Project: Sandia Exercise

Introduction

One of the goals of this class is to teach how to analyze the security of systems. The *penetration test* is a powerful *a posteriori* testing technique that examines not only the design and implementation of a system but also its maintenance and operation—the latter two often being overlooked in *a priori* evaluations (because the system is not yet fully operational, or is not yet in its production environment).

A proper penetration test devises specific goals that the testers are to achieve. Examples of such goals are to acquire *root* access on a UNIX system, to read a specific file, to create or delete a specific file, or to block access to the system for some period of time. The goals vary depending upon the security policy of the site and upon the reasons for the test.

We will use the Flaw Hypothesis Methodology for this study. The project grade does *not* depend upon achieving the stated goals It depends upon the correct implementation of the Flaw Hypothesis Methodology. You are required to keep a notebook to demonstrate that you are using the methodology. If the notes show what you thought of, tried (and how it was tried), the results, and any ideas that spring from the test, you will get a good grade. If the notes do not show this, you will receive a bad grade even if some of the goals of the test are met. In other words, your grade depends upon your use of the methodology.

Background

This class section will analyze a set of systems (the “targets”) set up at Sandia National Laboratories by Fred Cohen and his College Cyber Defenders (CCD). Dr. Cohen is the scientist who has led research into computer viruses since his seminal paper in 1984 (see Chapter 18 of the class text), and has done wonderful work in studying how people attack and defend computer systems. So this should be a fun exercise!

You are required to follow certain rules. If you break the rules, we can (and will) take appropriate action, up to and including removing your access to the Sandia systems and filing a complaint with the SJA. For whatever it is worth, we have never had to do this before, and I’d like to keep this class’ good record intact. So, your classmates, the teaching assistants, Dr. Cohen, the CCD members, and I would appreciate your cooperation.

You are being granted access to a portion of the CCD network at Sandia National Labs for the purposes of this course. You will be given an IP address, a user ID, and a password. Use there as your *exclusive* means to access the systems.

- These user IDs and accounts may ***only*** be used for the purposes of this class and not for any other purpose.
- User IDs and passwords ***must not*** be shared with others. They are for your class use and no other use.
- Some of the software on these systems is also copyrighted and it may be a legal violation to take it back without permission to copy.
- You must not advertise, in any way, your use of these systems to others. They are just more class systems used in your education and they happen to be operated by and located at Sandia.
- Federal law explicitly prohibits the placement of, or transfer of, any pornographic or “inappropriate” material in, to, or from these systems.
- The defenses in these computers do not necessarily reflect any particular defenses used in any other particular computers or networks.
- These computers may include experimental defenses, intentionally weakened or artificially strengthened defenses, and ***any*** defense technologies that the defenders see fit to place in them.
- Sandia, and Dr. Cohen’s research groups, will be recording your uses of these computers for their research. ***Do not*** attempt in any way to circumvent this recording, unduly obfuscate your methods, or pander to them. Just go about your business as if we never told you this ….
- Neither Sandia nor UC Davis is in any way liable for any harm due to emotional stress, physical consequences of typing, seizures, feinting spells, lack of sleep, collapse of ego, loss of significant other, ridicule from your classmates, poor grades, loss of future income, or death that may result from your attempts to attack these systems.

You will only be able to reach the Sandia systems by using SSH to enter a secure login server (more on this below). Please ***do not*** proxy back X11—use this SSH connection only for terminal sessions and securely copying files. This is necessary because 70 students proxying back X11 will make the connection to UCD collapse.

Think of access to the login server as insider access to the network under attack, obtained from an insider who has loaned you her computer so you can break in. The login server is ***not*** to be attacked in any way by your teams because it is a shared resource for your classmates.

All of the computers in the exercise are defended. They may "fight back." Be aware of this while you are carrying out your attacks. Your login server is open game and in their territory.

Do not count on the safety of any software you might bring back from the CCD network. It may contain computer viruses, Trojan horses, or any other sort of nasty thing. We recommend you not bring any software back unless doing so enhances your chances—and even then, see if you can gain the same advantages in other ways.

You may only use informational attacks against these systems. No physical or electronic attacks are permitted. Perception management can be attempted if so desired—as long as all of your efforts remain within the network under attack. According to Dr. Cohen, Sandia students and staff are not all that well paid, so bribes may be accepted, but services are will not be rendered for them. (All bribes, extortions, perception management, *etc*. must be offered and fulfilled through the information link into the network under attack. Also, they must be recorded in your notes.)

