all software, but a representative sampling is good for my purposes. We'll discuss this more shortly.

Often, Red Teams will run their commands in their lab before the commands are used in the target environment. This ensures the Red Team has the correct syntax and outcomes so as to reduce the number of artifacts on the target. If the Red Team executes the wrong command or has a typo, the Red Team will need to run another command, thereby leaving additional artifacts and increasing the likelihood of being caught.

Virtual machines are a great way for offensive personnel to quickly build, test, revert, and clone test systems. The VMs can be run on old hardware, or even the testers laptop. Jeff McJunkin has a webcast on "Building Your Own Super Duper Home Lab" located at:

youtube.com/watch?v=uzqwoufhwyk
Short link: redsiege.com/560/lab

Useful Lab Systems

Infrastructure

Servers	Windows Domain (and Domain Controller) Windows File Share and IIS (Web) Server Linux – A few different distributions (e.g., Ubuntu, Fedora)
End User	Windows 10, 11

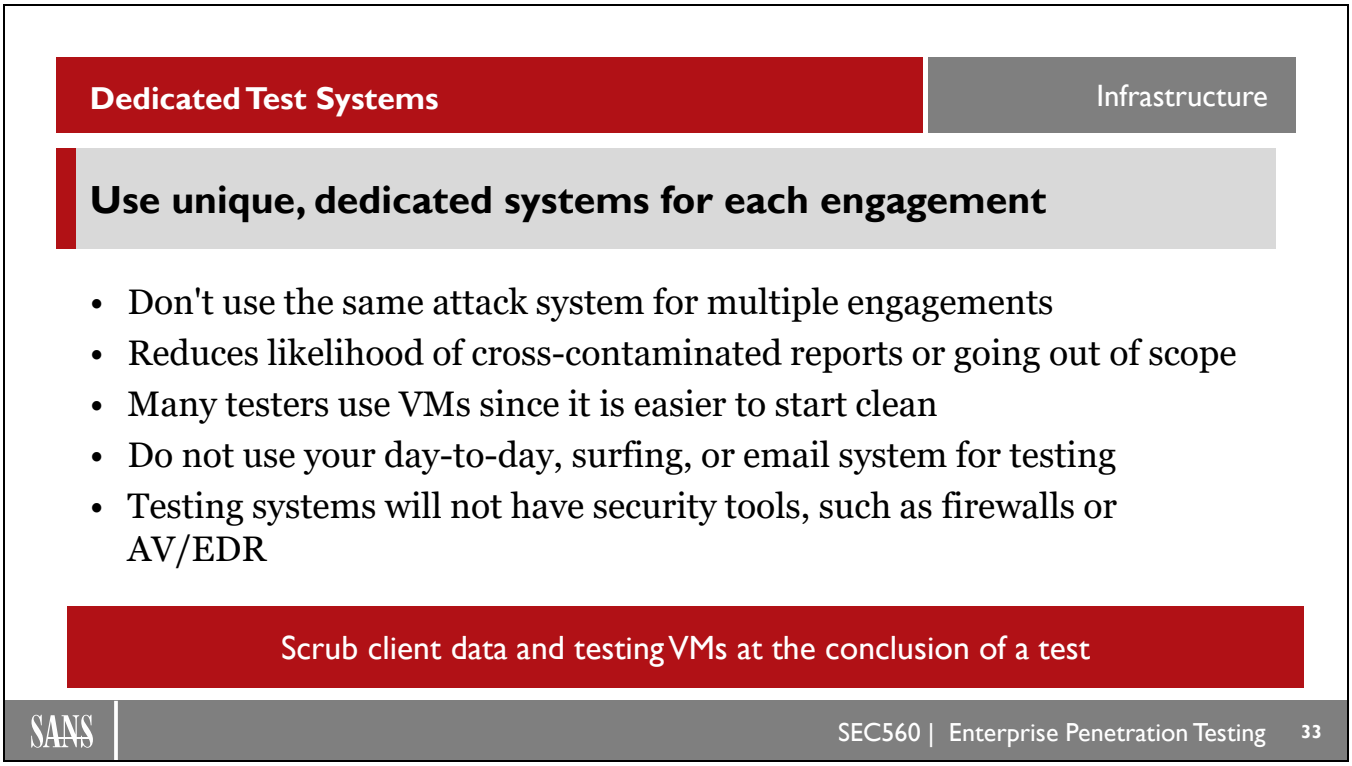
Linux	Free (usually) RHEL requires a paid subscription (Fedora is similar and free)
Windows	Buy licenses Use a free trial – limited versions available and they expire

As previously mentioned, it is important that we have systems that resemble a wide range of target networks. Most networks use a Windows Domain and, therefore, a Domain Controller (we'll discuss attacking the Windows Domain later in this course). Similarly, most of these networks include Windows file servers and web servers. You could combine these two roles into one system to save on physical resources and licensing. Of course, the most widely used end-user system in the enterprise is Windows, so we'll need one of those systems in our lab.

Unfortunately, using Windows in a lab is more difficult due to licensing. Of course, you can pay for licenses and can have as many different targets as you like, including older server and end-user systems. Many people have much tighter budgetary constraints that limit them to free systems only. We do not recommend or condone stealing or otherwise bypassing the legal licensing requirements set by Microsoft (or other vendors). Microsoft does offer free virtual machines for testing; however, it is limited to the latest operating system, and the licensing window is limited.

On the Linux side, we can use a range of free Linux systems. For example, Ubuntu is a common free Linux distribution found in enterprises. While most of the installs and updates for Linux are free, some are not. For example, Red Had updates are not free, but it is very similar to CentOS and Fedora. For more details on this relationship between Fedora, Red Hat, and CentOS, you can read more here:

> https://danielmiessler.com/study/fedora_redhat_centos/
> Short Link: redsiege.com/560/centos



Next, you need the actual systems that will be doing the testing. We recommend that you use systems that are dedicated to testing and are unique for each engagement.

- **Disabled defenses:** The attack systems will not have defensive software such as AV/EDR, and it will often have its firewall disabled. These modifications decrease the defensive strength of the system should it come under attack from an adversary. You don't want these features disabled on your day-to-day system.
- **Uptime:** The testing systems will often need to be available and online for weeks or months. If you were to run the attack from your laptop (or even a VM on your laptop), the test could be impacted if you carry your laptop to another location (network change) or perform your day-to-day tasks (installing software, rebooting, patching).
- **Prevent data leakage:** If you use unique systems for each test, it is less likely that the data from one test will end up in the data from another test. This is especially important for third-party testers where the test data is likely to be from different organizations where such an action could be considered a breach of confidentiality. Such a cross-contamination could lead to embarrassment, loss of trust, and even legal action.

At the conclusion of a test, be sure to clean the target data from the test systems. You can't lose data you don't have.

Systems Used for Internal Testing

Infrastructure

Use a "leave behind", "drop box", or "drop" for long running tasks and remote access

- Physical: Laptop or Minicomputer
  - Use FDE or scrub unencrypted data before return shipping
  - Note: Internal policies often allow corporate-owned devices on the network but require extra paperwork for third-party-owned devices
  - Third-party testers will sometimes expense or sell the physical device to avoid return shipping and thereby make it easier to get on the network
- Virtual: Custom VM
  - Target is responsible for destruction
  - Requires internal team to provision, setup, and configure

If you are physically on the same network as the target systems, use Bridged network and not NAT!

Next, we need the actual systems that will be used for the testing. The needs for internal testing and external testing are different.

For internal testing, we can use a "leave behind" or "drop box" for long-running tasks. We can also use this for remote-internal or pseudo-internal testing, which allows the penetration testers to work remotely but have internal access to their desired testing systems. The drop box could be physical, such as a laptop or a minicomputer. If you use full disk encryption (FDE) on the system, the target will then have to unlock the disc when initially booted or on reboot. Alternatively, the host OS could be booted without a key and then the sensitive client data could be stored in an encrypted container using a key known to the pen testers. If there is any unencrypted data on the system, ensure it is properly scrubbed before shipping. You never want to ship unencrypted sensitive target data through the mail or a public shipping company. If that system is lost, it could be considered a reportable breach.

To get around the shipping issue, some third-party testers will sell or expense the drop box to the client as part of the contract. This also solves another common issue: internal access. Some organizations have strict policies as to what systems are allowed on the network. Corporate assets are allowed on the network, but third-party owned assets require extra verification, documentation, and paperwork. If the organization owns the physical drop box device, then it is often easier to get the device on the network.

You could create a custom VM configured to connect back to the pen testers and no shipping is ever required. However, this often requires extra action from the internal team to provision, setup, and configure the VM.

Harden Testing Systems Carefully

Infrastructure

- Make sure you harden the testing machines (to a point)
  - Don't be the breach! Don't leave your self open to attacks
  - Disable or filter unneeded services
- Harden too much and you can break your tools
  - The latest version of OpenSSL doesn't support SSLv2 or 3, so tools that use the latest OpenSSL libraries can't test for these SSL vulnerabilities

The Center for Internet Security has free templates for hardening Windows and Linux (as well as other OSes and environments) https://www.cisecurity.org/cis-benchmarks/

Although we recommend that you avoid using a network or personal firewall to protect your testing machines because of the potential impact on your test results, we do caution you. Make sure you carefully harden the testing machines before starting a test. You must guard against compromise of the attacking machines during a test by either a third-party malicious attacker or even an overexuberant system administrator in the target organization. There have been cases of an administrator on a target network launching a counterstrike during a test, hacking back to the penetration tester's machines and compromising them. If someone compromises your testing machines, they could steal your interim test results and even alter the results they leave behind. Such exposure of test results could be, at best, embarrassing, and at worst, catastrophic for your career as a penetration tester or ethical hacker.

For this reason, keep patches up to date on all your testing machines, and shut off unneeded services. For a penetration testing system, you likely need no or only minimal listening services on the machine. The only services that should be listening are specific ones you need for your test, such as a web server set up to deliver a client-side exploit or a file server needed to serve up files to compromised target machines. You want to increase security settings of the testing machines beyond the defaults for the given operating system, but make sure that you don't inhibit the functionality of your testing tools. Harden the boxes but verify on lab systems that your hardening process doesn't break needed functionality of your test tools.

For example, if you are testing from a fully updated and patched Linux system will not be able to interact with systems running SSL 3.0. As of OpenSSL 1.1.0 (released September 2019), SSL 3.0 is disabled due to multiple vulnerabilities in the encryption. Not being able to interact with these services means you will be unable to identify and exploit vulnerabilities on these systems. Similar scenarios exist on Windows, such as NTLMv1 authentication and the SMBv1 protocol, both of which should be disabled on secure systems. However, we need these insecure features so that we can test other systems for their existence, and interact with those systems, with the goal of identifying vulnerabilities.

To help with this hardening process, the Center for Internet Security (www.cisecurity.org) has a large number of free templates for hardening various kinds of systems, including Windows and Linux. Download and use these templates: https://www.cisecurity.org/cis-benchmarks/

Systems Used for External Testing

Infrastructure

- Cloud infrastructure (more common)
  - Pros: Easier to build and destroy
  - Cons: Advanced defenders carefully monitor these addresses
- Self-hosted (less common)
  - Pros: Not associated with a cloud provider
  - Cons: Clean up can be difficult and IP addresses can be branded with a bad reputation

Next, we need the actual systems that will be used for the testing. The needs for internal testing and external testing are different.

Internal Testing

For internal testing, we can use a "leave behind" or "drop box" for long-running tasks. We can all use this to allow remote-internal testing, which allows the penetration testers to work remotely but have internal access to their desired testing systems. The drop box could be physical, such as a laptop or a minicomputer. If it is a physical device, make sure you encrypt or wipe data from the device before it is returned from the target. Alternatively, you could create a custom VM configured to connect back to the penetration testers.

External Testing

For external testing, we have two options: cloud infrastructure or self-hosted. Cloud infrastructure is much more common due to the ease of creating and destroying the test systems. Cloud has the additional benefit of allowing the penetration testers to use setup testing infrastructure all over the world and change IP addresses/subnets.

Alternatively, you could self-host the testing infrastructure, either in your datacenter or a co-location facility. If you host your own systems, it can be more difficult to perform a full cleanup of the testing system. The benefit of self-hosting is that the IP address is not associated with any cloud service provider. Some advanced defenders perform additional monitoring on traffic to cloud providers due to its heavy usage by attackers for command and control.

Sources for Free Tools and Exploits

Infrastructure

- **Exploit-DB**: exploit-db.com
  - Sorted by remote, local, web app, denial of service, shellcode, and papers
- **SEEBUG Vulnerability Database**: seebug.org
  - Hundreds of categories, split by OS and product
- **Packet Storm Security**: packetstormsecurity.com
  - Vast history of attack and defense tools
- **US-CERT:** www.us-cert.gov/ncas/alerts
  - Latest information about vulnerabilities
- **MITRE CVE Repository:** cve.mitre.org
  - Latest information about vulnerabilities

We are not endorsing these sites or the tools they distribute.
Remember to be careful!

Although there are numerous exploit and attack tool repositories on the internet, some of the most comprehensive archives that are updated on a regular basis include the Exploit Database and Packet Storm Security. Several other sites come and go on a regular basis, but these sites are long-standing and tend to have relatively higher quality tools.

The Exploit Database (exploit-db for short) is maintained by the same group that maintains Kali Linux, Offensive Security. Its site hosts more than 10,000 exploits and sorts them into useful categories, such as Remote Exploits, Local Exploits, Web Applications, Denial of Service/Proof of Concept, Shellcode, and Papers. For each exploit in these categories, it lists the platform (Windows, Linux, PHP, and so on) and the author.

Also, the SEEBUG site has hundreds of categories of vulnerabilities, including exploit code for many different issues that they inventory.

Packet Storm Security has an archive of attack and defense tools that spans over a decade. It's quite an impressive assortment of useful tools, exploits, and security research papers.

Note that we are not endorsing these sites or the tools that they distribute. These sites have been quite controversial, and you need to be careful with any code you download from them. Still, ethical hackers and penetration testers need to know about these sites to do their jobs.

Beyond the tool and exploit sites, numerous vulnerability research sites are also available. Although these sites do not distribute exploit code freely, they do publish information about vulnerabilities. These detailed vulnerability descriptions are invaluable in letting a tester know that there is an issue with a system type or service version discovered in a test. Even though an exploit might not be available (in fact, an exploit may have never been publicly released or even created), the tester still needs to understand the vulnerabilities so that they can be included in the test report.

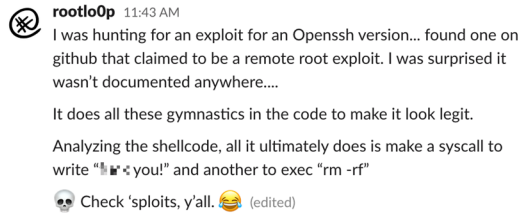
Some of the best sites with vulnerability research and detailed descriptions are the following sites:

- The United States Computer Emergency Readiness Team (US-CERT), maintained by the US Department of Homeland Security (DHS)
- The Common Vulnerabilities and Exposures (CVE) repository operated by Miter

Free Software Tools

Infrastructure

- Most testers use at least some free tools in their testing
- Determine your organization's policy for using free tools
- Be careful! Backdoors are possible!
  - Test the tools in a lab against a sample target first
  - Analyze the code of the tool or exploit if possible
  - Sniff the traffic to see if it sends extra packets to unanticipated destinations
  - Look at the file system changes on both the attacker and the target
  - Microsoft Sysinternals' Procmon and Sysmon are helpful



In addition to the SANS Slingshot image and other free bootable Linux environments, a variety of websites offer vast arsenals of free tools and exploits, which can be incredibly helpful. The vast majority of professional penetration testers and ethical hackers rely on at least some of these free tools when doing their jobs. Before considering whether you can run such tools in your environment, you need to determine your organization's policy regarding the use of such third-party security assessment and exploitation tools. Some organizations strictly forbid the running of any tools beyond a standard baseline of already-approved tools. Others allow additional tools to be used, but only if they are carefully vetted.

Consider this scenario: A tester scans a target environment, discovering a listening service that has a version number that is known to be exploitable. With a little research, the tester discovers a freely downloadable exploit for that specific version of the service from an exploit distribution site. Suppose further that the Rules of Engagement for the test allow the actual exploitation of the target machine, and furthermore, the tester's own organization allows for the use of third-party free exploits. What should the tester do?

We strongly urge you to be careful with free downloaded tools and exploit code. Historically, some of the tools and exploit code freely distributed on the internet have included backdoors that let an adversary control any system on which the tool was run or even control the target machine against which the tool was run. Also, some of the tools may cause a crash in a target service or system.

Thus, we recommend that testers analyze all free tools carefully in a test lab before using them in a test. If you have the skills, review the source code for the tool before using it, making sure it does exactly what it says it does, with no hidden backdoor functionality or other trojan horse capabilities in the tool. If you cannot review the code, then at a minimum, run the free tool in a laboratory environment, carefully reviewing the traffic it sends across the network (looking for unexpected packets going to unexpected destinations) and any changes it makes in the file system of the attack and target machines. The free Microsoft Sysinternals' Procmon and Sysmon tools are helpful in analyzing file system and Registry interactions. Procmon has subsumed the earlier tools, filemon and regmon, extending their functionality in a single tool.

Commercial Tools

Infrastructure

- Commercial tools may be expensive, but often give you:
  - Higher quality (not always)
  - More frequent updates
  - Support
- Vulnerability Scanners: Nessus, Nexpose, Qualys
- Exploitation Toolkits: Metasploit Pro, Core Impact, Canvas
- C2 Frameworks: Cobalt Strike, Red Team Toolkit

In addition to the free tools we've been discussing, some penetration testers and ethical hackers rely on commercial tools for testing. There is a large number of commercial tools, with new ones released on a regular basis. The advantages of commercial tools include generally higher quality (but not always), typically more frequent updates (given the vendors' paid teams of software developers), and technical support if issues arise during testing.

Although this course is taught from a vendor-neutral perspective, professional penetration testers and ethical hackers do need to know about some of the commercially available tools, even if they don't use them. That way, they can make sure that their test regimen made up of noncommercial (free or in-house) tools includes similar concepts and capabilities as the commercial tools. This slide lists a few of the more popular and comprehensive tools for testing.

Some of the commercial tools commonly used by penetration testers are:

- **Vulnerability Scanners:** Nessus, Nexpose, Qualys
- **Exploitation Toolkits:** Metasploit Pro, Core Impact, Canvas
- **C2 Frameworks:** Cobalt Strike, Red Team Toolkit

Tools for Penetration Testing Teams

Infrastructure

- Knowledge base
  - Must be easy to search and update or it won't be used
  - Share tools and techniques for specific attacks
  - Internal teams can share remediations and solutions
  - Tools: OneNote, Confluence, Wiki
- Storage
  - In-house tools, code, artifacts found in target environment
  - Tools: File share, SharePoint, cloud storage, Git
- Central code repository (Git, CVS, SVN, Mecurial, TFS/VSTS)
- Chat (secured)
- Secure credential sharing
  - You need a secure way to share credentials with your team
  - Tools: 1Password (includes 2FA), LastPass, KeePass
- Common finding repository
  - Don't rewrite the same finding every time, tweak it for the specific case
  - Tools: Excel, Word, DB, Git
- Collaborative report writing
  - Tools: MS O365, Google Docs, Git
- Pen Test specific (less common)
  - Tools: Dradis

Most penetration testers work as part of a team. Even if you are working by yourself, it is important to take good notes so you can save yourself time on future engagements. We recommend you have a knowledge base containing tools, techniques, and command examples to make your attacks more efficient. Also, this is a great way to share with your team, especially new or junior members. The tools are as important, but some of the common tools used are Confluence, a wiki, or OneNote. The exact tools isn't as important as the process. This must be easy to use, search, and update or your team won't use it.

You will need storage for data specific to the engagement. You will often find keys, certificates, and other artifacts that are useful in the penetration test that you need to (securely) share with other team members. This could be an internal file share, SharePoint, cloud storage, or a Git repository. This needs to be easily accessible and updateable by the team members. Remember to clean up this data when the test is complete!

Your team will often have in-house code that needs to be shared with team members. Most teams these days use Git. Other common version control software includes CVS, SVN, Mercurial, and the Microsoft Team Foundation Server (TFS) or Visual Studio Team Services (VSTS).

Your team needs to communicate, so a secure chat mechanism is important. Signal and Telegram offer great security, but the scalability for an enterprise ready tool is lacking. Microsoft Teams, while somewhat behind the curve in the world of chat, offers retention policies to make cleanup automatic. Similarly, Slack offers a retention policy at their paid tier. Many teams using internal services, such as IRC, allow the pen testers to have more control over the data.

Clients will give you credentials, you will likely acquire new credentials (cracking, guessing, or even cleartext), and you possibly will even create accounts in the target network. It is important to securely share that information with your teammates. There are a number of password vaulting tools that can be useful here, including 1Password, LastPass, and KeyPass. One nice benefit of 1Password is that it also allows you to store some 2FA tokens.

Finally, we need ways to write reports in a consistent manner. It is a timesaver to have a repository for common vulnerabilities and findings. These findings are pre-vetted prior to going into the report. This allows you to have a good quality base for the finding and you can tweak the finding for the particular scenario or target. This is a

tremendous time saver. The findings in the repository should not contain any target information, making it much safer to copy into a report. Be very careful never to copy data between reports as this can lead to data leakage. The authors of this course have heard of multiple instances were one target's name or details ended up in a report for a different target. This can be embarrassing and undermine your credibility.

To make report writing easier for multiple people, a collaborative reporting environment is a nice time savings. Ideally, this allows multiple people to work on the report (and even the same portion of the report) at the same time. Services like Google Docs and SharePoint with Microsoft Office allow for this kind of shared editing. Some penetration testers use version control software (e.g., Git, SVN) to manage reporting, but in the author's experience, this can lead to a number of code/edit conflicts that take extra time to resolve.

Some penetration testers use collaborative tools specifically designed for penetration testing. The two most common ones are Dradis and Lair.

The Dradis tool is a Ruby-on-Rails project designed for recording information among multiple penetration testers working on one or more projects together. The Dradis server runs on Windows, Linux, or macOS and features multiple client options: a command line client, several different thick client applications, or a web-based interface.

