Lab 1: Security Audit

Mark Gius

January 16, 2017

1 System Description

The system is a low-powered Atom PC, small form factor (it is less than 12" on each side, and about 3" tall), with a 320GB hard drive and 2GB of ram. The system is located in my bedroom in a house in San Luis Obispo, on top of another computer. It is plugged into a surge protector, but not a UPS. The surge protector is shared with several other computers. The system has no input devices plugged into it (keyboard, mouse, webcam, etc).

The system is connected to the internet via a switch in my bedroom that is shared with several other desktops. The bedroom switch connects to a shared house switch via a cable that runs along the floor of the hallway leading to my bedroom. The shared workgroup switch has connects to a Linksys router running DD-WRT (linux), connected to a DSL modem (local ATT affiliate ISP).

2 Threat Analysis

Threat			
Power Loss	Goals violated	Availability	The system is completely unavailable in the event of a power loss
		Integrity	Power loss is well known to cause data corruption
	Vulnerability	Random power outage (or malicious conduct) could cause power loss to the system	

Continued...

Threat			
	Controls	Availability	Power outages can be mitigated by attaching the server to a UPS. There is not currently a UPS on this system.
		Integrity	Steps can be taken to ensure data integrity, such as journaling filesystems or battery backed disk cache. A journaling filesystem is currently in place (ext3) and past experience has shown that the journal is reasonably effective at preventing integrity errors.
Shared Network	Goals violated	Availability	A malicious or incompetent roommate can use all available resources, limiting system's access.
		Confidentiality	Data on shared equipment can be snooped.
	Vulnerability	Network on which computer is located is untrusted.	
	Controls	Availability	Reasonably sane QoS policies on router reduce competition for network resources.
		Confidentiality	All sensitive data is sent and received via encrypted, and possibly authenticated channels (HTTPS, SSL, etc).
Public OpenSSH	Goals violated	Availability	Compromised accounts could consume system resources or damage services
		Confidentiality	Compromised accounts could grant information to unathorized persons
		Integrity	Compromised accounts could modify system configurations and data
	Vulnerability	A user account with a bad password could become compromised and allow unathorized access to the system	

Continued...

Threat			
	Controls	General	OpenSSH is configured to disallow remote "root" login. "sshblackd" script monitors for repeated failed logins (brute force attempts) and adds IPs to a "banned" iptables chain. All admin accounts have reasonably secure passwords. OpenSSH is listening on a nonstandard port.
		Confidentiality	Sensitive data is not publically readable via file system permissions, so compromised non-admin accounts are prevented from accessing restricted data
		Integrity	Same as above, except read/write
Theft	Goals violated	Availability	A missing computer is an unavailable computer!
		Confidentiality	With physical access to the hardware an attacker would have full access to any data on the system.
	Vulnerability	The system is small and unlocked. Any person who gains access to the home (locked external door) would be able to easily take the system.	
Continued...			

Threat			
	Controls	General	The house is generally locked, and roommates are trustworthy. Roommate's guests' are also generally trustworthy. Strangers in the home would be confronted by roommates (assuming somebody was home.
		Availability	A lock could be placed on the system to prevent theft. Currently unlocked, and more valuable systems would be higher in the priority list.
		Confidentiality	Whole-disk encryption with required password to unlock would protect data. Currently unused because I don't care and data on disk is not important enough to encrypt.
"Public" Fileshares	Goals violated	Confidentiality	Files that are not supposed to be accessible can be made accessible, either by exploit or accidental publishing of files.
		Integrity	Read-only shares may be written to.
	Vulnerability	Samba exploits could allow access to restricted files, or allow read-only shares to be written to.	
	Controls	Confidentiality	Not much I can do here
		Integrity	Backups to another system can ensure that a valid version of the file is always available.
Dynamic DNS	Goals violated	Availability	If DNS service goes down, or if DNS is not updated properly, the system will be unavailable except by IP address.
	Vulnerability	System is connected to the internet via a "home" DSL connection, with a dynamically assigned IP address. Dynamic DNS service is used to provide a consistent address on the internet for the system.	
	Controls	Availability	Choose a reliable DynDNS service. Purchase a static IP

Continued...

Threat			
Compromised Roommates' PCs	Goals violated	Availability	Compromised PCs can consume unreasonable levels of shared resources.
		Confidentiality	Compromised PCs can expose "internal"-only resources to unathorized persons.
	Vulnerability	Resources which are supposed to be "internal" can be exposed if another machine on the network is compromised. This bypasses any rules that grant internal machies more access than an external one.	
	Controls	General	Could hold house-wide security audits.
		Availability	Detect and Take down any compromised machines.
		Confidentiality	Treat internal and external users as the same, assume that anything published for an internal user will be accessed by somebody on the outside.
IRC Bot	Goals violated	Confidentiality	Exploited IRC bot could be used to gain access to system resources.
		Integrity	Exploited IRC bot could change IRC logs or communicate as me
	Vulnerability	IRC software is probably not designed with high-security in mind. It is likely that the IRC client can be exploited.	
	Controls	Confidentiality	Run the IRC bot as an unprivileged user to prevent access.
		Integrity	Keep backups of the logs that the IRC user can't touch. I'm not doing this because I really don't care.

3 Conclusions

Overall, I would rate the risk level of this system as "low". Almost all of the threats evaluated are known, and are mitigated in some fashion. The only publically accessible route into the system is an OpenSSH service. OpenSSH has a long (long) history of being coded by paranoid security fanatics. In addition, OpenSSH is listening on a non-standard port, which mitigates the effects of automated bots. In addition, the machine only really

provides one important function, that of a source code repository for myself. Because I recently switched from subversion to git, I now have several such repositories that I sync to, so the importance of this machine has diminished.

The data on the system consists primarily of these repositories of program code and a collection of MP3s. These MP3s are backed up to other locations periodically, so their loss is not critical.

Although incompetence, maliciousness, or malfeasance on the part of my roommates could cause damage to my systems, they are all generally trustworthy in this regard. Between technical competence, and a "cold-war"-esque mutually assured self-destruction, we tend to leave each other's machines alone. I take mild precautions (local firewalls, antivirus for windows, read-only network shares) regardless, but I'm certain that none of my roommates are sitting in their rooms trying to hack me. They might as well walk into my room and steal the hardware.

In short: nothing particularly sensitive or important, the only write-only access is via OpenSSH, which is coded by paranoid security experts, and reasonably trustworthy roommates.

[M= $\begin{bmatrix} 1 & 2\ 3 \\ 4 & \end{bmatrix}$]