U.S. Department of State, Information Resource Management,
Office of Information Assurance, May 14, 2010

Enterprise Network Management

iPost:
Implementing Continuous Risk Monitoring at the Department of State

Version 1.5
May, 2010

United States Department of State

Introduction and Executive Summary

iPost is the custom application that continuously monitors and reports risk on the IT infrastructure at the Department of State (DoS).

OpenNet, the DoS Sensitive But Unclassified (SBU) network, serves several hundred foreign posts and domestic bureaus, and consists of 5,000 routers and switches, and more than 40,000 hosts. For foreign posts, almost all network and system administration is the responsibility of the local staff at each post. There is much more centralization domestically, but even there, the administrative responsibility is somewhat dispersed. iPost provides these local administrators with a single interface for direct access to the monitoring data for objects within their scope of responsibility. It also provides a single source for Enterprise-level management reporting across a variety of monitoring data.

The Risk Scoring program uses the data integrated into iPost from the various monitoring tools to produce a single holistic view of technical vulnerabilities. Each host and user account is scored in multiple categories, using a scoring method based on the National Vulnerability Database's scoring system for vulnerabilities, where higher scores mean higher risk. Scores are then aggregated across categories to give a risk score for a host, a site, a region, or the Enterprise. Scores are normalized so that small and large sites can be equitably compared. Letter grades, based on the normalized scores, are provided to both administrators and senior management, leveraging natural competitiveness to encourage risk reduction. Finally, there is an extensive "exception" process that accommodates anomaly situations where the risk cannot be reduced by local administrators because of technical or organizational impediments beyond local control. In such cases, risk is transferred to the proper owner and local administrators are left to deal only with those weaknesses within their control.

The Risk Scoring program at DoS evolved in three separate stages.

- Deployment of Enterprise management tools to monitor weaknesses
- Delivery of operational monitoring data to the field in an integrated application, iPost
- Establishment of a risk scoring program that fairly measures and assigns risk

The Department's Risk Scoring Program, which is implemented via iPost, was awarded the National Security Agency's 2009 Frank B. Rowlett award for Organizational Achievement, recognizing the U.S. Government organization making the most significant contribution to the improvement of national information systems security, operational information assurance readiness, or the defensive information operations posture of the United States.

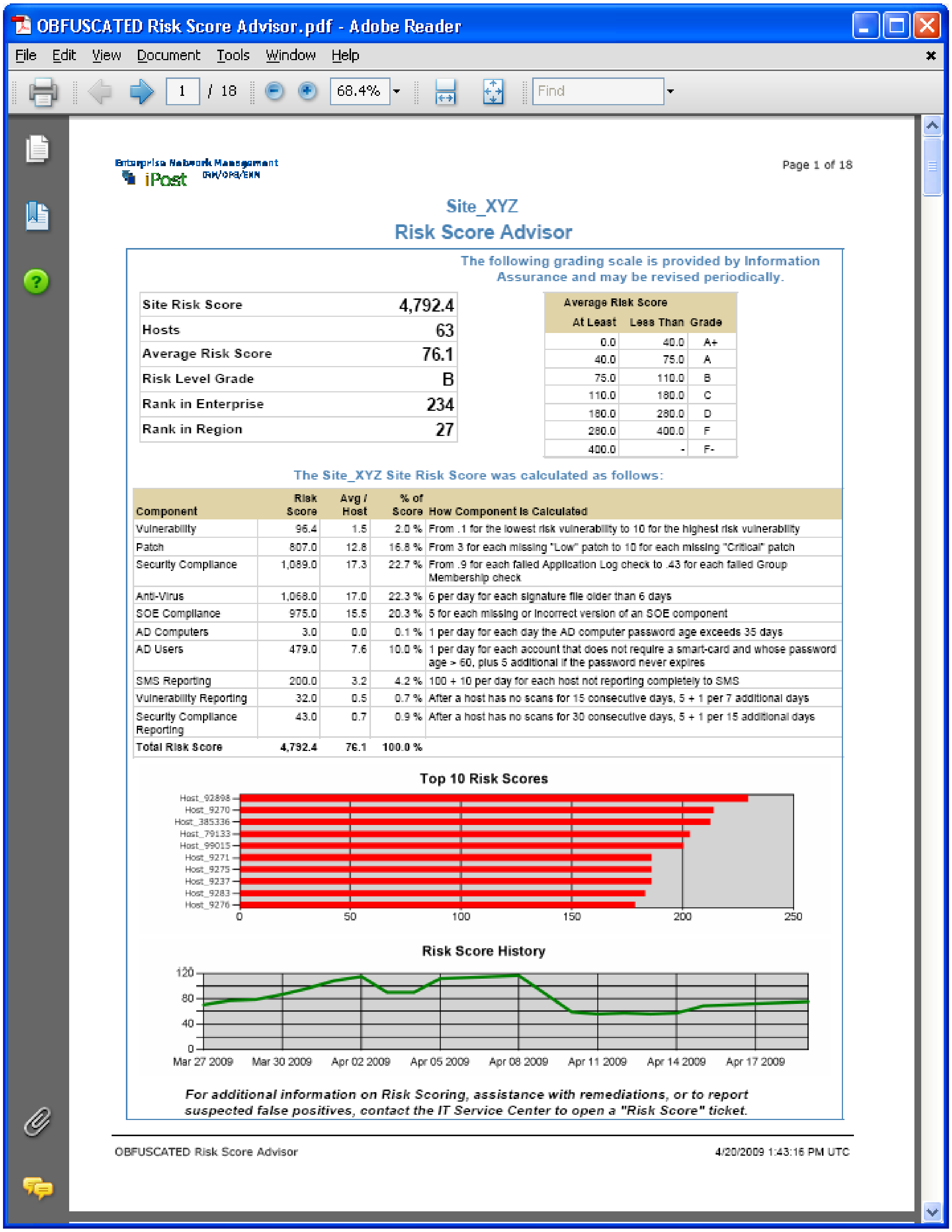
Part 1: The Risk Scoring Program

Objectives of Scoring

The Risk Scoring program at DoS is owned and directed by the Department's Computer Information Security Officer (CISO) and the Office of Information Assurance (IA) in support of its organizational mission. The program is implemented through iPost and is intended to meet the following objectives:

- Measure risk in multiple areas
- Motivate administrators to reduce risk
- Motivate management to support risk reduction
- Measure improvement
- Inspire competition
- Provide a single score for each host
- Provide a single score for each site
- Provide a single score for the enterprise
- Be scalable to additional risk areas
- Score a given risk only once
- Be demonstrably fair

Figures 1 and 2 show two pages from the Risk Score Advisor, a report that summarizes all the scoring issues for a site and provides advice on how to improve the score.


OBFUSCATED Risk Score Advisor.pdf - Adobe Reader
File Edit View Document Tools Window Help
1 / 18
68.4%
Find
Enterprise Network Management
iPost
Page 1 of 18

Site_XYZ

Risk Score Advisor

The following grading scale is provided by Information Assurance and may be revised periodically.

