Most important cybersecurity vulnerability facing it managers computer science es...

Profession, Manager

Vulnerabilities to exploitation in modern computers are varied. They range from web server vulnerabilities that allow attackers to take over the web server to very sophisticated side channel exploits that use things like packet timing or instantaneous power consumption to glean confidential information from computers. Vulnerabilities appear in the client software that members of an organization use to get their jobs done. The conclusion of this paper is that unpatched client side software is the most important cybersecurity vulnerability facing the IT community today. Since all modern organizations (companies, non-profits or government entities) use computers and networks as part of everyday operations, this vulnerability is applicable to all of them. For this reason, this paper does not focus on a particular organization or industry.

Vulnerability vs. Threat

Cybersecurity vulnerability is defined as weakness in a computer hardware or software system that can be exploited. This is different than a threat. A threat is the way in which vulnerability is exploited. An example of a cybersecurity threat is spyware or malware being introduced into a computer. Vulnerability is the weakness in the computer's systems that allowed the threat to succeed. This paper focuses on the vulnerabilities, not the threats. Vulnerabilities can be very expensive. The 2009 Computer Security Institute / Federal Bureau of Investigations Computer Crime and Security Survey reports that average losses per respondent were $234, 244, although that number was down from the previous year (Peters, 2009). Cybersecurity vulnerabilities can be present in any part of a computer

system’s software or hardware. According to the SANS institute, the number of vulnerabilities discovered in software applications far outnumber those found in operating systems. (“ Top security risks-vulnerability exploitation trends”). This is because operating systems tend to be more long lived and therefore more tested than applications. Vulnerabilities can also be more sophisticated than the normal vulnerabilities we read about often. For example, one can determine what operands are being processed by a computer by monitoring it instantaneous power consumption. This, along with a knowledge of what algorithms are being processed can lead to the guessing of an encryption key (Brooks, 2010). Once the encryption key is guessed, files and communications involving that host could be decrypted. Another unusual vulnerability is the fact that keystrokes are sent across communications networks one at a time, so that if one captures the communications of an ssh session, the keystrokes can be guessed based on the time between them and the layout of a QWERTY keyboard (Brooks, 2010).

The Origin of Vulnerabilities

Most vulnerabilities occur because of programmer error. One of the most common errors that cause cybersecurity vulnerability is called buffer overflow. In buffer overflow, more data is provided as input than the program is expecting. This causes a corrupted stack and can allow an attacker to inject rouge code. The use of modern programming languages and proper coding techniques can eliminate the possibility of buffer overflow, but there is vast amount of software out there that has this vulnerability, Much work

has gone into mitigating and preventing this type of vulnerability to exist in software, or if it exists, to not be exploited. Vulnerabilities that appear in software may not be the result of programmer error. They may be inserted into software applications intentionally by dishonest employees of software vendors. The fact that there is not much reporting of the discovery of such vulnerabilities does not mean they don't exist. Consider the factors that might prevent a software vendor from publicizing the discovery of deliberate malicious code in one of their products. There are liability issues and the company's reputation would suffer if such a thing became known (Franz, 2008).

Human Vulnerabilties

Vulnerabilities that allow malicious actions to take place on an organization's computer systems sometimes have nothing to do with hardware or software. An organization's personnel can be a large cybersecurity vulnerability as well. Since it is the organization's personnel who implement any cybersecurity measures that are dictated from the CIO staff, it is they that are the key to the cybersecurity plan's effectiveness. If people are practicing dangerous activities on the organization's computers, then all the planning in the world won't prevent bad things from happening. There are factors that contribute to the cybersecurity vulnerabilities that personnel contribute to. One study divided these factors into nine areas, external influences, human error, management, organization, performance and resource management, policy issues, technology, and training (Kreamer, Carayon, & Clem, 2009). The authors make the point that not all vulnerabilities are caused by bad

programming. Personnel issues are a big factor, also. Take, for example, the Stuxnet worm that infected the Iranian nuclear facilities and has reportedly caused lots of damage and has delayed the Iranian nuclear development. The cyberdefenses that the Iranian IT security staff put in place were circumvented by the actions of at least one employee. The worm was introduced via an infected flash drive (Paulson, 2010). All the perimeter defense in the world won’t work if an insider does something wrong either intentionally or unintentionally.

Impacts of Vulnerabilities on Organizations

Some of the cybersecurity vulnerabilities faced by an organization largely depend on what type of business that organization is engaged in. For example, if an organization has a large presence in online commerce (Amazon, New Egg) it has more vulnerability to web based attacks than an organization that doesn’t use the internet for commerce. An organization that possesses unique hardware, for instance an electric utility or a hospital, has vulnerabilities that most organizations don’t face.

Regardless of the type of business an organization engages in and the associated vulnerabilities that are unique to that type of business, a modern organization’s day-to-day operations are performed on computers. Computers and networks are at the core of every process that a company uses to do business. Most managerial and technical employees of any organization have access to and use a computer for performing his or her work. There are internal web sites and email systems that allow

communications between employees. Employees use these computers to do research and purchase products from web sites. This requires that these computers be connected to the internet.