The exercises for this class are part of an experiment that Dr. Cohen is running. Your actions on Sandia's systems will be monitored, and Dr. Cohen and the other researchers may use those actions and the results of those actions in their work. Your names will not be used for any purpose except to grant you access to the computers, or to take action if you break the rules.

WARNING. In sniffing traffic, ***please do not sniff*** the devices *eth0* or *eth1* on the login servers. They are used for logging and *ssh* traffic. If you sniff the traffic, you will be sniffing your own sniffing, causing an infinite loop which will consume all of your bandwidth and cause others in the class to be unable to work. Naturally, it will not affect the inside network, so you will only get yourself into trouble and cause others inconvenience.

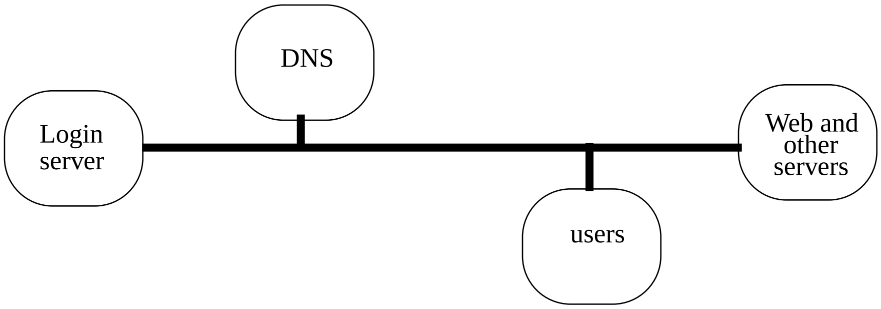
The Sandia Systems

To reach the Sandia systems, you must first log into a CSIF computer, and then use *ssh* to connect to the login servers at Sandia. You will be assigned a unique IP address, login name, and password for this purpose. Please use only the ones assigned to you.

When you log into the login servers, you will be able to *su* to *root* (no additional password required). Use this carefully! If you crash the server, everyone on it will be very, *very*, unhappy with you for a while. (It won't take them long to figure out who did it, either.)

The login servers have very limited disk space, so please don't leave unnecessary files on the login servers. Moreover, to ensure that crashed systems become available again quickly, the servers are periodically rebooted and cleaned out. When this happens, some files on the servers may be deleted. This means that any data you leave on a server may be deleted periodically, possibly as often as once every hour. So make sure that you keep copies of your work on a local machine. Also, if you are connected to a server that reboots, you will lose your connection.

The initial network configuration will look like the diagram below, with all hosts on a single subnet. As the term progresses, we plan to add a firewall at various places on the network. For example, one may be added in front of the "web and other servers", or between the DNS server and the users systems. When there is a change in the network configuration, we will announce it.



A few more comments about this network may help you.

Resilience

All systems will be duplicated with a stand-by system. When a system fails, the stand-by system takes over its function. This has interesting consequences if you decide to try a denial of service attack.

DNS Server

The source for the DNS server being used in the internal network is located on the *heat.ca.sandia.gov* web site. Look for SDNS (*S*ecure *DNS*). You are free to download the source and try to figure out an attack against it, and then use that attack on the internal network. Dr. Cohen and the CCD would love for someone to find a weakness in it!

Web Servers

The web servers may run Apache, or may run a special web server called *thttpd*, the *t*rivial *http d*aemon. Some of the web servers may also run a perl script or a lisp program; you may find either (or both). The servers may have additional services; if so, it is legal for you to attack those services.

User Processes

The users will be doing things. What they will be doing is left for you to discover. Presumably, it will be comething interesting, just as your use of the computer would be interesting to others (right?)

Goals

There are four possible desired outcomes for this penetration study. You should try to do as many as possible, but expect not to do all. The goals are, in (what we think is the) order of difficulty:

1. Locate, identify, and characterize all of the computers in the network (by reporting their IP addresses, type, and any other characteristics), including security devices and show sample traffic that indicates how to proceed in your attack.
2. Deny services in a controlled and repeatable manner to the user box or DNS server. To do this you must:
    a. identify when which services will stop and start (in defending network time) ahead of time (make the times for 5 minutes at a pop and make sure you demonstrate the ability to repeat the deny/allow cycle 3 times);
    b. say how you will determine whether your attack succeeds; and
    c. carry out the attack, and show it succeeded.