Site Risk Score	4,792.4
Hosts	63
Average Risk Score	76.1
Risk Level Grade	B
Rank in Enterprise	234
Rank in Region	27

Average Risk Score		
At Least	Less Than	Grade
0.0	40.0	A+
40.0	75.0	A
75.0	110.0	B
110.0	180.0	C
180.0	280.0	D
280.0	400.0	F
400.0	-	F-

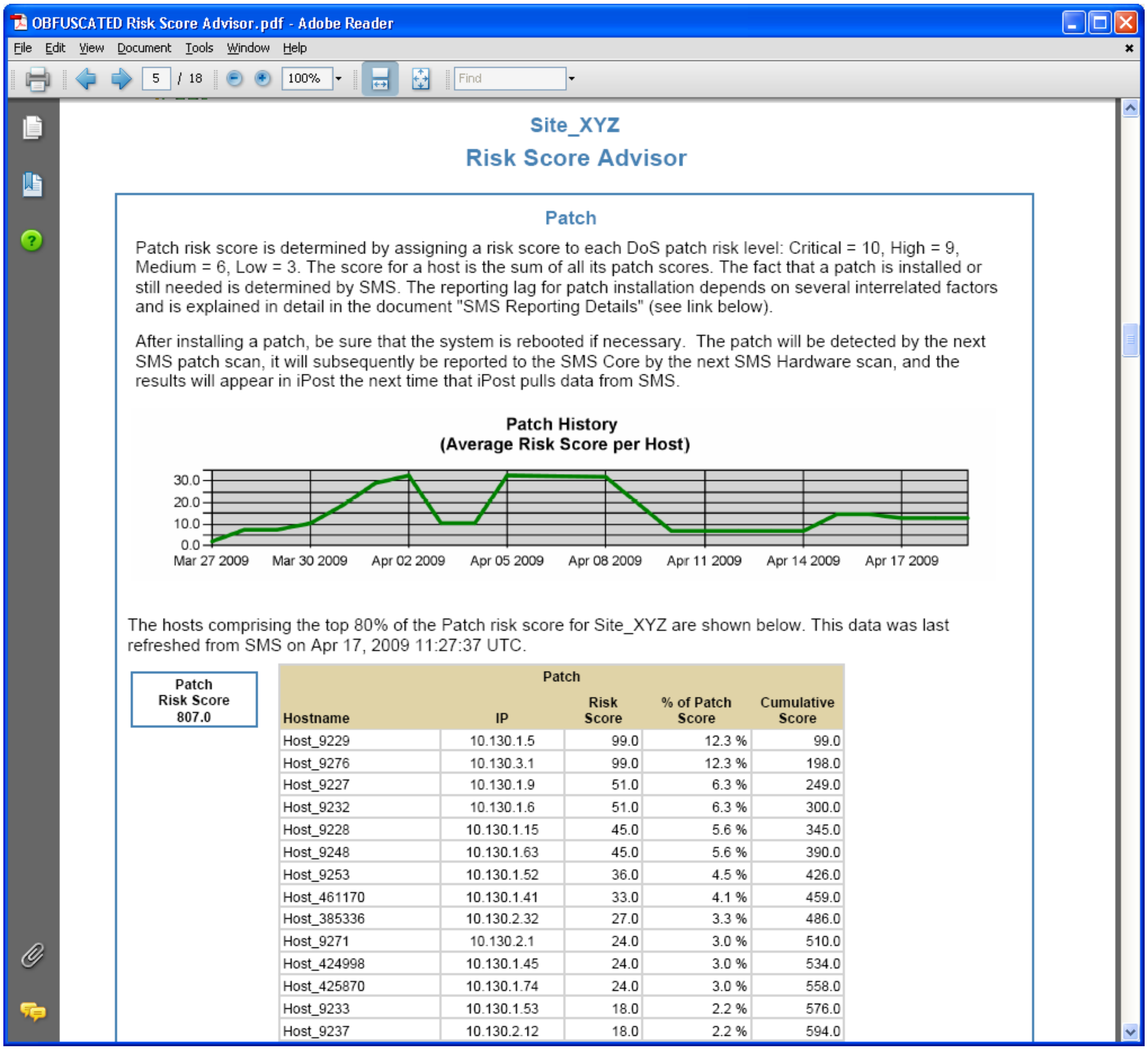
The Site_XYZ Site Risk Score was calculated as follows:

Component	Risk Score	Avg / Host	% of Score	How Component Is Calculated
Vulnerability	96.4	1.5	2.0 %	From .1 for the lowest risk vulnerability to 10 for the highest risk vulnerability
Patch	807.0	12.8	16.8 %	From 3 for each missing "Low" patch to 10 for each missing "Critical" patch
Security Compliance	1,089.0	17.3	22.7 %	From .9 for each failed Application Log check to .43 for each failed Group Membership check
Anti-Virus	1,068.0	17.0	22.3 %	6 per day for each signature file older than 6 days
SOE Compliance	975.0	15.5	20.3 %	5 for each missing or incorrect version of an SOE component
AD Computers	3.0	0.0	0.1 %	1 per day for each day the AD computer password age exceeds 35 days
AD Users	479.0	7.6	10.0 %	1 per day for each account that does not require a smart-card and whose password age > 60, plus 5 additional if the password never expires
SMS Reporting	200.0	3.2	4.2 %	100 + 10 per day for each host not reporting completely to SMS
Vulnerability Reporting	32.0	0.5	0.7 %	After a host has no scans for 15 consecutive days, 5 + 1 per 7 additional days
Security Compliance Reporting	43.0	0.7	0.9 %	After a host has no scans for 30 consecutive days, 5 + 1 per 15 additional days
Total Risk Score	4,792.4	76.1	100.0 %	

For additional information on Risk Scoring, assistance with remediations, or to report suspected false positives, contact the IT Service Center to open a "Risk Score" ticket.

OBFUSCATED Risk Score Advisor 4/20/2009 1:43:16 PM UTC

Figure 1 – Summary Page (Risk Score Advisor)


OBFUSCATED Risk Score Advisor.pdf - Adobe Reader
File Edit View Document Tools Window Help
5 / 18
100%
Find

Site_XYZ

Risk Score Advisor

Patch

Patch risk score is determined by assigning a risk score to each DoS patch risk level: Critical = 10, High = 9, Medium = 6, Low = 3. The score for a host is the sum of all its patch scores. The fact that a patch is installed or still needed is determined by SMS. The reporting lag for patch installation depends on several interrelated factors and is explained in detail in the document "SMS Reporting Details" (see link below).

After installing a patch, be sure that the system is rebooted if necessary. The patch will be detected by the next SMS patch scan, it will subsequently be reported to the SMS Core by the next SMS Hardware scan, and the results will appear in iPost the next time that iPost pulls data from SMS.

The hosts comprising the top 80% of the Patch risk score for Site_XYZ are shown below. This data was last refreshed from SMS on Apr 17, 2009 11:27:37 UTC.

