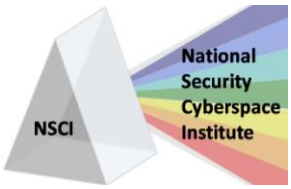

Software (In-)Security

Kandice McKee, NSCI
September 23, 2011

Introduction

Software is a critical necessity in today's Information Age. While the number of operating systems, programs and web applications continues to rise, unfortunately so do software vulnerabilities and threats. When these vulnerabilities are exploited, a number of malicious attacks can result: malware, rogueware, spyware, trojans, and viruses can infect downloads; data can be exposed and stolen; and phishing and other social engineering techniques can expose gaping holes in software security. What's even more devastating is that such attacks are often initiated in the early stages of software development. This poses an enormous problem and threat to U.S..national security. As stated by the CSO of Tenable Network Security, Marcus Ranum, "If we're going to maintain our place in the world, software is not *a* strategic problem; it is *the* strategic problem going forward."[1] So what's even more crucial, and rare, than the everyday software is *secure* software.

Recent Research

In its annual report released in June, The MITRE Corporation released the 2011 CWE/SANS Top 25 Most Dangerous Software Errors. Included in the "list of the most widespread and critical errors that can lead to serious vulnerabilities in software" are:

1. SQL injection
2. Command injection
3. Classic buffer overflow
4. Cross-site scripting
5. Missing authentication for critical function
6. Missing authorization
7. Use of hard-coded credentials
8. Missing encryption of sensitive data
9. Unrestricted upload of file with dangerous data
10. Reliance on untrusted inputs in a security decision
11. Execution with unnecessary privileges
12. Cross-site request forgery
13. Path traversal
14. Download of code without integrity check
15. Incorrect authorization
16. Inclusion of functionality from untrusted control sphere
17. Incorrect permission assignment for critical resource
18. Use of potentially dangerous function
19. Use of a broken or risky cryptographic algorithm
20. Incorrect calculation of buffer size
21. Improper restriction of excessive authentication attempts
22. Open redirect
23. Uncontrolled format string
24. Integer overflow or wraparound
25. Use of a one-way hash without a salt

In addition to the 25 listed vulnerabilities, another 16 were considered. The top 25 errors "are often easy to find, and easy to exploit," according to the report. "They are dangerous because they will frequently allow attackers to completely take over the software, steal data, or prevent the software from working at

[1] "Software Insecurity is Our Biggest Weakness," Dennis Fisher, ThreatPost, May 12, 2010, available at http://threatpost.com/en_us/blogs/software-insecurity-our-biggest-weakness-051210 (herein after "Software Insecurity is Our Biggest Weakness")

all."[2] MITRE's study resembles another, released in April 2010 by The Open Web Application Security Project. In its Top 10 Web Application Security Risks for 2010, OWASP also includes injection as the top risk for software security.[3]

Web applications that are often exploited by the techniques listed in the aforementioned studies have also been examined. Released in November 2010, Bit9's "dirty dozen" exposed the top applications with security vulnerabilities. "The report represents a 'who's who' of venerable tech companies and the applications most popular with enterprises and consumers alike, and contradicts the perception that Apple software is the most secure," according to DarkReading.

The list includes:

1. Google Chrome (76 reported vulnerabilities)
2. Apple Safari (60)
3. Microsoft Office (57)
4. Adobe Reader and Acrobat (54)
5. Mozilla Firefox (51)
6. Sun Java Development Kit (36)
7. Adobe Shockwave Player (35)
8. Microsoft Internet Explorer (32)
9. RealNetworks RealPlayer (14)
10. Apple WebKit (9)
11. Adobe Flash Player (8)
12. Apple QuickTime (6) and Opera (6) – TIE

Not only are the most problematic software applications abundantly popular, the number of vulnerabilities is increasing. As technology is being quickly developed, hackers seem to be staying ahead of software developers. As written by Michael Cheek: "The number of vulnerabilities recorded in the first half of 2010 is almost equal to the total number of vulnerabilities recorded in 2009, according to a report by Secunia, a security notification firm." Secunia has also noted a shift in which software code is being most exploited, which is reflected in Bit9's study: "The company sees the threat landscape as shifting from targeting operating system vulnerabilities to hitting third-party applications. Secunia estimates that a user with 50 programs installed will be faced with 3.5 times more security flaws in the 24 third-party programs running on the systems than in the 26 Microsoft programs."[4] One reason why web applications are being targeted is due to the complexity in fixing the problem. "Attackers target this exploit because third-party software tends to be harder to patch. As a result, the probability of finding a vulnerable host is higher compared to software included with the operating system, which tends to be patched faster."[5]