    The system logs will show us whether you succeeded.
3. Gain access to the user box so that you can cause desired changes on it. Desired changes (in increasing difficulty) are:
    a. alter any file (*proof*: identify the altered file by name and its previous and subsequent content along with how you did it);
    b. gain superuser access (*proof*: show the contents of a root-only accessible file);
    c. add an account for yourself (*proof*: show the new password file and provide us with the name, UID, and password of the new account);
    d. sustained re-entry at will (*proof*: show a TA or the instructor how you can re-enter the system at will);
    e. sustained access across reboots (*proof*: same as for d but across reboots); and
    f. alter the system for sustained reentry across reboots without the need to repeat the attack that got you in in the first place (*e.g.*, leave a permanent login for yourself across reboots; *proof*: tell us how to get in after a reboot).
4. Same as 3 except for the secure DNS and the protected server(s).

    In what follows, a "vulnerability" is any method that will achieve any of these goals.

Notebooks

We will discuss the Flaw Hypothesis Methodology in class. It is also covered in the textbook (see section 19.2.4), Your notebook must document your use of this methodology. It should consist of four types of entries.

Knowledge of the System

These entries document things you have learned about the system. For example, if you establish that the server is an SCO system, you would create an entry saying that and explaining how you know it is an SCO system. Your observations may either be used to reach one of the goals, or to provide background information of other entries.

Hypotheses

These entries document suspected vulnerabilities in the system. In addition to the hypothesized vulnerability, you must say why you think the vulnerability might exist (for example, the system is a Linux Red Hat 6.2 system, and the vulnerability is known to exist for that system). You also have to say what the consequence of the vulnerability would be (for example, if this buffer overflow in the setuid to *root* program succeeds, you can execute an arbitrary program and thereby get access to *root*). Indicate which goal (or goals) the hypothesized vulnerability would help you achieve.

Testing

Choose at least 10 hypothesized vulnerabilities and design tests that will tell you if the vulnerability exists and is exploitable. You must be able to carry out the test. For example, you cannot say, “the system administrators should check the contents of the protected file /etc/errors, and if the contents of that file begins with ‘abracAdabra’ the system is vulnerable.” If you can’t read the file, you’ll have to devise some way of testing whether the file contains what you think. One way is to try to exploit the vulnerability … you get the idea.

Document each test in detail. We will need to be able to repeat it. For each test, document the relevant parts of the system and environment settings, the arguments to the program, the input, and the output, and any relevant side effects (like creating s setuid-to-*root* shell). You will need to submit the test program, and document how you used it to test the hypothesized vulnerability.

Generalization

Some of the tests you documented in the previous entries will succeed. Others will fail. From this, you may get ideas for other vulnerabilities, or be able to generalize to find new vulnerabilities. For example, if a buffer overflow allows you to obtain *root* privileges, and the bug is in a library call, check other programs for use of that library call. They are also vulnerable, in all probability. Document any generalizations, and hypothesize new vulnerabilities based upon them.

Organization

For this project, you will need to work in teams of 2. You must pick someone in your own section to work with (the other section is doing a different penetration study). Once you have selected your team, please email the team member names and email addresses to cs153@cs.ucdavis.edu. Members of a team may exchange information freely; the project is their joint work. However, different teams may *not* communicate. Each team works independently of all other teams. This is necessary for two reasons. First, if two teams independently come up with the same vulnerability, they can both use it; were the teams communicating, there would be questions of priority that we want to avoid handling. Secondly, your efforts will be monitored for research purposes, and a copy of your notes will be sent to the researchers. Collaboration across teams will bias the results and limit the effectiveness of the research.

Conclusion

The goal of this project is to help teach you ways to analyze systems for security. It will also bring to life some of the concepts we will discuss in class. Also, in order to defend a system, you need to know how to figure out where you might be attacked. In the instructor’s experience, those who know how to attack are much better defenders because they can react, and understand, much more quickly and effectively than can those who have only check lists of things to look for, or who do not know how attackers think.

User Accounts and IP Addresses

Your Sandia IP address, user ID, and transformed password are available on the class web page. Please go to *http://my.ucdavis.edu*, sign in, go to the class web page, and from there to *project/sandia/ids.html*. Look for the last 6 digits of your student ID. Your IP address and user ID are as shown. To get your password, subtract your *full 9 digit* student ID from the number in the "transformed password" field.

When you pair up into teams, you *must* choose your team mate from the users whose account numbers have the same parity (odd or even) as you. For example, UCD333 may pair up with either UCD117 or UCD21, but not with UCD38 or UCD 134. This is necessary because of the way Sandia has set up its tracking mechanisms.