CS 444/544 Spring 2011 Lab 2: Exploits

Due: 10 February 2011, at 11:59pm

100 points

I will assign groups, but your deliverable will be an individual report that will be graded separately, and you're not obligated to work in the assigned groups. Your writeup should be sent to me in PDF format at my gmail account, jedcrandall@gmail.com.

The purpose of this lab is to learn about some common security mechanisms (process separation, access controls, *etc.*) and understand why security mechanisms fail. Since understanding vulnerabilities and exploits requires understanding the underlying data and what it means, we'll be learning a lot of systems design topics such as system call traces and debugging.

I've broken you up into groups of four. This is a very informal group arrangement, this lab is essentially an individual effort. I expect you to exchange contact information with your group and try to help each other out, but feel free to work with others in the class and seek help from your group, other classmates, myself, or anybody you feel comfortable asking for help. You should do your own work on your own laptop and only present results in your final writeup that are your own.

Your assignment is to document at least 6 exploits for real vulnerabilities in UNIX or UNIX-like systems. Two of the exploits should be remote memory corruption vulnerabilities, one should be local privilege escalation based on concurrency issues, one should be web-based (SQL command injection or cross-site scripting), and the last two can be anything but you should try to have some variety. Two exploits for the same vulnerability is okay, but only counts as one exploit towards your six.

I'll provide a tar ball with some virtual machine images, particularly the ones that should interest you for this lab are Red Hat 6.2, FreeBSD 4.2, OpenBSD 3.1, and Damn Vulnerable Linux, but it's possible that exploits exist for FreeBSD 8.1, OpenBSD 4.8, or Solaris 11 since I have not installed patch updates for any of these virtual machines. If you're using your own laptop, I can give you ISOs for any of these OS versions. You may install other OS versions to run exploits against with my permission. For the most part the answer will be “no” for anything Windows-based and “yes” for everything else, but get permission before doing so in any case. All exploits must be done in a virtual environment.

For each exploit/vulnerability, you should show me concrete evidence like system call traces, annotated tcpdumps, and dumps from gdb debugging sessions. You should try to address the following kinds of questions...

- What data was interpreted differently than the programmers of the system expected? What were they expecting? What did the exploit string give them instead as input?
- What kinds of things are anomalous about the exploit? For example, you can strace a normal FTP session and strace an exploit on the FTP service and show the differences.
- Was any information related to the exploit interpreted differently over time even though it didn't change?

- What parts of the exploit string cannot be changed without the exploit being caused to no longer work?
- Was there any data that had multiple meanings throughout the course of the exploit?
- Is the part of the service being exploited, or the particular functionality, really critical or could it be removed?
- Was the exploited service running with more permissions than it really needs?
- Were there any “secrets” to the exploit that could not be figured out with tools like strace and gdb but would actually require source code to know? What about in terms of discovering the vulnerability in the first place, what are your thoughts on the possibility of that discovery occurring without access to the source code?
- What could the programmers have done differently to avoid the vulnerability, in terms of recommended programming practices?
- For the key parts of the service being exploited, in terms of library functionality and APIs, were they shared resources with the rest of the system or private resources existing only for the exploited process?
- How does this exploit relate to the classical vulnerability studies in Chapter 23 of Bishop's book or the design principles in Chapter 13?

Your writeup should be well-annotated with technical information, I don't need a lot of text with background information and everything, just show me what you did and how you addressed the above kinds of questions for each exploit, with plenty of your own thoughts and ideas as context. You'll turn in an individual writeup and be graded individually.

I'll assign groups of four, you should trade contact information but these groups are fairly informal. You can work with anyone you want in the class, you're not obligated to work with the people in your group but you should at least keep in touch with them and help them out when you can. Remember that you can always come to me for help, send an email to the secpriv-chat list, or get help in a number of other ways. The work you present should be your own in the end, though, with only results presented from your own virtual environment. All text should be your own writing in the writeup. In terms of source code, feel free to use Google and grab all the source code and tools and information you want from the web or from your classmates. Do not give other people concrete results such as system call traces, gdb output, tcpdumps, or anything you generate from within your own environment. In this case you will be helping them to cheat and you will receive a 0 on the assignment if they use your results. Also, don't present others' results as your own, this will be considered cheating and you will receive a 0 on the assignment and I may pursue the matter further as per University policy. If you're not sure whether a particular kind of data/code is okay to share or not, ask first. In general, if you typed it or downloaded it, it's okay to share, if you generated it by executing something in your virtual environment, it's not okay to share. You can look at each others' generated output on-screen and discuss it, but do not make copies of it.

Lab policies for this lab:

- All exploits should be carried out in virtual environments only.
- You are not to carry out any kinds of attacks on real, external networks as part of this lab, all

exploit traffic should traverse virtual networks only.

- If you are not sure about containment of a particular exploit, ask first.

Some resources that will help you:

- Try Googling “Metasploit” and “BackTrack Linux.” You can use these tools if you like, I prefer to download exploits from Bugtraq. For Bugtraq IDs of some exploits for the OS versions we're working with check out the tables in the CCS 2005 paper, “On Deriving Unknown Vulnerabilities from Zero-Day Polymorphic and Metamorphic Worm Exploits.”
- Refer to chapters 4 through 12, and 15 in the Gray Hat book. 16, 18, and 19 might also be helpful.
- Read Chapter 13 of Bishop's book, and skim or read Chapter 23.
- I'll provide virtual machines soon, as well as a walk-through example for one exploit to show what I'm looking for. You can do this walk-through and include it among your exploits, but be sure to only include your own results and add your own thoughts.
- Don't forget about man pages, e.g., “man strace”.
- Get help early and often, and from multiple sources.
- I'm fairly familiar with the specific OS versions and vulnerabilities we'll be looking at, so if you get stuck on something or are not sure how to do some administrative task (like enable a service), shoot me an email.