All results are organized as a hierarchical tree, typically organized starting by overall project, then split according to functional areas of the target infrastructure (for example, DMZ/intranet/extranet, servers/network devices/clients, or other applicable divisions of the test's scope), then separated by individual devices, down to individual ports on those devices, and then through findings and notes associated with each port.

With Dradis, a tester can import results from the Nmap port scanning tool, the Nessus or Qualys network vulnerability scanning tools, the Nikto web server scanning tool, or the Burp web application attack tool. In addition, the tester can manually enter findings and notes or add analytical notes to results already imported.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

Linux is a very common attack platform for penetration testers. It is also quite common as a server platform, and you will encounter it on your penetration tests. In this section, we'll go over a refresher on Linux with an offensive spin.

Linux is the OS of penetration testers

- Many of the tools work better or work only in Linux
  - Windows is still a useful tool
- We will briefly cover some of the basics of Linux and the command line (bash) here, but with an offensive twist
- Various Linux distributions and shells do things a little differently, but most of the core functionality is the same

We need to briefly go through Linux basics to make sure everyone is on the same page

In this course, we will be using your Linux VM to perform our attacks. Linux is the Operating System (OS) used by the vast majority of offensive personnel. Many tools work better or work only in Linux. Also, the package management system in many of the Linux distributions make configuration and setup of the tools quite easy. While we will be using Linux a lot, remember that both Windows and Linux are simply tools, and we should switch between them as needed to be efficient.

Slingshot is based on Ubuntu. Inside our VM, we will be using the Bourne Again Shell (BASH) as our shell on the command line. There are other Linux distributions and shells, each of which does things a little differently, but the core functionally is the same.

Fun Ease-of-Use Shell Tips

- Command history, accessible via up and down arrows
- Tab autocomplete for directory and filenames
  - Tab once to expand to unique
  - Tab twice to show nonunique matches
- CTRL-R history search
- CTRL-L to clear screen
- CTRL-C to clear current command (no need to press Delete key)
- Home key and Ctrl+A to go to start of command line
- End key and Ctrl+E to go to end

Throughout this session, we use bash, one of the most common command shells in Linux distributions today, as our command shell. This shell includes many ease-of-use features that make interacting with Linux simpler. You should memorize each of these items, as they will save you much time and effort, making Linux a lot friendlier for you.

Bash, like many other shells, remembers your shell history, letting you access it by pressing the up and down arrows to access and edit recent commands, which you can rerun by simply pressing Enter.

After you choose a previous command, you can press the left and right arrow keys to position your cursor to edit the command.

Also, bash supports tab autocomplete for the names of directories and files. When accessing something in the file system, just press Tab for the shell to expand it to a unique name that matches what you've typed so far. If there are multiple items that match what you've typed (that is, there is nothing unique yet), you can press Tab again to show the names of all files or directories in your current working directory that match what you've typed so far. That is, Tab expands to a unique value, and Tab-Tab shows all items that match what you've typed so far if nothing is unique.

You can also search your history in bash by pressing CTRL-R at the start of a command line. Then start typing characters, and bash jumps back to the most recent command that has the characters you typed in that order. You can then press Enter to rerun that command or the left or right arrow key to edit the command.

The CTRL-L option clears the screen, or you can simply type clear. The CTRL-C command lets you abandon the current command and get back to the command prompt. There is no need to delete the current command by holding down the Backspace or Delete key. Just press CTRL-C to get rid of the current command.

The Home key included on some keyboards lets you jump to the beginning of a command line, whereas the End key lets you jump to the end. These options can help you jump around in long commands to make altering them easier.

Users: Root and Non-root

- Principle of least privilege – only use elevated privileges when necessary
  - As attackers, we can often exploit deficiencies in least privilege
- Non-root users
  - Have a prompt with a "**$**" (sometimes "**%**" in other shells)
  - Home directory similar to: **/home/*username***
- Root users
  - Have a prompt with a "**#**"
  - Home directory: **/root**
- As root, create an (backdoor) account: **useradd *username***
- Change your password with **passwd**
  - Change another user's password (must be root): **passwd *username***

Pen Test Tip: See if you can access other user's home directories, history files, and keys

The principal of least privilege "states that only the minimum access necessary to perform an operation should be granted, and that access should be granted only for the minimum amount of time necessary" (US-CERT). Attackers will often abuse shortcomings in the execution of the principal to escalate permissions or move from a non-privileged user to a privileged user. On a Linux system, the prompt will change depending on the user's access level. Also, the home directory for "root" and regular users are, by default, in different locations.

Non-root users

Have a prompt with a "**$**" or "**%**"

Home directory similar to: **/home/username**

Root users

Have a prompt with a "**#**"

Home directory: **/root**

If you want to create an account, such as a backdoor for persistence, you can use the "useradd" command. The command takes the following options:

useradd [options] LOGIN

-d HOME_DIR

-e EXPIRE_DATE (format YYYY-MM-DD)

-u UID (numeric user ID, must be unique unless -o is used. Users with a UID of zero are granted root level access)

See more options by running **man useradd**

Who Am I?

- Attackers will often gain access to a system and have no idea who they are or what access they have
  - The shell prompt is not always visible to an attacker
- Get your username with **whoami**
- Get your user id and group ids with **id**

```
whoami
clark
id
uid=1000(clark) gid=1000(clark) groups=1000(clark),24(cdrom),27(sudo),29(aud…
./exploit
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
```


When you successfully exploit a system, often, you will not know the level of access you have or the user you are running as. One of the first things to do is run the **whoami** command to determine your user. To get more information about your access, you can use the **id** command to get your numeric user ID and your group memberships. In the example below, we have access as a regular user (**clark**) and then, using a privilege escalation exploit, we gain root permissions.

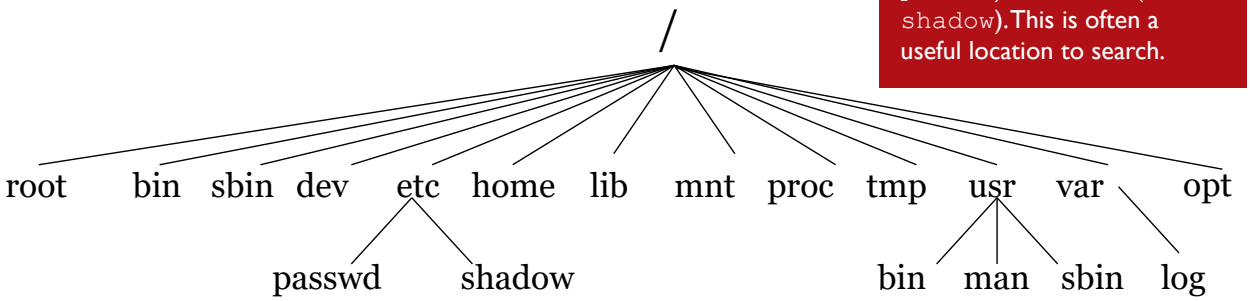
```
whoami
mike
id
uid=1000(clark) gid=1000(clark) groups=1000(clark),24(cdrom),27(sudo),…
./exploit
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
```

Linux File System Structure

- The top of the file system is called / (root)
- The filesystem can vary for different distributions of Linux and BSD

TIP

The `/etc` directory contains a number of interesting files, including configuration files, account information (in `passwd`), and hashes (in `shadow`). This is often a useful location to search.



Executable programs are stored in /bin and /sbin.

/root is the root login account's home directory. This is hugely important because if you log in directly as root, this will be your initial location in the directory structure. If you log in as an individual user other than root, you'll be put in that user's directory, typically somewhere inside of /home.

/dev stores devices (drives, terminals, etc.).

/etc holds configuration items, like the account information (stored in /etc/passwd) and hashed passwords (stored in /etc/shadow).

/home contains users' home directories.

/lib contains common libraries.

/mnt is where various remote and temporary file systems (CD-ROMs, floppies, etc.) are attached.

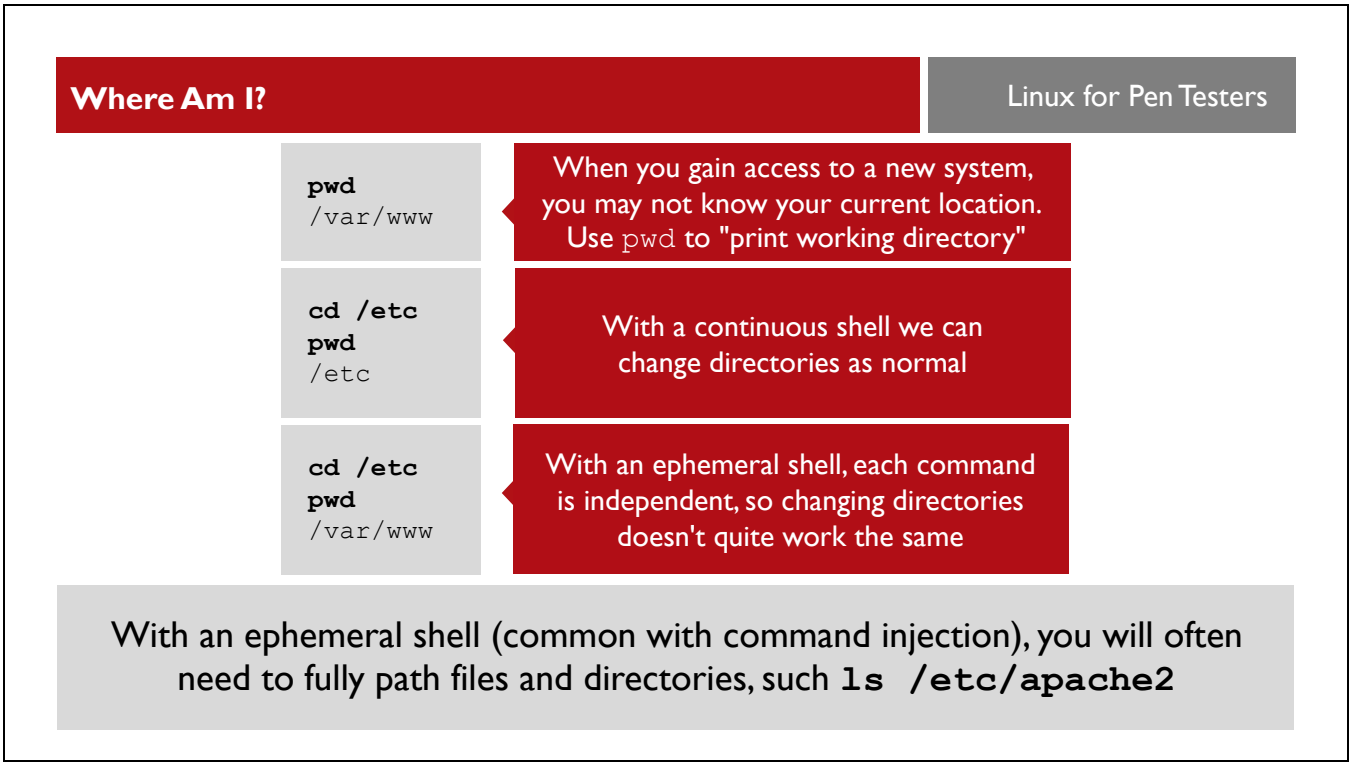
/proc is a virtual file system used to store kernel info.

/tmp is for temporary data and is usually cleared at reboot.

/usr holds user programs and other data.

/var holds many different items, including logs (/var/log/).

/opt stores optional items and is often a location for specialized tools that have been added to a distribution.



When you gain access to a system, you often do not know your current working directory. To get this location on Linux, we can use the **pwd** (print working directory) command. In a continuous shell, such as with the terminal, you can change directories and the shell remembers the change. The example below shows a continuous shell where we can't see the shell prompt.

```
pwd
/var/www
cd /etc
pwd
/etc
```

With an ephemeral connection, each command is run independently of the others. This means if you change directories, the next command has no knowledge of the change because it is using an entirely new shell. Command injection and web shells are often ephemeral. The example below demonstrates an ephemeral shell.

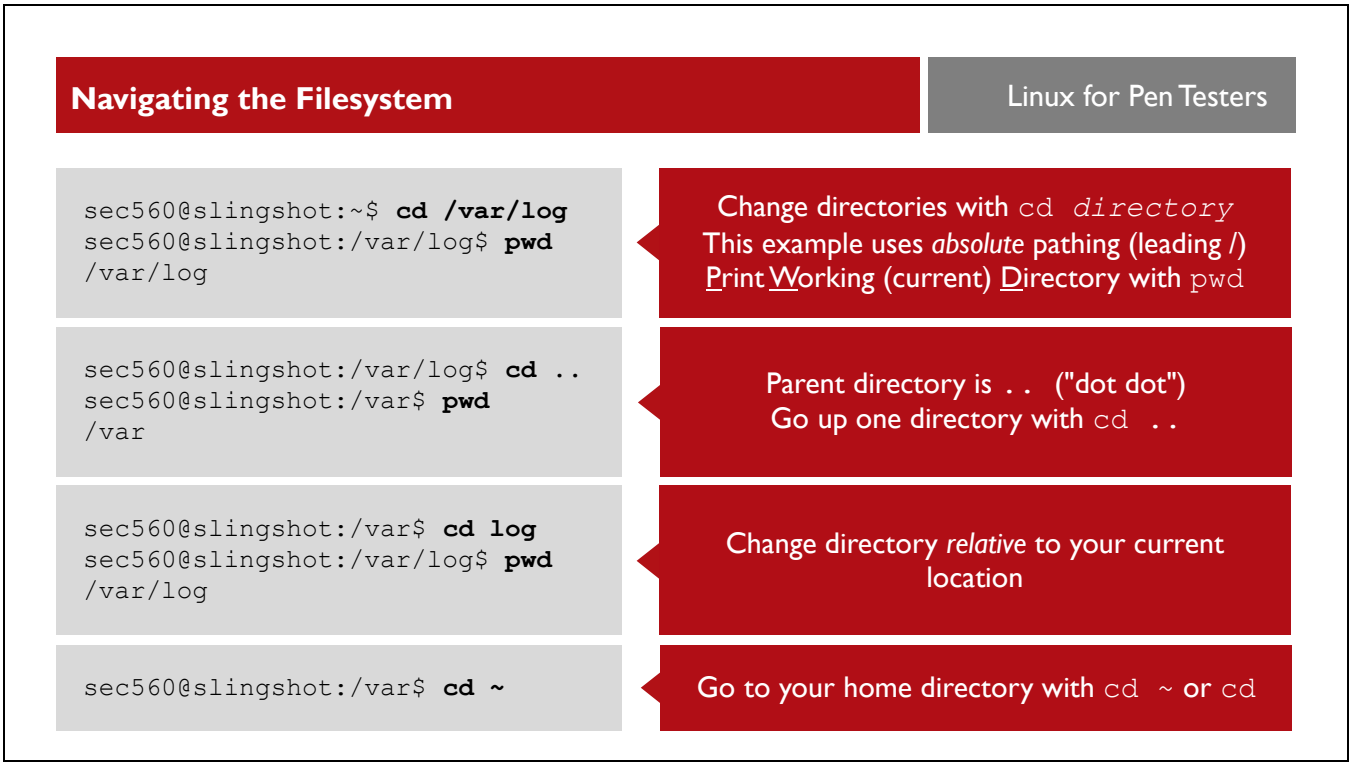
```
pwd
/var/www
cd /etc
pwd
/var/www
```

Notice in the command above that the directory didn't appear to change. If you have an ephemeral shell, you will often need to full path your files and directories. Instead of doing this:

```
cd /etc/apache2
ls
```

You need to do this:

```
ls /etc/apache2
```



We can use the `cd` command to change directories. This moves our current location to the new location. The shell will often show us our current directory, but if we want to print working directory, we can use the `pwd` command.

When referencing files or directories, we can use their absolution path or the location relative to our current location. The absolute path is the full path to the object. The command below uses absolute pathing:

```
sec560@slingshot:~$ cd /var/log
sec560@slingshot:/var/log$ pwd
/var/log
```

Relative pathing references the location relative to our current location and does not use the leading slash (/). We can use `..` ("dot dot") to access the parent directory (go up one level). This command uses relative pathing and changes our current working director to `/var`, but only because we are currently in a directory under `var`:

```
sec560@slingshot:/var/log$ cd ..
sec560@slingshot:/var$ pwd
/var
```

This command also uses relative pathing to move to the `log` directory in our current directory (`var`):

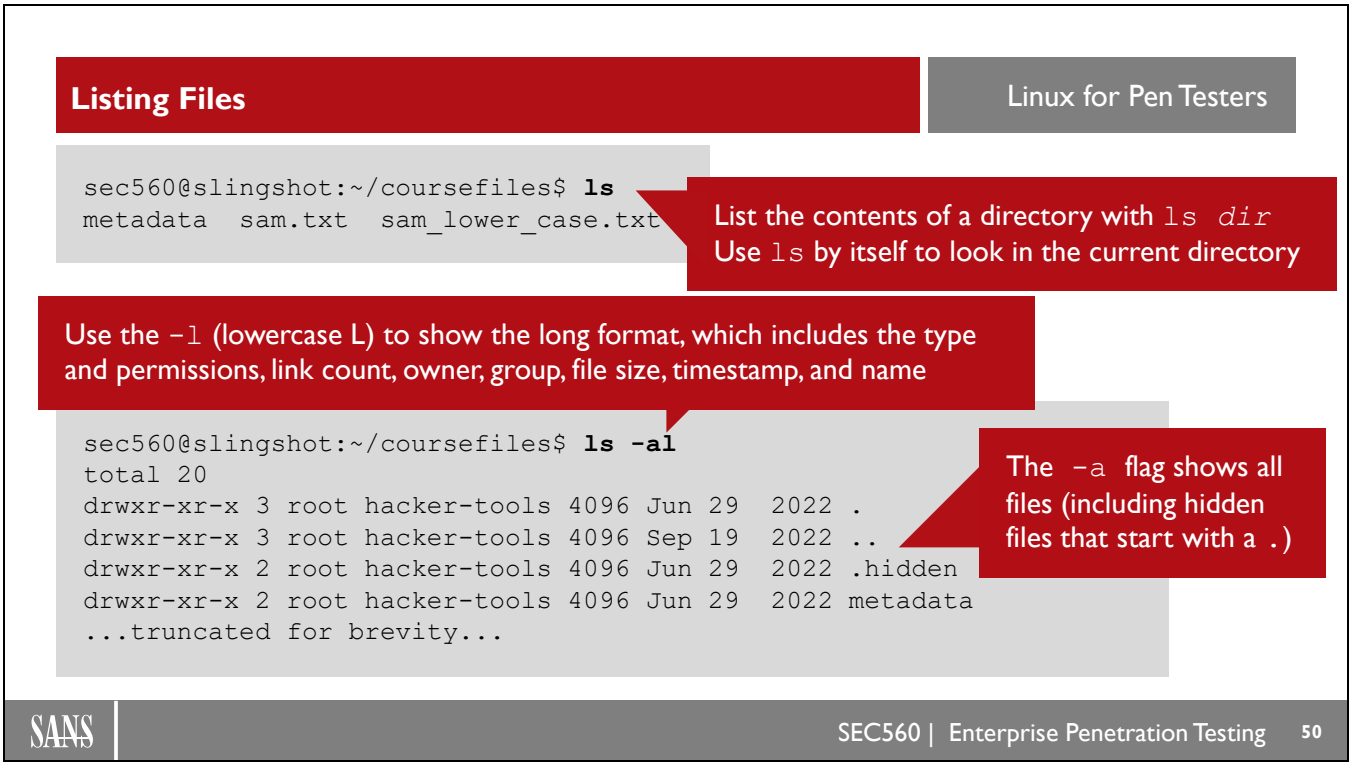
```
sec560@slingshot:/var$ cd log
sec560@slingshot:/var/log$ pwd
/var/log
```

Your home directory is aliased as ~ (tilde). To move to your home directory, you can use

```
sec560@slingshot:/var/log$ cd ~
sec560@slingshot:~$
```

If we use the cd command without a directory, it will take us to our home directory.

```
sec560@slingshot:/var/log$ cd
sec560@slingshot:~$
```



The `ls` command is used to "list" files. By default, the command will only show the file names. The default option will also skip hidden files. On Linux, an object is hidden when the first character is a dot (.). To see these hidden files, we can use the `-a` option to show "all" objects. Use the `-l` (lowercase L) to show the long format, which includes the type and permissions, link count, owner, group, file size, timestamp, and name.

Permissions


```
drwxr-xr-x  2  root  sec560  4096  Jun 2 2022  metadata
-rwsr-xr-x  1  root  root 149080  Jan 1 2022  /usr/bin/sudo
```

Permissions are broken into **4 parts**

Type directory (d), regular file (-), symbolic link (l), FIFO (p), ...

User Read (r), Write (w), and Execute (x) – Also known as owner

Group Read (r), Write (w), and Execute (x)

Other Read (r), Write (w), and Execute (x)

s is executable, but you gain the permission of the owner (SETUID) or group (SEGUID)

Change permissions with the **chmod** command

Permissions are broken into into four parts.

1. **Type**: The first letter in the permission is the type of the object. The most common letters you will see are:
   - Directory: d
   - Regular file: -
   - Symbolic link: l
2. **User** (owner): These are the permissions given to the user who owns the file. We use the term "user" since "owner" starts with an "o" and we need to differentiate between the user (u) and other (o) when using commands like "chmod" (discussed later).
3. **Group**: These are the permissions given to the members of the group assigned to the file.
4. **Other**: These are the permissions given to everyone who is not an owner or the relevant group. In certain cases, this is referred to the "world" permissions.

The User, Group, and Other permission are broken down in to three pieces. If there is a letter in the relevant position, then the permission is granted. If it is a minus/dash (-), then the permission is not granted. The permissions in order are:

1. **r**: Read Grants the permission to read the file or directory
2. **w**: Write Grants the permission to write to the file or directory
3. **x**: Execute For files, it grants the permission to run or execute the file. For directories, it grants permissions to enter the directory and access files and directories inside.

There is another special permission that can be applied to files instead of the "x". An "s" in the first (user) or second (group) is the SETUID or SETGID bit respectively. SETUID means the executable runs under the context of the user (owner). Likewise, the SETGID bit means the executable will run under the context of the specified group.

