

Effective Threat Management Using Vulnerability Correlation

By Joe Minieri

The Challenges of Making IDS Effective

According to a 2003 FBI/CSI survey, 73% of respondents have deployed and used some type of intrusion detection system (IDS). Yet, the same survey reports that only 29% of those surveyed claimed to have had no unauthorized use of computer systems within the last 12 months. This leaves an alarming 71% of organizations represented in this survey admitting to some kind of security breaches. While some of these may be undetectable to intrusion detection systems (e.g. laptop theft), the vast majority includes denial of service, worms, and unauthorized access. Why is it that IDS does not appear to be effective at detecting these intrusions soon enough for them to be contained?!

Tuning IDS is the best mechanism to make it more effective. Limiting the active signatures is one way to increase the relevancy of alerts generated by a NIDS. In an ideal world, the NIDS would filter out attacks that will not succeed. However, tuning an NIDS on a host by host basis is impractical for anything larger than a trivial network. Worse yet, constant tuning would be necessary as new patches are applied to hosts and new vulnerabilities are uncovered. In 2003, CERT reported 3,784 new vulnerabilities². That is over 10 per day, every day, on average.

A Dose of Protection with Vulnerability Scanners

Primarily as a tool to target patch management activities, vulnerability assessment (VA) scanners are being increasingly deployed to address the issue of the vast number of vulnerabilities being discovered constantly. Yet even with automated vulnerability and patch management systems, it is not always possible—or permissible—to patch a system immediately.

Moreover, a recent Forrester study shows that it is common for organizations to use multiple vendors for the same point solution—sometimes as a result of policy decisions (i.e. redundancy by design), or the lack of centralized standards. 27% of respondents have two or more types of NIDS, 44% have two or more types of host based IDS, and 57% have two or more types of Anti-Virus protection. Unfortunately, the lack of interoperability and integration between these multi-vendor solutions often makes security less effective overall³.

Consider an analyst using two vendor's NIDS and two vendor's vulnerability scanners. The morning's VA report will have to be scrutinized comparing the results from Vendor A's and Vendor B's vulnerability assessment. Did they both report the same thing? Maybe. Do they both report them in the same way so this is easily deducible? Unlikely.

To see if a system was compromised, the analyst needs to check the NIDS logs. After wading through several thousand lines and 'grepping' for appropriate signatures, the analyst does the same for the other NIDS device—and then has to be able to map the IDS signature to the VA scans.

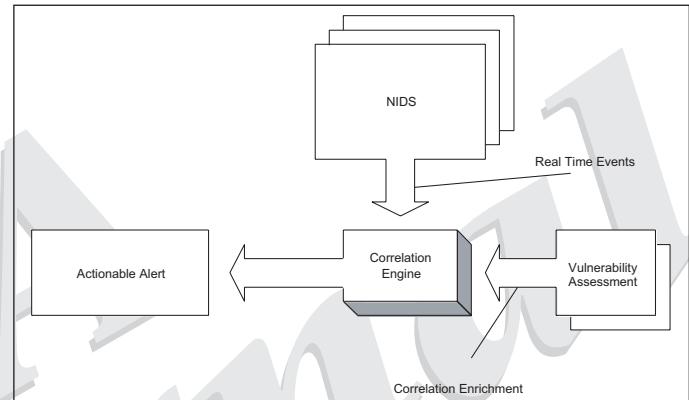


Figure 1: Vulnerability Assessment Correlation

This is an impractical approach which does not yield effective or timely results—and if something is found, it's too late.

Mixing and Matching IDS and VA with Real-Time Correlation

The solution to this problem is a real time correlation product which maps the vulnerability assessment information to NIDS events in real-time, with the goal that events are only escalated when they use an exploit the target is vulnerable to.

The correlation engine filters IDS alerts through the VA information, thus validating an alert's accuracy. As a result the analyst can receive fewer but more accurate alarms.

Correlation Components—IDS Normalization

The goal is to provide context to the alert received by the NIDS, and to determine its relevance and importance. In order to do this, components of this event must be extracted and examined using a process called normalization, since no two product vendors (and often, products by the same vendor) report events the same way. It is this normalized form that enables vulnerability correlation. A translation layer is therefore implemented to convert events from their native format to a normalized format. For vulnerability correlation specifically, the following fields are key:

- ▲ **Target IP.** The target IP is necessary since it will be used to index other databases of information that might be used in correlation and weighting.
- ▲ **Target Port.** In some cases, the IDS attack signature will not match any known vulnerabilities, but vulnerability assessment scanners will typically report open ports. This information can be used if the target port matches a known open port.

