Vulnerability Trends - Measuring Progress

Rick Kuhn and Chris Johnson

In the first installment of this column (Jan/Feb 2009) [1], we reviewed trends in software vulnerabilities. Now that roughly one Moore's Law generation has passed, it seems appropriate to revisit vulnerability trends, with an eye towards measuring the IT industry's progress in building security into products. What is the state of security engineering today? Are we as an industry making progress? What are prospects for the future? To address these questions we analyze data from the National Vulnerability Database (NVD). Operated by the National Institute of Standards and Technology (NIST) and the Department of Homeland Security (DHS), the NVD provides fine-grained search capabilities for all publicly reported software vulnerabilities since 1997, a total of 41,810 vulnerabilities for more than 20,000 products. Many vulnerabilities affect a large number of products, for example where the fault occurs in a library function.

As we will see, the news is encouraging despite increasingly sophisticated attackers. At one time the typical attacker was most likely a petty criminal or cracker, but today systems are targeted by organizations with significant resources. Software may be vulnerable as a result of design or implementation errors, but also because better attacks have been discovered, just as armor that is proof against small arms fire may be penetrated by more powerful munitions. Eliminating flaws that lead to vulnerabilities is vitally important for IT systems, but making those that remain less potentially damaging and harder to exploit can improve security as well.

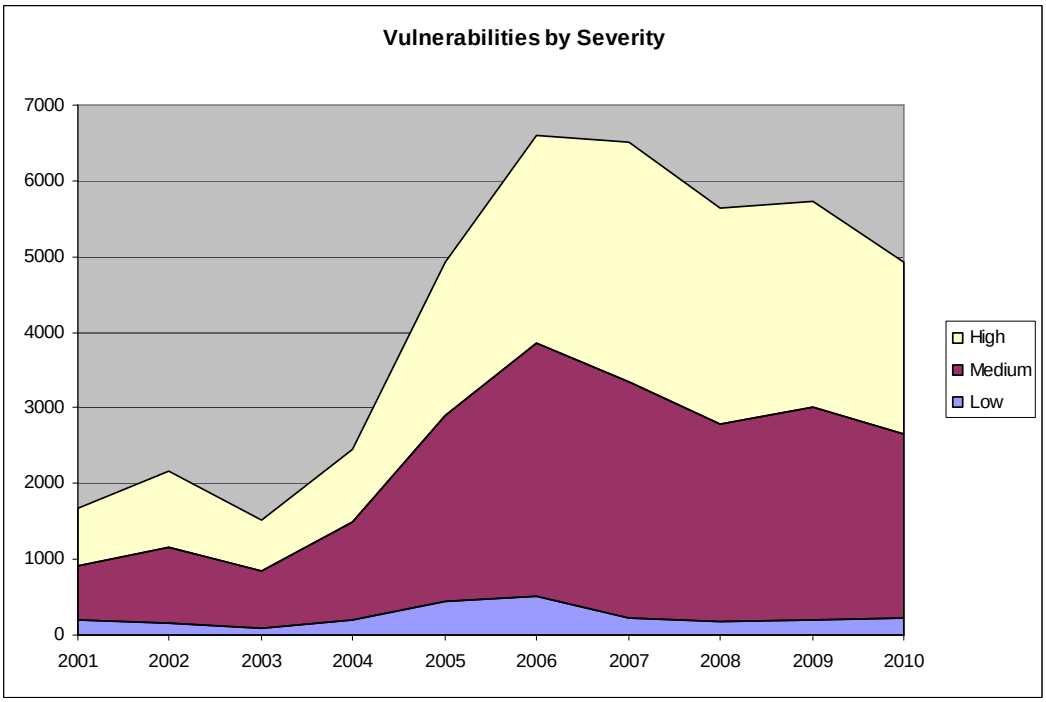

Figure 1. Vulnerabilities have declined about 30% since 2006, but the proportion of High, Medium and Low severity has remained relatively constant.

Figure 1 shows that real progress is being made. Since 2006, vulnerabilities have declined by 26%, despite the ever-growing number of applications. (It is important to note that figures for 2010 are projected based on four months, January to April. NVD data vary little by quarter and are approximately normally distributed with a standard deviation of 3 percentage points.) Separate

figures are given in Figure 1 for High, Medium, and Low severity vulnerabilities, based on the Common Vulnerability Scoring System (CVSS) [2], which assigns a numeric composite score that considers the impact on confidentiality, integrity, and availability. Impact for these three aspects of security may be none, partial, or complete, and scores combine these impacts into a single number. Essentially, a Low score means limited adverse effect on the organization; Medium is a serious adverse effect; and High is considered catastrophic. Although vulnerabilities have been declining, it is apparent from the data that the proportion at each severity level has changed relatively little in the past 10 years.

