COMP 4108: Computer Systems Security

Winter 2018 Mid-term Exam Solutions

1. [6] Please answer the following questions using one attack scenario.
   A: For this question many answers are possible; below is one consistent scenario.

   (a) [2] What is one attack that a student in a coffee shop is vulnerable to when using free wifi to surf the web on their Windows 10 laptop? Please describe the attackers goals and means of attack.
   A: An attacker controlling the access point redirects traffic to malicious versions of normal websites using a rogue DNS server and counterfeit TLS certificates. The attacker wants to gather passwords for banking sites so they can steal money using interac e-transfers.

   (b) [2] Would this attack be prevented or mitigated if the student first connected to the Carleton VPN? Explain.
   A: This attack would be mitigated by using the Carleton VPN because then the attacker will not be able to impersonate banking websites, because they will no longer be able to observe or modify the local traffic. From their perspective, it will all be encrypted and going to a single destination. At most, they can block the traffic, not tamper with it.

   (c) [2] Could this attack be mitigated by a commercial security (anti-malware) product? Explain.
   A: Commercial anti-malware suites sometimes include software for detecting malicious websites using a known blacklist. If the attacker was using websites on such a blacklist, then they could be protected. Also, if the suites did additional verification of web site certificates, the impersonation could be detected. It is likely, though, that the attacker's websites are not on a blacklist (because they are new) and the suite has no effective means of detecting rogue certificates.

2. [6] As a regular, non-privileged user on a UNIX-like system (Ubuntu Linux, FreeBSD, or even MacOS X), which of the following can you do? Why? (Assume that this user is only part of non-privileged groups such as "users".)

   (a) [1] Run a program binary or script downloaded from a remote site.
   A: You can create and run arbitrary programs on UNIX-like systems, so it is possible to download a program and run it as a unprivileged user. This program will only run with the user's permissions. Note that some systems can display warnings for downloaded programs (e.g., MacOS X); however, these warnings can be easily bypassed.

   (b) [1] Make a binary available to be run by any user on the system.
   A: Yes, they can do so, they simply need to place the binary in a directory they can make accessible to others on the system (e.g., make a bin directory in their home directory that can be accessed by any user on the system). By default this directory won't be in the search path of other users, but this is easily changed.

   (c) [1] Place files in a system temporary file directory, e.g. /tmp.
   A: Any user can place files in a temporary directory, that is what they are for.

   (d) [1] Restrict access to files in a directory called "Private" in your home directory. (Who will and will not be able to access these files?)
   A: By simply running chmod 600 on the directory, other users won't be able to access the fles in Private—as long as they don't have root privileges. The root user could access the files in Private as the root user can access all files on the system.

(e) [1] Share files with one other user on the system such that only that other user (and you) can access the files.
A: A regular user cannot do this on their own on a standard UNIX-like system because they would need to create a group for the two people, and creating a group is a privileged operation. If the root user defined an appropriate group these users could use it to share files. (If the system supports access control groups as many Linux systems do, then it is possible for two users to share files privately without root's help.)

(f) [1] Perform an operation that requires root privileges. (Give an example.)
A: Regular users can perform operations requiring root privileges simply by running a setuid-root binary or by sending a message to a program running as root. For example, they can run passwd to change the their password stored in /etc/shadown (which can only be read and modified by root). They do not have the ability to perform arbitrary operations as the root user, however.

3. [2] Argue for or against: On a properly configured system, host-based firewalls are useless.
A: For: On a properly configured system, no unneeded services should be running, so there should be no daemons that need to be blocked. Against: A host-based firewall can block outgoing connections and can prevent rogue applications from receiving incoming connections—these are protections that are useful even if the system is properly configured.

4. [2] How is remote host authentication similar between SSH and TLS (in a web browser)? How is it different?
A: They are similar in that both receive a public key from the remote host, and they both then verify that the remote host has the associated private key. They differ in that TLS verifies the key by checking a chain of signatures that should be rooted in a certificate authority's key, while SSH checks whether the key is the one that was previously used by that host (and thus on first connection SSH provides no guarantees about the remote host's identity).

5. [2] How could a backdoor in Firefox allow a malicious web application to bypass Firefox's page-level sandbox? What would be the potential impact of such a backdoor?
A: A backdoor in Firefox could be very simple, e.g., a special string contained in a web page could allow Javascript in the page to have access to the local filesystem (giving it similar privileges to a browser extension). The impact would be the same as installing a malicious application with the same privileges as Firefox (i.e., the attacker could run arbitrary code as the user and potentially as the Administrator depending upon how Firefox self updates).

6. [2] How does downloading an authentic copy of Firefox help prevent backdoors in Firefox? How could this protection be bypassed (e.g., how could an authentic copy of Firefox have a backdoor in it)?
A: An authentic copy of Firefox would normally not have a backdoor because Mozilla would be seriously harmed if it became known that Firefox had a backdoor in it. The process of authenticating the binary (using a digital signature or by comparing hashes from a trusted source) ensures that nobody else has tampered with the binary. Thus, for an attacker to get a backdoor into the authenticated binary, they either need to compromise the authentication process or they need to modify the code at Mozilla before it is sent out. (Normally this is hard with an open source project; however, the build process could be compromised so the malicious code is inserted without being part of the publically visible source code.)

7. [2] Assume that you download a malicious application into a clean virtual machine (i.e., one with no sensitive information or special access to other hosts). You then install the application in the virtual machine and run it. If you then destroy the virtual machine, are you secure (assuming your system was not compromised before downloading the malicious application)? Explain.
   A: You should be secure, assuming that 1) the malicious VM didn't compromise other resources via the network or other shared resources (e.g., access to part of the host filesystem), and 2) the malware wasn't able to exploit a vulnerability in the VM that gave it access to the host operating system. If either of these assumptions is violated you're in trouble.

8. [2] What is one mechanism that prevents Windows from installing malicious Windows updates, even if an attacker controls the machine's network connections? How could this protection be bypassed?
   A: Windows has the public keys of Microsoft built in; it uses these keys to verify the authenticity and integrity of updates to Windows. This protection could be bypassed if the key store on Windows is compromised, or if the signing key of Microsoft is compromised in some way.