▲ **Event signature.** Each IDS vendor will report its method of signature identification using some unique identification scheme.

The correlation engine for use in other correlation rules may normalize other fields. These other approaches aren't discussed in this article.

Note that the frequently used terms "source" and "destination" are not used; instead, we use "attacker" and "target." In some cases, an IDS will match on the response traffic from the target, rather than on the initial attack itself. The IDS typically reports the 'target' of the attack as the 'source' of the event (since it is the 'target' that is responding to the attack). For vulnerability correlation, the target of the attack needs to be determined, regardless of whether it is the source or the destination of the detected event. In order to enable timely responses to potentially successful intrusions, it is vital that all of this event data should be collected, normalized, and correlated in real time.

Correlation Components—Vulnerability Scan Normalization

Consider now the vulnerability information that should be collected from the VA scanner. Unlike NIDS events, VA result data is typically *not* real time. Generally, scans/reports are run at scheduled intervals or repeatedly. As with the NIDS event normalization, a translation layer must be implemented to convert vendor-specific VA reports from its native format into a standard form. For VA correlation, we are interested in collecting the following components:

- ▲ Scanned IP. The scanned node is used to correlate with the NIDS event target.
- ▲ Vulnerability ID. The vendor-specific reference ID to the vulnerability.
- ▲ Open Ports. Useful for correlating without an exact signature match.
- ▲ ID References. References to standard vulnerability sources (such as CVE).

Vulnerability scan data must be regularly updated in the correlation system to be accurate. In cases where multiple scanners are being utilized, the timestamp of the scan could be used to choose one scanner's vulnerability over another. Alternatively, a scanner "accuracy" rating may be applied, causing the more accurate scanner to "win" in cases of a tie.

Implementing a Unified Threat Identification System

The most important component for vulnerability correlation is the mapping of vulnerability and attack identifiers. As discussed, IDS event signatures and vulnerability identifiers have vendor-specific reporting identifiers. The key to mapping these together is the included references with each event.

For example, consider just some examples of how various popular NIDS report the SASSER Worm and related exploits, as shown in Figure 2.

Attacks exploiting this single vulnerability trigger many different signatures. Each NIDS reports a specific ID for each specific signature. In this case, SNORT detects LSASS exploits no less than 11 different ways, but ISS has a single LSASS overflows/activity alert.

Figure 3 shows the vulnerability SASSER exploits as reported by different vulnerability systems and third party dictionaries.

Many of the scanners reflect the existence of this vulnerability as a single identifier, rather than multiples based on how it might be exploited. Nessus, however, provides two IDs: one for the existence of

Vendor	Identifier	Description
Cisco Systems	3030/0	IDS Signature TCP SYN Host Sweep
Cisco Systems	3338/0	IDS Signature Windows LSASS RPC Overflow
Cisco Systems	3142/0	IDS Signature Sasser Worm Activity
SNORT	2507	NETBIOS DCERPC LSASS bind attempt
SNORT	2508	NETBIOS DCERPC LSASS DsRolerUpgradeDownlevelServer Exploit attempt
SNORT	2509	NETBIOS SMB DCERPC LSASS unicode bind attempt
SNORT	2510	NETBIOS SMB DCERPC LSASS bind attempt
SNORT	2511	NETBIOS SMB DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt
SNORT	2512	NETBIOS SMB-DS DCERPC LSASS bind attempt
SNORT	2513	NETBIOS SMB-DS DCERPC LSASS unicode bind attempt
SNORT	2514	NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt
SNORT	2524	NETBIOS DCERPC LSASS direct bind attempt
SNORT	2525	NETBIOS SMB DCERPC LSASS direct bind attempt
SNORT	2526	NETBIOS SMB-DS DCERPC LSASS direct bind attempt
ISS Real Secure	15699	Microsoft Windows LSASS buffer overflow

Figure 2: Some Intrusion Detection System Identifiers for Sasser Worm.

Vendor	Identifier	Description
Mitre (CVE)	CAN-2003-0533	Stack-based buffer overflow in certain Active Directory service functions in LSASRV.DLL
SecurityFocus	BID-10108	Microsoft Windows LSASS Buffer Overrun Vulnerability
nCircle IP360	3643	Worm: Sasser
Nessus	12219	Sasser Worm Infecting
Nessus	12220	Sasser Worm Infection
ISS/XForce	15818	Microsoft Windows MS04-011 patch is not installed

Figure 3: Vulnerability Identifiers for LSASS Vulnerabilities.

the vulnerability and one to alert that the system appears to be already compromised. Alternatively, the ISS vulnerability ID refers instead to a patch not being installed. Note too that ISS reports this ID for fourteen different vulnerabilities existing on the scanned system, all addressed by the same patch.

