

A contemporary approach to network vulnerability assessment

Andrew Stewart



Andrew Stewart

Modern network vulnerability assessment tools suffer from an “information overload” problem

The roots of this problem lie in the fact that the competitive and commercial drivers that shaped the early market for network vulnerability assessment products continue to have influence today.

These historical goals no longer reflect the needs of modern businesses, however. A shift in requirements has occurred, due to the now widespread use of patch management technologies.

In this paper I describe the advantages in using patch management technologies to gather vulnerability data. I also propose a lightweight method for network vulnerability assessment, which does not rely on signatures, and which does not suffer from information overload issues.

The effect of historical market forces

In the formative years of the commercial network vulnerability assessment market, the number of vulnerability “checks” that vulnerability assessment tools employed was seen as a key metric by which competing products could be judged. The thinking was that the more checks that were employed by a tool, the more comprehensive it would be, and thus the more value its use would provide.

Vendors were also evaluated on how quickly they could respond to newly publicised security vulnerabilities. The quicker a vendor could update their product to incorporate the checks for new vulnerabilities, the better they

were perceived to be. In some respects this is similar to the situation today where software vendors are judged by the security community on their timeliness to release patches for security problems that are identified in their products.

The market's desire for a comprehensive set of vulnerability checks to be delivered in a timely fashion spurred the manufacturers of network vulnerability

“A network vuln. scanner report can be as thick as a phone directory”

assessment tools to incorporate ever-larger amounts of checks into their products, and to do so with increasing rapidity. Some vendors even established research and development teams for the purpose of finding new vulnerabilities. (An R&D team was also an opportunity for vendors to position and publicize themselves within the marketplace.) Vendors were said to have sometimes sought competitive advantage through duplicitous means, such as by slanting their internal taxonomy of vulnerability checks in order to make it appear that

they implemented more checks than in reality.

A common practice was for vendors to create checks for any aspect of a host that can be remotely identified. This was often done regardless of its utility for security. As an example, it is not unusual for network vulnerability scanning tools to determine the degree of predictability in the IP identification field within network traffic that a target host generates. While this observation may be useful in certain circumstances, the pragmatic view must be that there are far more influential factors that can influence a host's level of vulnerability. Nonetheless, network vulnerability assessment products typically incorporate hundreds of such checks, many with similarly questionable value.

Information overload

The result of these competitive drivers has been that when a network vulnerability scanner is run against any network of reasonable size, the printout of the report is likely to resemble the thickness of a telephone directory. An aggressive approach to information gathering coupled with an ever increasing set of vulnerabilities results in an enormous amount of information that can be reported. Such a large amount of data is not only intimidating, but it severely limits the ability to make key insights about the security of the network. The question of “where to begin?” is a difficult one to answer when you are told that your network has 10,000 “vulnerabilities”.

Vendors of network vulnerability assessment products have tried to address this information overload problem in several ways. One approach has been to attempt to correlate the output of other systems (such as intrusion detection systems) together with vulnerability data to allow results to be prioritised. Another approach has been to try and “fuse” data together on the basis of connectedness, in order to

increase the quality of data at a higher layer. These approaches have spawned new categories of security product, such as “Enterprise Security Management” (ESM), “Security Information Management” (SIM), and “Vulnerability Management”.

But rather than add layers of abstraction (and products to buy), the solution would logically lie in not gathering so much data in the first place. This has now become a viable strategy, because of the capabilities provided by modern patch management technologies.

The rise of patch management

The widely felt impact of Internet worms has opened the eyes of businesses to the importance of patching systems. Host-based patch management products such as Microsoft’s SMS (Systems Management Server) and SUS (Software Update Services) are now in wide deployment, as are other commercial and freeware tools on a variety of platforms. See for example, PM (2005) and Chan (2004).

In many respects, this increased focus on patch management has diminished the traditional role of network vulnerability assessment tools. If the delta between current patch status and the known set of vulnerabilities is already being directly determined on each individual host, then there is less need to use a network vulnerability assessment tool to attempt to collect that same information (and to do so across the network and en masse).

An advantage here is that it is a relatively straightforward task for a software agent running on a host to determine the host’s patch level. A network vulnerability scanner has to attempt to remotely infer that same information, and this task is made more difficult if the vulnerability scanner has no credentials for the target host.