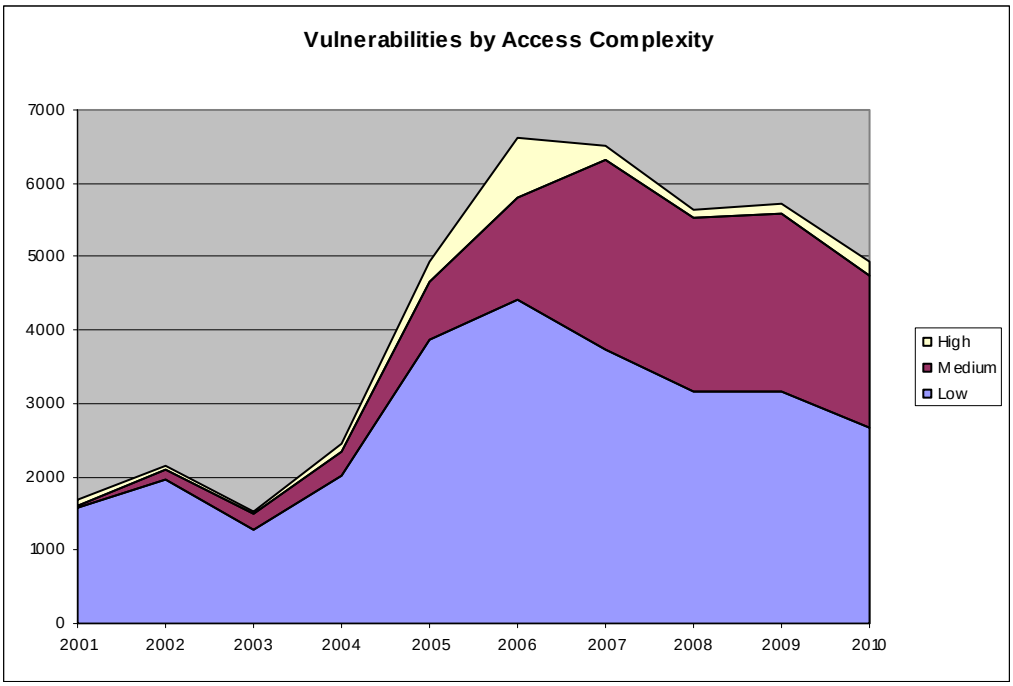

Figure 2. Before 2006 almost all vulnerabilities were easy to exploit.

In Figure 2 we see additional progress over the past decade. Until 2006, vulnerabilities rated as low access complexity tracked closely with the total, i.e., almost all were easy to exploit. For this component of the CVSS, a Low attack complexity means one that involves no specialized conditions, such as a default configuration or an attack that can be conducted manually and requiring little skill. (As a conservative measure, the Low complexity totals include cases where there is not sufficient information to assign a category. This situation may occur when the mechanics for exploiting a vulnerability are not well understood or in cases where a vendor chooses not to fully disclose detailed information for a vulnerability.) Medium complexity means that access conditions are somewhat specialized, for example, involving non-default conditions or requiring specific system knowledge in advance. High complexity refers to specialized access conditions, such as rarely seen configurations or race conditions with a narrow window. Since 2006, Low access complexity vulnerabilities have dropped as a percentage of the total. In other words, developers appear to have begun making significant progress in 2005 in making it more difficult to attack systems, but this progress has leveled off, with Low and Medium complexity vulnerabilities accounting for roughly 55% and 40% of the total respectively.

The IT environment has become increasingly complex in the past 10 years, primarily with the growth of internet commerce. The number of web servers on the internet has increased from

roughly 26 million in 2001 to 74 million in 2006, to 205 million as of April 2010 [3]. As shown in Figure 3, the number of vulnerabilities that are locally exploitable has fluctuated around an annual average of about 500 for the 10 year period, while network based vulnerabilities have increased by a factor of four or more. In this case, Local access means either physical access to the machine or availability of a shell. Network access, often referred to as "remotely exploitable" means that an attack does not require local access or local network access. An Adjacent Network refers to a local network such as a TCP/IP subnet, wireless, or Bluetooth. Adjacent Network vulnerabilities are nearly invisible in Figure 3 but are detailed in Table 1. It is interesting that these vulnerabilities peaked in 2007 and now appear to be declining, suggesting that developers are implementing appropriate controls when integrating these technologies into applications. In today's increasingly networked world, the overall reduction in vulnerabilities of the past few years is encouraging.

