Editors: Matt Bishop, bishop@cs.davis.edu
Deborah Frincke, deborah.frincke@pnl.gov

Guarding the Castle Keep: Teaching with the Fortress Metaphor

Deborah A. Frincke
Pacific Northwest National Laboratory

Matt Bishop
University of California, Davis

The computer security field is replete with metaphors—the original and most commonly used metaphor is the computer (or network) as a fortress, the walls of which must be guarded against potential breaches. This metaphor is useful, but like all metaphors, it is not precise. Understanding the differences between the metaphor of a fortress and the realities of securing a system is crucial to students understanding the subtleties of computer security. In this department, we discuss the fortress metaphor as a pedagogical tool, both how it succeeds and how it fails to aid student understanding.

Fortress-based security

A goal of computer security is to prevent people from violating a site's security policy. Managers and security experts believe that the greatest threat arises from unauthorized system access or use, or authorized limited system usage.

This leads to the paradigm of fortress-based security. Fortifying a structure provides safety because defenders believe that attackers will have difficulty overcoming the fortifications. Fortifying usually involves layering the defenses: a moat, for example, surrounds a castle wall, and a castle wall might consist of several different walls. This is analogous to traditional computer security mechanisms.

In computer security terms, the principle of separation of privilege requires an entity to satisfy multiple conditions to obtain privileges (such as access). For instance, to log in to a system, someone must have both a valid username and password. This idea leads to security mechanism layering. Using a firewall to protect a site is such an example. The fortress metaphor can explain this concept further. The firewall acts as the fortification's outside wall. Each computer system inside the defensive perimeter has its own security mechanisms in place. In some installations, multiple firewalls provide a (restricted) area for external access and a secure inner area for the site.

For those students who know mythology, we can use our fortress metaphor to explain the reason behind the name "Trojan Horse" and how that attack works. This story can be particularly effective in livening up an otherwise dry discussion of technique: "The Greeks could not breach the walls of Troy, so they used deception to enter the city. They built a wooden horse big enough to hide soldiers inside it, and tricked the Trojans into dragging the horse through the city gates. That night, the soldiers crept out of the horse, and opened the gates for the Greeks, who promptly sacked the city."

The computer security analogy is, of course, a program that a user executes. The program performs some action a user intends (displaying a mail message), and some other unintended action (mailing a copy to everyone in the user's address book). The program breaches the walls set up to protect the system because an authorized user executes it, just as the Trojan soldiers were authorized to bring the wooden horse through their defensive perimeters to prevent the Greeks from placating the goddess Athena. The Anderson report, in 1972, christened programs that acted this way as "Trojan horses."[1]

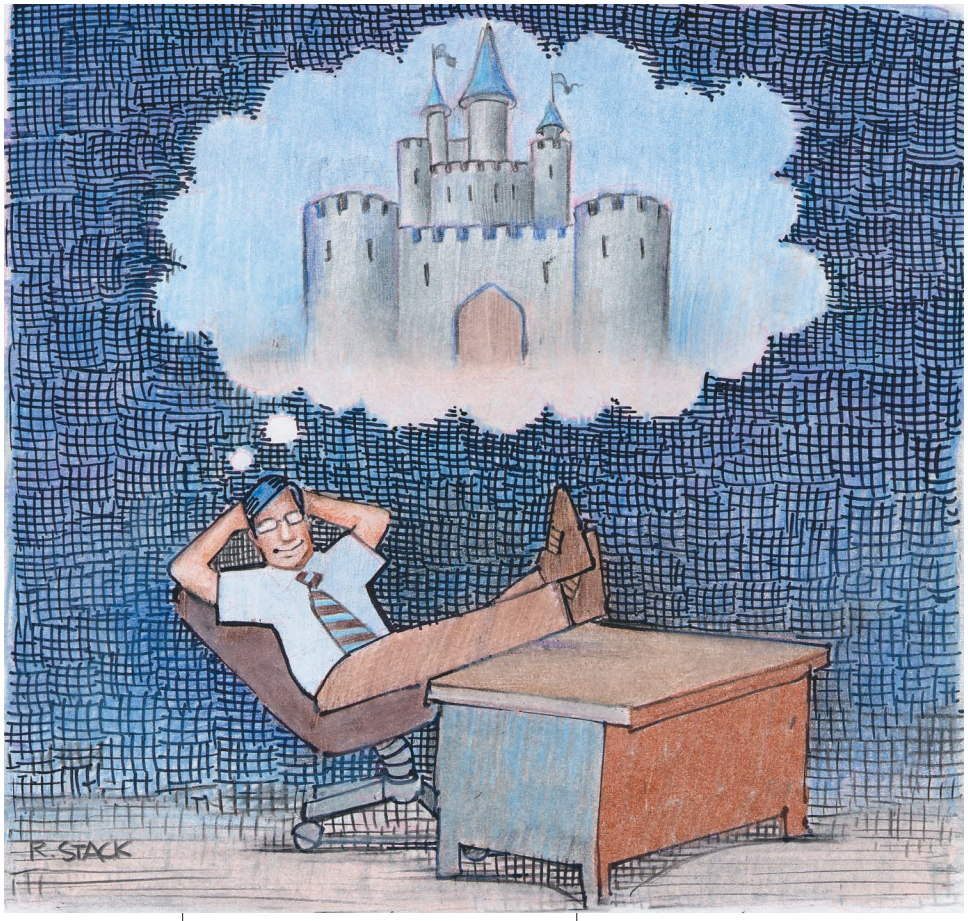
This analogy also works well when discussing malicious code with non-computer science student audiences—and stays with them longer than a dry definition and a warning not to click on attachments.