Patch Risk Score
807.0

Patch				
Hostname	IP	Risk Score	% of Patch Score	Cumulative Score
Host_9229	10.130.1.5	99.0	12.3 %	99.0
Host_9276	10.130.3.1	99.0	12.3 %	198.0
Host_9227	10.130.1.9	51.0	6.3 %	249.0
Host_9232	10.130.1.6	51.0	6.3 %	300.0
Host_9228	10.130.1.15	45.0	5.6 %	345.0
Host_9248	10.130.1.63	45.0	5.6 %	390.0
Host_9253	10.130.1.52	36.0	4.5 %	426.0
Host_461170	10.130.1.41	33.0	4.1 %	459.0
Host_385336	10.130.2.32	27.0	3.3 %	486.0
Host_9271	10.130.2.1	24.0	3.0 %	510.0
Host_424998	10.130.1.45	24.0	3.0 %	534.0
Host_425870	10.130.1.74	24.0	3.0 %	558.0
Host_9233	10.130.1.53	18.0	2.2 %	576.0
Host_9237	10.130.2.12	18.0	2.2 %	594.0

Figure 2 - Patch Score Details (Risk Score Advisor)

Raw Score

A typical scoring method deals in percentages, with 100% being a perfect score. iPost had used such a scoring method for some time, but there were computational and conceptual issues when attempting to combine scores for objects and compare scores for different areas. Especially problematic were issues around objects that were not properly reporting the very data needed to score them.

Risk Scoring, on the other hand, assigns a score to each vulnerability, weakness, or other infrastructure issue based on its risk. The higher the score, the greater the risk; a perfect score is zero. This simple concept is very powerful, because the score for an object can be defined simply as the total of the scores of all its weaknesses. The score for a set of objects, e.g., for a site, is the total of the scores of all its objects. The score for the Enterprise is the total of all the scores of all the sites in the Enterprise.

Score for an aggregate = SUM(scores of the constituent parts)

Scoring Components

The risk scoring program is built on a set of areas of interest, each of which is known as a scoring ***component***. A scoring component represents an area of risk for which measurement data is readily available. The method used to score each component is unique to that component but is based on an overall view that a specific numerical score in one area is about the same overall risk as a similar numerical score in another area.

Score for an object = SUM(scores of its components)

Component selection at DoS was also influenced by organizational mission. Data on detection of vulnerabilities was readily available and was an obvious scoring component. However, data on the installation of patches to the Operating System and specific well-known software products was independently available; vulnerabilities and patches were managed by two different organizations using two different tools. Since every patch remediates a known vulnerability, there was overlap in this data that would result in double scoring if both were used. This was accommodated by making “patch” and “vulnerability” separate scoring components but also discarding the vulnerability data associated with patches already in the scope of the Patch Management program.

The issue of certain objects not correctly reporting the data required to compute a score for a component is handled by treating “not reporting” as a separate component to measure the risk of the unknown. Much of the data at DoS is collected by Microsoft System Management Server (SMS) through client agents installed on each host. Occasionally these agents are incorrectly installed or become otherwise broken, so there is a separate component for “SMS Reporting.” Vulnerability and Security Compliance data is collected – in separate scans – by Tenable Security Center. When scans are missed for some reason, hosts incur separate risk scores.

The complete set of scoring components at DoS is as follows:

Component	Abbre-viation	What is Scored	Source
Vulnerability	VUL	Vulnerabilities detected on a host	Tenable
Patch	PAT	Patches required by a host	SMS
Security Compliance	SCM	Failures of a host to use required security settings	Tenable
Anti-Virus	AVR	Out of date anti-virus signature file	SMS
SOE Compliance	SOE	Incomplete/invalid installations of any product in the Standard Operating Environment (SOE) suite	SMS
AD Users	ADU	User account password ages exceeding threshold (scores each user account, not each host)	AD
AD Computers	ADC	Computer account password ages exceeding threshold	AD
SMS Reporting	SMS	Incorrect functioning of the SMS client agent	SMS
Vulnerability Reporting	VUR	Missed vulnerability scans	Tenable
Security Compliance Reporting	SCR	Missed security compliance scans	Tenable

There are still other prospective risk scoring components that may be implemented in the future.

Prospective Component	What Would Be Scored
Unapproved Software	Every Add / Remove Programs string that is not on the “official” approved list
AD Participation	Every computer discovered on the network that is not a member of Active Directory
Cyber Security Awareness Training	Every user who has not passed the mandatory awareness training within the last 365 days

Grace Period

For a given scoring component, the score may be assessed immediately upon detection of a problem or deferred for a short time. This is measured in days and is based on how frequently the underlying monitoring tools report and other reasons of practicality. For example, it may take SMS several days to discover and configure a host, once the host is added to Active Directory, so the “SMS Reporting” score has a 5-day grace period. Another example is how patches are scored: Patches are tested centrally and made available in a standard way to the entire Department. Each patch has a due date, based on its risk level, by which it is required to be installed. There is no patch score for a given patch until its due date has passed.

Aging

For a given scoring component, the risk it measures may increase over time. As a result, some components have ***aging*** built in, so that the assessed score increases over time. The rate of increase depends on the component. In order to age a score, the component must have an identifiable date associated with it. For example, the "Anti-Virus" score is based on the date of the current signature file. The score ages because new vulnerabilities are introduced into the wild daily. All age measurements are in terms of days.

Average Score

The type of score described above is called the ***raw score*** to distinguish it from the ***average score***. The average score takes into account that the raw score for a site with many objects would almost certainly be higher than the raw score for a site with fewer objects. The average score eliminates site size as a consideration and is obtained by the formula:

Average Score = Raw Score / #hosts

The number of hosts was selected as the denominator for this average because (a) most of the components actually score hosts, and (b) it is a good, simple, measure of size and complexity of a site. Even when assigning raw scores to user objects, the raw score is still averaged over the number of hosts. By using this common denominator for all averages, the average score for a site can be equivalently computed in either of two ways:

Average score for an aggregate = Aggregate raw score / Aggregate #hosts

or equivalently

Average score for an aggregate = SUM(average scores of its scoring components)

Letter Grades

While the average score provides a way to compare sites (or any aggregates) of varying sizes, it is not intuitive that a site score of 50.0 is good – or is it bad? In particular, the score would be meaningless to those who do not work with scores on a frequent basis.

To provide a more intuitive way for management to track progress, a global grading scale is managed by IA to assign a familiar letter grade to an average score. The possible grades include the typical A-F, but also include A+ for especially low scores and F- for especially high scores. The scale can be modified periodically to adjust for changing conditions without modifying any other part of the scoring program.

Fairness and Exceptions

It was acknowledged during piloting the program that while the original intention of risk scoring was to measure risk, it was nonetheless being perceived as assessing performance of administrators and their management. Therefore, fairness had to be built into scoring.

Once fairness was established, administrators took the scores much more seriously and thrived on competition with their peers and pride in a job well-done.

An example of a fairness-based modification that was made directly in the scoring methodology is the following:

> Tenable, the scanner for Security Compliance, can return three types of results when checking a security setting: PASS means the setting conforms to the required template; FAIL means the setting does not conform; and NO CHECK means the scanner, for one of a number of reasons, was unable to determine a PASS or FAIL result. Originally, NO CHECK was scored as a FAIL, based on the principle it is best to assume the worst when dealing with security. However, some of the reasons for NO CHECK were due to issues with the scanning tool itself, and it was impossible to separate these cases from those that could be resolved by local administrators. Therefore, the scoring algorithm was modified so that a NO CHECK no longer results in a score increase.