Escalating with SETUID

Linux for Pen Testers

Some commands run as root even if the user executing the command is NOT a root user

```
$ find / -perm -4000 -ls 2>/dev/null
-rwsr-xr-x   1 root     root     43088 Sep 16  2020 /bin/mount
-rwsr-xr-x   1 root     root     44664 Mar 22  2019 /bin/su
-rwsr-xr-x   1 root     root     64424 Jun 28  2019 /bin/ping
-rwsr-xr-x   1 root     root     26696 Sep 16  2020 /bin/umount
-rwsr-xr-x   1 root     root     30800 Aug 11  2016 /bin/fusermount
-rwsr-xr-x   1 root     root    149080 Jan 19  2021 /usr/bin/sudo
-rwsrwxrwx   1 root     root     50472 Jun  5  2021 /usr/local/bin/updater
```

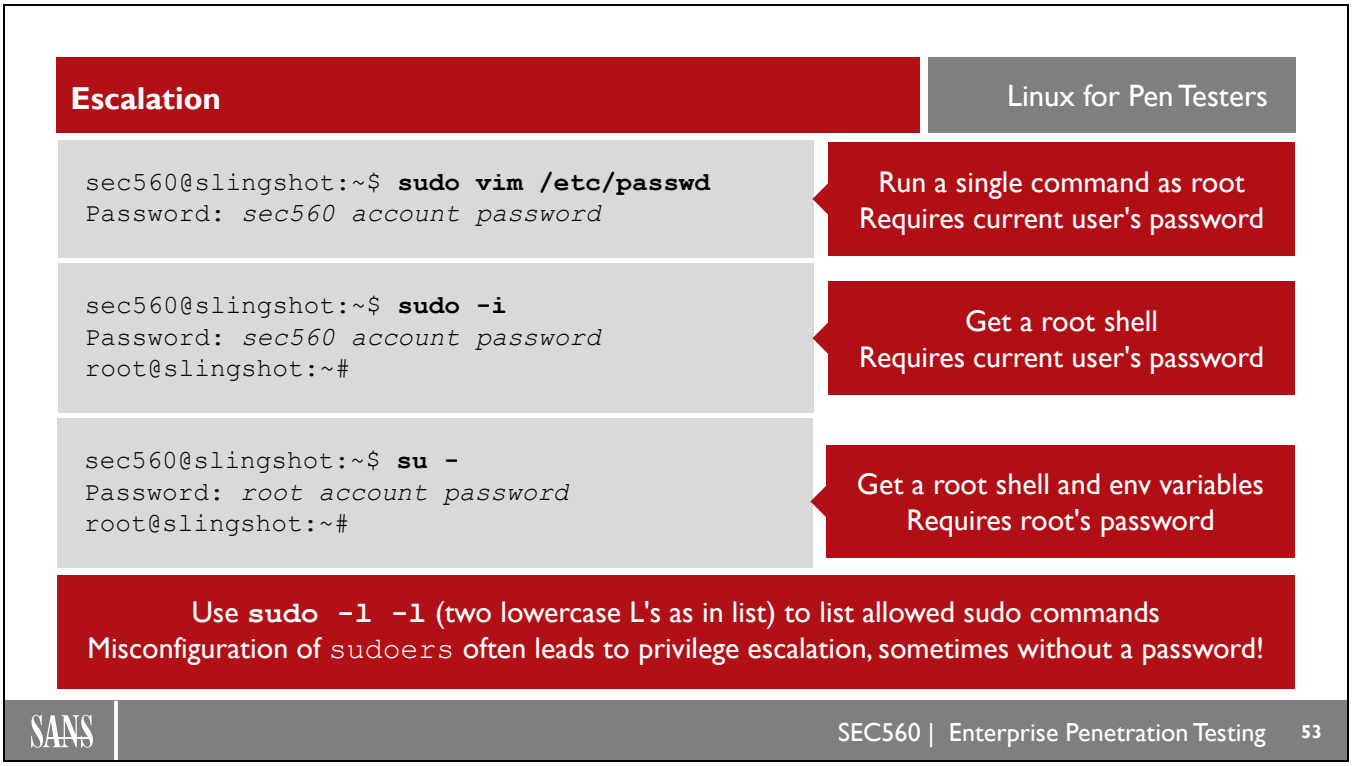
Look for files with the SETUID and SETGID bit set as they may allow escalation!
In this case, we can overwrite "updater" and escalate!

When you run an executable that has the SETUID bit set, you become the owner of the file. This is mostly commonly the powerful root user. To find these files, we can use the "find" command.

```
$ find / -perm -4000 -ls 2>/dev/null
-rwsr-xr-x   1 root   root    43088 Sep 16  2020 /bin/mount
-rwsr-xr-x   1 root   root    44664 Mar 22  2019 /bin/su
-rwsr-xr-x   1 root   root    64424 Jun 28  2019 /bin/ping
-rwsr-xr-x   1 root   root    26696 Sep 16  2020 /bin/umount
-rwsr-xr-x   1 root   root    30800 Aug 11  2016 /bin/fusermount
-rwsr-xr-x   1 root   root   149080 Jan 19  2021 /usr/bin/sudo
-rwsrwxrwx   1 root   root    50472 Jun  5  2021 /usr/local/bin/updater
```

In the command above, we specify the starting search location, where "**/**" means the entire drive. Next, we specify we want to look for files with the SETUID bit set with **-perm -4000**. The leading dash/minus before the 4000 is important as it specifies a mask, and not the exact permissions.

Some of these files may have been modified by an administrator to enable the SETUID bit. In the example above, the "updater" executable has insecure permissions. The file is what we call "world readable", meaning any user (other) can write to the file and the SETUID and SETGID bits are set. An attacker can overwrite this file with an executable of their choosing and then run it as root. This is a very easy privilege escalation!



There are a few ways to execute commands with elevated permissions. The first option is to use `sudo`. The sudo command allows us to execute a command as another user, most commonly root. To use sudo, simply prefix another command with sudo. For example, the command below will execute the vim editor as root to allow editing of /etc/passwd.

```
sec560@slingshot:~$ sudo vim /etc/passwd
Password: sec560 account password
```

You can use the `-i` option with sudo to run an interactive shell instead of a single command.

```
sec560@slingshot:~$ sudo vim /etc/passwd
Password: sec560 account password
root@slingshot:~#
```

When you run the two commands above, you are prompted for the current account's password (sec560). Assuming the user has the ability to use sudo, the commands would then run successfully as root. The privilege is set in the `/etc/sudoers` file. In this file, you can specify the executables you are allowed to run and the target user account. There are additional tags that can be set on commands, such as the `NOPASSWD` tag which tells sudo not to ask for a password. For more details, look at the contents of "man sudo" and "man sudoers".

Another way to run commands as root is with the "su" (substitute user) command. When this command is run, you will enter the password of the target account, not the original account. Many Linux distributions (including macOS) require you to be in the "wheel" or "admin" groups to run this command. The `-l` option (lowercase L, also can be shorted to just a minus "-") will simulate a full login and set the environment variables. The command below will give the user a full root shell.

```
sec560@slingshot:~$ su -
Password: root account password
root@slingshot:~#
```

Looking at Network Configs (ifconfig)

- View or change network interface configuration with **`ifconfig`**
- The **`ip`** command is newer, more flexible, and more powerful
  - This command has many options and subcommands
  - All of these commands are identical

```
ip address show
ip addr show
ip addr
ip a
```

- To look at the brief output use the **`-br`** option

```
sec560@slingshot:~$ ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0             UP             10.10.75.102/16 fe80::20c:29f0…:f3df/64
```


We can look at our network configuration using the ifconfig command.

```
sec560@slingshot:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:e6:f3:df
          inet addr:10.10.75.102  Bcast:10.10.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fee6:f3df/64 Scope:Link
          ...truncated for brevity...
```

The `ifconfig` has been superseded by the `ip` command. This command has a number of options and subcommands. It also allows us to shorten the command such that all of these commands are identical:

```
ip address show
ip addr show
ip addr
ip a
```

We can look at the brief output with the `-br` option.

```
sec560@slingshot:~$ ip -br a
lo     UNKNOWN   127.0.0.1/8 ::1/128
eth0   UP        10.10.75.102/16 fe80::20c:29ff:fee6:f3df/64
```

Reference: https://www.howtogeek.com/657911/how-to-use-the-ip-command-on-linux/

Short link: redsiege.com/560/ip-command

Command	Description
`mkdir directory`	Create a directory
`cp file1 file2`	Copy
`rm file`	To delete a directory, use the -rf (recursive, force) flags
`mv obj1 obj2`	Move a file or directory (also for renaming)
`grep somestring [file]`	Search through output for somestring in output or file
`echo sometext`	Display a line of text
`ps aux`	List processes
`cat filename`	Show the contents of a file
`head filename`	Show the first 10 lines of a file, use -n X to specify the number lines
`tail filename`	Show the last 10 lines of a file, use -n X to specify the number lines
`netstat -nap`	Show connections. -n numeric ports. -a listening and established. -p show PID.
`lsof -Pni`	Show connections. -P numeric ports. -n IP address instead of name. -i network.
`man command`	Look at the manual (help) for a command

Multiple Linux Cheatsheets available at: https://www.sans.org/blog/the-ultimate-list-of-sans-cheat-sheets/
Shortlink: https://redsiege.com/sans-cheat

The above table shows other common commands used in Linux.

Software for Testing: Prepackaged Testing Suites

Infrastructure

- SANS Slingshot includes many of the tools to get you started in pen testing

- Other Linux distributions can also be helpful
  - Kali Linux by Offensive Security: http://kali.org
  - Parrot Linux from Parrot Security: https://parrotlinux.org/

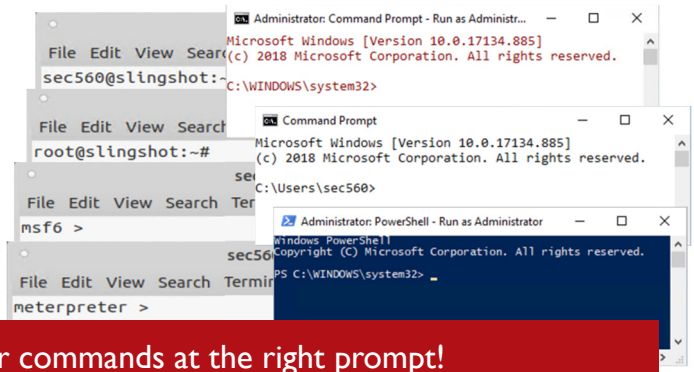
You can use another Windows or Linux system during the labs, but remember, we have tested the provided Slingshot and Windows VMs. Also, these VMs have the EWB, and other files needed for the labs!

First, you need software for your testing regimen. With this course, you received a copy of the SANS Slingshot image, which is full of tools used in ethical hacking and penetration testing. Furthermore, this VMware image includes tools pre-installed, and in many cases, preconfigured so that you can apply them directly in your own testing.

Another useful source of tools is the bootable Linux distributions various people have made freely available, loaded with useful assessment and attack tools. A solid set of tools is included in Kali Linux, created and maintained by Offensive Security. Numerous similar Linux images for pen testing are also available, but Kali is one of the best because of its comprehensive set of tools, compatibility with a wide range of hardware, and carefully designed organization and layout.

An Important Note: Command Prompts

- We work with numerous different shells and switch often
- Be aware of shells within shells: Linux → Metasploit → Meterpreter →…
- The labs and notes indicate the prompts
  - Windows cmd.exe: `C:\>`
  - Windows PowerShell: `PS C:\>`
  - Linux (non-root): `$` or `%`
  - Linux (root): `#`
  - msfconsole: `msf6 >`
  - Meterpreter: `meterpreter >`



Please make sure you enter commands at the right prompt!

Throughout this course, we use numerous different shells, both in our operating system and within Metasploit. We frequently change between these different shells as we switch back and forth between Linux and Windows, as well as within different aspects of Metasploit. Sometimes, even on a single page in the book, you use two or even three different types of shells to do something and then observe the results.

All the labs and notes were carefully written to indicate the proper shell you are supposed to use at any given time by including the shell prompt right before each command you are supposed to type. That is, each lab command is preceded by the prompt indicating which shell to use. The shell types you encounter throughout this class include:

Windows cmd.exe with the prompt:

```
C:\>
```

Windows PowerShell with the prompt:

```
PS C:\>
```

A Linux bash shell running as a regular user:

```
$
```

A Linux bash shell running as root:

```
#
```

The Metasploit Framework Console with the prompt:

```
msf6 >
```

A Meterpreter shell with the prompt:

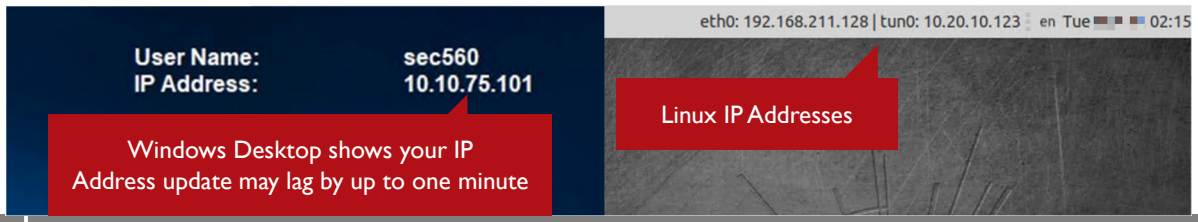
```
meterpreter >
```

DOUBLE-CHECK AT EACH LAB STEP THAT YOU ARE ENTERING THE PROPER COMMAND INTO THE PROPER SHELL. Otherwise, a given lab step will not work for you properly.

Networking

USB and Targets

- VPN address will be 10.254.X.X
  - **`Tun0`** on Linux, **`Ethernet2`** on Windows
- NAT addresses: will vary depending on your system!
  - It will commonly be 172.16.X.X or 192.168.X.X, but it could be different
- Netmask and DNS are set automatically
- In your VMs, we have helpers that show your IP address in the UI
  - Windows IP address can lag by up to a minute



All the VMs will use DHCP to dynamically acquire IP address and network information.

You VMs will be configured to use "NAT", where the network connectivity will be shared with the computer and the VMs are not directly accessible on the local network. The VMs will be able to communicate with each other.

Many of the labs will require that you connect to the SANS hosted infrastructure using a VPN. To configure the VPN, follow the directions in your workbook (electronic or paper).

You will not use your host system in any of the lab networks; however, your host will get an IP address on the lab network (for in-person classes), so make sure you have the latest software patches and host protections in place.

Be sure to use the correct interface and IP address depending on target

Between local VMs	Between VM and remote targets (VPN)
• Linux: `eth0`	• Linux: `tun0`
• Windows: `Ethernet0`	• Windows: `Ethernet2`
	• Address will be `10.254.X.X`

You will need to connect to your VMs differently depending on the target

If you are taking the course live in the classroom and the labs are hosted locally in the room, you can skip this page. This page is important if you are using remote labs via VPN (LiveOnline or OnDemand).

Depending on how you are using your local VM, you may need to use the IP address for the local interface or the VPN interface. If you are setting up a connection directly between your VMs, then use the address assigned to the `eth0` interface on Linux or the `Ethernet0` interface on Windows.

If you need to have a remote target connect to your VM, such as an exploit callback, use your Windows or Linux `tun0` interface and associated IP address.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

Let's now practice the Linux skills we learned with a lab exercise.

Please work on below exercise.
Lab 1.1: Linux for Pen Testers

Please go to Lab 1.1: Linux for Pen Testers in the SEC560 Workbook.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

With our infrastructure in place, let's now go over the overall ethical hacking and penetration testing process. This process should be applied to all the testing that you do. Be careful in skipping any of the steps we describe. Some steps are designed to ensure that you've conducted a test with appropriate legal protections. Furthermore, other steps help ensure that you are providing demonstrable value to the organization you are testing.

Overall Penetration Testing Process

Overall Process

Preparation	• If applicable, sign Non-Disclosure Agreement (NDA) • Discuss nature of test with target personnel – Identify most salient threats and business concerns – Agree on Rules of Engagement – Determine scope of test • Sign off on permission and notice of danger of testing • Assign team
Testing	• Conduct the test
Conclusion	• Perform detailed analysis and retest • Reporting and (possible) presentation

The overall penetration testing process involves preparation, testing, and conclusion phases.

During the preparation phase, the parties participating in the test may sign a Non-Disclosure Agreement (NDA), especially if the test is conducted by a third-party organization. Then the testers and the target personnel discuss the most significant concerns of the target organization. What are the biggest threats? Which systems are the most sensitive? What kind of information is the most valuable? We also agree on Rules of Engagement that describe how the testing will occur. Next, the scope of the test is determined, a process we'll discuss in depth later.

The next step is absolutely crucial. You need to get official, written permission to conduct the test, even if it is against targets in your own organization. This permission should notify the personnel associated with the target systems that there is some danger of their systems being crashed or impaired by the testing. We'll discuss this permission memo in more detail shortly. Then, based on the nature of the test, a team of appropriate testers is assigned based on its technical areas of expertise and business knowledge of the target environment.

The test then occurs, potentially lasting from a day to many months.

To conclude, the team then analyzes the results, trying to discern their business implications. The technical details and business implications are described in detail in a final report. As findings are addressed, single-issue retests could occur, or an entire comprehensive retest may happen. Some tests conclude with a wrap-up final presentation.

Resources: OWASP Testing Guide

Overall Process

Focus is on Web Application Testing
Gets quite deep into techniques and tools

- Info gathering
- Business logic testing
- Authentication testing
- Session management testing
- Data validation testing
- Denial-of-service testing
- Web services testing
- API testing

Also includes a great discussion of determining risk severity

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Free at https://owasp.org/www-project-web-security-testing-guide/
Shortlink: redsiege.com/560/owasp-testing

Next is the *Open Web Application Security Project (OWASP) Testing Guide*. Unlike the other broad and general-purpose methodologies, we've touched on so far, the OWASP guide focuses purely on web application security testing. From a web app perspective, this document is an excellent description of the various kinds of testing that need to be done, providing great depth and a wide variety of tools to use in the process.

One of the best aspects of the OWASP guide is its detailed description of determining the business risk posed by findings. The OWASP guide rates risk based on the impact it could have to the business and the likelihood it has of occurring. From those two aspects, the overall risk rating of a given finding is derived, giving the enterprise appropriate guidance on prioritizing the findings.

References:
Free at https://owasp.org/www-project-web-security-testing-guide/
Shortlink: redsiege.com/560/owasp-testing

Resources: MITRE ATT&CK

- MITRE ATT&ACK framework is a knowledge base of attack tactics based on real-world attackers
- "Techniques" are grouped together into "tactics"
  - Phishing and Valid Accounts *techniques* are part of the Initial Access *tactic*
- The framework includes "Groups", which groups set of related activity tracked to a common adversary group (e.g., APT1, FIN7)
- Great resource for learning specific tools and techniques as well as understanding the playbook of real-world attackers

MITRE ATT&CK Matrix for Enterprise can be found at https://attack.mitre.org/

MITRE ATT&CK (pronounced "attack") is a knowledge base of attack tactics based on real-world attackers. The framework includes many techniques (and sub-techniques) used by good and bad attackers. It is valuable resources for learning new tools and techniques. The techniques are grouped together into "tactics". For example, the "Initial Access" tactic includes the techniques Exploit Public Facing Application, Phishing, Valid Accounts among others.

The tactics and techniques are pieces used by an attacker. ATT&CK also includes "Groups", which is a set of related activity tracked to a common adversary group (e.g., APT1, FIN7). These provide the story of various breaches, which we can use to learn when and how to use specific attacks.

The combination of the Groups and the Tactics/Techniques make ATT&CK a valuable resource to develop our own offensive skills.

MITRE ATT&CK Matrix for Enterprise can be found at https://attack.mitre.org/beta/.

Documented permission is critical for pen testers

Internal testers can use an internal permission memo or "Get Out of Jail Free Card"

- Sample: www.counterhack.net/permission_memo.html

External testers need more protections and assurances

- Contract language limiting liability should be drawn up by a lawyer
- Liability is commonly limited to the value of the contract
- Pricing, payment terms, and other such business requirements
- Ownership of intellectual property
- Insurance: Liability, Errors and Omissions, Cyber/Breach

It is absolutely vital that you get signed permission from an authorized person at the target organization giving you permission to test its environment. This memo is sometimes called a "Get Out of Jail Free Card".

There is a free sample memo on the Counter Hack website at www.counterhack.net/permission_memo.html.

Among other things, this memo states:
The purpose of this memo is to grant authorization to specific members of our information security team to conduct vulnerability assessments and penetration tests against this organization's assets. To that end, the undersigned attests to the following:

1) *[Insert name of tester], [Insert name of tester], and [Insert name of tester] have permission to scan the organization's computer equipment to find vulnerabilities. This permission is granted for from [insert start date] until [insert end date].*
2) *[Insert name of approver] has the authority to grant this permission for testing the organization's Information Technology assets.*

Have your legal team review, tweak, and approve this language. Then, print it on corporate letterhead and have a Chief Information Officer or similar level of management sign-off on it. Note that this memo is suitable for employees to test the security of computer equipment owned by their employers, not for third-party tests.

Third-party testers require significantly more paperwork since they do not work directly for the company. The contract language should be drawn up by a lawyer to ensure neither party is put at undue risk. Common language includes a limitation of lability (commonly limited to the value of the contract), ownership of intellectual property, and payment terms. The report is often owned by the penetration testing company and then licensed to the client in perpetuity. If the client owned the report, then there could be an issue reusing language in the report (such as text associated with common finding). Additionally, many clients will require insurance of between US$1M and 5M+. The key to contracting is that they are not designed to help you when things go well, they are designed to handle things when things go poorly. Too many times, pen testers and/or clients have been stiffed in payment or deliverables because the request was not properly documented or approved. While contracting can be daunting, do not overlook it or you may not get what you expect.