Imperva's Web Application Attack Report, released July 2011, also found that web applications are being attacked more frequently. According to the report, average Internet applications, like those listed in Bit9's study, are attacked about once every two minutes or 27 times per hour, often by automated cyberattacks. "The level of automation in cyberattacks continues to shock us," Amichai Shulman, Imperva's lead researcher and chief technology officer, said in a statement. "The sheer volume of attacks that can be

[2] "2011 CWE/SANS Top 24 Most Dangerous Software Errors," The MITRE Corporation, June, 29, 2011, available at http://cwe.mitre.org/top25/archive/2011/2011_cwe_sans_top25.pdf
[3] "OWASP Top Ten Project," The Open Web Application Security Project, April 19, 2010, available at https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
[4] "Software Vulnerability Numbers Increase Dramatically in 2010," Michael Cheek, The New New Internet, July 13, 2010, available at http://www.thenewnewinternet.com/2010/07/13/software-vulnerability-numbers-increase-dramatically-in-2010/
[5] "Software Exploits Running Wild," Adam Ross, March 30, 2010, NextGov.com, available at http://cybersecurityreport.nextgov.com/2010/03/software_exploits_running_wild.php (herein after "Software Exploits Running Wild")

carried out in such a short period of time is almost unimaginable to most businesses. The way hackers have leveraged automation is one of the most significant innovations in criminal history...Automation will be the driver that makes cybercrime exceed physical crime in terms of financial impact."[6]

Imperva's study also looked at the type of attacks and concluded that there is often a combination of attacks used to "find and exploit" applications. The four most common – directory traversal, cross-site scripting, SQL injection and remote file inclusion – resemble The MITRE Corporation study's findings.[7]

This means that a vast majority of computers, whether they have a Microsoft or Apple operating system, have at least one vulnerable software application and the likelihood that the vulnerability has been exposed and taken advantage of is great. While many of these applications have offered patches to the security loopholes, patches only work if the software user is aware of and installs the patch[8] – and herein lies a major problem that will be further discussed later.

Issues

As software security improvements and user awareness continue to "lag the fight" against the threat, the exploitation of insecurities is rising. As of October 2010, nearly half of all computers worldwide have been infected with malware. Growing in popularity, a relatively new type of malware flooded cyberspace in 2010. While rogueware accounts for only about 12 percent of malware infections, almost 40 percent of detected rogueware was created in 2010.[9] Taking advantage of the average computer users' naivety and concern for security, rogueware, malware and spyware creators code software that resembles true security updates and/or anti-virus software.[10] Not only do these programs download, install and attack computer systems, they often require users to "buy" a license, which exposes the users' personal data to cybercriminals that can be used for further cybercrimes.[11]

Not all software threats are lurking in the Internet to take advantage of uninformed surfers, though. There are also user- and developer-created vulnerabilities. User-created vulnerabilities occur frequently when users take advantage of software customization. While customization of software often allows for higher efficiency, customization also poses problems for security.[12] By having a few companies developing multipurpose software that can be customized by users, the U.S. has often sacrificed security for convenience. "The country's reliance on commercial off-the-shelf software has made us more susceptible to attack, not to mention less innovative and creative." By relying on commercial software, companies and government agencies alike have "…fewer and fewer people inside the agencies who understand what it takes to write and deploy good software." And for the government, not having "people inside" has translated to more money being spent – "the software they're [government agencies] getting is costing