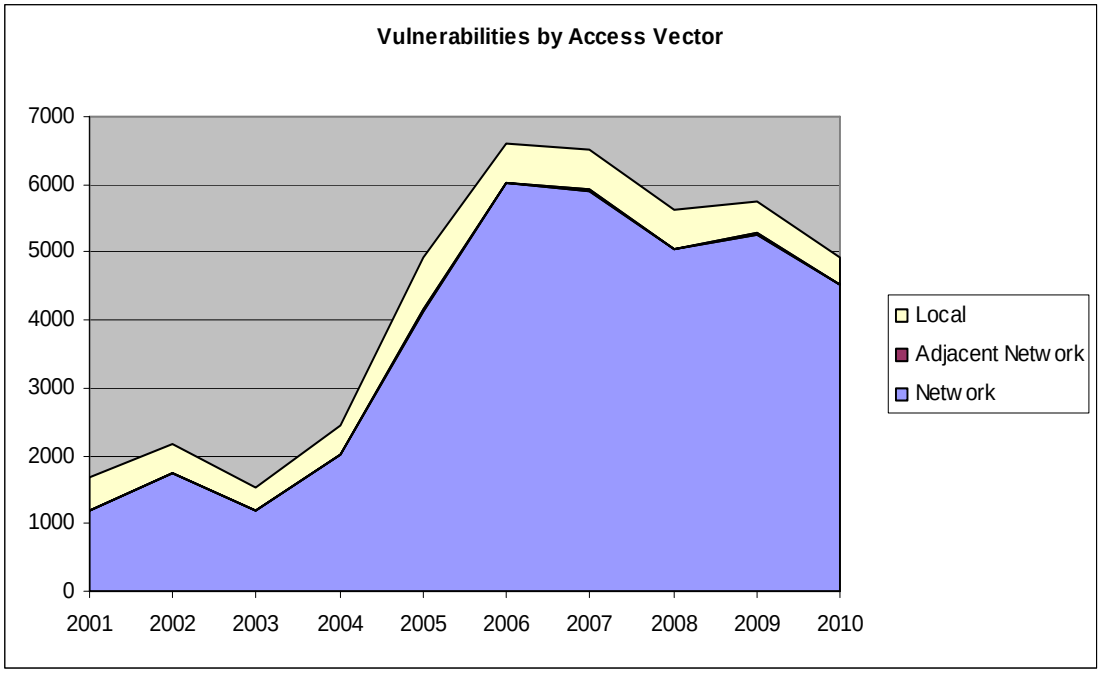

Figure 3. The increase in vulnerabilities has been almost entirely in those exploitable by network access.

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Network	1193	1754	1194	2028	4140	6012	5887	5037	5270	4518
Adjacent Network	0	0	2	0	4	18	38	10	16	0
Local	484	402	331	424	788	578	589	585	447	405

Table 1. Adjacent network vulnerabilities remain a small component.

Despite solid progress, it is too soon to declare victory. The proportion of high and medium severity vulnerabilities has changed little in a decade (Fig. 1), and roughly half of the vulnerabilities are easy to exploit (Fig. 2), suggesting that many developers are still ignoring security basics. In a separate analysis, NIST looked at more than 3,000 NVD reports for denial of service vulnerabilities and found that 94.7% involved only a single condition, nearly always a too-long input string [4] (a few were exploitable only when two or three conditions held).

As we have seen in this brief review, software developers are making real progress in securing systems, but broader adoption of secure programming practices, even simple measures such as input validation, could bring more dramatic improvements. Luckily large developers are rapidly improving their practices. The NVD contains data on thousands of software products, but only a few browsers, servers, and office packages account for the vast majority of software in use. A preliminary analysis – which we will expand and report on in a future column – shows that easily exploitable vulnerabilities are being reduced significantly in applications with large user bases, to the point where a third or fewer of their vulnerabilities are in the easily-exploitable Low Access Complexity category. Clearly there is much room for progress, but it appears that developers are taking security engineering seriously and meeting with success.

Acknowledgments: We are very grateful to Harold Booth for consultations on the NVD data.

We identify certain products in this document, but such identification doesn't imply recommendation by the US National Institute of Standards and Technology or other agencies of the US Government, nor does it imply that the products identified are necessarily the best available for the purpose.

References

1. R. Kuhn, H. Rossman, S. Liu, "Introducing 'Insecure IT'", *IEEE IT Pro* Jan/Feb 2009

2. P. Mell, K. Scarfone, S. Romansky, A Complete Guide to the Common Vulnerability Scoring System Version 2.0 http://www.first.org/cvss/cvss-guide.html

3. Netcraft, "April 2010 Web Server Survey", www.netcraft.com

4. NIST (2010). Number of Interactions Involved in Software Failures - empirical data, http://csrc.nist.gov/groups/SNS/acts/ftfi.html

Sidebar (if there is space)

Definitions from NIST IR 7298, Glossary of Key Information Security Terms.

- Exploit: a program [or method] that allows attackers to break into a system.
- Threat: Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.
- Vulnerability: An error, flaw, or mistake in computer software that permits or causes an unintended behavior to occur.