Metaphor's benefits

We can use this fortress metaphor as a unifying concept in the classroom. The metaphor suggests many consequences. Among them is the cost of constructing and maintaining strong defenses, detecting breaches in defenses, distinguishing combatants (attackers) from noncombatants (others), and maintaining internal services.

Building the fortress

A useful lesson drawn from this is the cost of securing systems. Building fortresses is not cheap and requires care. All that an intelligent



adversary needs to breach a defense is one error or undiscovered weakness. Requiring the enemy to overcome multiple defenses lessens the risk, but the risk does not vanish. The extra layers of defense add to the cost of the security. Students with technological backgrounds sometimes tend to brush aside cost and convenience issues as unimportant, or assume that constant vigilance is easy. This is a harder opinion to hold when confronted with a metaphor in the physical world. Building a fortress requires learning what the threats are. A fortress that defends against stones and arrows might not protect against cannons or bombs. In some environments, the former are the primary threats and the fortress walls need not be extraordinarily thick. But if the latter are the primary threats, then the fortress' interior must not be open to the air (lest bombs fall into it) and the walls must be strong enough to withstand a fusillade of cannon shot. Attackers' strength and weaponry drive the specifications for protecting the people and resources within the fortress, much as attackers' skills and goals drive the specifications to protect a computer system.

Furthermore, defenders' resources affect what defenses they can construct because they must pay for them, with money, labor, or some other resource. Their ability to obtain nonfinancial resources is also an issue. For example, if the walls must be constructed of stone, and the defenders do not have an available supply, they must locate and obtain the stone. This introduces a risk: will the stone be high quality? Will it withstand a cannonball's impact, or will it crumble? Can the firewall withstand a determined attack, or will it crumble and allow illicit connections and packets through?

Defending the barrier

Once defensive barriers are in place, how will defenders maintain them? The walls must be strengthened as attackers learn to scale them; other defenses must hinder them as they reach the top. Similarly, we must continuously assess the perimeter, ensuring that the defenses will block the expected, and unexpected, attack; likewise, we should modify firewalls' parameters as attacks become more sophisticated.

But you can go too far—you can build a seemingly impregnable fortress, and yet be easily overcome if you do not plan for how to provide food, water, and other supplies to both inhabitants and defenders. Early warriors learned this lesson well. Tales of fortress inhabitants being starved into submission are legion, as are tales of fortresses that were planned around wells that could provide the needed water for defenders and inhabitants. Likewise, if a protected site requires email and Web access, blocking all incoming traffic defeats the protection's purpose, causing a denial-of-service attack. At some point, site defenders must allow email to enter the defensive perimeter—which would breach the defense. But if the defenders could monitor the incoming email and block any potentially dangerous letters, then email entry would pose an acceptable risk. Even with fortresses under siege, defenders would try to find ways for people to smuggle food through the siege lines. The trick was to ensure that the food was not poisoned. The fortress metaphor lets us paint a vivid picture that effectively brings home the need to appropriately balance costs and benefits, and to think through the problem of what is being defended and why.

Who's a danger?

The greatest threat to a fortress can be the people inside because defenses are designed to keep people out. A fortress cannot be isolated

from the outside, however. One purpose of medieval fortresses, for example, was to provide a safe place for those who lived outside the fortress. If these people are noncombatants, the fortress is not in danger. But if any are attackers, then they have penetrated the fortress' defenses and it is thus weakened. Furthermore, if the defenders cannot distinguish noncombatants from attackers, they have two choices. First, they can establish the credentials of all who seek entry into the fortress, turning away those who aren't suitable. The problem here is that the defenders might make mistakes. Falsely identifying someone as an attacker might breed ill will among the population, and could hinder those who wish to help the defenders (or, worse, turn them into new adversaries). Moreover, falsely identifying someone as a noncombatant allows an enemy into the fortress. A second approach is to allow everyone to enter the fortress, which again raises the possibility of letting in an enemy.

Computer security mechanisms have similar problems. The fortress approach handles external attackers: it does not inhibit internal attackers from damaging systems within the secured perimeter. Nor does it prevent internal users from giving access, or making resources available, to external attackers. The Trojan Horse is a classic compromise.