---

[6] "Web apps attacked every two minutes, study finds," Lance Whitney, July 26, 2011, CNET News, available at http://news.cnet.com/8301-1009_3-20083576-83/web-apps-attacked-every-two-minutes-study-finds/ (herein after "Web apps attacked every two minutes, study finds")
[7] "Web apps attacked every two minutes, study finds"
[8] "Web Browsers, Desktop Software Top 'Dirty Dozen' Apps List," DarkReading, Nov. 16, 2010, available at
[9] "40 Percent of All Rogueware Strains Created in 2010," Nov. 18, 2010, The New New Internet, available at http://www.thenewnewinternet.com/2010/11/18/40-percent-of-all-rogueware-strains-created-in-2010/
[10] "Spyware hidden in Mac software and apps, says security firm," David Chartier, June 2, 2010, TechWorld, available at http://news.techworld.com/security/3225444/spyware-hidden-in-mac-software-and-apps-says-security-firm/
[11] "Cybercriminals use fake Windows update to push bogus security software," Warwick Ashford, Computer Weekly, March 11, 2010, available at http://www.computerweekly.com/Articles/2010/03/11/240572/Cybercrimals-use-fake-Windows-update-to-push-bogus-security.htm
[12] "Dangers of Software Customization Exposed," Eric Chabrow, GovInfoSecurity, July 13, 2010, available at http://www.govinfosecurity.com/articles.php?art_id=2750

several times what it used to because it's coming from contractors rather than internal employees," says Tenable's Ranum..[13]

Not only is the software more expensive, it's also more threatening. Ranum adds, "…our own software is probably a greater threat to us than anything other people can do to us." [14] Users are doomed from the beginning, however, when software is developed with the intent to be insecure. The Department of Homeland Security has recognized that technology components have been knowingly tainted with insecurities. "At the time of a January federal report on the U.S-China supply chain, conversations had been largely hypothetical about 'backdoor' mechanisms, where outsiders insert faulty programming into foreign-manufactured devices to, for example, shut down systems remotely or leak information," according to a report. However, by the time of a July 2011 congressional hearing, Acting Deputy Undersecretary of the Homeland Security Department National Protection and Programs Directorate Greg Schaffer said he was aware of "software or hardware components that have been embedded with security risks."[15] Flawed software and/or hardware can pose serious risks as "a contaminated device can act as a 'Trojan horse' for foreign hackers that could jeopardize the entire network. These types of attacks are hard to detect and could allow malicious actors to steal mass quantities of information without being noticed."[16]

These types of developer-created vulnerabilities not only create risks in cyberspace but also in the physical realm. U.S. Associate Deputy Attorney General James A. Baker told U.S. Congressmen that "acts [in cyberspace] that would be equivalent . . . to kinetic attacks on the Unites States" would constitute an "act of war."[17] While overseas wars are less visible to many Americans, a vast majority of Americans are dependent on utility grids – grids that depend on software to correctly run. However, utility grids aren't immune to insecure software and attacks.[18] And some have speculated that attacks on utility grids could lead to a war outside of cyberspace.[19]

Ideas and Answers

As noted in the first paragraph of this paper, software security is key in today's globalized world – and the U.S. is failing. "The advantage that our adversary has right now is absolute. They can break into our systems pretty much whenever they want. It's really a simple thing you're saying, which is every agency or business that is entrusted with privacy of data and information must have the capability to continuously find and detect vulnerabilities," says Roger Thornton, founder and chief technology officer of Fortify Software.[20] As mentioned, one way our enemies are staying ahead of us is by creating and selling to us software intentionally designed to be insecure. "In place of this current model, Ranum suggested that it may be time for a centralized federal development organization that focuses on writing custom software," Fisher writes.[21]

---

[13] "Software Insecurity is Our Biggest Weakness"
[14] Ibid.
[15] "Threat of destructive coding on foreign-manufactured technology is real," Aliya Sternstein, Nextgov, July 7, 2011, available at http://www.nextgov.com/nextgov/ng_20110707_5612.php (herein after "Threat of destructive coding on foreign-manufactured technology is real")
[16] "Foreign made chips could be allowing hackers into U.S. networks," Homeland Security Newswire, July 11, 2011, available at http://www.homelandsecuritynewswire.com/foreign-made-chips-could-be-allowing-hackers-us-networks
[17] "Threat of destructive coding on foreign-manufactured technology is real"
[18] "More Secure Software Needed for Utilities, NERC CSO Says," Dennis Fisher, Threat Post, Oct. 7, 2010, available at http://threatpost.com/en_us/blogs/more-secure-software-needed-utilities-nerc-cso-says-100710
[19] "Hacks and Cyber Attacks," Tom Ashbrook, On Point with Tom Ashbrook: NPR Boston, June 1, 2011, available at http://onpoint.wbur.org/2011/06/01/hacks-and-cyber-attacks
[20] "Software Security a Growing Concern for Businesses, Government, Expert Says," Matt Korade, CQ Homeland Security. (herein after "Software Security a Growing Concern")
[21] "Software Insecurity is Our Biggest Weakness"