The issue of fairness arose in other contexts, and the necessity of providing a scoring ***exception*** in certain instances was acknowledged and implemented. An exception is a way to track a real risk, yet remove the associated score from the responsibility of a local administrator who cannot address the problem. In essence, the risk is transferred to the organization responsible for fixing the problem. The most compelling and prevalent type of exception at DoS is for a software vulnerability.

Vulnerabilities are initially identified by the software vendor, industry, or a trusted government source. They are quickly assessed and find their way into the National Vulnerability Database and the scanning engines of commercial vulnerability scanners. Originally, software vulnerabilities were scored immediately upon detection with the stated remediation to “upgrade the product to version x.y”. However, administrators are prohibited by DoS policy from installing unapproved versions of products, and the approval process can be complex and lengthy when the upgrade is a major one. Thus, for a time, sites were being scored for the vulnerability but were unable to remediate it. In such cases, IA now approves global exceptions. The score for such a vulnerability is not included in the scores for any sites. Instead, the score is associated with the organization responsible for having the upgrade approved. The exception then expires a short time after this approval, giving local administrators time to install the upgrade before scoring resumes.

There is also a formal process for individual sites to submit requests to IA for local scoring exceptions. An example of an approved local exception is a single Microsoft patch that “broke” a certain critical financial application running on a single host. Since the application was not owned by DoS, IA approved a scoring exception so that this one host is not assessed a score for missing this one patch.

Exceptions can be implemented for any scoring component. The hosts to be excepted can be specified in any of the following ways:

- A set of specific hostnames and/or IP addresses
- All hosts that have a specific software product in their Add / Remove Programs list
- All workstations or all servers at a specific set of sites
- All workstations or all servers in the Enterprise
- All hosts with a given Operating System / Service Pack

Part 2: Scoring Details

This section describes exactly how the score for each component is calculated.

Vulnerability (VUL)

Vulnerability detection is provided by Tenable Security Center.

Each vulnerability is initially assigned a score according to the Common Vulnerability Scoring System (CVSS) and stored in the National Vulnerability Database (NVD). Scores range from 1.0 to 10.0. Tenable, McAfee, and other vulnerability scanning products provide these scores as part of their vulnerability databases.

Scores of 1.0 – 3.0 are considered LOW, 7.0 – 10.0 are considered HIGH. The rest are considered MEDIUM. However, IA desired greater separation between HIGH and LOW vulnerabilities so that aggregating many LOWs would not overwhelm the score. The transformation created to accomplish this is:

DoS VUL Score = (CVSS Score)^{N} / 10$^{(N-1)}$ where N=3

The result of this transformation is shown in the following table. Note that using the DoS score, it takes many more LOW vulnerabilities to equal the score of a single HIGH vulnerability.

CVSS Score	DoS Score
10.0	10.00
9.0	7.29
8.0	5.12
7.0	3.43
6.0	2.16
5.0	1.25
4.0	0.64
3.0	0.27
2.0	0.08
1.0	0.01

In DoS risk scoring, “vulnerability score” always implies use of the DoS score, not the CVSS score.

Host VUL Score = SUM(VUL scores of all detected vulnerabilities)

Patch (PAT)

Patch detection is provided by SMS.

Each patch is assigned a score corresponding directly to its risk level.

Patch Risk Level	Risk Score
Critical	10.0
High	9.0
Medium	6.0
Low	3.0

A patch is scored if it is not fully installed. For example, if SMS says the patch is installed on a host but the host requires a reboot, the patch is still scored.

Host PAT Score = SUM(PAT scores of all incompletely installed patches)

Security Compliance (SCM)

Security Compliance detection is provided by Tenable Security Center.

Each security setting is compared to a template of required values. The template used to score a host is based on the operating system of the host.

Scores for security settings are based on a set of categories into which the settings are partitioned. For example, all Registry Settings have the same score. The scores for each category were determined in three steps:

1. The scores were initially determined by Diplomatic Security (DS), the organization that manages the scanning tool and performs the scans. The scores were based on general comparable risk to the CVSS vulnerability scores.
2. These scores were then transformed in the same way as the CVSS vulnerability scores.
3. When the total Security Compliance score was computed, it was found that because of the large number of settings, the score was an unacceptably high percentage of the total Enterprise score. As a result, IA scaled the category scores uniformly to bring the total Security Compliance score into an acceptable range when compared with other scoring components.

Security Setting Category	Initial CVSS-Based Score	Adjusted CVSS-Based Score	Final DoS Score
Application Log	2.0	0.862	0.0064
Event Audit	6.0	2.586	0.1729
File Security	10.0	4.310	0.8006
Group Membership	10.0	4.310	0.8006
Privilege Rights	8.0	3.448	0.4099
Registry Keys	9.0	3.879	0.5837
Registry Values	9.0	3.879	0.5837
Security Log	5.0	2.155	0.1001
Service General Setting	7.0	3.017	0.2746
System Access	10.0	4.310	0.8006
System Log	3.0	1.293	0.0216

SCM Score for a check = score of the check's Security Setting Category

Host SCM Score = SUM(SCM scores of all FAILed checks)

Note that there is no SCM score for a check that cannot be completed (NO CHECK). Only a FAIL is scored.

Anti-Virus (AVR)

Anti-Virus signature file age detection is provided by SMS.

The date on the signature file is compared to the current date. There is no score until a grace period of 6 days has elapsed. Beginning on day 7, a score of 6.0 is assigned for each day since the last update of the signature file. In particular, on day 7 the score is 42.0.

Host AVR Score = (IF Signature File Age > 6 THEN 1 ELSE 0) * 6.0 * Signature File Age

Because this component requires SMS to be functional, a host that has an SMS Reporting score has its Anti-Virus score set to 0.0.

SOE Compliance (SOE)

SOE Compliance uses SMS data to monitor the installation on workstations of a specific set of software products and versions known as the Standard Operating Environment. Each product is required and must be at the approved version. When a product upgrade is approved, there is generally some overlap where both the old and new versions are approved. After a time, only the new version is approved.

SOE Compliance scoring assigns a distinct score to each specified product that is either missing or has an unapproved version. Currently, each product has an identical score of 5.0. There are 19 products in the

SOE, so the highest score for this component is 19 * 5.0 = 95.0, a workstation with no correctly installed SOE products.

Product SOE Score = 5.0 (for each product)

Note: There are two Public Key Infrastructure products in the SOE. Since many locations do not yet have the physical card readers to utilize the software, these products have been temporarily assigned a score of 0.0. This is another example of a fairness consideration.

Host SOE Score = SUM(SOE scores for each product)

The SOE concept is a function of Operating System. If a host does not have an approved operating system, then there is no score. This gap will be corrected when Unapproved Software is added as a separate scoring component.

AD Users (ADU)

AD Users monitors the age of user account passwords (PWs). DoS policy requires all passwords be changed every 60 days. This includes service accounts.

The date the password was changed is compared to the current date. There is no score for 60 days. Beginning on day 61, a score of 1.0 is assigned for each day since the last password change. However, under any of the following conditions, the score remains 0.0:

- The user account is disabled
- The user account requires two-factor authentication for login

Account ADU Score = (IF PW Age > 60 THEN 1 ELSE 0) * 1.0 * (PW Age – 60)

If iPost cannot determine the date of the last password reset, e.g., if the user account has restrictive permissions, then a flat score of 200 is assigned.