Another advantage to using a host-based model for gathering patch data is that with an ever-increasing set of vulnerability checks being built into network vulnerability assessment tools, the probability increases that a check might adversely affect a network service on a box. The result might be that the scan causes services to crash, restart, or otherwise misbehave. The days when port scanning would crash the simplistic network stack within printers and other such devices are probably behind us, but a business might rightly question the use of increasingly complex vulnerability checks to interrogate production systems.

With an ever-increasing number of checks, the impact on network bandwidth when a network vulnerability assessment tool is run also climbs. (Rate-limited and distributed scanning can help here, but these involve additional complexity.)

There are *disadvantages* to employing a host-based model, however. Products which require that an agent be installed on hosts have usually been seen as time-consuming to deploy and complex to manage. Indeed, the value proposition of network vulnerability assessment tools was, in part, that they

did not require a roll-out of host-based agents. With the now widespread use of agent-based patch management technologies, this barrier has been overcome.

Given the advantages in using a host-based model to gather patch status information, do network vulnerability assessment tools still have a role to play? In discovering new vulnerabilities, or for discovering vulnerabilities in bespoke applications (such as Web applications), network vulnerability assessment tools clearly add value. But this is somewhat of a niche market. These are not activities that businesses typically wish to perform against every device within their network environment, or on a regular basis. (Scanning a DHCP allocated network range provides little value if the DHCP lease time is short, just as one example.)

A modern approach

It is a widely held belief amongst security practitioners that the majority of security break-ins take advantage of known vulnerabilities. While there is no concrete evidence for this claim, on an intuitive basis it is probably correct. In most cases, the patch for a known vulnerability already exists, or the vendor affected is in the process of creating the patch. (In that latter scenario, the version numbers of the particular operating systems or applications that are known to be vulnerable are usually known, even if the patch itself is not yet available.)

A patch management solution can determine the presence or absence of patches on hosts, and can also identify the current version number of operating systems and installed applications. A patch management solution can therefore be used to determine vulnerability status. The depth of reporting that modern patch management tools provide in this area has in many respects already surpassed the capabilities of conventional network vulnerability assessment tools. This is possible because of the advantages inherent in a host-based model.

service	count
telnet	20
ssh	79
rlogin	3
http	52
https	26
ldap	8
vnc	9
ms-term-serv	30
pcanywheredata	2
irc	1

Table 1: Display of services running on hosts

os	count
HP embedded	26
Cisco embedded	33
Linux	42
Windows	553
OpenBSD	1
No match	2

Table 2: Number of operating systems found in a particular network

However, host-based patch management tools only have visibility into the hosts onto which an agent has been installed. Organizations still need some form of network assessment in order to detect changes that lie outside the visibility of their patch management infrastructure.

I suggest that this task can be accomplished using traditional network interrogation techniques, and does not require a library of vulnerability checks. Well-documented techniques exist for gathering data related to the population of a network, the services running on hosts within the network, and the identification of operating systems type

“what data is still valuable to gather across the network?”

(Fyodor, 1997, 1998). These techniques do not require a constant research effort to develop new vulnerability checks. A port scanner written in 1990 could still be used today, whereas a vulnerability scanner from the same year would be considered woefully inadequate because it has no knowledge of modern vulnerabilities.

The information that can be gathered using these relatively simple techniques has enormous utility for security. Consider [Table 1](#), which displays data

gathered on the number of different services running on hosts within a network.

The policy on this network is to use Microsoft's Terminal Services for remote administration, and therefore the two installations of pcAnywhere and the nine installations of VNC that were detected are policy violations that need to be investigated then corrected. Running pcAnywhere or VNC is not a security “vulnerability” per se, but remote administration software certainly has a security implication. That is the difference between looking for specific vulnerabilities and gathering general data on the network.

As a further example, the IRC server that was found on the network would probably raise the eyebrow of most security practitioners.

Note how simple it is to perform this analysis, in contrast to having to wade through hundreds of pages of vulnerability assessment report. If a patch management solution is being used to detect weaknesses in the patch status of hosts, then this is the type of data that it is valuable to collect across the network. This is not traditional vulnerability assessment data, but rather *foundational* data about the network.

[Table 2](#) shows data on the number of operating system types found within a particular network. Again, this data was collected using simple network information gathering techniques.

This network employs both Linux and Windows machines as its corporate standard. We can therefore say that the detection of a device running OpenBSD warrants investigation. Similarly, it would be valuable from a security perspective to investigate the two devices for

which there was no fingerprint match. An all-Linux organization might worry about the presence of a Windows 95 machine on its network (and vice-versa, of course).