Developing and implementing secure software is the first, and most vital, step in warding off system attacks.[22] However, several vendors agree that developing secure code often takes more resources – whether it's people, time or money – than is budgeted.[23] With a rampant market for the latest piece of technology, companies are in a frenzy to get "the next best device/program/software" out into users' hands.

The growing market, combined with low resources and budgets, is a recipe for a security disaster. Often times, security is sacrificed while after-the-fact security patches are developed. Software companies and developers have become known for their patches and response times. For example, the second Tuesday of every month has become known as "patch Tuesday" for Microsoft and its users. Several other companies and organizations have lured hackers to work for them, rather than against, with money; Google, [24] Facebook[25] and Mozilla[26] are among those that have offered cash rewards, which range from hundreds to thousands of dollars, in exchange for reporting security loopholes. Unfortunately, patches are often developed long after the fact, sometimes six months to a year after the first attack.[27] This gap can make systems even more susceptible to attack. As noted by Cor Rosielle, writing at Infosec Island, "After the patch becomes available the whole world can know about the weakness and often a working exploit is available as well. So between the release and the installation of the patch, the system is even more vulnerable."[28]

Once the patch becomes available, its productivity is up to the user, which has an unsatisfactory rate. A "best case scenario" showed that less than two percent of PCs were fully patched in a 2008 study done by Secunia. While the stats are dated, the company's commentary regarding the study posted on its blog can lead one to infer that the situation probably hasn't gotten much better. As Secunia's Jacob Balle has written, "…these numbers are worse now than previously (11 months ago) when we generated these numbers initially." [29]

Because patch success is reactive and dependent on action by users,[30] it's clearly not the best solution. Some may recommend that software be coded to automatically patch itself so it's not dependent on user action.[31] [32] However, even this has its problems as automatic updates can be exploited.[33] While patches can fix software insecurities, "patch management and software/operating system selection layered with intrusion prevention is a good first line of defense against vulnerability-based attacks."[34]

---

[22] "Software [In]Security: Cyber War – Hype or Consequences?", Gary McGraw, InformIT.com, June 17, 2010, available at http://www.informit.com/articles/article.aspx?p=1597476
[23] "Why Can't Johnny Develop Secure Software?"
[24] "Google Pays Cash to Hackers for Finding Web Security Flaws," Jared Newman, PC World, Nov. 2, 2010, available at http://www.pcworld.com/article/209548/google_pays_cash_to_hackers_for_finding_web_security_flaws.html
[25] Facebook dangles cash rewards for bug reports," Dan Goodin, The Register, July 29, 2011, available at http://www.theregister.co.uk/2011/07/29/facebook_bug_bounties/
[26] "Mozilla Web Application Security Bug Bounty FAQ," available at http://www.mozilla.org/security/bug-bounty-faq-webapp.html#eligible-bugs
[27] "Changing botnets, spam & software vulnerabilities," Derek Manky, CXOToday, May 21, 2010, available at http://www.cxotoday.com/story/changing-botnets-spam-software-vulnerabilities/ (herein after "Changing botnets, spam & software vulnerabilities")
[28] "Do You Always Need to Install Software Updates?" Cor Rosielle, Infosec Island, Sept.13, 2011, available at https://www.infosecisland.com/blogview/16401-Do-You-Always-Need-to-Install-Software-Updates.html (hereinafter "Do You Always Need to Install Software Updates?"
[29] "1.91% of PCs are fully patched!" Jakob Balle, Secunia, Dec. 3, 2008, available at http://secunia.com/blog/37/
[30] "Web Browsers, Desktop Software Top 'Dirty Dozen' Apps List"
[31] "Automated patches necessary for true endpoint security," Greg Masters, Nov. 10, 2010, SC Magazine, available at http://www.scmagazineus.com/automated-patches-necessary-for-true-endpoint-security/article/190574/
[32] "Software Exploits Running Wild"
[33] "Do You Always Need to Install Software Updates?" Cor
[34] "Changing botnets, spam & software vulnerabilities"