AD Computers (ADC)

AD Computers monitors the age of computer account passwords. These passwords are not changed manually; they are normally automatically refreshed by Active Directory, although occasionally this does not occur. The password is used to establish a trust relationship between the host and AD. By means of Group Policy Objects (GPOs), workstations should refresh passwords every 7 days, while the server refresh is set to 30 days.

The date the password was changed is compared to the current date. There is no score for 30 days, regardless of host type. Beginning on day 31, a score of 1.0 is assigned for each day since the last password refresh. However, the score remains 0.0 if the computer account is disabled.

Account ADC Score = (IF PW Age > 30 THEN 1 ELSE 0) * 1.0 * (PW Age – 30)

SMS Reporting (SMS)

SMS Reporting monitors the health of the SMS client agent that is installed on every Windows host. This agent independently reports the following types of information:

- Hardware inventory data, e.g., installed memory and serial number
- Software inventory data, i.e., .EXE files
- Patch status, i.e., which patches are applicable to the host and the installation status of those patches

If the agent is not reporting all of the expected data, the SMS database is populated with one or more 3-digit “error codes” that describe the action necessary to remediate the problem on the host. **Error codes are not part of SMS.** They were added by the DoS SMS team to assist in identifying the specific reason that a client was not reporting.

For each host, its error codes, if any, are examined to determine the score. If an error code has the form 1xx or 2xx, a score is assigned to the host. Other error codes do not result in a score, as they indicate an anomaly situation that can only be repaired by the SMS team.

In addition, the last time that the agent was correctly reporting is tracked and used as the base date to age the score. If the agent has never reported correctly, e.g., the host is new on the network, the current date is used as the base date. The base date is compared with the current date to determine the SMS Reporting age. There is actually a grace period built into the population of the error codes. Therefore, the error codes themselves completely determine the score.

Host SMS Score =
(IF Error Code = 1xx/2xx THEN 1 ELSE 0) * (100.0 + 10.0 * (SMS Reporting Age))

SMS is necessary to score each of the following other scoring components:

- Patch
- Anti-Virus
- SOE Compliance

If there is a score for SMS Reporting, the scores for the dependent scoring components are set to 0.0, since any residual SMS data is not reliable.

Vulnerability Reporting (VUR)

Vulnerability Reporting measures the age of the most recent vulnerability scan of a host. This scan is conducted from outside the host rather than by an agent. It is therefore possible that a host may not have recent scan information for one of the following reasons:

- The host was powered off when the scan was attempted.

- The host's IP address was not included in the range of the scan.
- The scanner does not have sufficient permissions to conduct the scan on that host.

The date of the most recent scan is used as a base date and compared to the current date. There is a conceptual grace period of 2 consecutive scans. Operationally, each host is scanned for vulnerabilities every 7 days. Therefore, a grace period of 15 days is allowed for VUR. After this period, a score of 5.0 is assigned for each subsequent missed scan.

If a host has never been scanned, e.g., the host is new on the network, the current date is used as the base date.

Host VUR Score =
(IF VUR Age > 15 THEN 1 ELSE 0) * 5.0 * FLOOR((VUR Age – 15) / 7)

Security Compliance Reporting (SCR)

Security Compliance Reporting measures the age of the most recent security compliance scan of a host. This scan is conducted from outside the host rather than by an agent. It is therefore possible a host may not have recent scan information for one of the following reasons:

- The host was powered off when the scan was attempted.
- The host's IP address was not included in the range of the scan.
- The scanner does not have sufficient permissions to conduct the scan on that host.

The date of the most recent scan is used as a base date and compared to the current date. There is a conceptual grace period of 2 consecutive scans. Operationally, each host is scanned for security compliance every 15 days. Therefore, a grace period of 30 days is allowed for SCR. After this period, a score of 5.0 is assigned for each subsequent missed scan.

If a host has never been scanned, e.g., the host is new on the network, the current date is used as the base date.

Host SCR Score =
(IF SCR Age > 30 THEN 1 ELSE 0) * 5.0 * FLOOR((SCR Age – 30) / 15)

Part 3: Underlying Data and Tools

Continuous monitoring requires data, a set of Enterprise-level tools to deliver it, and a team of people to maintain those tools. However, DoS did not create iPost by setting out to find tools to deliver a desired set of data. There were already multiple tools that had been selected over time by multiple organizations. In each case, a specific organizational mission drove the implementation of a specific tool.

The data described in this section is selected to manage the DoS infrastructure, irrespective of the creation of risk scoring and iPost.

Microsoft Active Directory (AD)

AD was deployed to authenticate and authorize network services. As a source of data, it contains every network user account, the account password age, and various other security-related attributes. It also contains purely descriptive data about user accounts, e.g., name and organization.

AD provides an example of a tool fulfilling its mission yet not, at least yet, being the expected authoritative source for some of its data. The name of a user in AD is used to populate the Global Address List (GAL) used in email. A user's name in the GAL is inherently correct because the user wants it to be. However, the organization to which the user belongs, which also populates the GAL, can be blank, out of date, or inconsistently entered, and is therefore unreliable for operations like counting, sorting, and filtering.

AD also contains computer objects, but it does not contain every hostname. It is possible, although unusual, for a server that does not use AD authentication to be on the network yet not be in AD.

Microsoft System Management Server (SMS)

SMS was deployed to distribute patches. In addition to a regional hierarchy of servers, SMS relies on an agent installed on each Windows host to send information up the hierarchy to the core server in Washington. As a source of data, SMS contains a wealth of information about each host: serial number, RAM, BIOS date, needed patches, Add/Remove Programs entries, etc.

Unfortunately, not all hosts report properly to SMS. There are a variety of reasons for this, but over the years, the SMS infrastructure at DoS has made steady progress on reducing and even eliminating some of these "not reporting" issues. Still, problems remain and so while SMS is authoritative for the hosts reporting to it, it is not suitable by itself for a complete host inventory, even for Windows machines.

Tenable Security Center (Tenable)

Tenable was deployed to scan every device on the network for vulnerabilities, and every Windows workstation and server for compliance against a checklist of security settings. Tenable performs agentless network scans of specific IP ranges, so hosts that are powered off during the scan are missed. Although Tenable scans include non-Windows machines, printers, and even hosts not in AD, it is authoritative only for vulnerability and compliance data for hosts that are scanned, not for a complete host inventory.

The data listed above is integrated into iPost and used in the risk scoring program, but there are additional tools used to manage the infrastructure that have also been integrated into iPost

NetIQ AppManager

AppManager was deployed to manage and monitor servers. The AppManager agent on a server monitors resource utilization and alert conditions. AppManager is used primary for domain controllers, SMS servers, and Exchange servers. Its data is authoritative for those servers on which it is deployed, but the data is not available for all servers Enterprise-wide.

NetIQ Security Manager

Security Manager was deployed to provide Information System Security Officers a convenient way to monitor security alerts on domain controllers. These alerts occur when a user attempts to login outside of approved hours, when a user account is enabled, etc.

HP OpenView

OpenView was deployed to actively manage the centrally managed network devices and to monitor network traffic volume and faults. It sees all wide area network traffic and is the only tool that does so.