Containment methods

Finally, the fortress should provide a containment method in case the defenses are overcome. If attackers breach one wall, defenders should be able to detect this and respond accordingly. Furthermore, other defensive components should continue to function correctly, letting defenders concentrate their counterattacks on the breached portion without scattering forces to meet the enemy at many breaches.

The computer security analogy provides heterogeneity in defensive mechanisms. If one mechanism is overcome, attackers should not be able to use the same methods to overcome other mechanisms. Defenses can be crafted to detect a breach and respond appropriately.

Comparing computer security to defending a fortress illuminates many similarities between the two. Pedagogically, it demonstrates to students that many classical, traditional ideas of security carry over from the physical world to the computer world. But the analogy is not perfect, and teaching the differences encourages students to think critically. Failing to teach the differences can lead students to make unwarranted assumptions, or to develop blind spots that might later lead to serious consequences.

Fortress metaphor's drawbacks

Metaphors, by nature, represent reality imperfectly. They emphasize only the reality that is relevant to the analysis, which is true for our fortress and secure site metaphor. The differences are critical to understanding the analogy's limits.

The primary difference between securing sites and building fortresses is the fortress' fixed defensive policy, which is designed to prevent external attackers from entering the stronghold. Multiple-entry barriers are intended to be insurmountable. But computer security policies are considerably more complex. Only the simplest policies consist solely of preventing intrusions by external attackers. Most policies involve protection against authorized users, which corresponds to the legitimate fortress inhabitants. These users have limited, or full, access to the defensive site's interior, and because they are already behind the fortress walls, the defensive perimeter cannot protect the site from them. This is why some security experts consider firewalls to be so dangerous—they think security officers will be encouraged by their presence to focus on the perimeter and not the interior. Given the widespread belief that the "insider problem" is far more serious than an external attack, firewalls' effectiveness is questionable.

A second, related, difference is the defenses themselves. They are built to withstand a concerted attack on the fortress. But they are not built to hold out indefinitely against other attacks, such as siege warfare, in which attackers keep the fortress defenders bottled up until their supplies run out (a distributed-denial-of-service attack). The DDoS attacks on Amazon.com and eBay bear witness to its effectiveness. Those sites were defended in ways that attackers simply did not care about because their goal was to interdict communications (network connections) and supplies (orders).

Perhaps the most important difference is the static nature of defense and the changing nature of the threat. A fortress cannot suddenly change its walls. The battle's nature might require soldiers in the fortress to charge against the attackers. But the walls keep the soldiers in the fortress, as well as keep attackers out, so the soldiers must exit through the castle gate. There, the attackers can battle the soldiers: the exit's narrowness lets the attackers focus their efforts on limiting the number of soldiers that can safely leave the keep. Computer security defenses must be flexible. They must allow defenders to change their defensive strategy to cope with changing attacks. This ability is critical to meet previously unknown types of attacks and to cope with changing user needs for a system behavior and what programs can be executed. The "build well and maintain indefinitely" nature of a fortress does not suffice.

Common theme

From the naming of the Trojan Horse virus to the development of the firewall to the idea of securing the Internet, the impenetrable bastion of safety idea dominates computer security. The metaphor provides a paradigm for practicing

computer security. But the paradigm is not comprehensive; it has some serious deficiencies. It assumes a single policy—that external attackers are a more serious threat than insiders. Moreover, it handles external attackers, whose goal is to penetrate the bastion, but not those whose goal is to disrupt commerce or communications, whether or not they enter the bastion. Finally, it promotes fixed defenses, whereas in attacks, flexibility is far more beneficial than rigidity. When teaching with this or any metaphor, it is important to identify the gaps in advance, and to ensure that other metaphors—and other examples—are put forward to combat them. Otherwise, this useful pedagogical tool can become a hindrance to understanding.

Other paradigms exist for security, such as the biological model, in which the computer system and networks are seen as biological organisms and attacks are considered infections. A second is the airplane model, in which teachers compare the airplane and civil aviation system with securing computers and networks. Like the fortress model, these paradigms have benefits and drawbacks. Perhaps future columns will suggest ways to use these paradigms for instruction. ▢

Reference

1. J. Anderson, *Computer Security Technology Planning Study*, tech. report ESD-TR-73-51, Electronic Systems Division, Hanscom Air Force Base, 1974.

Deborah A. Frincke *is chief scientist for the Pacific Northwest National Laboratory's cybersecurity group in Richland, Washington. She is currently on leave from the University of Idaho, where she is an associate professor and was previously the director of the Center for Secure and Dependable Systems. Her research interests emphasize system defense, especially intrusion detection, and the security of high-speed systems. Contact her at deborah.frincke@pnl.gov.*

Matt Bishop *is an associate professor in the department of Computer Science at the University of California, Davis, and a codirector of the Computer Security Laboratory there. His research interests include vulnerabilities analysis, the design of secure systems and software, network security, formal models of access control, and intrusion detection. He is the author of* Computer Security: Art and Science, *published by Addison-Wesley. Contact him at bishop@cs.ucdavis.edu.*