This example illustrates the challenge of multi-vendor vulnerability mapping. It does not mean to imply that one method of reporting and/or identifying is better or worse than another. Rather, the point here is to demonstrate that individual vendors have different methods for detecting and reporting the same events, exploits, and vulnerabilities. In order for the vulnerability correlation system to effectively correlate between these systems, a single scheme must be developed to map an event from *any* IDS source to a vulnerability reported by *any* supported vulnerability scanner.

Fortunately, many vendors provide references to CVE, CAN, and/or BugTraq IDs, which makes building a cross reference table easier, although some vendors still use other ID schemes for reference. For example, many SNORT IDs still map only to the ArachNIDS database, while many ArachNIDS IDs can be mapped to CVE/CAN IDs. Likewise, ISS tools report the ISS XForce IDs, which, in general, can also be mapped to CVE or BugTraq IDs. Figure 4 shows a partial relationship mapping for a single vulnerability-exploit (in this case, the "trin00 DDOS" mapping) using four different systems. Vulnerability correlation can rely on CVE/CAN IDs where they exist, since they are used by the preponderance of vendors as references. However, the list of CVE/CAN IDs is short in comparison to the number of known vulnerabilities⁴. In Open's vulnerability correlation implementation we fall back to other ID sources when CVE IDs are not available, including BugTraq and XForce⁵. Clearly, ensuring the completeness of the IDS to vulnerability signature map is vital to the accuracy of the correlation process.

The signature maps need regular updates as new vulnerabilities are added, new exploits detected, and new vendor products supported. The quality and completeness of this mapping relies on the accuracy and completeness of the individual vendor mappings. Not all VA vendors provide complete mappings.

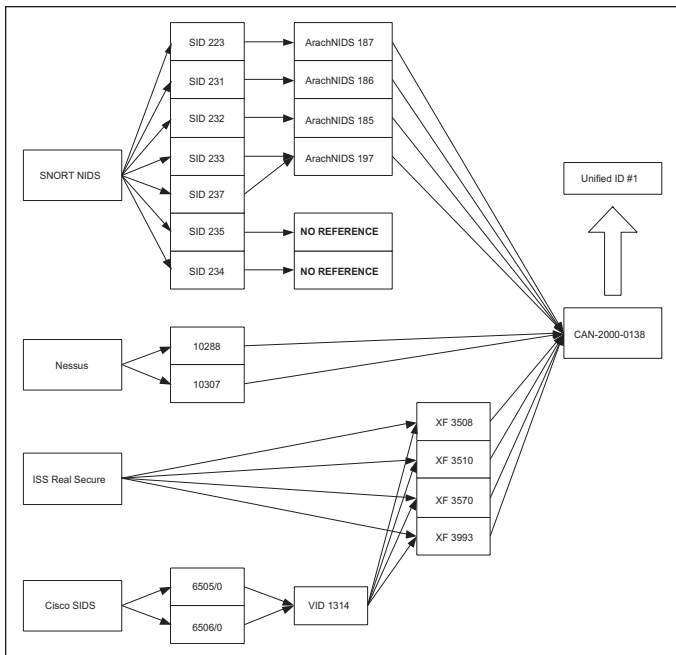


Figure 4: Vulnerability/Exploit Relationship Mapping

More importantly, not all detected exploits correspond to code-based vulnerabilities. Consider Cisco's Secure IDS alert ID 3171/0. The alert is generated when detecting an ftp login for a privileged user (root or administrator)⁶. This alert warns about a potentially misconfigured FTP server, not software vulnerability. As a result there may not be a corresponding vulnerability ID or signature to map this event to. Whether and how to correlate this event may be a function of other correlation rules, and serves as a reminder that correctly tuning your IDS to your security policies is not a task that can be eliminated simply by deploying a vulnerability correlation solution.

As a result of mapping the IDS and VA systems together, an implementation of this relationship map involves collapsing down the multiple layers and mapping the native ID directly to the unified ID (the correlation engine's "internal" ID). So, our example in figure 4 will yield a lookup table similar to Figure 5.