CiscoWorks Campus Manager

Campus Manager was deployed to centrally manage switches. It rediscovers all switches on the network several times per day. Its database includes the various hardware components inside a switch and the hosts connected to each interface. Campus Manager is authoritative for this data.

Tavve PReView

Tavve PReView was deployed to monitor network latency. Its server continuously pings all DoS routers and provides historical latency (round trip time) for analysis and troubleshooting. It is authoritative for this data.

Niksun NetOmni

NetOmni was deployed to facilitate reporting of detailed network traffic statistics. It manages a database that is fed by NetVCR sniffer appliances. It captures most, but not all, wide area traffic and maintains data at the source-destination-port level of detail. For example, the data can show which local IP addresses initiate the most web traffic at a given DoS site.

BMC Remedy Service Desk

Remedy was deployed to manage trouble tickets and service requests. iPost integrates this data to provide users with the convenience of a single interface.

Utopia: One Framework Tool

Periodically, DoS reviews industry framework tools and tool suites. These are tools providing the combined functionality of several of the individual tools already in use. While there are some perfectly good, viable candidates, none has so far overcome a cost/benefits analysis.

The infrastructure data underlying a risk scoring program can be provided by tools other than the ones used in iPost. But even if a completely integrated tool suite is used, there is additional data necessary that is specific to risk scoring. This additional data, described in more detail in Part 4, must be kept somewhere, and that means integration work. When evaluating tools, their ability to allow such integration may be just as important as other features. A tool suite can eliminate some integration effort by providing all the infrastructure data in one place, but unless the tool already has a suitable scoring mechanism, one will have to be bolted on somehow. There was significant effort required to add just risk scoring data to iPost, even though all the necessary infrastructure data had already been integrated.

Note that the set of multi-vendor best-of-breed tools used at DoS cannot talk to each other for management purposes. However, integrating their data for reporting purposes is significantly easier, as is proven by the success of iPost.

The Need for iPost

The need for an integrated application like iPost was driven by some very specific requirements that pre-dated the desire to establish a risk scoring program:

- Each area of infrastructure was managed by a specialized central team that selected, engineered, and maintained the best-of-breed tool(s) within its scope of responsibility.
- Hundreds of local administrators were responsible for managing the infrastructure at hundreds of local sites, yet they did not have visibility into the data provided by the Enterprise tool suite or individual components of that suite.
- The best-of-breed tools each had their own authentication and authorization mechanisms, so it was not practical for the tool owners to manage the accounts for the various local administrators, and those local administrators would have to login separately to each tool.
- The best-of-breed tools were designed for Enterprise use and so did not have the ability to prevent one user from inadvertently affecting another administrator’s site, or for that matter, the entire Enterprise.
- Individual administrators, let alone their management, could not be expected to learn to effectively use so many different tools, each with its own quirks.

iPost Data and Features

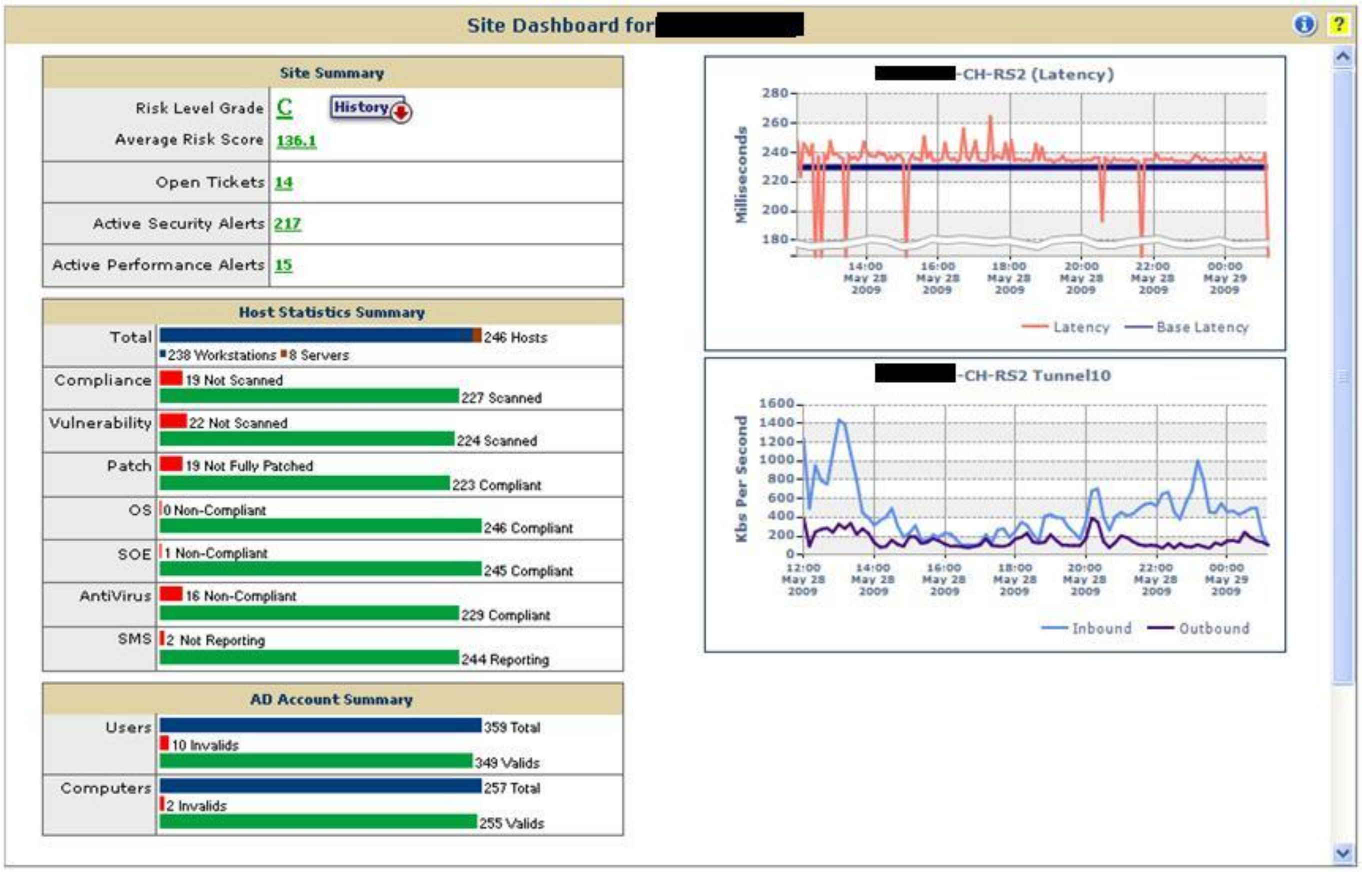
The iPost concept is to select some of the data available from the Enterprise tools and integrate it into a single user interface. In most cases, data is obtained by scheduled tasks and integrated directly into the iPost database. In some cases, data is accessed on demand directly from a source system.

iPost is built on a platform of ASP .NET (C#), Microsoft SQL Server 2005 (including Reporting Services), and Dundas Chart. The users of iPost are 2500 local and Enterprise IT administrators and their management.