Follow the Law

- Many (but not all) countries have laws regarding cyber crimes and privacy
- Which countries' laws do we need to follow?
  - Tester country
  - Target country
  - Infrastructure country
  - Packet traversal country?
- Consult with your lawyer before testing in a new country
- The texts of the dominant cybercrime laws of more than 100 countries (including the US, Canada, UK, Germany, Singapore, Australia, India, and more) have been gathered by the United Nations at https://www.unodc.org/cld/v3/cybrepo/legdb/

Operate only within a clearly defined scope, against machines for which you have explicit permission to test from both their owners and operators

Many countries have instituted laws for dealing with crimes committed using a computer, so-called cybercrime laws. Not all countries have such laws, and indeed, attackers sometimes move to countries or operate through countries without such laws or where cybercrime laws are not enforced.

As penetration testers and ethical hackers, we want to make sure we carefully adhere to the laws of the countries in which we operate. In essence, conduct all your tests according to agreed-upon Rules of Engagement with a clearly defined scope. Attack only machines for which you have explicit permission, in written form, from both their owners and operators.

Your Permission Memo (the Get Out of Jail Free Card) is a helpful thing in ensuring that you have the permission of the target organization that owns and operates the systems you will test.

It is important to note that the tester has to follow not only the laws where the tester is located but also the laws in the country where the target machines are located. Furthermore, some countries (such as the United States) have indicated that any packets associated with a computer crime that traverse the country's borders fall within their law enforcement jurisdiction, regardless of where the packets originate or terminate. In other words, an attack from Germany against targets in Japan that traverse US networks would be subject not only to German and Japanese law but also to US law.

Because some of the legal issues associated with testing in different countries can grow complex, we strongly recommend that you consult a lawyer when testing in a country where you haven't tested before. The lawyer can help explain the rules of the road to you, helping to make sure you don't run afoul of the law. For the most part, with a carefully documented Permission Memo, Rules of Engagement agreement, and Project Scope, you can operate in most countries legally without incident. Still, a lawyer's review can be helpful in establishing peace of mind and making sure you've taken any late-breaking legal issues into account when formulating your test plans.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

Next, we will cover what happens before we begin the penetration test, the pre-engagement phase. This phase includes the legal paperwork, contracting, rules of engagement, scoping and the kick-off. Both the people responsible for the target environment and the testing team must agree on these rules. Without properly agreed upon test, a penetration test or ethical hacking project could go seriously awry, resulting in devastating consequences for the target organization and the testers, including system downtime, financial damage, reputational damage, personnel firing, and possibly civil lawsuits or even criminal prosecutions.

Pre-Engagement Steps

1) **Goals** – Discuss the goals of the test to frame the rest of the discussions
   a. Important data and/or processes
   b. Why is the test being performed
2) **Type of tests** – What type of test do they need? (Network, Web, SE, …)
3) **Scope**
   a. In-scope IP addresses, subnets, URLs, people/roles (social engineering)
   b. Select exclusions
4) **Rules of Engagement**
   a. Usually, a checklist
   b. Add additional exclusions or exceptions
5) **Kickoff Call** – Final planning, establish communications channels and people

You may go back and forth a bit, that's OK. The key is to get details here!

Many third-party penetration testers have gone through the test hundreds of times, but that doesn't mean you can't be prepared as well, even if it is your first time procuring a penetration test. This process is important for both the penetration testers and the recipients of the test. Don't be afraid to ask questions of the other side. It is always better to ask "dumb" questions and get the answer you expect than to assume and be wrong.

The process to define a high-value penetration test can be daunting for people new to the process. The most important thing for both sides to understand are the goals. First, what is the sensitive data or process that if compromised, stolen, or destroyed would cause the greatest impact to the organization. Second, why is the test being performed (compliance, new system deployment, audit requirement, …)?

The next question to discuss is the type of tests to perform. If the test is executed for compliance reasons, then the type of tests often are straight forward. Otherwise, the aforementioned goals can help channel this discussion.

Once the tests are decided upon, the scope is often self-evident. As the penetration tester, make sure you ask about systems that are out of scope and the systems owned by third parties.

The Rules of Engagement is usually a checklist based on the test types selected. There are a few discussion points that we will go into in more depth later.

Finally, you will need to determine communication methods and points of contact.

We'll go through each of these items in more depth in the following section.

Goals

Pre-Engagement

- Always ask what **data or process is most important**
- It is alright to **ask the question** and get the answer you expect rather than to **assume and be wrong**
- Focuses the rest of the discussion and the assessment
  - Higher value penetration test
  - Focus on organizational goals, not technical goals (e.g., Domain Admin)
- Also, ask why they are doing the test
  - Compliance, audit, new systems, ...
- Goals will guide the rest of the discussion

If the penetration testers **don't ask** your goals, **find new testers!**

It is very important to have and share the goals of the penetration tests before having further discussions. Understanding and acknowledging these goals will help guide the entire test as well as the preparation discussions. As the tester, never assume you know the important data or processes of the target, even if they are part of the company. The target has inside knowledge of the systems that you do not have. Even if you are an in-house tester, you may not know the details of the target. Ask the question, "What data or process if lost, stolen, compromised, or destroyed would cause the greatest impact to your organization". If you aren't comfortable asking a "dumb" question then try leading with, "I think I know the answer, but I'd rather ask than assume and be wrong".

You might get the answer you expect, but that's alright. It is better to ask the question and get the answer you expect rather than to assume and be wrong. Being wrong means the focus of the test is incorrect and can greatly impact the value of the test.

Types of Tests and Allowed Actions

Pre-Engagement

- Ping sweep and port scan of target networks and hosts
- Vulnerability scan of targets
- Penetration via listening network services
- Penetration via client-side software
  - A dominant attack vector today
  - Assumed breach or phishing?
- Post-exploitation
- Application-level manipulation
- Physical penetration attempts
- Social engineering of people

After establishing the goals, we should talk through the tests that meet those goals. Will the test merely be a network scan for targets and vulnerabilities, which would include a ping sweep, port scan, and vulnerability scan? Or should the testers go further and actually penetrate the target systems, getting access (such as a remote shell) on the targets if possible? If penetration is allowed, should it focus on listening network services, or will client-side software exploitation be allowed as well? Client-side software exploitation is a dominant attack vector today. If client-side exploitation is allowed, which client machines will be included in the scope? If exploitation of any kind is allowed, should the attackers continue to pivot through the organization?

Will the test include any application-level or client-side web component testing? Will it include physical penetration attempts, with the team trying to walk into an environment? Should social engineering, which involves duping human beings, typically via the telephone, be attempted?

Your Rules of Engagement should specify each element on this list that is included in the scope.

Scope

- Pre-Engagement
  - What is to be tested?
    - Specific domain names
    - Network address ranges
    - Individual hosts
    - Particular applications
  - **Exclusions** – What should be explicitly avoided?
  - Check when additional items are discovered before attacking them
  - A tight definition here will help prevent "scope creep" or missed targets
  - Penetration tests typically time-boxed 1 to 3 weeks

A tight definition of scope is better for everyone. One of the most important elements to include in the project scope is a succinct statement of what is to be tested. Spell out explicitly those domain names, network address ranges, individual hosts, and particular applications that are included in the test.

Also, if there are particularly sensitive elements of the organization that should not be tested, explicitly spell out that list of off-limits machines.

While the team is performing the test, they may discover additional systems within network address ranges or domain names that weren't considered in the original formulation of the scope. Make sure the Rules of Engagement direct the testing team to check with the target organization before testing any other machines outside of the original scope that are discovered through reconnaissance or scanning after the test starts.

Even if the test is a "zero-knowledge" assessment, get approval for new targets as you find them. It is better to confirm than to attack outside assets and risk legal action.

Even with a defined scope, the test is often restricted to a specific duration. The scope and duration should be defined together to ensure sufficient coverage of the targets.

Scope: Third Parties

- Pre-Engagement
  - Cloud services
    - Some Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) providers do allow application-level testing, but no network service testing
    - Azure and AWS allow pen testing (subject to limitation) without pre-approval
    - Most Software as a Service (SaaS) providers prohibit customer testing
  - ISPs: routers, switches, mail servers, DNS servers, and more
  - It is unlikely that the target organization can authorize a penetration test of a third party's assets

The only difference between the "good guys" and the "bad guys" is permission

If any systems owned or managed by third parties are included in the scope, make sure to get documented permission from these parties before the test begins. The target organization is responsible for getting this permission, and the testers are responsible for making sure the target organization does this.

Many major organizations today use at least one third party to manage at least part of their computing and network infrastructure. Some companies outsource this altogether, with their infrastructure actually being owned by the third party itself. Examples include Internet Service Providers, whose routers, switches, mail servers, and DNS servers may fall into the scope of a test.

Always remember that permission is the only difference between a "good guy" and a "bad guy". If the target organization has assets in the cloud, that client is unlikely to have permission to authorize a penetration test. Check the contract language before beginning a pen test. Of course, it never hurts to double check to make sure everyone is legal and has permission. An extra check up front can reduce the likelihood of the cloud provider taking your systems offline or shunning the penetration testers (and wasting the tester's time).

For third-party-provided clouds, check the contract to see if it allows or explicitly forbids security assessments or penetration tests. Most cloud providers explicitly forbid such tests unless you specifically ask for and receive written permission. Some cloud providers will simply provide the results for their most recent assessment to their customers upon request or choose a specific set of testers to honor a client's request for a needed test. We've seen many cloud providers that offer underlying virtualized systems, in a Platform as a Service (PaaS) or Infrastructure as a Service (IaaS) arrangement, allow customers to perform application-level testing (such as web app testing) but prohibit them from testing for underlying network and listening service flaws. Both Microsoft Azure and Amazon AWS allow penetration testing without pre-approval (subject to limitations).

https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement
https://aws.amazon.com/security/penetration-testing/

Many Software as a Service (SaaS) providers prohibit customers from conducting their own tests, whether at the network/service layer or the application layer. Make sure you get appropriate written permission before launching any tests.

Rules of Engagement

Pre-Engagement

- **Pre-approve safer attacks** like scanning, password guessing, etc.
- Attacks with an **elevated chance of a negative impact** require **specific approval** and/or testing during a maintenance window
  - Make sure it is documented (email is usually fine)
  - Responder (with the wrong options), Password guessing (lockout), certain exploits
- Denial of Service (DOS) is rarely authorized
  - Asking is likely a wasted effort unless you have a specific useful attack

It is important to establish the Rules of Engagement (ROE) before beginning the test. It is always better to ask simple questions rather than to assume and later be wrong. You could end up reducing the value of the penetration test, wasting both time and money. The consequences could be far worse, you could receive angry calls from other business units, other companies, service providers, or other third-party companies. In effect, someone attacked them, possibly giving them legal standing for a suit against you! Plan carefully in advance to minimize that risk.

Documentation of the ROE, and a target signature is a useful tool. At some point in your career, something bad will happen to a target, and that signature can be a powerful tool, especially if the authorizer conveniently says they did not authorize a specific action. This is never a fun experience, but that signature could save your job or career.

The ROE discussion doesn't have to be a complex discussion. Describe the basic process of the test and the safe actions, such as scanning, password guessing, and normal use of protocols (login via RDP, access via SSH, etc.). Of course, never guarantee your actions will never cause harm. Systems crash on their own and we can't control that; unforeseen circumstances always exist. Informing the target that you will take care to perform only reasonably safe actions and any riskier options will require additional approval. Higher-risk issues will often require justification (outline of the goal) to even be considered. Be prepared to do this testing during a maintenance window.

You can ask about Denial of Service (DOS) attacks but know that it is very likely the target will decline such testing. We recommend not pushing the target on denial of service (DOS) attacks unless you can demonstrate an attack that will help you complete your objective.

Dates and Time of Day

- Agree upon an explicit start date and a finish date
  - Never let these things go as a total surprise
  - At least one person in the target organization should know when the test begins and ends
- Agree upon acceptable times for testing
  - For particular production environments, some target organizations request evening-only or weekend-only tests
  - Off hours testing will usually incur additional fees (up to 25%)

Of course, the Rules of Engagement should explicitly state the start and end dates of the test, as well as valid test times.

Some penetration tests and ethical hacking projects run around the clock, whereas others with more sensitive infrastructures and business needs require testing during off-hours or weekends only. When such off-hours tests are conducted by a third-party company, there is typically a slight, additional charge for such off-hours testing, but it is usually quite reasonable. Also, such limited testing time windows require a longer total duration to complete the test.

Announced vs. Unannounced Tests

Pre-Engagement

- Will the target's system admins, security team, hunt team, and incident handlers be informed of the test?
- Or will their response to a surprise test be measured?
- There are benefits of each
- However, be careful with an unannounced test!
  - Defenders may discover the scans and then shun all traffic
  - Actions taken after shun are useless (waste of time/money!)
  - If shunned, what happens next?

Here's another point that can cause controversy with some target organizations: Should the penetration test be announced to the people responsible for running the target infrastructure? Or will the test be a surprise to them?

Performing an unannounced test does have some advantages. First, if any of your admins are purposely running backdoors or side businesses on your servers, the testing team might find them during an unannounced test. If you announce the test, the admins will likely temporarily shut off their shifty activities during the test duration. Some penetration testers have discovered deliberate, sys-admin-planted backdoors during an unannounced test. Other testers have identified pay-for-porn services on target web servers that were run by the web administrator. Such findings are important results of a penetration test.

Second, an unannounced test gives you a chance to evaluate the detection and response mechanisms and processes of the target organization. Does information flow properly through the organization concerning a computer attack?

However, unannounced tests also have a downside. The system or network administrators might detect the attack and start shunning the traffic, blocking it from reaching the target systems. Then any testing activities after the shunning is applied are invalid, a waste of time and money. If you do perform an unannounced test, make sure the system and network administrators are watched to verify that no shunning occurs.

Also, to prevent controversy about such tests, make sure that target personnel know that they could occur at any time and are just a normal part of the way the organization does business.

Zero-Knowledge vs. Full-Knowledge Testing

Rules of Engagement

- Will the testers be given network diagrams and system info?
- Reasons for Zero-Knowledge testing:
  - "More like the real-world attackers"—but is that true?
  - Non-existent or deficient architecture documentation
- Reasons for Full-Knowledge or Limited-Knowledge testing:
  - More efficient and cost-effective
  - Attackers may have this info (dumpster diving, insider attacks, time)
  - Less chance of an error causing damage to systems
- Hybrid approaches are possible, but take more time (costlier)
  - Daily debriefing can help foster knowledge transfer

Here's another point of some controversy. Should the testing team be given a copy of the network diagram, listing hosts, topology, operating systems, and so on? Such crystal-box testing allows testers to see inside the target before launching any scans. Alternatively, the test might be a zero-knowledge engagement, where the testing team is given only a domain name and is expected to figure out the network topology and targets through reconnaissance and scanning.

The reasons some organizations say that they opt for zero-knowledge testing is that it is "more like what a real-world hacker would see in our environment". Unfortunately, that's not necessarily true. A real-world attacker might have a picture of your network architecture, stolen from a dumpster, faxed from a duped employee, or swiped by a temporary employee. Working from a network diagram lets the penetration testing team analyze a worst-case scenario from the target's perspective.

A better argument for zero-knowledge testing is that some organizations are worried that a network diagram may bias the testers so that they miss items. Many network diagrams have significant limitations or errors, missing a lot of detail and sometimes overlooking hosts or whole networks.

In the end, full-knowledge testing tends to be more cost effective because the attackers can quickly perform their reconnaissance and scanning. There is also a lowered chance of an error causing damage to an out-of-scope system. Although most penetration testers perform both zero-knowledge and full-knowledge testing, the latter approach is usually recommended. Of course, you could take a hybrid approach in which the testing team starts out with nothing more than a domain name and performs detailed recon and scanning. Then after a week or so, the target organization provides a detailed diagram for further testing and analysis. When performing such hybrid testing, explicitly label in your report the elements the testers discovered, and which items were given to them by the target organization. Hybrid tests usually take a little longer and therefore cost somewhat more. You can use the daily debriefing conference call on a zero-knowledge penetration test to start asking more questions, which could transition the project into a more full-knowledge-style test.

Be Careful Viewing Data on Compromised Systems

Pre-Engagement

Do not view or download protected or sensitive data

PHI: Are you a medical practitioner in care of the patient (No!)
PII: Be very careful with credit card and banking information
HIPAA, GLBA, GDPR: Technically, you are a breach if you access this data
Intellectual Property: How should you handle this sensitive data?

- Accidents happen, so be careful
- Do NOT include the sensitive data in the report (redact!)
- Demonstrate access and quantity (access to a DB table and record count)
- Sometimes (with appropriate permission) it may be acceptable to sample small amounts of data to confirm access and assess business impact

Here is a crucial issue to emphasize in your Rules of Engagement: If the team successfully penetrates a target, should they avoid viewing data on the target? Think about it: You might have sensitive employee or customer data on the target. A penetration tester has just gotten root on the box and might be able to view all that data. Only your Rules of Engagement prevent that from happening, so make sure they are clear on this topic. For some kinds of information, including healthcare and personal financial data, only duly authorized representatives with a need to know are allowed to see such information, under various regulatory initiatives such as the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act (GLBA) in the United States and the EU General Data Protection Regulation (GDPR).

We recommend that your default Rules of Engagement allow the testers to view configuration data from a machine but to avoid looking at any customer and user data on the machine. Also, if sniffers are used on the compromised box, make sure the team explicitly documents that any personal data captured from the network will be ignored by the testing team.

Although the default policy is usually to avoid accessing sensitive user data, in some tests, it may be appropriate to sample small amounts of sensitive user data to confirm access and assess the business impact. Such access should be done sparingly and only with written permission from target system personnel. Also, try to focus more on counting the number of sensitive records you can access (based on file length, number of lines, number of files, number of rows in a database table, or other metric based on the type of data and the form of access you've gained) than on retrieving the sensitive content.

Common Signed Paperwork

Pre-Engagement

- Non-disclosure Agreement (NDA) – Shhh! No talking about the test!
- Master Service Agreement (MSA) – Items broader than a single test
  - Payment terms
  - Intellectual Property (IP) – Who owns the report? If the client owns the report, can the testers reuse findings? Common solution, the testers own the report and give a non-revocable license to the report
  - Notice of possible damage or disruption – Unintended consequences can and will happen. Testers need to limit liability if unforeseen circumstances happen
  - Example: https://redsiege.com/msa *
- Statement of Work (SOW) – Details specific to the test
  - Scope, ROE, Methodology, types of tests to be performed

In the report, state that you may miss vulnerabilities and that the test is a point in time snapshot!

When an organization engages third-party penetration testers, there is a common set of documents that are signed. This may vary somewhat, but these three documents are very common.

Non-disclosure agreement (NDA) – Also known as a mutual non-disclosure agreement (MNDA). This document limits what each part can share about the test including data the testers may find during the test.

Master Service Agreement (MSA) – The MSA contains items that are much broader than a single test. For example, an organization will negotiate common payment terms for all outside vendors to make it easier on the organization. This includes critical penetration testing specific pieces such as:

- Intellectual Property (IP) Ownership – Who will own the report? If the client owns the report, does that mean the that if the testers reuse general findings they are infringing on a copyright? To resolve this issue, many testers retain ownership of the report and offer a non-revocable license to the report.

- Another important piece is limitation of liability for damage or disruption. If damage is caused through unforeseeable circumstances, then the testers are not held liable. You should never guarantee 100% uptime, as systems will crash even under the best operating conditions.

Statement of Work (SOW) – Contains the details specific to the test including the scope, methodology, and types of tests to be performed.

Finally, never state that you will find all the vulnerabilities. The MSA, SOW, and/or report should include a clause stating that issues may have been missed and that the test is a point in time snapshot. Additionally, make it clear that your service does not guarantee the security of any system, or that computer intrusions will not occur.

If you would like to see a live contract, including sections on intellection property and damage/disruption, Red Siege's default Master Services Agreement* is available at: https://redsiege.com/msa.

*Note: No warranty is associated with this document, or the links provided herein. This should not be considered as legal advice. Talk with a qualified lawyer in your jurisdiction for any contracting questions. This document, the example links, and instructor commentary is solely provided as an example.

Kickoff Call

Pre-Engagement

- Go over the test scope and RoE as a reminder to everyone
- Exchange contact info (Name, Role, Mobile/Cell, Availability)
  - Pen testing team may notice erratic behavior or a crash in a target system
  - Penetration test might discover evidence of previous intrusion
  - Intruder could be attacking at the same time and target needs deconfliction
- Agree upon secure data transmission method for the report, vulnerability data, passwords, etc.
  - Encrypted email, secure content management (e.g., Box), GPG/PGP (less common), Encrypted Zip (less secure due to shared passwords and leakage of filenames in zip files)
- Schedule debriefing calls

The planning and pre-engagement phase usually concludes with the kickoff call in which all the testers (or at least test leads) and the target personnel discuss any final planning and details. The call usually reminds everyone of the scope and rules of engagement as a final discussion of any potential issues.

During this call (or in a follow-up email), the testers and the target organization will exchange contact information. During a test, the testing team may crash a target; discover an urgent, high-risk vulnerability; or find evidence of a previous intrusion. In such cases, the team may immediately need to contact personnel in the target company to report the issue.

Sometimes the target company personnel needs to reach the testing team to verify that an observed attack is coming from the testers. What if an evil attacker starts hitting the target at the same time that the penetration testers begin their work? The target organization needs to be able to contact the testing team 24/7 during the duration of the test. Therefore, both sides need to be available around the clock during the test.

After you identify points of contact, be sure to agree upon a secured method of communication regarding vulnerability details and the final report. The penetration testing team will be handling some sensitive data, which you may need to email to the points of contact among target system personnel. You do not want to send this information in cleartext. Instead, choose a suitable encryption solution. Many times, teams will use encrypted email. Increasingly, teams are moving to security content management technology (such as Box).

The free, open-source Gnu Privacy Guard or commercial PGP is an acceptable solution here. However, due to the additional technical requirements and the waning usage, this is less often used these days. If you do choose to use PGP/GPG, make sure to exchange public keys and verify their fingerprints. Encrypted ZIP files are an option but are less secure because the shared password used to protect the symmetric key in the ZIP file could come under attack. A malicious actor could steal the shared password, or it could leak if it is shared among too many people. For example, sometimes target system personnel may be tempted to send the password for the ZIP file in email, potentially exposing it. The more common issue with ZIP files is that many email servers will not allow attachments of this type.

Another element that we find useful to define in our Rules of Engagement is to require a daily debriefing conference call. At the beginning or end of each testing day, schedule a brief session between the testing team and

one or two representatives of the target organization. These calls help to make sure that everyone understands the progress of the test and to identify any issues early on.