The Most Important Cybersecurity Vulnerability: Unpatched Client Software

Because internet connected computers are ubiquitous in an organizational setting, these computers must be kept up to date with relevant security patches to prevent attacks against known vulnerabilities. For a large organization, this can be a daunting task. The fact that a patch exists for a vulnerability means that the vulnerability has been found and probably publicized. This means that the entire hacker community has access to the exploit and there is a good chance more attacks exploiting this vulnerability will be launched. This makes it imperative that the patch be put in place quickly. Failure to do this leaves an organization open to This is why the SANS institute ranked as the number one vulnerability facing organizations today (as of 2009) unpatched client side software (“ Top security risks – executive summary”, 2009). The number two ranked vulnerability was internet facing web sites. SANS also stated that on average, major organizations are taking at least twice as long to patch client side vulnerabilities than they are to patch operating systems (“ Top security risks – executive summary”, 2009). Because the unpatched client software vulnerability is not industry or business class dependent it is applicable to any company, non-profit organization or government entity. For this reason, the discussion of unpatched client side software does not focus on a particular class of organizations.

Unpatched client side software can be exploited in many different ways. One of the more popular methods is by use of directed email attacks called spear phishing. In a spear phishing attack, a computer user is sent an email intended to entice the user into opening an attachment or clicking on a link that results in malware being installed on the user's computer. When the user opens the attachment or clicks on the link, vulnerabilities in the client software on his or her computer are exploited to gain access to the user's machine or the entire corporate network. The exploited vulnerabilities may be in any client software such as browsers, document readers, or image viewers. These types of attacks are a common method of gaining footholds into corporate networks (ICS-CERT, 2011) and were the method used to launch some well publicized attacks, like the Aurora attack against Google, Adobe and other tech companies (Zetter 2010). While the Aurora attack was not enabled by unpatched client software (it used previously unknown, or zero day vulnerabilities in Microsoft Internet Explorer to enable the exploit), it is relevant to this discussion because the methods used in this attack have been published, making it easy for other attackers to replicate it. This makes it imperative that patches are applied in a timely manner to prevent it.

There are two main problem areas that contribute to the large amount of unpatched client software that remains in use in an organization. The first is that the software vendors sometimes do not publish patches in a timely manner. The second is that once a patch is issued by a software vendor, the patch does not get deployed to the organization's computers for various reasons. As an example of software vendors not fixing vulnerabilities quickly

enough, a company called TippingPoint (now a part of Hewlett Packard) recently released the details of 22 unpatched security vulnerabilities. Some of these vulnerabilities had been reported to their developers over two and half years ago (Keizer, 2011). TippingPoint's Zero Day Initiative buys exploits from independent researchers. They also sponsor contests that reward the best exploits. They then provide their customers protection from these exploits and notify the developer of the targeted software of the existence of the vulnerability that allowed the exploit to work. When a patch is issued by a software vendor, it then has to be applied to an organization's infrastructure in order to be effective. The application of patches does not always happen quickly for several reasons. One reason is that the application of patches is disruptive to the organization's operation. The patches must be vetted by the security personnel and tested by the IT department. Testing patches prior to deployment is critical in avoiding incompatibility problems which would disrupt the organization even more. Another reason that patches don't get applied quickly is that they may not be compatible with in-house operating software. For instance, if Microsoft announces an upgraded browser that fixes many security holes, an organization may not be able to use it because internal software such as an accounting or HR system that they use is not compatible with it.

How to Prevent Unpatched Client Software Vulnerabilities

Organizations can deal with the problem of unpatched client software by being proactive in subscribing to a service that informs them of the existence of new vulnerabilities and in creating and implementing a patch

management process. A patch management process is a multifaceted one. The following elements must be included in the patch management process (Gerace and Cavusoglu):

Senior Executive Support. Without which this, no process can succeed.

Dedicated Resources and Clearly Defined Responsibilities. If there is no staff assigned to the patch management process, it won’t get done.

Creating and Maintaining a Current Technology Inventory. This helps the patch management team determine which and how many systems need to be patched.

Identification of Vulnerabilities and Patches. This allows the team to be aware of what patches are applicable to the organization’s machines.

Pre-deployment testing of patches. This should be done in a controlled environment to prevent adverse side effects.

Post-deployment scanning and monitoring. This gives an indication of the effectiveness of the patch.

As with any other business process, the patch management process must be audited by the use of measurements and metrics. Key metrics include severity/priority incidents associated with mission-critical application outages for inaccurate patching (Colville, 2010). Measuring the effectiveness of the patch management process then leads to modifications to it that improve the effectiveness.

Conclusion

Of the many different cybersecurity vulnerabilities that face organizations in today’s world, unpatched client side software is the most dangerous. This is because this type of vulnerability threatens all organizations, regardless of the type activities they are engaged in. If they utilize computers, then this vulnerability must be addressed to prevent cybersecurity exploitation.