The iPost database has the following properties:

- All data is obtained from servers in Washington. It is up to the underlying tools to scan individual systems and report results to their own core servers in Washington.
- All data is read-only. Essentially, iPost is a reporting tool – actual system or network management is still done using the underlying tools.
- All data is associated with individual sites. The structure in AD is designed specifically to create administrative boundaries, so the definition of an iPost site is taken primarily from existing AD structure. In some instances there was a need to augment this with IP assignments.
- Users are authorized to see data only for specific sites. This authorization is implemented by AD groups.
- Users can be authorized to see only certain types of data. This authorization is implemented by AD groups.
- When new data is integrated into iPost, it automatically inherits the site security model.
- Navigation is both vertical (drill-down) and horizontal. For example, when looking at a specific patch, a user can drill down to see which hosts need that patch. Once a specific host is selected, the user can see not only all of the patches needed by that host, but all of the data iPost has for that host, regardless of what type of data it is.

Figure 3 below shows the iPost Dashboard, the primary screen for a specific site. Figure 4 shows the detail available at the host level.



Site Dashboard for [redacted]

Site Summary

Risk Level Grade	C History
Average Risk Score	136.1
Open Tickets	14
Active Security Alerts	217
Active Performance Alerts	15

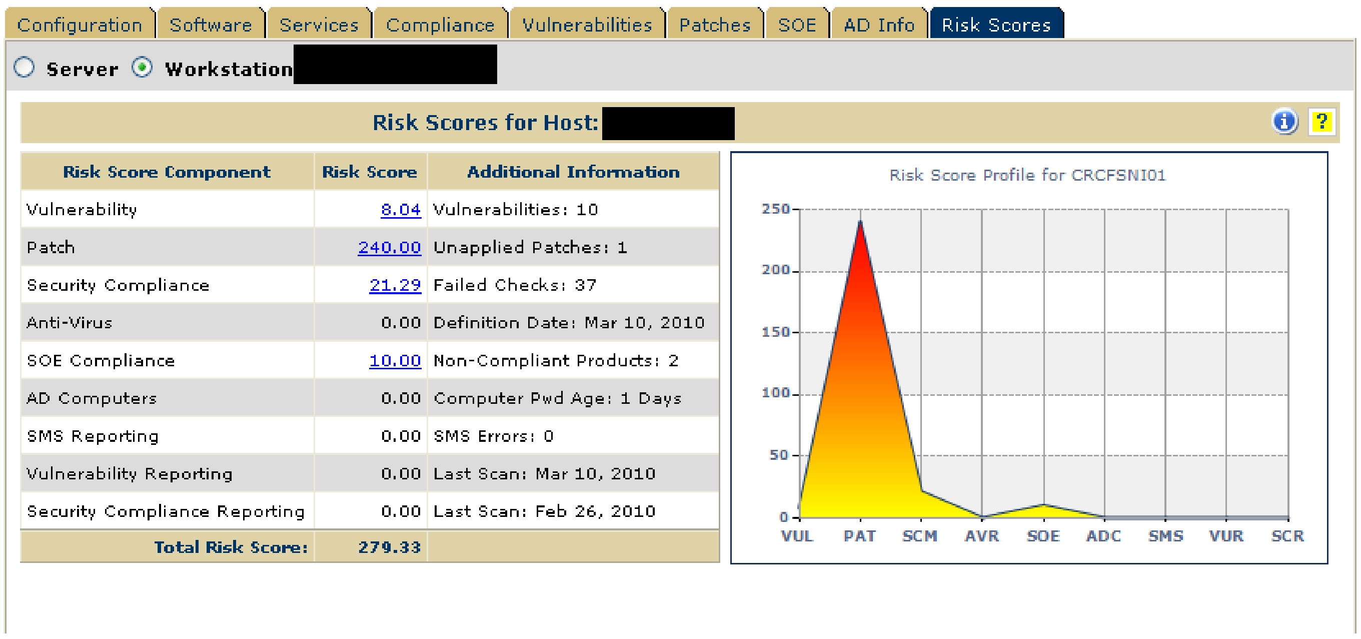
Host Statistics Summary

Total	246 Hosts; 238 Workstations, 8 Servers
Compliance	19 Not Scanned; 227 Scanned
Vulnerability	22 Not Scanned; 224 Scanned
Patch	19 Not Fully Patched; 223 Compliant
OS	0 Non-Compliant; 246 Compliant
SOE	1 Non-Compliant; 245 Compliant
AntiVirus	16 Non-Compliant; 229 Compliant
SMS	2 Not Reporting; 244 Reporting

AD Account Summary

Users	359 Total; 10 Invalids; 349 Valids
Computers	257 Total; 2 Invalids; 255 Valids

Figure 3 - Dashboard (iPost)



Server [✓ Workstation] [redacted]

Risk Scores for Host: [redacted]

Risk Score Component	Risk Score	Additional Information
Vulnerability	8.04	Vulnerabilities: 10
Patch	240.00	Unapplied Patches: 1
Security Compliance	21.29	Failed Checks: 37
Anti-Virus	0.00	Definition Date: Mar 10, 2010
SOE Compliance	10.00	Non-Compliant Products: 2
AD Computers	0.00	Computer Pwd Age: 1 Days
SMS Reporting	0.00	SMS Errors: 0
Vulnerability Reporting	0.00	Last Scan: Mar 10, 2010
Security Compliance Reporting	0.00	Last Scan: Feb 26, 2010
Total Risk Score:	279.33	

Figure 4 - Host Level Details (iPost)

General Integration Issues

Each time a new source of data is integrated into iPost there are the following minimum considerations:

- The initial pulling of raw data to gain familiarity and resolve apparent anomalies. This item alone invariably uncovers misunderstandings about the underlying data and long-ignored or unknown quirks.
- The quantity of data and how much should be periodically integrated directly into the iPost database.
- The frequency with which the source system's data is updated.
- The frequency with which a scheduled task should pull data.
- The mapping of the new data to existing iPost objects.
- The relationship of the new data to other data already in iPost.
- The need for rolling up and/or partitioning the data for better performance.
- The design of front-end display screens.
- The site and Enterprise-level reporting to make the data useful.

Data Source Change Management

It is essential to maintain close and frequent coordination between iPost and each of its data sources. Each source has its own schedule of upgrades, and it is possible that an upgrade will impact iPost's interface with that source. In addition, purely operational changes in a product source system may impact the iPost back-end and/or front-end. So far, iPost has been able to keep up with source system changes, although each additional source system that is integrated increases the effort of doing so.

Data Source Operational Status

Since iPost is fed by other systems, it is dependent on the operational status of those systems. For example, the source of iPost security compliance data is Tenable. The iPost database processes Tenable data daily but pulls over only the status (PASS/FAIL/NO CHECK) of each security setting on each host. When an iPost user wants to see exactly why a particular host failed a particular check, iPost accesses the detailed explanation directly from the Tenable data source. If Tenable is experiencing an outage at that moment, iPost cannot fulfill the request. iPost has to test for this type of condition for all its data sources, and whenever possible, alert users beforehand when a data source will be unavailable to some degree.