During the debriefing, the team should discuss what the team has done so far and what they plan to do next. In addition, any major findings can be reviewed. Finally, the target organization can confirm whether its detection mechanisms (IDS, IPS, log review, and so on) have been triggered by the test.

If a daily call is too onerous given the busy schedules of target environment personnel, consider conducting a debriefing call twice per week during the duration of the test.

Finalizing Pen Test Planning

Rules of Engagement

- You should agree on all these issues before you start
- Document your agreements and have everyone sign off
  - Testing dates
  - Target organization
  - Head of the test team
  - Possibly the individual testers
- Armed with a well-defined project goals, scope, and rules of engagement, your penetration tests will be more thorough and valuable

To conclude this segment of the course, make sure you have a solid, signed set of Rules of Engagement before you embark on any penetration test or ethical hacking project. You need to make decisions about these crucial issues in advance. If you do, you'll have a high-value ethical hacking and penetration testing experience.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

Because establishing firm Rules of Engagement and scoping a project properly are so important, we will now conduct a lab on the topic. In this lab, class attendees will formulate questions and responses associated with a penetration testing Request for Proposal (RFP). The goal of this lab is to help testers get a feel for the kinds of questions and answers that should be asked to scope a project and to determine the Rules of Engagement, sharing information about risks and benefits between clients and testers.

Please work on below exercise.
Lab 1.2: Scope and RoE Role Play

Please go to Lab 1.2: Scope and RoE Role Play in the SEC560 Workbook.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

The next portion of the course deals with the first phase of the test: Reconnaissance. During this phase, the tester learns as much as possible about the target organization. One important aspect of this recon phase is building an inventory of potential target machines that are likely associated with the target organization. This inventory must be vetted carefully to ensure that it is indeed in scope before any scanning activities (the next phase of the process after reconnaissance) can begin.

When planning a test, we recommend that you budget and schedule at least 8 to 10 hours of detailed reconnaissance work or more if resources allow. Don't skip recon. It can provide crucial insights for the remainder of the entire test.

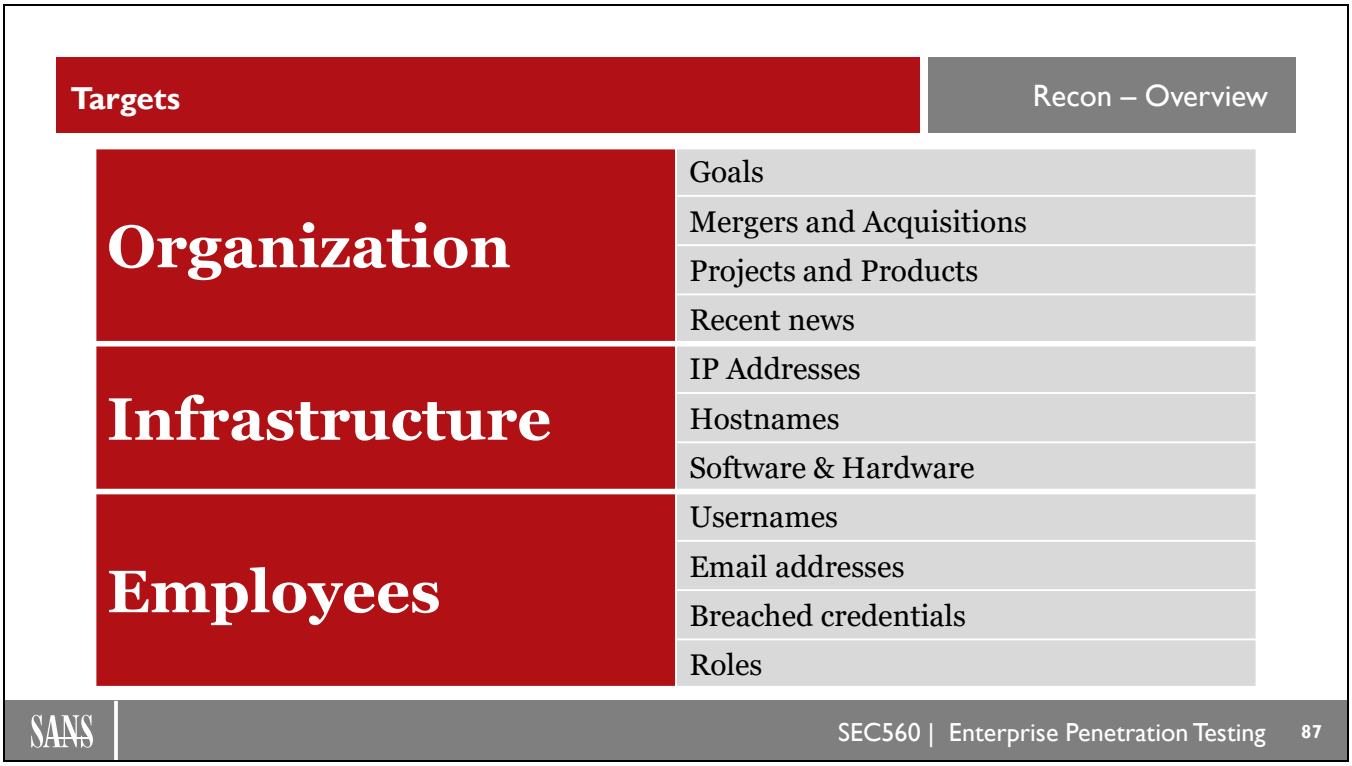
Recon helps us make intelligent targeting and attack decisions

- What do we know about the user or organization that will increase our likelihood of success of a phish or SE call?
- Knowledge of the username format and a list of users will make guessing more effective and efficient
- What do we know about the hardware and software that could make lateral movement easier?

We use the data found during the recon phase to help use make better targeting and attack decisions. If we are going to do phishing, we need to research the users in the organization so we can develop a phish that is likely to be opened. Recon helps us decide on the user and select a ruse that will increase our likelihood of success.

Another common recon goal is to discover the format for usernames. This will aid in developing a list for wide-scale phishing attacks, since the first portion of the email address is often the same as the username. Knowledge of the format also means password attacks are more effective. Using the wrong username format could mean that all the password guessing attacks are wasted effort

Knowledge of the internal infrastructure can help us save time by informing testers of the systems and software (and ports) to look for. For example, if you know the organization uses Microsoft SQL Server, you can save time (and keep a lower profile) by only looking for TCP/1433 instead of this and all the other common database ports.



When it comes to the data we are after, it can be grouped into three categories: organization, infrastructure, and employee data.

Organizational information is not going to to be technical details, but it can still be very helpful. This information is tremendously useful for phishing attacks as it allows us to develop relevant emails in hopes of getting users to open the email. It can also be useful for targeting and goal setting. For example, if the organization has a press release on a new development or technological breakthrough, we can use that as a phishing pretext; and, once inside, we can target our pillaging and pivoting towards the sensitive data.

Information on the target infrastructure helps us target the right software through client-side attacks as well as network-based attacks. If we can learn the AV/EDR in use, we can tailor our payloads and actions to avoid detection. We may even learn about network topology and architecture, which is very useful when pivoting inside the organization. This information is also valuable for social engineering (phone and email), if you are impersonating IT or pretending to be a user calling the help desk.

User information is the often the most directly useful information. If we, the testers, are lucky (and the target isn't) then we may be able to find breached credentials for users at the target organization and immediately login. If if the credentials don't immediately work, we know the types of passwords used by people in the organization. We can also learn about the hierarchy and organizational structure which can help with phishing or social engineering as well as "admin hunting".

Types of Shared Data

Intentional Sharing

- URLs and websites
- Project names
- Annual reports
- Press releases
- Job requirements

Unintentional Sharing

- Account information in third-party breaches
- Employees on social media
- File metadata
- Server banners

Nearly every modern organization has a website with publicly available content that is intentionally shared with customers and clients. The website and the linked pages are intended to be shared with the world. On the website is often information about upcoming projects and promotions. Depending on the size of the company, they may publish annual reports and press releases that offer insight into various aspects of the company. Some organizations have a jobs/careers portion of their site that describes the job and candidate requirements, which may include details about the organization's hierarchy, people, and technology.

Not all publicly available information is intentionally shared. It could be that a sensitive document is accidentally posted on the website. In addition, users will sometimes discuss sensitive information about their company on social media or public forums. Data may even be dumped online as part of a breach.

Servers will often announce the software they are running. For example, web servers will, by default, include the "Server" header that describes the server version. Web servers may send other headers to the client but reveal information about the servers as shown below:

```
Server: Apache/2.2.15 (CentOS)

X-Powered-By: PHP/5.6.29
```

The above headers are not useful to the client (browser). Other protocols will reveal similar data about the server that is not useful to the client.

Many file types include metadata that include information about the user, the user's computer and software, and other details that go beyond the standard file contents. As we will see, that data can reveal usernames and file locations.

Dedicate some of your pen testing time for reconnaissance

- "This is a <*compliance*> test, we don't need recon"
  - Some budgets won't allow for testing time
  - Half day?
  - Borrow some time from the rest of the test?
- Junior testers and interns can be a valuable asset
- Recon is less valuable on internal (only) tests
- Tests with a social engineering component need more time
- Red Teams need even more time

Ideally, we should dedicate some time for reconnaissance during our testing. If the target pushes back on spending time performing reconnaissance, we could cut back the recon portion of the test or borrow time from the rest of the test. While performing good OSINT is a valuable skill to develop, you can often get good results from junior testers and interns. In a compliance-based test (i.e., internal PCI) you could skip the recon phase, but for tests that include social engineering, this phase is a must. If the engagement is a full Red Team, even more recon time is necessary.

There is little to no traffic sent to the target, this is not a scan

- **Zero touch:** Ideally, you send no packets to the target
  - Find data from third parties
- **Light touch:** If you do interact with the target, it should be (mostly) normal, light traffic
  - Browse the target's website
  - DNS lookups

If stealth is key (e.g., red team), recon and testing infrastructure should be separated

In the reconnaissance phase, you are sending little to no traffic to the target network. Ideally, it is zero-touch, meaning no data is sent from your systems to the target. In this case, all data is gathered from third parties, or third parties are used to access the targets. Only rarely can we get all the information we need without interacting with the target or will be prohibited from accessing the target as a normal user.

If you interact with the target systems directly, make sure you do so in a way that looks like normal traffic (no scanning, no attacking). If you browse the target website just as a normal user would, then it will be impossible to detect, and your testing systems will not be detected. This is especially important in Red Team engagements where stealth is paramount. If you need stealth, we recommend using dedicated recon systems/IP addresses that are not used for other portions of the assessment.

Ethics: Do not use fear, anger, or extreme emotions

- Be respectful of your target, especially the people
- Remember the goal of testing the organization's security and processes
- Example bad pretexts: Car accident, sick/injured child, threatened job
- The malicious actors do not have rules, but we are better than they are
- Social Engineering Code of Ethics
  https://www.social-engineer.org/framework/general-discussion/social-engineering-code-of-ethics/
  Short link: redsiege.com/560/se-ethics

Penetration testers aim to mimic and model real-world adversaries, but there are limits. We should always be respectful even though we are attempting to trick users. This can be an ethical, moral, and/or spiritual dilemma, and some penetration testers will not take part in such activities, even while others pen testers see no such issue. We do not intend to resolve the conflicts here, but we do want to address that these debates do exist.

The evil attackers are already breaking the law and may use tactics and techniques that are ethically, morally, and legally off limits. We need to act better than the evil attackers and operate within a safe environment. Our end goal is to help the organization determine its weaknesses. The organization and its people should be better off after working with us.

We should not put undue stress on the targets of phishing or social engineering tests. As such, pretexts that can cause extreme emotion should be avoided. Example bad pretexts include using someone being injured in a car accident, anything involving children, or direct threats against someone's job. We highly recommend reading the Social Engineering Code of Ethics before you do any social engineering or phishing:

References:
https://www.social-engineer.org/framework/general-discussion/code-of-ethics/
Shortlink: redsiege.com/560/se-ethics

Targeting Caveat

We will be using live organizations in this section

- You are not authorized to attack the mentioned orgs
- Never exceed "normal" traffic
- Slides/book, lab, and the live network will likely differ
- This is research into publicly available information

DO NOT ATTACK! You are just examining freely available data, not attacking!

Throughout this section, we will be referring to live networks, systems, companies, and targets.

You are not authorized to attack any of the third parties mentioned in this section or in the course in general. Please to not misconstrue this or other slides as license to attack any organization that you do not have permission to attack. The examples here use data that is exposed on the internet.

The organizations and systems mentioned in this portion of the book are from live production systems. As such, they may change at any time and may not reflect what is in the book.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

The first step in our reconnaissance is to learn about the target organization. Later, we'll investigate their technology and people.

Information on the Organization

- High-level information about the target
- What are their goals and how can we use them in our testing?
- The data is there, figuring out how to use it is the fun part!
- Think like an attacker

At this point, we need to find out information about the target that can give us a better understanding of their goals, processes, and initiatives, which, in turn, helps us better understand the business risks. Armed with this knowledge, we can write a more effective executive summary by more accurately describing the risk and impact in their terms. This context also allows us to write findings that more accurately describe the potential organizational risk and impact.

This is also the beginning of the fun! It is time to start thinking like an attacker! We can look at this information to help us better target the organization from a social engineering or phishing perspective. When you start looking at this information, start thinking of ways an evil attacker would or could use it against them.

Where to Start?

Recon - Organization

- Start with a basic web search on the target
- Look for company biography and history
- List of domains (may be limited)
- Affiliations, mergers, and acquisitions
  - Use the list of companies to find new domains both internally and externally

A simple starting point is a search engine. This can give you some basic information on the organization as well as a history that may not be on the target's website. You may find domains, but this list is likely to be quite limited. You may find sub-companies or affiliations that could lead you to other domains. These affiliations may also be easy sources for phishing or social-engineering pretexts.

Maybe the company purchased or merged with another organization. It could be that domains associated with the old company are available for purchase and are still trusted by mail servers or other systems.

Press Releases and Annual Reports

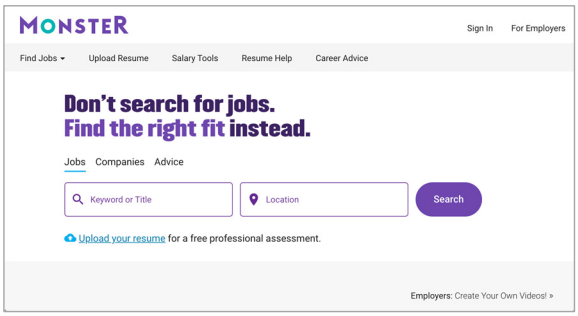
- What has the company done recently?
- Often useful as a social engineering pretext
- Example:
  - Organization receives an award
  - Send employees an invite to a banquet and ask them to fill out an RSVP (with HTA attachment)

This page intentionally left blank.

Look for Open Job Requisitions

Recon - Organization

- Job requisitions can help us get information about the information technology products used in a target organization, such as:
  - Web server type
  - Web application dev environment
  - Firewall type
  - Routers
- Google searches to find job reqs
  - site:[companydomain] careers
  - site:[companydomain] jobs
  - site:[companydomain] openings
- Also, searches of job-related sites
  - www.monster.com: Search on Info Tech and Internet/E-commerce



Most organizations have job requisition information available on the internet as they look to hire new staff. These job requests often contain detailed information about the technical environment of the enterprise. For example, if the target organization is looking for IIS administrators, we now know something about the web servers it uses. If it seeks skilled Checkpoint firewall admins, we have information about at least some of its firewalls. If it is looking for developers with Cold Fusion experience, we now know a little more about some of its web applications. What's more, if the job req is still active, we know that the target organization does not have enough experienced staff members to handle that part of its infrastructure. After all, if it did have the expertise already in-house—why would it be seeking to hire people with those skills?

To search for job requisitions, you could use Google with the "site:" directive focused on the target's domain, followed by common terms used on pages for hiring. We recommend searches like the following:

```
site:[companydomain] careers
site:[companydomain] jobs
site:[companydomain] openings
```

You can narrow down your results further by inserting terms such as Information Technology, internet, e-commerce, firewall, and so forth.

In addition, a thorough tester should look for job reqs on various job-hunting sites, such as Monster.com. On Monster.com, the Information Technology, Internet/E-commerce, and Telecomm categories are helpful. You can even narrow down your searches based on geographic areas.

Gather Competitive Intelligence

Recon - Organization

- Using search engines, determine the target organization's:
  - Major businesses
  - Major products or services
  - Corporate officers and other VIPs
  - Major competitors
  - Physical locations
  - Recent press releases

This information can be used for building a solid pretext for phishing or voice calls

As a start of the recon phase, the tester can use a search engine, such as Google, to learn more about the target organization. In particular, we recommend conducting searches on the target organization's name to gather the following information, which should be recorded in the tester's results:

- **Major businesses:** What is the industry or industries associated with the target? Financial services? Government agency? Manufacturing?
- **Major products or services:** What does the target organization produce? What are the brand names of its products or services?
- **Corporate officers and other VIPs:** Who is most important in the target organization? Who are its leaders? Who is associated with its technical infrastructure?
- **Major competitors:** Who competes with the target organization? What is the target organization's relative performance vis-à-vis its competitors? Is it the market leader?
- **Physical locations:** Where are the major facilities of the target organization?
- **Recent press releases:** What has the target enterprise told the public lately about itself? What does it consider important from an image and marketing perspective?

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

This page intentionally left blank.

Please work on below exercise.
Lab 1.3: Organizational Reconnaissance

This Please go to Lab 1.3: Organizational Reconnaissance in the SEC560 Workbook.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

Let's now look at the target's infrastructure. This will help us identify targets and more efficiently attack them.

Learn about the live systems, including software and hardware

- Find IP addresses and subnet ranges
- DNS and host names – these are a must for web attacks
- Listening ports and services
- Determine software and hardware in use

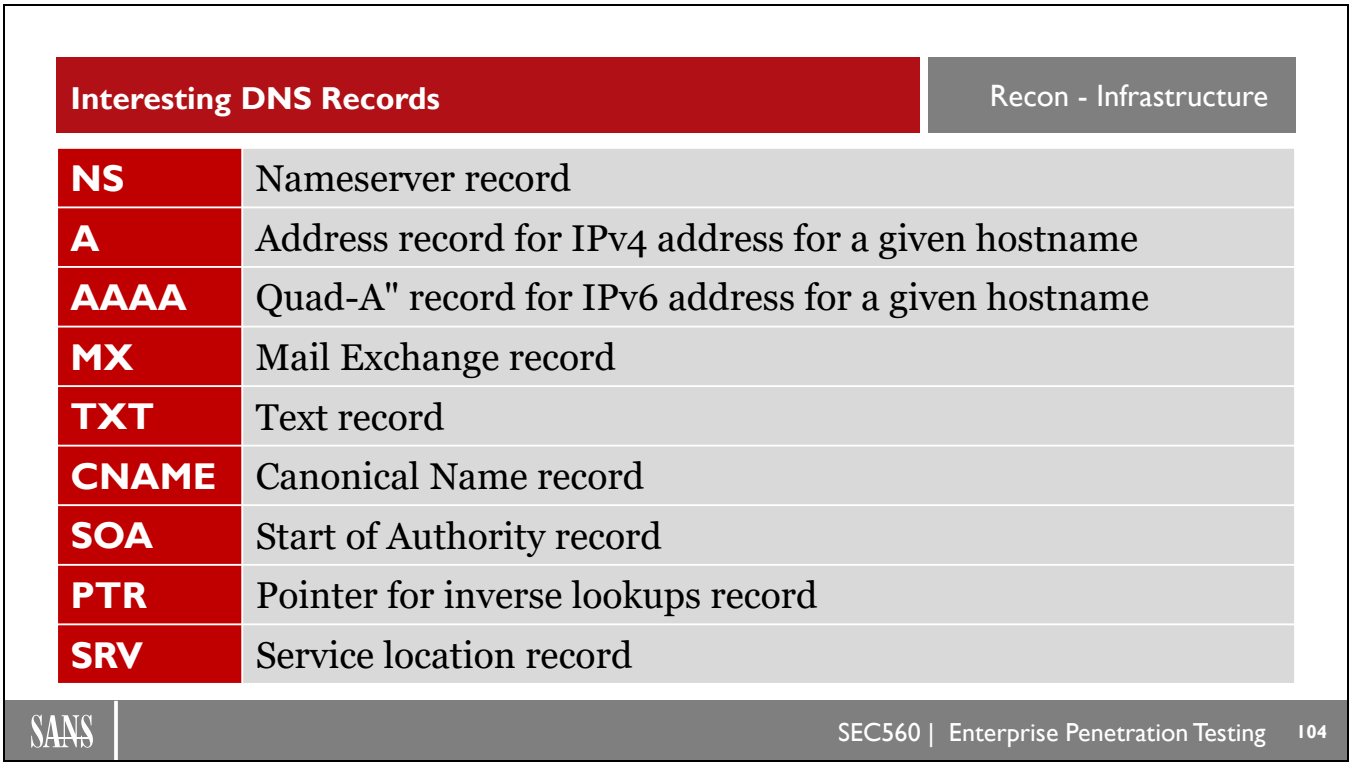
This page intentionally left blank.

Hostname Information

Recon - Infrastructure

- Hostnames often indicate their purpose
- For password spraying, look hostnames containing the following:
  - VPN sign-on portals: `vpn`, `access`
  - Citrix StoreFront portals: `ctx`, `citrix`, `storefront`
  - Online email: `mail`, `autodiscover`, `owa`
  - Hostnames containing `login`, `portal`, `sso`, `adfs`, or `remote` are also good targets

Once we've compiled a list of hostnames, we can start prioritizing which hosts to focus our efforts towards in later steps of the attack chain. The first type of hostnames to look for would be those containing 'vpn'. If we successfully guess a user's credentials and the VPN service is not protected by multifactor authentication, we have made it into the internal network in one shot. Another service to look for is external Citrix services. Citrix services are used to host VPN services, remote virtual desktop services, and access to single apps. Successfully password spraying a Citrix service can give access to valid internal credentials, VPN network access and/or access to a live domain-joined host inside the network, and potential for exercising Citrix environment breakouts when attacking a hosted application. A successful breakout can give us access to high-level credentials for the internal network through the underlying Citrix host. Next, we can focus on email services. Very commonly, we'll see hostnames containing 'autodiscover', 'owa', or 'mail' that led to on-premises Microsoft Exchange services. Password spraying against the auto-discover and exchange web services (EWS) endpoints of an Outlook Web Access (OWA) service can lead to gaining internal domain credentials that will work for other external services. Finally, we look for hosts that indicate use of organizational credentials such as 'login', 'portal', 'sso', 'adfs', or 'remote'. Single Sign-On (SSO) and Active Directory Federated Services (ADFS) are goldmines for access since they both allow signing in once to access multiple services. Login, portal, and remote are also worth investigating because they can indicate VPN access or other organizational services.