This approach is well-suited for detecting the *decay* in security that computers tend to suffer over time. Most businesses employ a standard build for desktop and server machines to reduce complexity and increase ease of management, but day-to-day administrative activities can negatively impact that base level of security. Temporary administrative accounts are created but then forgotten; services such as file transfer are added for ad hoc purposes but not removed, and so on. A vulnerability scanner is overkill for detecting this kind of “policy drift”. By employing more simplistic network information gathering techniques, the run time of a scan can be reduced, as can the impact on network bandwidth. The duration of the information gathering loop is shortened, and this allows results to be provided quicker, which itself reduces risk by allowing remediation activities to be carried out sooner.

Conclusions

Patch management technologies and processes now deliver to businesses the core capability of traditional network vulnerability assessment tools; namely, the identification of vulnerabilities that are present due to missing patches. Patch management solutions can be used to accomplish this task by identifying the delta between the set of patches for known vulnerabilities and the current patch status of hosts within the environment.

For network-wide vulnerability assessment, the question that businesses need to ask is: what data is it still *valuable* to gather across the network? There is little value in employing a noisy, bandwidth-consuming network vulnerability scan to interrogate production systems with an ever-increasing number of vulnerability checks, when

patch status data is already being collected through patch management activities.

Employing simple network information gathering techniques in this supplementary role is easier, takes less time, has less impact on network bandwidth, does not require a constantly updated set of vulnerability "checks", and provides more intuitive results.

About the author

Andrew Stewart is a Senior Consultant with a professional services firm based in Atlanta, Georgia.

References

Chan (2004), "Essentials of Patch Management Policy and Practice", Available: <http://www.patchmanagement.org/pmessentials.asp>

Fyodor (1997), "The Art of Port Scanning", Phrack Magazine, Volume 7, No. 51, September 01, 1997.

Fyodor (1998), "Remote OS detection via TCP/IP Stack FingerPrinting", Phrack Magazine, Volume 9, No. 54, 25th December, 1998.

PM (2005), Mailing list archive at <http://www.patchmanagement.org>

Crypto race for mathematical infinity

Sarah Hilley



Sarah Hilley

A newly emergent country has begun to set the pace for cryptographic mathematicians...

Chinese infosec research efforts are fixated on cryptography and researchers are already producing breakthroughs. A group of researchers from Shandong University in China stunned the established crypto community at the RSA conference in February by breaking the integral SHA-1 algorithm used widely in digital signatures. This SHA algorithm was conceived deep within the womb of

Even more proof of the hive of crypto activity in China is that 72% of all cryptography papers submitted to the Elsevier journal, *Computers & Security* last year hailed from China and Taiwan. And cryptography papers accounted for one third of all the IT security research submitted to the journal.

The Chinese are determined to get into the subject, says Mike Walker, head of Research & Development at Vodafone, who studied cryptography at Royal Holloway College, London. "If you attract the best people from one fifth of the world's population, you are going to sooner or later make a big impression." Walker would like to see more young people venture into cryptography in the UK. He believes the general decline in interest in science and maths is to the detriment of the country.

But no such lack of interest is evident in China. And the achievement in cracking the SHA-1 hash function is an earthquake of a result. "The breakage of SHA-1 is one of the most significant results in cryptanalysis in the past

decade," says Burt Kaliski, chief scientist at RSA Security. "People didn't think this was possible."

Shelf-life

"Now there is no doubt that we need a new hash function," says Mette Vesterager, chief executive officer at Cryptico. Vesterager says a competition will probably be launched to get a new

“It is a race between mathematicians and computers”

replacement for SHA-1. Such a competition generated the Advanced Encryption Standard (AES), from two Belgians in 2000 to replace the Data Encryption Standard (DES). DES was published in 1977 and had 72,000,000,000,000 possible key variations, making it difficult to break.

NIST have now taken DES off the shelf, however. No such retirement plan has been concocted for SHA-1 yet. As of yet the outcome for the broken algorithm is still undecided. But Fred Piper, at Royal Holloway says that people will migrate away from it in the next year or so if the Chinese research is proven. In

“The Chinese are determined to get into the subject”

the US National Security Agency's cryptography labs. It was declared safe until 2010 by the US National Institute of Standards and Technology (NIST). But this illusion was shattered last month.