## Risk Assessment and Vulnerability Scoring Systems

Various systems exist which attempt to categorize and score each vulnerability. Qualys have their own scoring system as do Tripwire ® (and nCircle), but there are also the consensus-based systems, presided over by NIST, which reference the three earlier definitions of vulnerability classes. In turn these are


*Figure 4: Assigning a risk score to a vulnerability is a logical step in order to prioritize remediation work, however the scoring systems are necessarily complex and nearly always too opaque to interpret for your environment*

- ***Common Configuration Scoring System (CCSS)***, used to score the severity of security configuration-based vulnerabilities
- ***Common Vulnerability Scoring System (CVSS)***, used to score the severity of software flaw-based vulnerabilities
- ***Common Misuse Scoring System (CMSS)***, used to score the severity of software misuse-based vulnerabilities

At a high level, the intention is clear – define how potentially dangerous each vulnerability is. But that isn't such an easy assessment to make and scoring vulnerabilities starts to get very complicated, very quickly.

Each of the Common Scoring Systems factor in the context of the threat: ***'Just how likely is it that this exploit can be used?', 'How real is the exploit?', 'How available are the fixes, and how risky are they?', 'How much damage could be done using the exploit?'***

In the CCSS system the vulnerability is given a 'Base Score' based on the

- ***Access Vector*** (Local, Adjacent Network or Network)
- ***Access Complexity*** (High, Medium or Low)
- ***Authentication requirements*** (Multiple, Single or None)
- ***Confidentiality Impact*** (Complete, Partial or None)
- I***ntegrity Impact*** (Complete, Partial or None)
- ***Availability Impact*** (Complete, Partial or None)

Next, there is a ***'Temporal Score'*** applied, based on

- ***Exploitability*** (Not Defined, Unproven that exploit exists, Proof of concept code, Functional exploit exists or High)
- ***Remediation Level*** (Not Defined, Official fix, Temporary fix, Workaround or Unavailable)
- ***Report Confidence*** (Not Defined, Unconfirmed, Uncorroborated or Confirmed)

Then there is the ***Environmental Score***….do I need to go any further?!

### Do vulnerability scoring metrics work?

Consider CVE-2004-2761: The MD5 Message-Digest Algorithm is not collision resistant, which makes it easier for context-dependent attackers to conduct spoofing attacks, as demonstrated by attacks on the use of MD5 in the signature algorithm of an X.509 certificate

This one comes up a lot because it is typically reported with respect to a Web Server using a self-signed certificate. An internet-facing website needs to be operated with an SSL certificate generated using a strong signature algorithm to prevent certificate forgery/spoofing. The CVSS Score is around 5 i.e. Medium to High – Scary!

However, all manner of other web-enabled systems may well use self-signed certificates which are typically MD5-based. So if the website in question is actually a web interface to a non-business critical, internal-use only system, that doesn't hold any PII or other confidential data, and is on an internal web segment behind an internal firewall and the external internet firewall, is this still a Severity 5 Vulnerability?

## Context is everything - Intelligent compliance beats vulnerability scanning

From an academic standpoint, all the factors outlined should be taken into account as they allow a quantitative score for any vulnerability to be derived based on its qualitative attributes.

But as the consumer of the scan report you just want a High, Medium or Low severity rating - you don't need to worry too much about how the Vulnerability Score was calculated.

Or do you? Without the context of your estate and network architecture, the risk-level of a vulnerability can only be calculated on a theoretical, not empirical, basis.

Now, the point is that there are no vulnerabilities that should be ignored, but there are any number that within the context of your estate might be tolerated temporarily or permanently due to compensating controls that are in place. SCADA infrastructure components subject to NERC CIP compliance will require the highest level of security, while user workstations segregated from confidential data systems can be treated as lower priority, lower risk items.

With scan results highlighting hundreds of vulnerabilities across the estate, the last thing you need therefore is to be re-reminded every time you scan of the same known-and-acknowledged vulnerabilities. The concept of improvement-based vulnerability management starts with the need to address this issue as a key objective.

## Continuous improvement is key - Measure compliance for *your* estate with *your* hardened build standard, taking into account the context of *your* systems

For example, with a large compliance initiative, there could be any number of reasons why servers or network devices will remain in a non-compliant state for months – resource constraints, application compatibility, network architecture – so the requirement to either suspend or exclude compliance requirements for certain hosts or device groups is essential. If we think it will take us 3 months to remediate all vulnerabilities across all systems then we can set time-based milestones for minimum levels of compliance to be achieved, and in doing so, give a realistic set of targets to hit progressively over time without being repeatedly beaten on over all vulnerabilities outstanding.

Similarly there may be a need to make exceptions or adjust compliance requirements, for example, allowing permissions to additional Groups over and above the standard settings advocated by the CIS Benchmark.

Finally the ability to also extend the compliance standard to include additional file integrity monitoring checks over and above the **STIG** or other secure build standard is valuable. For example, security best practices may recommend removing or disabling unnecessary daemons and services, but you can also use your compliance audit to ensure that other essential services are enabled and running, such as encryption, syslog forwarding agents, DLP or AV products. Likewise, ensuring a functional build standard for a host is implemented and maintained in terms of installed software, filesystem structure and network settings gives a dimension of quality control that will eliminate downtime/reduce troubleshooting.