Data Latency

When users make local changes, they want to see the results relatively quickly. The question most frequently asked by iPost users is: "How long will it take for my change to show up in iPost?" In general, the answer is:

UserPerceivedLatency = SourceSystemLatency + iPostLatency

Over time, the frequency with which iPost pulls data from various sources has been increased. At the same time, source system owners have responded by increasing the frequency with which their core servers are updated by network and system scans. While this continues to improve and is by now generally acceptable, latency is inherent in the iPost concept.

Data Surprises

As various data sources were integrated, it was expected that there would be some inconsistencies among the various tools. What was not expected was that the very process of testing the integration of new data would uncover items like the following, making reconciliation and reporting a challenge:

- Two different AD computer objects (in different domains) with the same hostname
- A hostname associated with multiple IP addresses
- A hostname associated with multiple MAC addresses
- A hostname with non-ASCII characters
- Two distinct computers with the same SMS GUID (supposedly a unique key)
- A network switch with two serial numbers
- An AD computer object with no Operating System
- Two hosts with the same software product installed but whose Add / Remove Programs entries had the product names spelled differently
- A non-Windows storage appliance interpreted by AD as having an Operating System of Windows NT

Reporting Lesson – Unassigned Objects

The original mission of iPost focused on bringing data directly to local administrators, hence the organization of all data into "sites." As iPost grew in significance and usefulness, the reporting requirements at the Enterprise level expanded. In particular, there was a need for Enterprise totals of various things, with the totals being broken out “by site.” However, there were gaps due to hosts being created in areas of AD where there was no defined site. This resulted in undercounted Enterprise totals.

It became necessary to create a virtual “unassigned” site for every domain in AD. As iPost processes AD objects, they are assigned to their designated site, and if no such site exists, they are put in Unassigned, which is then treated as another line item in the list of sites.

This facilitated and motivated a parallel task: review the hosts in Unassigned and either get them moved into more appropriate locations in AD or delete them. This resulted in the removal of an enormous amount of deadwood from AD.

Basic Principles

It bears repeating that there are two principles that iPost does not violate.

- iPost does not look directly at an individual host and does not have the permissions required to do so. Users must wait for the changes they make to percolate into the core server of the appropriate Enterprise tool, and from there to iPost.
- iPost does not allow users to change the underlying data. For example, if a user sees in iPost that a host needs a patch, the user cannot click on something in iPost to patch the host.

These principles are periodically reviewed. While they would increase the usefulness of iPost and are technically achievable, they are currently outside iPost’s mission.

Part 4: Beginning a Risk Scoring Program

The purpose of this section is to describe some of the requirements, issues, and actions for establishing a basic continuous risk monitoring program using scores and grades.

When the DoS risk scoring program was beginning, there was a 12-month pilot during which the only mechanism for viewing risk scores was a small set of reports:

- The Risk Score Advisor report (Figures 1, 2) provided site administrators and management with all the details of the score and how it was derived.
- The Risk Score Monitor Report provided Enterprise management with a table of the scores and grade for each site.

In addition, there were two reports, one for site administrators and one for Enterprise management, showing details of scoring exceptions.

Thus there is no need for an interactive application like iPost to begin a viable risk scoring program, although it has been extremely helpful to those who must remediate issues to have interactive drill-down and navigation features. All that is truly required to begin is a database and a small set of reports. This section will therefore focus on the data and the process. The reports can be provided by any robust reporting package (DoS uses SQL 2005 Reporting Services).

Assumptions

- The host operating systems are versions of Microsoft Windows
- Active Directory is implemented
- All data collected by monitoring tools is available in databases
- Available data includes
    - Vulnerability data, including CVSS scores and scan results for each host (at DoS, this is provided by Tenable)
    - Security Compliance data, including FDCC checks and scan results for each host (at DoS, this is provided by Tenable)
    - Anti-virus signature file dates for each host (at DoS, this is provided by SMS)

With this basic data, the following risk scoring components can be realized:

- Vulnerability
- Security Compliance
- Anti-Virus
- AD Computers
- AD Users
- Vulnerability Reporting

- Anti-Virus Reporting (at DoS this is called SMS Reporting)

Of course, if additional data is available there can be additional scoring components.

Process Considerations

Begin with a small pilot.

Determine the algorithm to be used for each scoring component. Begin with the raw CVSS scores for vulnerability and determine how they should be transformed to be meaningfully added. Determine the parameters for other scoring components by contrasting them with vulnerability scores.

Establish a formal process for requesting, reviewing, and approving/rejecting scoring exceptions.

Engage the owners of the underlying data so that the potential impact of tool upgrades on the scoring program can be analyzed. Having scores suddenly worsen across the board will generate trouble tickets, ill will, and a feeling that the scoring program is unstable.

Establish a team to which scoring questions can be directed. Initially, there will be many questions due to misunderstandings, issues with the underlying data, and issues with the scoring program implementation.

Additional Lessons Learned

There must be continued and close coordination among all of the participants in implementing the Risk Scoring program. This includes the tool owners, who may only be indirect participants. The actual end users see a single program, and at DoS this program would not have succeeded with this cooperation

As much as possible, the design should accommodate the addition of new scoring components and changes in the calculations for components. Establishing this type of flexibility was one of the most difficult challenges at DoS.

There must be qualified database expertise available; otherwise this effort will surely fail. As the program progressed, reporting needs increased and several times there had to be significant changes in the database to maintain acceptable performance.

Reporting history is important, even at the site-component level. For example, this allows tracking changes in a site’s Anti-Virus scores and may be helpful in uncovering system issues in local processes.

The ability to handle scoring exceptions was absolutely essential to success. This was a double win: it not only proved to administrators that their scores were completely under their control, but also exposed serious problems that could only be solved by the central bureaucracy.

The structure of “Organizational Units” (OUs) in AD is the logical basis for defining a “site” as an area of administrative responsibility. Once the scoring program became established, there was significant creation of new OUs and movement of objects from one OU to another, as administrators attempted to ensure that only objects within their scope were counting against their scores.

Scoring of a site is accomplished by summing the scores of all the objects at the site. This created the unexpected desire to be able to combine any desired set of objects into a kind of virtual site so that it could be viewed, reported, and scored just like a normal site. For example, risk scores can be used to assign a score to an application or to an arbitrary organization (not a defined site) within the Enterprise.

Another unexpected need was to be able to report scores for any sub-collection of sites. For example, DoS has a partially implemented program of Centralized Patch Management (CPM), where certain sites have turned over their patching responsibility to the CPM team. It was useful to be able to report scores for the CPM sites and contrast the overall score for CPM sites with the overall score for the Enterprise.

Almost all reporting needed to be broken out by servers vs. workstations. This is because the issues are generally quite different, and it more clearly exposes systemic root causes of risk. This facility should be designed into the database from the beginning.

Monitoring daily score changes proved to be an excellent way to expose systemic issues in local and/or global processes.

Re-imaged hosts may cause issues. At DoS it was found that, depending on how this was done (e.g., was the object first removed from AD or not), it could create confusion in SMS for days between the data for the old host and the data for the new host.

It took approximately 18 months after all the data was integrated into iPost to develop, pilot, and firmly establish the scoring program at DoS. Your mileage may vary significantly..

Contact Information

Dr. Ronald Rudman
TASC, Inc.
Department of State
7374 Boston Blvd
Springfield, VA 22153
(703) 912-8010