The last elements of the WHOIS record include the Domain Name System (DNS) servers associated with the target organization, listed in the order of primary, secondary, and tertiary (if it exists) DNS servers. We will next try to harvest records from those name servers.

Name servers are focused on resolving domain names into IP addresses, but that isn't their sole function. They also indicate which machines are mail servers for a given domain, among other useful information. DNS servers house a variety of different records, including:

- **NS:** Nameserver record, which indicates the name servers associated with a given domain.
- **A:** Address record, which maps a domain name into an IPv4 address.
- **AAAA:** "Quad-A" record, which maps a domain name into an IPv6 address.
- **HINFO:** Host Information record, which associates an arbitrary set of information with a domain name, formerly used to indicate system types.
- **MX:** Mail Exchange record, which identifies the mail servers for the given domain.
- **TXT:** Text record, which includes an arbitrary text string for the domain.
- **CNAME:** Canonical Name record, which indicates aliases and alternative names for a given host.
- **SOA:** Start of Authority record, which indicates that a server is authoritative for that DNS zone (set of records).
- **RP:** Responsible Person records, which are informational, not functional (that is, they have no impact on DNS functionality) and indicate the human responsible for a given domain (seldom used).
- **PTR:** Pointer for inverse lookups records, also called reverse records, indicating an IP address to domain name mapping.
- **SRV:** Service location records, which provide information about available services, including the port and hostname (seldom used).

The Dig Command

Recon - Infrastructure

- The dig command in most Linux can perform zone transfers

```
$ dig @[server] [name] [type]
```

- The type can be ANY, A, MX, and so on; the default is A records
- With a -t flag, we can specify zone transfer

```
$ dig @1.2.3.4 mydomain.com -t AXFR
```

- Use **+norecursive** or **+recursive** (default) to toggle recursion
- Simplify output with **+noall +answer**

In many recent Linux and UNIX systems, the nslookup command has been altered so that it can no longer perform zone transfers. On these systems, we can use the dig command for various kinds of DNS research, including zone transfers.

The dig command has the following syntax:

```
$ dig @[server] [name] [type]
```

The types we can specify include the abbreviations listed earlier, including A, MX, SOA, and such. To receive all kinds of records, we use the ANY type. If no type is specified, dig defaults to A (Address) records.

To get dig to perform a zone transfer, we invoke it with the -t AXFR notation as

```
$ dig @[server] [domain] -t AXFR
```

This syntax will pull all information about a given domain. Alternatively, dig can perform an incremental zone transfer, pulling only recently updated records, using this syntax:

```
$ dig @[server] [domain] -t IXFR=[N]
```

N is an integer that refers to the serial number of a Start of Authority record. The incremental zone transfer request will pull all records that have changed since the SOA serial number was the N we specified in our dig request.

Dig also supports turning off or on the Recursion Desired (RD), with the +norecursive or +recursive syntax. By default, dig performs recursive searches.

DNSRecon

Recon - Infrastructure

- Multi-threaded DNS recon tool by Carlos Perez (@darkoperator)
  - Available at https://www.github.com/darkoperator/dnsrecon

```
dnsrecon -d domain.tld -t type
```

- Enumeration includes standard DNS records (default), reverse IP address lookup (`rvl`), zone transfers (`axfr`), DNSSEC zone walks (`zonewalk`), and cache snooping (`snoop`)
- Dictionary-based subdomain brute forcing
- Output can be in XML (`--xml`) or SQLite database formats (`--db`) for use with other tools

DNSRecon is a Python3 tool for performing multi-threaded DNS reconnaissance. Since it is developed with Python3, DNSRecon can run on any OS that supports Python. DNSRecon includes several tools for reconnaissance:

- **dnsrecon -d [domain]** – Displays SOA, NS, A, AAAA, MX, and SRV of the target domain
- **dnsrecon -d [domain] -t rvl** – Performs reverse DNS lookup for IP address or CIDR range
- **dnsrecon -d [domain] -t axfr** – Attempts a zone transfer of all NS record nameservers
- **dnsrecon -d [domain] -t zonewalk** – Performs a DNSSEC zone walk by querying for NSEC records
- **dnsrecon -d [domain] -t snoop -D [dictionary file]** – Scans for DNS cache snooping using a supplied dictionary file

DNSRecon can also perform subdomain brute forcing with a dictionary using the following command:

- **dnsrecon -d [domain] -t brt -D [dictionary file]**

Finally, DNSRecon can output the returned data to an XML file using the **--xml [output file]** flag or to an SQLite database using the **--db [output file]** flag

DNSRecon Usage


```
sec560@slingshot:~$ dnsrecon -d sans.org -n 8.8.8.8
[*] Performing General Enumeration of Domain: sans.org
[-] DNSSEC is not configured for sans.org
[*]     SOA dns21a.sans.org 66.35.59.7
[*]     NS dns31b.sans.org 204.51.94.8
[*]     Bind Version for 204.51.94.8 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]     NS dns21a.sans.org 66.35.59.7
[*]     Bind Version for 66.35.59.7 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]     NS dns21b.sans.org 66.35.59.8
[*]     Bind Version for 66.35.59.8 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]     NS dns31a.sans.org 204.51.94.7
[*]     Bind Version for 204.51.94.7 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]     MX sans-org.mail.protection.outlook.com 104.47.44.36
[*]     MX sans-org.mail.protection.outlook.com 104.47.73.10
[*]     A sans.org 45.60.31.34
[*]     A sans.org 45.60.103.34
```


Shown here is sample output from when we scan sans.org with DNSRecon. First, we see that sans.org is not using DNSSEC. Next, we see the nameserver (NS) records listed for sans.org including the Start of Authority (SOA) record. Following those, we have the mail exchange (MX) records indicating the use of Office365. Finally, we have the text (TXT) records showing different services that SANS is using such as Office365, ExactTarget, SalesForce, and ClearSlide.

```
sec560@slingshot:~$ dnsrecon -d sans.org -n 8.8.8.8
[*] Performing General Enumeration of Domain: sans.org
[-] DNSSEC is not configured for sans.org
[*]    SOA dns21a.sans.org 66.35.59.7
[*]    NS dns31b.sans.org 204.51.94.8
[*]    Bind Version for 204.51.94.8 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]    NS dns21a.sans.org 66.35.59.7
[*]    Bind Version for 66.35.59.7 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]    NS dns21b.sans.org 66.35.59.8
[*]    Bind Version for 66.35.59.8 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]    NS dns31a.sans.org 204.51.94.7
[*]    Bind Version for 204.51.94.7 9.3.6-P1-RedHat-9.3.6-25.P1.el5_11.12
[*]    MX sans-org.mail.protection.outlook.com 104.47.44.36
[*]    MX sans-org.mail.protection.outlook.com 104.47.73.10
[*]    A sans.org 45.60.31.34
[*]    A sans.org 45.60.103.34
[*]    TXT sans.org _globalsign-domain-verification=XbqPoFvyLnW1lHWyrKazU_F…
[*]    TXT sans.org _globalsign-domain-verification=Z0fOVJB0oLvstFlL9BBVBnL…
[*]    TXT sans.org v=spf1 mx ip4:66.35.59.0/24 ip4:66.35.60.0/24 …
```

Output truncated for brevity

DNSDumpster (1)

Recon - Infrastructure

- Provides a list of DNS A records for a given domain
  - Free version provides up to 100 A records.
  - The paid version of dnsdumpster at hackertarget.com provides the full list as well as additional services
- MX and TXT records disclose cloud email services and spam/malware filters
- Autonomous System Numbers (ASNs) can have the target's name
  - ASNs with the target's name provide proof of in-scope hosts
  - Can lead to additional domain name discovery
- DNSDumpster is located at https://dnsdumpster.com

DNSDumpster provides a wealth of information about a target domain. A list of subdomains will be shown with the corresponding IP address and Autonomous System Numbers (ASNs). Additional port scan information can be shown but it is usually very light. The MX and TXT records will show if the target is hosting their email on premise or in a cloud service. Spam and malware filters can give away key information such as product platforms that the target is using. For example, finding Cisco VPN endpoints in the subdomains and Cisco spam and malware filters on their email give good indications that their entire infrastructure is built on Cisco. Finally, finding ASNs with the target's name is a good indicator of additional hosts and IP address ranges that should be in scope. While researching live hosts in the ASNs, additional associated domains can be found leading to further recon and attack surface.

DNSDumpster is located at https://dnsdumpster.com.

DNSDumpster (2)

Recon - Infrastructure

Here we see DNSDumpster listing A records for sans.org. On the left, the domain and subdomain are listed. Below each domain name (listed from left to right), there are options to see:

- Domain names sharing IP addresses with the subdomain: This can help find additional domains associated with the target.
- Show the HTTP headers observed during interactions with the web service: This can help enumerate server software in use.
- Traceroute to the host: This can help identify hosting infrastructure.
- Discover hosts in the identified netblock of the subdomain: This will show the other hosts found in the netblock or ASN discovered by DNSDumpster.
- Perform a quick Nmap scan: This scans the 10 most common TCP ports. Full Nmap scans can be performed from the paid hackertarget.com service.

In the center, we see the IP address and DNS lookup information for the IP. On the right, we can see the detected ASN for the IP address.

DNSDumpster (3)

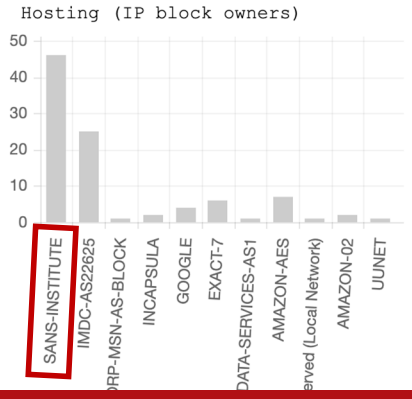
Recon - Infrastructure

MX Records ** This is where email for the domain goes...

0 sans-org.mail.protection.outlook.com.	104.47.73.10 mail-dm6nam080010.inbound.protection.outlook.com	MICRC CORP- AS-BI Unite State

TXT Records ** Find more hosts in Sender Policy Framework (SPF) configurations

"v=spf1 mx ip4:66.35.59.0/24 ip4:66.35.60.0/24 ip4:204.51.94.0/24 ip4:160.109.23 ip4:23.253.9.220 ip4:23.253.9.221 ip4:104.130.85.7" " ip4:108.171.167.255 ip4:161.47.83.173 ip4:162.209.38.195" " ip4:162.242.176.254 ip4:166.78.198.138 ip4:184.106.37.245" " include:amazonses.com include:stspg-customer.com include:c spf.exacttarget.com" " include:spf.protection.outlook.com include:spf.clearslide include:_spf.salesforce.com ~all"



Email Hosted in Office 365 (Cloud Attacks)

Block owner containing "SANS" (possible more targets)

On the left, DNSDumpster is displaying the MX and TXT records. According to the MX records, the target is using Office365 for email. We can use this information during social engineering attacks as well as user enumeration during later stages of recon. In the TXT records, we can see different methods of verifying email authenticity from sans.org through the Sender Policy Framework (SPF) declarations. They also give us an idea of what kinds of software platforms that the target is using such as (again) Office365, ExactTarget, Clearslide, and SalesForce. On the right, DNSDumpster displays a breakdown of the ASNs associated with the discovered subdomains. We can see that SANS has their own ASN which gives us further IP address ranges and netblocks to explore for additional attack surface.

IP Address Assignment WHOIS Databases

Recon - Infrastructure

Query the registries for IP ranges to find additional targets

- Regional Internet Registries (RIRs) offer Whois databases that store information about IP address block assignments
- Provide a company or domain name, and they tell you if there is an address range officially assigned to it
  - IPv4 and IPv6 address assignment and CIDR block
  - Autonomous System (AS) number assignment
  - DNS information
- Many orgs get addresses from their ISP (not self owned)
- Results may vary. You may get:
  - Actual addresses assignment
  - Nothing at all
  - A huge address space, far bigger than that allotted to this one organization (you are likely seeing whole ISP)

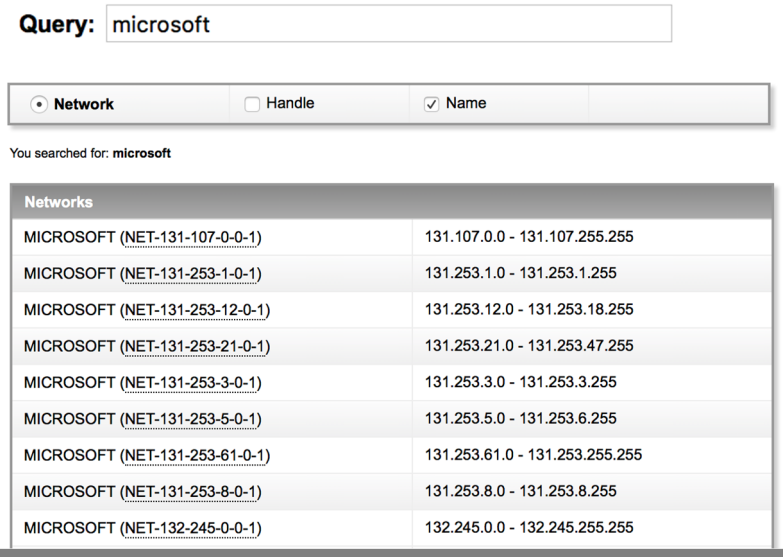
Another important element of reconnaissance involves determining the IP address blocks that are assigned to the target organization. There are several Regional Internet Registries (RIRs) that store this information in SHOIS databases. By surfing to the appropriate website, a user can provide a company name or domain name and retrieve official address assignments, including IPv4 and IPv6 addresses. Most records also include the CIDR (Classless Inter-Domain Routing) block, telling us the size of the target network. The American Registry for Internet Numbers (ARIN) covers North America, including the United States, Canada, and certain Caribbean islands. The Réseaux IP Européens Network Coordination Centre (RIPE NCC) is the RIR for Europe, the Middle East, and parts of Central Asia. The Asia Pacific Network Information Centre (APNIC) covers the Asia-Pacific region. The Latin American and Caribbean Internet Address Registry (LACNIC) encompasses Latin America and most of the Caribbean. AfriNIC covers the continent of Africa.

Also, these databases provide autonomous system (AS) numbers, sometimes known as ASNs. An AS is a collection of IP networks and their associated routers under the control of a single technical administrator, such as an ISP or enterprise, that has a common routing policy with respect to the internet. The AS will have its own internal routing policy but presents a separate routing policy to the internet, which moves packets between various autonomous systems using a routing protocol, like the Border Gateway Protocol (BGP). Each AS is assigned a unique ASN, which is stored in the Regional Internet Registries. These databases also store DNS information.

Not all organizations have an IP address block assigned to them. Some get IP addresses from their ISP. When searching a Regional Internet Registry for information, we may therefore not get exactly what we are looking for. When searching for an IP address block for a company, for example, we may get the actual results for that company, certainly a good thing. Alternatively, we may get nothing at all, implying that the given enterprise gets all its IP address space from its ISP. Furthermore, we may get a giant block of addresses back that do not apply to only the organization we searched for but instead apply to its entire ISP. Thus, we have to be careful when targeting the results we receive from a Regional Internet Registry, verifying that they are actually within the scope of our test.

Sample ARIN Lookups: Network

Recon - Infrastructure



The screenshot on this slide shows a Network search associated with Microsoft. For the Network search, we received more results by searching for the name microsoft than we did searching for the domain microsoft.com. It is a good idea to do searches for both name and domain to help ensure you get the data you need for your Reconnaissance phase.

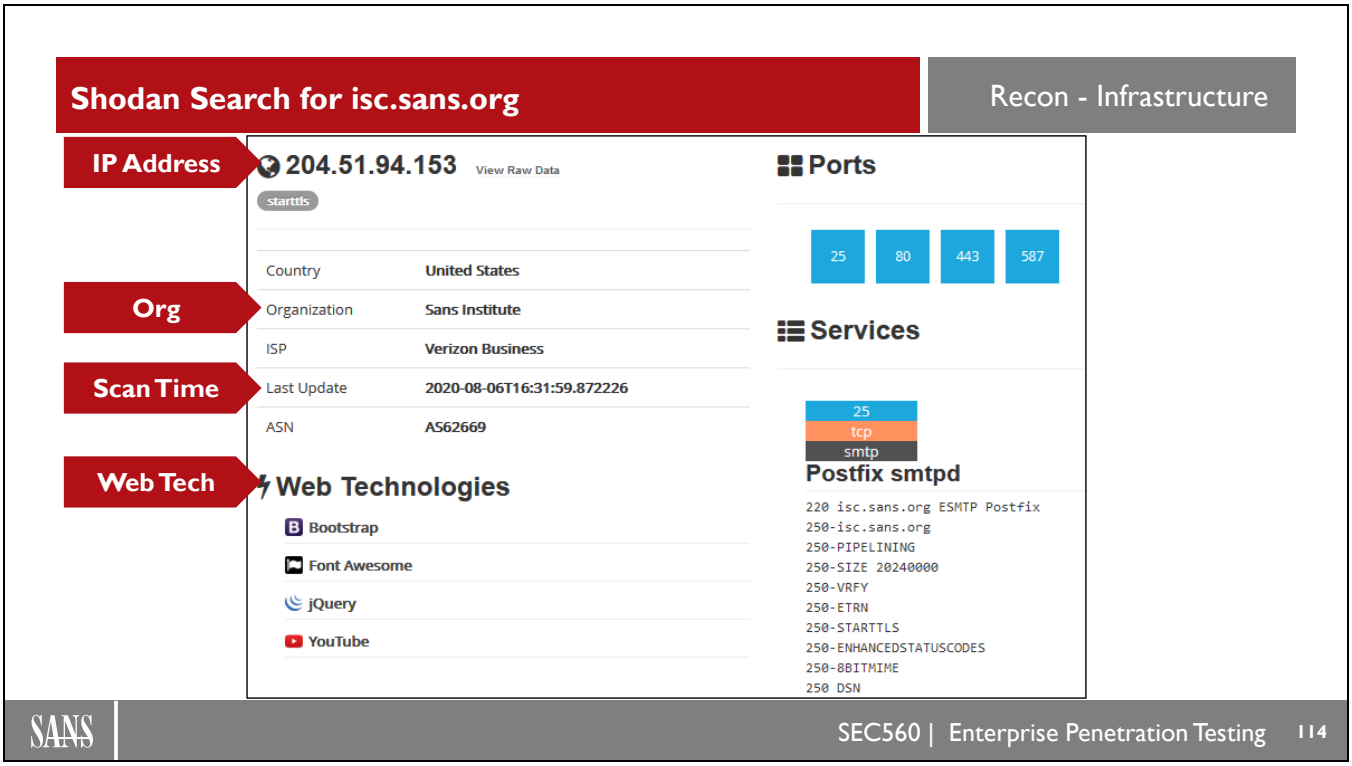
If we were targeting Microsoft, we have found IP addresses owned by them. It is highly likely that live systems in those ranges/blocks would be owned and operated by Microsoft. If we were performing an external penetration test, this would give us a good list of IP ranges in which we can find targets that are likely in scope.

Shodan

Recon - Infrastructure

- Regularly scans available services and ports on hosts connected to the internet
  - Port Scan results without accessing the target
- SSL Certificate Information
  - SSL certificate can reveal additional subdomains
  - Expired certificates can create social engineering scenarios
- IP Address Geolocation
  - Helps with verifying in-scope hosts when dealing with netblocks

Shodan is a search engine for devices on the internet. Shodan regularly scans the internet for connected hosts and then gathers information about listening ports and services. Hosts can be searched for by a number of identifiers such as hostnames in SSL certificates, IP address/CIDR ranges, listening ports, and service banner messages. The output from interactions with each service is captured, giving additional information for later attacks. We can also find additional subdomains through the SSL certificates that were observed during interactions with secure services. If the SSL certificates are expired, social engineering scenarios can be crafted around a one-off domain. If the user is desensitized to the SSL certificate errors being presented during interactions with the service, the user may not notice that the domain name is slightly changed and can lead to successful attacks. Finally, Shodan provides geolocation data about the IP address of the host. This can help identify hosts that may be out of scope such as ISP infrastructure inside of a target ASN or netblock.



Here we see a screenshot of Shodan results for isc.sans.org. In the upper-left corner, we see the IP address, ISP that is hosting the device, ASN, and last time Shodan scanned the device. The lower-left corner shows web technologies in use on detected web services. On the right, the open ports are listed with the output from service enumeration for each port listed below.