Focus, then, should be shifted away from patches and toward ensuring that software is securely coded from the get-go. Every software vendor "offers a different approach to automating the development of secure code, but they generally agree that the idea is to help developers identify and remediate the most common coding errors and fix them on the spot, during development, rather than waiting until after the code is complete."[35]

For a "preventative medicine" approach to work, however, software producers must be held accountable (and perhaps liable) for their flawed products. This means there must be a revision to the End User License Agreements – and users must more fully acknowledge these agreements. Companies are always developing new products for a market that is constantly demanding new technologies. If the market – and more concisely, the purchasers – demand security, the companies will be forced to deliver.[36] This means not only must EULA change, but software development budgets must also change.

One way this could get done quicker and more uniformly would be if the government demanded such security for its software uses. "If we can get every one of our government agencies – and this is important for business, too – to have this continuous capability to drive all the vulnerabilities, that's the first step in being ahead of our adversaries. Today we look for the vulnerabilities once a year, once a decade – it depends on the system and the requirements – and our adversaries find our vulnerabilities when they need them in between those times," Roger Thorton says.[37] The White House currently has a legislative proposal that would "mandate that all agencies continuously monitor federal computer equipment and software with automated tools to spot threats faster. Current law requires agencies to fill out paperwork to confirm compliance with security safeguards only once a year."[38] If governments demand software security for their own uses, the idea is that such secure practices would trickle into the private sphere, as well.

Again, this approach is reactive. Rather than looking for security loopholes once software is in use, there must be a demand for secure software from the beginning. Vehicles must pass state inspections prior to being allowed on the highway; if an illegal car is on roadways, a citation is issued or, if the vehicle doesn't pass inspection because of operational problems, it must be fixed prior to passing inspection and being allowed on roadways. Software use should perhaps be no different – operating systems and applications should be safe, secure and in working order before being unleashed to the public or put on the Internet highway. Manufacturers producing unsafe software should be held accountable. Users who do not meet minimum security standards should also be held accountable.

The government has responded to the increasing problems of software security and attempted to be proactive by creating, along with MITRE, a "new scoring system to help evaluate software projects against a list of common programming errors." The Common Weakness Scoring System, as it's known, "are strictly programming errors that DHS and MITRE are hoping to help developers avoid in future projects. In addition to the list of 25 programming errors, the initiative also includes guidance on how organizations can score their own software projects against the list."[39] Government can ensure these, and other errors, aren't repeated by mandating a third-party inspect software, either on a regular basis as with vehicle registrations or periodically as the FDA or OSHA does with its clientele.

---

[35] "Why Can't Johnny Develop Secure Software?"
[36] "Secure software development key to business, says Microsoft," Warwick Ashford, Computer Weekly, June 30, 2011, available at http://www.computerweekly.com/Articles/2011/06/30/247146/Secure-software-development-key-to-business-says-Microsoft.htm
[37] "Software Security a Growing Concern"
[38] "Threat of destructive coding on foreign-manufactured technology is real"
[39] "DHS Unveils Effort to Focus on Software Security," Dennis Fischer, Threat Post, June 27, 2011 available at http://threatpost.com/en_us/blogs/dhs-unveils-effort-focus-software-security-062711

Because the market will always be demanding the newest technologies, companies are likely to continue to be crunched for time. But, with more software security degree programs being offered and many of the courses including a focus on software security, the gap in knowledgeable and qualified coders may begin to diminish. And because the rarity of these degrees is declining,[40] the amount of software that is being developed overseas may also lower. This could counter not only those threats associated with customization, but also decrease the reliance on imported software. Organizations may also consider relying on in-house software developers.

Secure software isn't the end-all answer, though. "What we know from experience is if we secure the computers and the networks, the bad guys will go after the software. And if we secure the software, they'll go after the networks and computers. The system has to be secure," Thorton says.[41]

[40] "Software Security Degree Programs," Fergal Glynn, Information Security Short Takes, June 2011, available at http://www.shortinfosec.net/2011/06/software-security-degree-programs.html
[41] "Software Security a Growing Concern"