BuiltWith (1)

Recon - Infrastructure

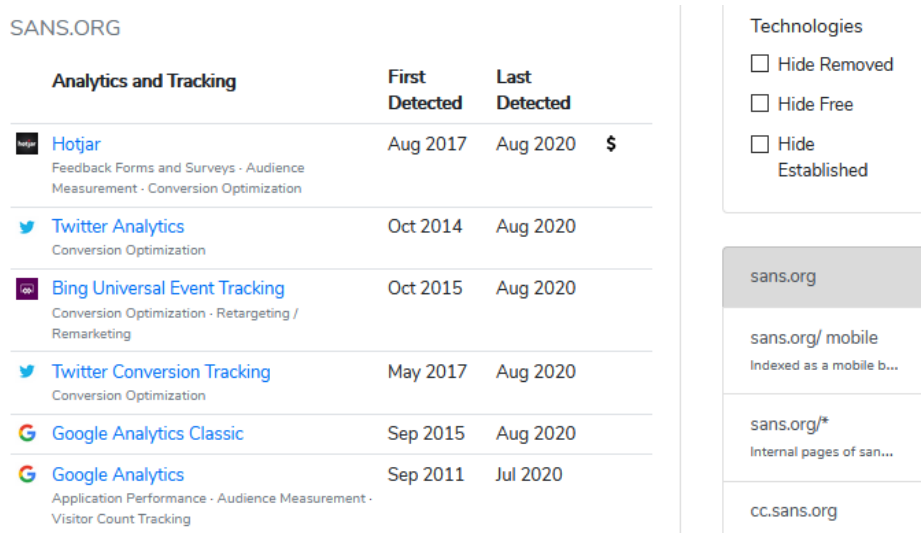
- BuiltWith.com compiles lists of technologies used in web server and software frameworks for target web services
  - List is broken down by the subdomain where the technology was observed, when the technology was first observed, and the last recorded time the technology was in use
- Maintains lists of related websites
  - Related site list contains domains directly related to the target domain
  - Also contains IP address history of each related domain

BuiltWith is a great website containing a large amount of data about a target's web presence on the internet. They compile lists of all the technologies in use for a target domain and each detected subdomain as well as breakdown the list into some of the following categories:

- Web Host
- Web Server
- Detected CDN
- Frameworks
- Widgets

On top of providing a list of subdomains, BuiltWith also provides related sites that are potentially owned by the same target. This can help with finding additional attack surface and additional domains for recon. Historical data about the IP address of the target and related domains is also kept. We can use this information to map out previous mergers and acquisitions, technology shifts between hosting and platforms, and potentially find forgotten hosts that are no longer in use but provide ingress vectors during later steps of the attack chain.

BuiltWith (2)



SANS.ORG

Analytics and Tracking	First Detected	Last Detected	
Hotjar Feedback Forms and Surveys · Audience Measurement · Conversion Optimization	Aug 2017	Aug 2020	$
Twitter Analytics Conversion Optimization	Oct 2014	Aug 2020	
Bing Universal Event Tracking Conversion Optimization · Retargeting / Remarketing	Oct 2015	Aug 2020	
Twitter Conversion Tracking Conversion Optimization	May 2017	Aug 2020	
Google Analytics Classic	Sep 2015	Aug 2020	
Google Analytics Application Performance · Audience Measurement · Visitor Count Tracking	Sep 2011	Jul 2020	

Here we can see a sample of the technology breakdown for sans.org, on the left, with links to each of the detected software platforms, the approximate date that the software was first detected, and an approximate date detailing the last time it was observed. This information can help us determine if research into each of the detected frameworks will be worth our time. On the right, a list of subdomains detected by BuiltWith can be seen.

Useful Google Search Directives: Sites and Links

`site:` – Searches only within the given domain

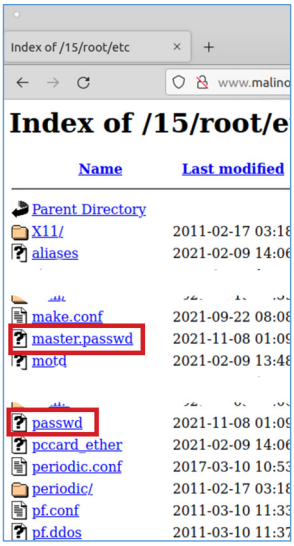
- Example: **`site:sans.org "web app"`**
- Find pages with the phrase "web app" that are on sans.org

`intitle:` – Page title matches search criteria

- Example: **`intitle:index.of passwd`**
- Finds indexed web directories with the word "passwd" in the directory listing, possibly an /etc/passwd file

`inurl:` – URL matches the search criteria

- Example: **`inurl:viewtopic.php`**
- Finds a script used in phpBB (a history of significant flaws)



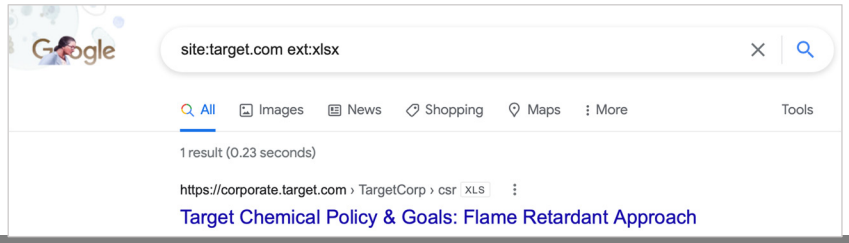
To understand how we can use search engines (and particularly Google) to find vulnerable systems, we are going to spend some time looking at more advanced search capabilities within Google. By taking the various search principles we'll discuss over the next several slides and combining them together in creative ways, we can use Google to find flawed systems in our target environments. Let's explore directives associated with examining specific websites and domains.

- **`site:`** – This 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 want to search pages only in the counterhack.net domain for the occurrence of the string "web app", you could do a search for site:**`sans.org "web app"`**. Have you ever seen a website that doesn't have its own built-in search capability (like sans.org) or one with a lame search capability? With Google's "site:" directive, you now can use Google to search for results associated with just that site, relying on the power of Google and its flexible search directives and operators. The "site:" directive can get specific, like **`site:example.exampleuniversity.edu`**, or broad, like **`site:.edu`**, which would cover all sites with a .edu suffix.
- **`intitle:`** – This directive searches for the criteria in the page title, which is displayed in the browser title bar or tab. One of the most useful title types to look for involves directory indexing on a website. On websites configured with directory index functionality, the web server will return to the browser the contents of that directory, with an autogenerated page with a title that includes the text "Index of". Thus, we can search for **`intitle:index.of passwd`** to look for directories that have a file in them called passwd. We may find an /etc/passwd file. Note that a number of the passwd files in interesting domains discoverable by Google searches are either honeypot /etc/passwd files or sites that may try to exploit your browser! So be careful when doing this kind of search, unless you restrict it using the "site:" directive to domains that are more trustworthy.
- **`inurl:`** – This directive lets us search for specific terms to be included in the URL of a given site. This can be helpful in finding well-known vulnerable scripts on web servers, including CGI, ASP, JSP, PHP, and others. For example, searching for **`inurl:viewtopic.php`** finds sites with URLs that contain "viewtopic.php" in them. That script is commonly associated with the phpBB suite of tools for implementing web-based discussion forums. Historically, there have been numerous flaws in phpBB implementations, so locating those sites is helpful if a new vulnerability is discovered.

Searching for File Types

Recon - Infrastructure

- Google identifies hundreds of different file types as it scours the internet, such as .pdf, .doc[x], .xls[x], .ppt[x], .cgi, .php, .asp, and many others
- "filetype:" and "ext:" directives search for only a specific kind of file
- Also, note that Google sometimes mistakes a given file type
- Combine with "site:" to restrict to your scope



Often, we want to search Google for some specific kinds of files. Most of the web consists of .html and .htm pages, but there are numerous other kinds of files that interest us. As Google scours the internet, finding new websites and adding their pages to its search directory, it recognizes several hundred different file types, allowing us to search for those types of files. For example, we can look for .pdf files. Or to make things more interesting, we can search for .xls files, which are commonly associated with Excel spreadsheets. We might also look for .ppt files to find PowerPoint presentations. Some organizations inadvertently put sensitive .xls or .ppt files on their websites, for which we could search.

To perform such searches, we have two options. The first is to rely on Google's "filetype:" directive followed by the suffix of the file we want to find. Note that the "ext:" directive does the exact same thing as the "filetype:" directive; it is exactly the same search. For example, we can search for PowerPoint files in the counterhack.net domain by looking for **`site:counterhack.net filetype:ppt`** or **`site:counterhack.net ext:ppt`**. Our second option is to look for the suffix of the file as a general search term. Sometimes Google gets confused about a file type, messing up the appropriate association and omitting that given file from the filetype: results. Using the file suffix as a general search term without the filetype: or ext: directive will usually give us more results because not only will we get files that have that suffix in their name, but we will also get webpages that merely include the text associated with that suffix. For example, if we search for **`site:counterhack.net ppt`**, we will not only get PowerPoint files, we will also get a series of webpages that include the text ppt.

Inventory of Discoverable Flaws via Google

Recon - Infrastructure

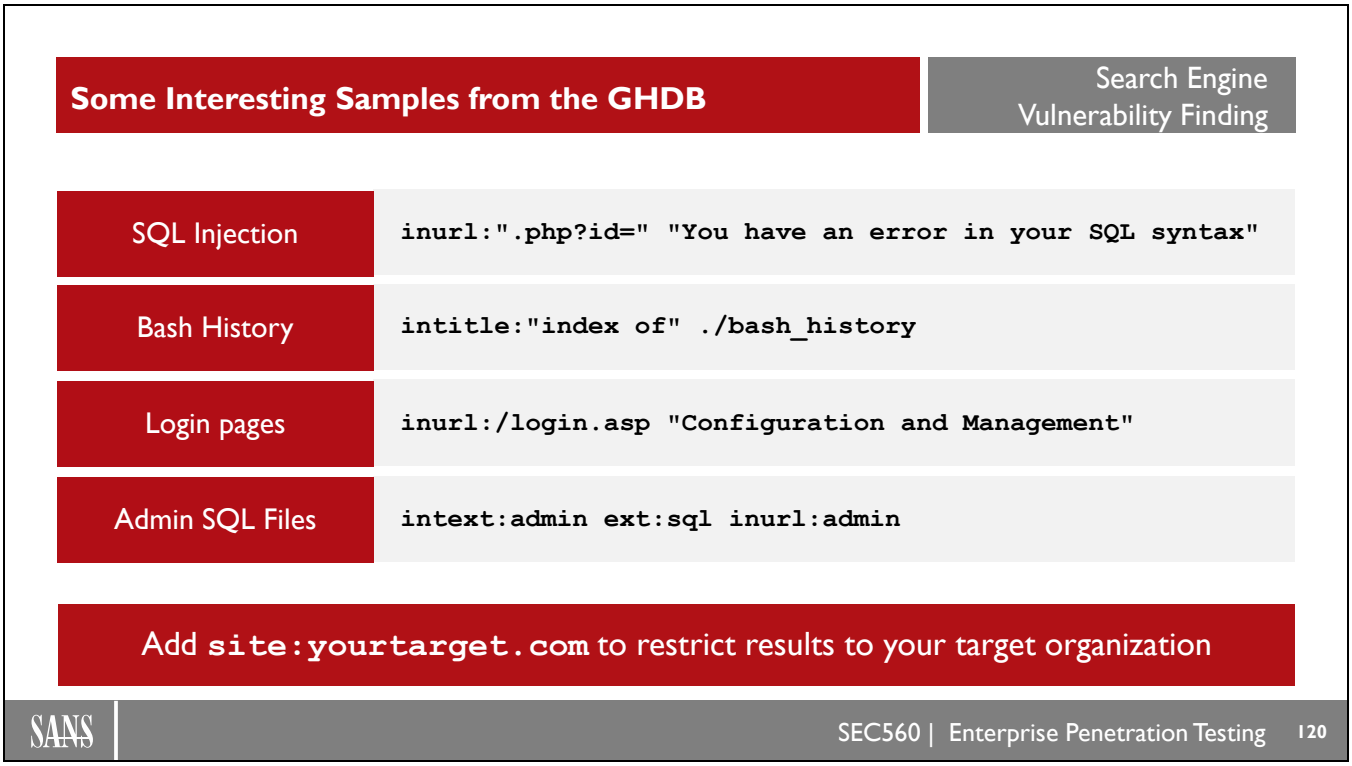
- Johnny Long created a huge inventory of Google searches to find vulnerable systems: the Google Hacking Database, with each search called a "Google dork"
- The folks at Exploit-DB took it over and now operate it at: https://www.exploit-db.com/google-hacking-database
- More than 1,000 entries in this database in the following categories:
  - Advisories and vulnerabilities
  - Error messages
  - Files containing juicy info
  - Files containing passwords
  - Files containing usernames
  - Footholds
  - Login portals
  - Network or vuln data
  - Sensitive directories
  - Sensitive online shopping info
  - Online devices
  - Vulnerable files
  - Vulnerable servers
  - Web server version detection

Several years ago, Johnny Long created a list of useful Google searches to find vulnerable systems. He called each individual search a Google Dork, and the entire inventory of all these searches is known as the Google Hacking Database (GHDB).

Today, the folks who run the Exploit-DB took over where Johnny left off and make a searchable list of the updated GHDB available online at the URL shown on this slide. There are more than 1,000 different searches in the GHDB that can find several varieties of security flaws and related issues, all by simply searching Google.

The GHDB is divided into numerous categories. All the individual categories are listed on the slide, but some of the most important and interesting to us are as follows:

- **Advisories and vulnerabilities:** These searches find vulnerable systems, usually by identifying a known flawed CGI script using the inurl directive or a page with known flaws identified with the intitle directive.
- **Files containing juicy info:** These searches find files that are often associated with caches and logging. Although they don't look for passwords directly, this cache and log information could be useful in learning more about the target organization.
- **Files containing passwords:** Numerous tools generate files that contain either cleartext passwords or encrypted/hashed passwords. These searches identify when such files are available via a web server.
- **Footholds:** These searches locate sites where an attacker may get a foothold that can later be used to compromise the server. A lot of these searches find admin login pages for various common web-based software environments. The Login portals category is similar.
- **Network or vulnerability data:** These searches find pages that hold logs and/or configuration information about network devices, such as firewalls, VPNs, routers, Intrusion Detection Systems, and so on.
- **Online devices:** This category of searches helps locate web-based video cameras, printers, and various kinds of appliances.
- **Vulnerable servers:** These searches locate web servers that may have a vulnerability, a category similar to Advisories and vulnerabilities.



Although the GHDB includes many hundreds of eye-opening Google searches to find vulnerable sites or sensitive data, let's explore a handful of them to get a feel of the power of the GHDB and Google.

The first search on this slide will identify php pages with the "id" parameter in the URL and a common SQL injection error message on the page. This would allow an attacker to very easily identify a severe web application vulnerability in a site that could expose sensitive data.

The second search identifies sites with directory indexing enabled and a file named "bash_history", which contains the list of commands typed by a user. The history file may include interesting targets or even passwords if the user typed the password in the wrong prompt.

The third search identifies login pages and "Configuration and Management" in the page. This is associated with administrative login pages.

The last search, looks for SQL files (typically defining database queries) and the word "admin" in both the text and the URL. According to the creator of the query:

> With the extension sql and intext admin and inurl admin, I was able to look into some of the admin sql files and even sql queries directly that reveals lots of sensitive information like login id, password in clear text. ~Anshul T

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

This page intentionally left blank.

Please work on below exercise.
Lab 1.4: Infrastructure Reconnaissance

Please go to Lab 1.4: Infrastructure Reconnaissance in the SEC560 Workbook.

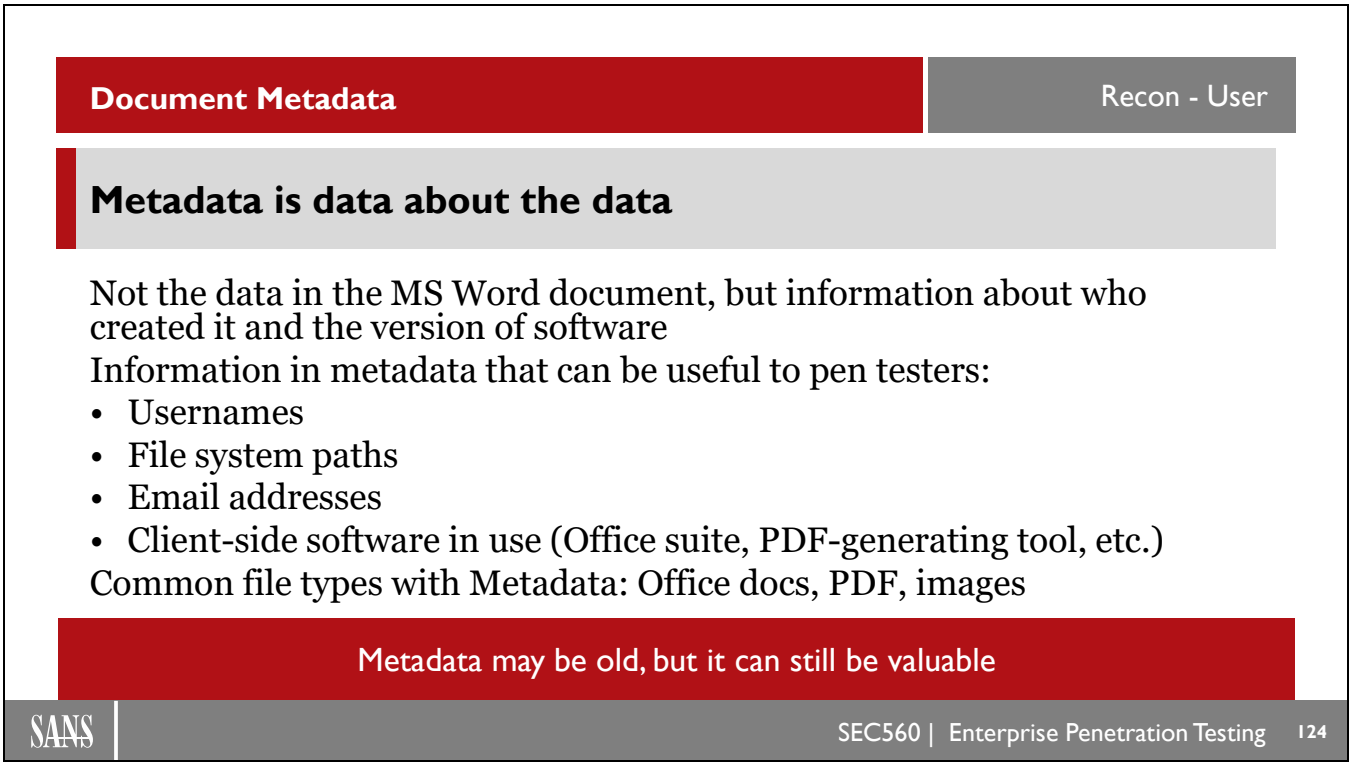
Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

Our next recon step will be to identify users in the organization. The goal is to identify names, email addresses, usernames, and the username format. Email addresses are very useful for phishing. If we can flesh out the phish by using the target's name, the phish will be more successful. Also, names and email addresses can be used to build a list of usernames which is very helpful in password guessing attacks.



As organizations create documents, the software that they use to create these documents embeds an enormous amount of information in the document files. Of course, much of this information is the contents of the file. But a good deal of metadata (that is, data about data or data describing other data) is also included in the file. A lot of file creation and editing tools include additional metadata entries that can be useful for pen testers, such as:

- **Usernames:** Often user password guessing attacks, and so exploits.
- **File system paths:** This helps the tester know where important data is stored in the organization as well as system naming conventions and maybe even important directories.
- **Email addresses:** Useful for phishing and password guessing.
- **Client-side software in use:** Useful for client-side exploitation, especially when the document is recent and includes the software version information.

Almost every document type has some form of metadata, but some are richer in metadata than others. The following types of documents, generated and used by most enterprises, are of particular interest to pen testers:

- **pdf files:** These files are associated with Acrobat Reader and a variety of other pdf creation and editing tools.
- **Microsoft Office files (doc, docx, dotx, docm, xls, xlsx, xslt, xlsm, ppt, pot, pttx, pttm):** These files are associated with Microsoft Office, including Word, Excel, and PowerPoint

In addition to these types of documents, there are hundreds of others that may be interesting. This list is not intended to be exhaustive but is instead designed to get the reader thinking about interesting and useful document types to analyze during a penetration test.

Retrieving Documents for Metadata Analysis

Recon - User

- Pull documents from the target's website (most common)
  - Curl or wget with recursive fetch
  - Grab files from search engine results with PowerMeta by Beau Bullock
- Review documents sent by target system personnel during the planning of the test (agreements, NDAs, contracts, etc.)
- In-house penetration testers can often harvest documents from a file server
- Common metadata extraction tools: ExifTool, FOCA, DIA-NZ

To perform metadata analysis, a penetration tester must first retrieve files to analyze. Numerous methods could be applied to gather these documents.

One of the most common and especially important methods for harvesting documents for metadata analysis is to use a web spider tool against the target organization's website, pulling all potentially interesting documents onto the penetration tester's machine for analysis. In our next lab, we discuss how the wget tool can be used for this. We could use a tool like PowerMeta by Beau Bullock to find data on search engines, download the files, and perform metadata analysis. PowerMeta is available at: https://github.com/dafthack/PowerMeta.

The penetration tester may have already received some documents generated or edited by target system personnel during the planning of the testing project. For example, the tester may have received Rules of Engagement agreements, scope information, diagrams, Non-Disclosure Agreements, contracts, policies and procedures, and other information.

And finally, if the penetration test is conducted by in-house testers (employees of the same target organization), they typically can get an ample supply of documents for analysis from file servers in the organization.

There are many tools to extract metadata, the most commonly used are:

- ExifTool
- FOCA – https://www.elevenpaths.com/labstools/foca/index.html
- Metadata Extraction Tool from the New Zealand Department of Internal Affairs - https://github.com/DIA-NZ/Metadata-Extraction-Tool

ExifTool

Recon - User

ExifTool: Reads, writes, and changes metadata

- Freely distributed, written by Phil Harvey at exiftool.org
- Runs on Windows, Linux, and macOS
- Supports more than 100 file types and many metadata formats
  - Original focus was on image and audio files
  - Expanded to include many file types, including various enterprise document file types (doc[x], xls[x], ppt[x], pdf, and so on)
  - Parses out specific fields and is handy for determining usernames and software versions used to create or edit files
  - Processes entire directories, with recursion supported
- PowerMeta uses ExifTool to extract metadata

The ExifTool program focuses on reading, writing, or editing the metadata in more than 100 different file types, including images, audio files, videos, Office documents (doc, dot, xls, ppt, and more), pdfs, and a multitude of other formats.

Written and freely distributed by Phil Harvey, ExifTool runs on Windows, macOS, and Linux.

When it was first released, the original focus of ExifTool was on image and audio files. For images, it focused on pulling out the camera type and details about the format of the image. It also pulls information about any tools that were used to edit the image or audio. If the image includes geotags indicating the latitude and longitude of where the photo was created, ExifTool retrieves that information.

ExifTool has been significantly extended beyond its original roots in image and audio metadata, now pulling data from the vast majority of file formats a penetration tester is likely to encounter. Of particular interest to pen testers is ExifTool's capability to discern usernames, email addresses, and document editing tools from the files it analyzes.

By default, ExifTool handles one or more files provided to it on its command line invocation. Alternatively, the tool processes entire directories on the local machine where it runs, handling every file in the directory, and it can even be set to recurse through a directory structure, analyzing all files it finds.

The aforementioned PowerMeta tool uses ExifTool to extract the metadata from the files it retrieves.

Strings Command Details

Recon - User

- The strings command displays printable text from a file
- Good for finding unstructured data or data with unknown structure
- Linux: Included in most Linux distributions and UNIX varieties
  - ASCII strings only by default
  - Four sequential characters by default (change with -n)
  - Can also be used to look for Unicode strings with-e b (for 16-bit big-endian Unicode) or
    -e l (for 16-bit little-endian Unicode)
- Windows: Separate download from Sysinternals
  - Both ASCII and Unicode strings by default (specify -a or -u to select only one)
  - Searches for both big-endian and little-endian Unicode strings by default
  - Three sequential characters by default (change with -n)

Unlike many metadata analysis tools that focus on structured data, the strings command is useful for finding unstructured data or data for which the structure is unknown. The strings command simply displays printable text from files. It is included in most Linux distributions and UNIX varieties. The strings command is available as a separate download for Windows in a variety of different packages. For example, it is available as a component of Cygwin, the free POSIX environment for Windows available at www.cygwin.com. Or strings is available as a free standalone download from Microsoft Sysinternals.

By default, the Linux version of strings looks for printable ASCII strings only. It searches through the file for four or more consecutive ASCII characters and then prints them to Standard Output. To change the default minimum string length, strings can be invoked with the -n X option to specify whatever string length the user wants (the X). The default of four characters is reasonable for most uses.

Many document types, especially those associated with Microsoft Office programs (doc, docx, xls, xlsx, and such) store some strings not as ASCII (an 8-bit character representation) but instead as Unicode (a 16-bit character representation). If you run Linux strings with its defaults, it will show you only ASCII strings, and you may miss out on some highly useful information. It's a good idea to run strings multiple times: once with its ASCII default, once with the -e b option (for an "encoding" type of big-endian 16-bit characters), and once with -e l (a lowercase "L", for 16-bit little-endian encoding). Big endian and little endian refer to the way the bytes are ordered for the given string in the file. Most Microsoft document editing tools use little-endian encoding but will sometimes (even in the same file) store some strings in big-endian format.

The Sysinternals version of strings looks for ASCII, big-endian Unicode, and little-endian Unicode strings by default (pulling each of those different formats in a single invocation), focusing on strings of three or more characters in length. To focus on only ASCII or Unicode, the tool can be invoked with -a or -u, respectively. And to change the minimum character length, we can invoke it with -n X.

Hunter.io (1)

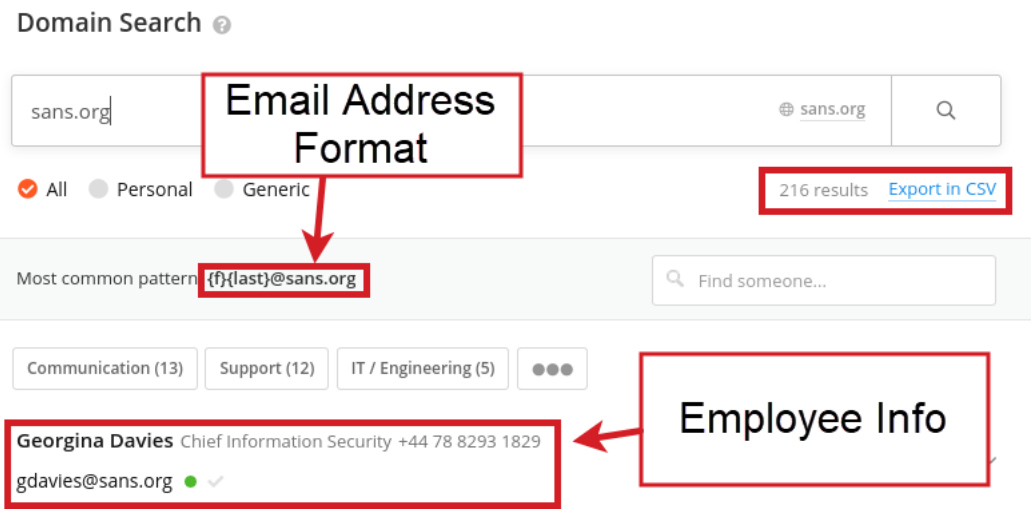
Recon - User

- Regularly performs OSINT scans and compiles lists of organizational data
  - Contact name, email address
  - Occasionally contains contact phone number and job title
- Provides common email format
  - Examples: [first initial][last name], [first name].[last name],
- Recently became a paid service

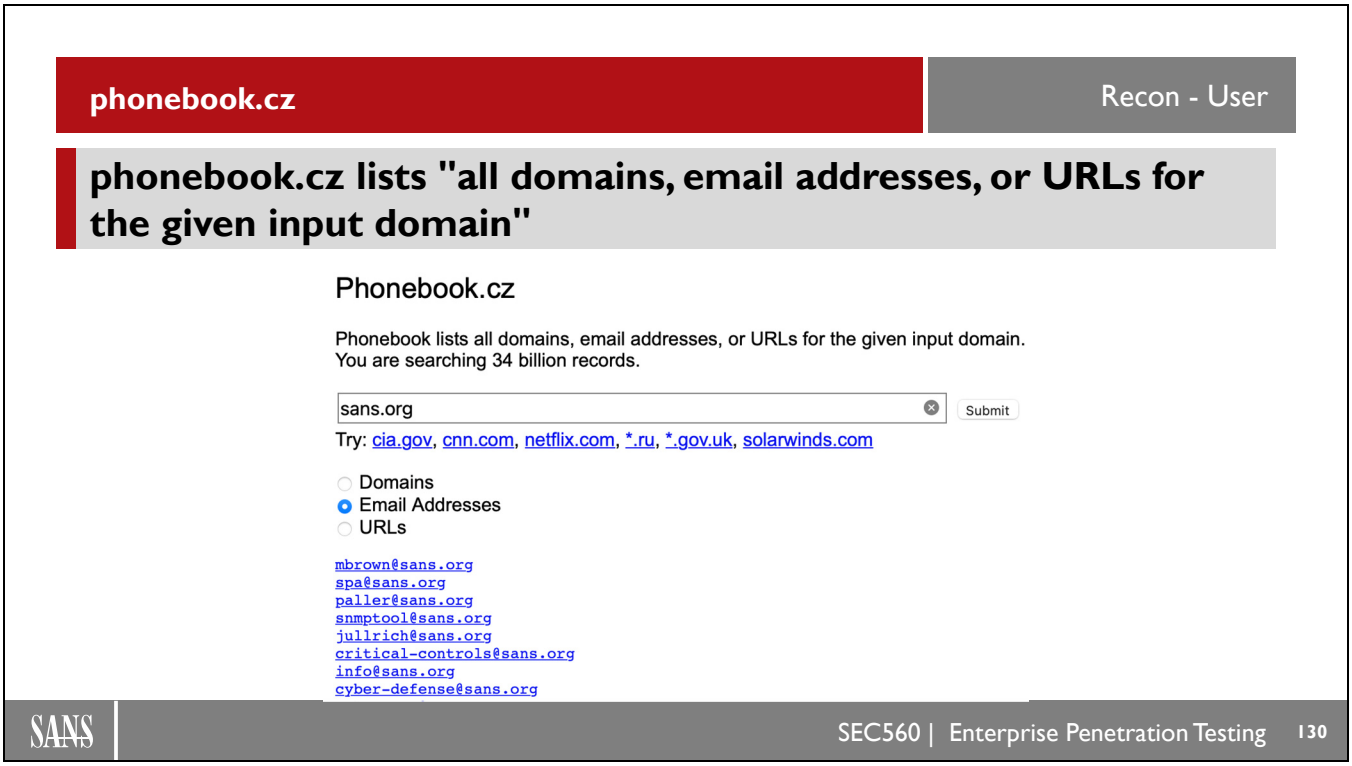
Hunter.io compiles large lists of intel on organizations based on OSINT scans performed by their team. These lists contain contact names and email addresses as well as, occasionally, phone numbers and job titles. The best part about Hunter.io is they show the common email address format for the detected email addresses. At a glance, you can find the format and then use that to build potential email addresses for employee names found in later recon steps. Unfortunately, the free access to Hunter.io recently came to an end. They are now offering 10 contact results and the email format for free. Additional contact info will require a subscription.

Hunter.io (2)

Recon - User



Here we can see Hunter.io search results for the sans.org domain. Hunter.io returned 216 results containing employee names, email addresses, and (in some cases) phone numbers and job titles. Another great thing about Hunter.io results is that they show you the observed email format. According to the search results, sans.org is using the format {first initial}{last name} for their email addresses. When combining this pattern with recon gathered in other steps such as LinkedIn recon and document metadata analysis, we can start building continually growing lists of valid email addresses for later social engineering attacks.



Phonebook (https://phonebook.cz) lists all domains, email addresses, or URLs for the given input domain. In the example above, we searched for sans.org and found over 1,100 email addresses. The address may not be recent or valid, but it does provide a good simple list of email addresses.

Public Breach Data of Credentials

Recon - User

- Data breaches are becoming increasingly common
  - Breaches often go unreported for months or years
- Breach data is sold on dark web markets and traded on password leak forums
- Limited shelf life means that the data is eventually shared with everyone for free (usually)
- A niche security market has sprung up around reporting breach data observed in the wild
  - Email addresses are still valid even if the breached organization required a password change

Every day, we hear about another astronomical amount of customer accounts that were leaked through publicly reported breaches. Companies often wait months or years to report the breach while they perform incident response. Customers are often notified if the breach data is found online being traded or sold in dark web marketplaces. As penetration testers, we're torn between sympathy for the companies who are experiencing the breach and the prospect of having a large trove of information that would otherwise not be available. Once the data is publicly available, the shelf life is usually extremely limited for the passwords that are leaked. However, that doesn't mean that they're not valid. Users will often stick to passwords that are very similar as well as reuse them across multiple sites. Even if the passwords are no longer valid, the email addresses are almost always valid. Additionally, password patterns can emerge from the leaked data leading to successful password guessing during later stages of the attack.

Public Breach Data

Recon - User

Find useful leaked data in public services

- HaveIBeenPwned.com
- Dehashed.com
- intelx.io
- PwnDb - https://pwndb2am4tzkvold.tor2web.io
- Public data dump forums
  - Searching for public data dumps will land you public dumps pretty easily
- Torrents -
  - 77gb Leaked Database Archive 7zip file, Archive.org is hosting Collections 1-5

It's all well and good to read about these companies experiencing tens to hundreds of millions of accounts being leaked, but where do we find these credentials?

- HaveIBeenPwned.com is a great first stop. They provide a search function on their website for individual email addresses. However, for roughly $3 a month, you can have API access which gives you the ability to search by email domain names. While you won't be able to download passwords, you can obtain lists of email addresses using their API.
- Dehashed is a site that makes a large number of public data dumps searchable and will include email address, usernames, and passwords. A paid subscription is required for access to the data though.
- Intelligence X (intelx.io) is a database with over 75B leaks. The free tier shows some breached credentials, while the paid tier shows even more.
- PwbDb is a free-to-use website onion site that allows you to search data dumps that they have indexed. The site doesn't claim to have every data dump, but they have a large number, and the price is right. Due to being a free service, access to the results can be spotty when they're under a heavy load of searches.
- Public data dump forums are another avenue for obtaining breach data. To encourage community growth, they often require posting to earn credits towards downloading the breach data. You can pay for credits and download the data as well. This is definitely entering a gray area legally and caution should be advised before participating in these forums.
- Finally, there are torrents with the data available. I'll leave it as an exercise to the reader to seek out links to find what is mentioned here. Again, caution should be taken when downloading these materials.

LinkedIn is can provide a lot of information on employees

- Employee names
- Organizational Position/Titles
- Email Addresses
- Posts containing charities or causes affiliated with the target
- Recent awards and accolades
- Unfortunately…
  - LinkedIn security team continually closes loopholes allowing access to organization employee lists
  - Often requires connections within the target organization to view the employees
- Also look on sites like Twitter, Facebook, Instagram, TikTok, and others

LinkedIn is a great platform for performing reconnaissance against an organization. Often, you can build almost an entire organizational chart of every employee while gathering their names, job titles, and email addresses. Additionally, you can profile charities and causes that the organization supports or view awards and recognition that the company has received. All of these bits of information go toward creating social engineering scenarios in later stages of the attack chain. Unfortunately, LinkedIn has become aware of these abuses of their platform and tries to close loopholes that we as pen testers use during profiling an organization. It is becoming increasingly difficult to masquerade as employees of an organization that your account is not a part of as well as viewing the employee profiles without having connections in their organization. There are ways around these limitations, but they require investments of time and effort to build LinkedIn profiles. As we will see on the following slides, there are still tools out there that can help with searching LinkedIn.

LinkedInt

Recon - User

- LinkedInt is a great tool for searching LinkedIn
  - Written by Vincent Yiu and hosted at https://github.com/vysecurity/LinkedInt
  - Automates finding the company ID
  - Searches by email domain once the company ID is found
  - Can include different search terms to narrow or broaden the search
- Returns a list of the discovered employees, email address, and reported job title
- Requires valid LinkedIn credentials
  - Burner account will work, but … burner account may have fewer results if it has small number of connections

LinkedInt is a great tool written by Vincent Yiu for scraping LinkedIn and gathering employee names/email addresses. LinkedInt accepts search terms for narrowing or broadening the search and automates finding the company ID used by LinkedIn. The results are further narrowed down to results containing email addresses with the target organization domain. Once the scan is complete, an HTML report is generated containing the profile picture, employee name, and email address. LinkedInt does require valid credentials to a LinkedIn account. Use caution when giving your personal LinkedIn account information as the scan can sometimes cause the account to be temporarily banned for scraping. Also, if the account does not have any connections, the results can be reduced.

GatherContacts (1)

Recon - User

- GatherContacts is a Burp Suite extension written by Carrie Roberts
  - Can be found at https://github.com/clr2of8/GatherContacts
- Scrapes LinkedIn search results from Google and Bing
  - Requires manual searches in the form of 'site:linkedin.com/in [target org]'
  - Does not require LinkedIn credentials
- Potential Downsides
  - No email addresses
  - Results can contain profiles who either no longer work for the target or mentioned them on LinkedIn

Another great tool for LinkedIn reconnaissance is the Burp Suite extension GatherContacts by Carrie Roberts. The great thing about this tool is it can gather employee names and job titles without needing LinkedIn credentials. Once GatherContacts is installed and configured, simply search for 'site:linkedin.com/in [target organization]' and scroll through the results. GatherContacts will automatically extract the relevant data to a text file determined during configuration. There are some downsides to using GatherContacts. Unlike LinkedInt, you will not be gathering email addresses against the target. Another downside is results can contain LinkedIn profiles that are not directly employed by the target organization. If the LinkedIn profile mentions the target organization as a previous employer or (in SANS' case) used their services and/or gained a certification from them, the results may have profiles of non-existent employees. This is especially prevalent when your target is an educational institution.

GatherContacts (2)

Recon - User

site:linkedin.com/in "SANS Institute"

Page 2 of about 28,700 results (0.31 seconds)

www.linkedin.com › johnlhubbard

John Hubbard - SANS Institute - LinkedIn

John Hubbard. Certified Instructor and Author @ **SANS Institute** | On a mis
Teams everywhere! **SANS Institute**Purdue University.
Philadelphia, Pennsylvania - Certified Instructor / Course Author - SANS Ins

www.linkedin.com › wanicha-owen-gisf-43693050

Wanicha Owen, GISF - SANS Institute - LinkedIn

SANS is a leading organization in information security training, the **SANS In**
providing intensive, immersion training designed to help you and ...
Denver, Colorado - Inside Account Manager, GISF - SANS Institute

B	C	D	
Column2	Column3	Column6	Column7
Name 1	Name 2	Description 1	Description 2
Johannes	Ullrich	Fellow	SANS Institute
John	Nix	Director, Federal	SANS Institute
Rob	Lee	SANS Institute	LinkedIn www.linkedi
Ray	Hawkins	Director	The SANS Institute / G
Jay	Armstrong	Director, SLED Partnerships	The SANS ... www.link
Tim	Conway	ICS	SANS Institute
Brian	Ventura	Certified Instructor	SANS Institute
Benjamin	Wright	SANS Institute	LinkedIn www.linkedi
Frank	Kim	Fellow	SANS Institute
Scott	Cassity	Managing Director	SANS Institute
Steve	Penny	Director	SANS Institute
Howard	Cribbs	CIO	SANS Institute
Wanicha	Owen	SANS Institute	LinkedIn www.linkedi
John	Hubbard	SANS Institute	LinkedIn www.linkedi

Here we can see a sample of the search results for 'site:linkedin.com/in "SANS Institute"'. While Google mentions that 28,700 results were found, Google will only allow you to scroll through 30 pages of 10 results. Afterwards, it will stonewall the search results. On the right, the resulting text file has been imported into Microsoft Excel as a tab delimited data file. The columns show the employees' names, titles, and places of business. The latter columns can help with determining whether the resulting LinkedIn profile is valid for your search terms.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

This page intentionally left blank.

Please work on below exercise.
Lab 1.5: User Reconnaissance

Please go to Lab 1.5: User Reconnaissance in the SEC560 Workbook.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

So far, we've done a lot of manual research. Let's now discuss automated reconnaissance using SpiderFoot.

SpiderFoot (1)

- Open-source automated OSINT tool developed by Steve Micallef
  - Available for free at https://github.com/smicallef
  - There is also a paid cloud version, SpiderFoot HX
- Minimal configuration and Python library installs required
  - Web interface and command line interface
- For maximum results, API keys to online information broker sites can be added
  - Total of 44 API keys can be added
  - Most only require a confirmed email address for access
- Options for quick to thorough scans available

SpiderFoot is a configurable automated OSINT scanner that uses more than 100 different data sources. To get started with SpiderFoot, we only need to clone the GitHub repository, run the python setup script, and launch the tool. To get the most out of SpiderFoot, 44 API keys can be entered for checking different data sources. Of these 44, only 8 required a paid subscription. SpiderFoot can be ran in 4 different modes ranging from very quiet and quick scans to extremely thorough as we'll see on the next slide.

SpiderFoot (2)

- All
  - This scan uses every module available
  - Takes a very long time to finish (hours up to days)
- Footprint
  - Scans for information about the target's network perimeter
  - Mostly search engine and web crawling results
- Investigate
  - Basic footprinting and querying block lists and malware intelligence feeds
- Passive
  - Never directly queries the target or affiliated sites

SpiderFoot has four preset scan configurations.

1. The 'All' scan will perform an extremely thorough scan including over 100 data sources and procedural scanning meaning that every bit of information that is found is also investigated. This can lead to scans taking multiple days to finish and can also contain irrelevant data.
2. The 'Footprint' scan will search for information about the target's network perimeter and will also look for associated identities such as email addresses and employee names. This scan is much faster than the 'All' scan but can still take some time depending on the size of the target organization.
3. The 'Investigate' module will query spam and malware block lists as well as malicious site intelligence feeds for information about the target. This scan can be helpful for Blue Team responders investigating an alert or report.
4. The 'Passive' scan disables all modules that could potentially interact with the target organization. This scan will use only pure OSINT to gather information and should not alert the target to your scans.

Course Roadmap

- ***Comprehensive Pen Test Planning, Scoping, and Recon***
- In-Depth Scanning and Initial Access
- Assumed Breach, Post-Exploitation, and Passwords
- Lateral Movement and Command and Control (C2)
- Domain Domination, Azure Annihilation, and Reporting

560.1: Comprehensive Pen Test Planning, Scoping, and Recon

- Defining Terms
- Types of Pen Tests
- Building an Infrastructure
- Linux for Pen Testers
  - LAB 1.1: Linux for Pen Testers
- Overall Process
- Pre-Engagement
- Rules of Engagement
  - LAB 1.2: Scope and RoE Role Play
- Reconnaissance Overview
- Organizational Recon
  - LAB 1.3: Organizational Reconnaissance
- Infrastructure Recon
  - LAB 1.4: Infrastructure Reconnaissance
- User Recon
  - LAB 1.5: User Reconnaissance
- Automated Recon with SpiderFoot
  - LAB 1.6: Automated Recon with SpiderFoot

This page intentionally left blank.

Please work on below exercise.
Lab 1.6: Automated Recon with SpiderFoot

Please go to Lab 1.6: Automated Recon with SpiderFoot in the SEC560 Workbook.

Conclusion for 560.1

Conclusion

- That concludes the 560.1 session
  - We've now covered some important definitions and concepts
  - We've looked at scoping and Rules of Engagement
  - We've also looked at methods for conducting recon to gather information that will serve as a crucial foundation for later components of testing projects
  - We've configured our machines for the lab work ahead
- In 560.2, we'll look at scanning and initial access

We now conclude our 560.1 section in which we've addressed some important concepts in penetration testing and ethical hacking. Among the most important topics for this section have been proper scoping and the formulation of Rules of Engagement. We also discussed the recon phase used in many penetration tests and ethical hacking projects, gathering information that will act as a firm foundation that testers will leverage for the remainder of a testing project. We've also configured our machines to prepare for the lab work we'll perform throughout the remainder of the class.

In our next section, 560.2, we'll take an in-depth look at scanning, the process used by penetration testers and ethical hackers to determine openings in the target environment. We'll also gain access to our first systems through exploitation and password guessing.

This Course is part of the SANS Technology Institute (STI) Master's Degree Curriculum.

If your brain is hurting from all you've learned in this class, but you still want more, consider applying for a Master's Degree from STI. We offer two hands-on, intensive Master's Degree programs:

Master of Science in Information Security Engineering

If you have a Bachelor's Degree and are ready to pursue a graduate degree in information security, please visit www.sans.edu for more information.

www.sans.edu 855-672-6733 info@sans.edu