Making it Happen—Vulnerability Correlation

With IDS events normalized, vulnerabilities normalized and mapping tables built, consider some useful correlation methods using this enriched vulnerability data. The simplest and most useful correlation is a direct association of the event data to the vulnerability map. The received IDS event has been normalized. From this, the lookup table is used to get the unified ID. The correlation engine also takes the event's target IP and retrieves the list of known vulnerabilities (again using the unified ID) for this target. Is the received event's unified ID a member of that list? If so, it appears that the target system is indeed vulnerable to the attack and is in the process of being compromised. Clearly, this is important information and should be immediately sent to the security analyst for investigation.

No Mapping, No Problem?

The lack of obvious and immediate association does not mean that the attack can be ignored. It may be that the scan data is old, that the scanner was not configured to look for the vulnerability being targeted by the exploit, that the scanner itself has not been updated recently, or the implemented IDS and VA scanner have a few mappings in our vulnerability data-

Vendor	Vendor ID	Unified ID
CVE	CAN-2000-0138	1
SNORT	SID 223	1
SNORT	SID 231	1
SNORT	SID 232	1
SNORT	SID 233	1
SNORT	SID 237	1
NESSUS	10288	1
NESSUS	10307	1
ISS RS	XF3508	1
ISS RS	XF3510	1
ISS RS	XF3570	1
ISS RS	XF3993	1
Cisco SIDS	6505/0	1
Cisco SIDS	6506/0	1

Figure 5: Unified ID Mapping for Exploit/Vulnerability

base. The event describes the destination port the exploit was attempting to connect to. The list of open ports can be retrieved from the scan of the target IP, and matched to see whether the attack targets an open port on the victim. If so, this information should be reported to the security analyst—but care is needed to avoid introducing false positives.

Vulnerability Correlation In Practice

To show a real-world solution in practice, we set up a controlled environment to examine the effects of a worm on a vulnerable system, and used Security Threat Manager from OpenService⁷ as the vulnerability correlation solution.

Using Nessus, both systems were scanned, with 192.168.0.2 being reported as vulnerable to an MS-SQL overflow that is exploited by a number of worms⁸. Additionally, Nessus reported that the other system, 192.168.0.1, is *not* vulnerable to this exploit. The completed scan data from Nessus was fed into the correlation engine (not shown in the diagram) for enrichment.

A worm that uses this exploit was released into the network. The Snort IDS reports the following attack signatures, which are fed into the correlation engine:

```
Jun 16 10:02:53 10.1.2.20 snort: [1:2003:2] MS-SQL Worm
propagation attempt [Classification: Misc Attack]
[Priority: 2]: {UDP} 172.16.0.1:1183 -> 192.168.0.1:1434
Jun 16 10:02:53 10.1.2.20 snort: [1:2003:2] MS-SQL Worm
propagation attempt [Classification: Misc Attack]
[Priority: 2]: {UDP} 172.16.0.1:1184 -> 192.168.0.2:1434
```

At this point, these two events have generated a single actionable alert on our event viewer.

The correlation engine produces its highest level of alert, showing that an exploit has been detected against a host that is vulnerable—in this case, 192.168.0.2. Note that although the IDS reported the same attack against 192.168.0.1, no alert appears because this host is NOT vulnerable to this exploit.

The security analyst may drill down into that high level alert and see the individual event(s) that caused it to be generated.

Seeing that there is a compromise under way, the security analyst should immediately perform remediation on this host. Assume for this

example that doesn't happen, and our compromised host now attempts to infect other systems. As it does so, the Snort IDS reports the following events:


```
Jun 16 10:04:55 10.1.2.20 snort: [1:2003:2] MS-SQL Worm
propagation attempt [Classification: Misc Attack]
[Priority: 2]: {UDP} 192.168.0.2:1345 -> 192.168.0.10:1434
Jun 16 10:04:55 10.1.2.20 snort: [1:2003:2] MS-SQL Worm
propagation attempt [Classification: Misc Attack]
[Priority: 2]: {UDP} 192.168.0.2:1346 -> 192.168.0.11:1434
Jun 16 10:04:55 10.1.2.20 snort: [1:2003:2] MS-SQL Worm
propagation attempt [Classification: Misc Attack]
[Priority: 2]: {UDP} 192.168.0.2:1347 -> 192.168.0.12:1434
Jun 16 10:04:55 10.1.2.20 snort: [1:2003:2] MS-SQL Worm
propagation attempt [Classification: Misc Attack]
[Priority: 2]: {UDP} 192.168.0.2:1348 -> 192.168.0.13:1434
```

Since the target host is now the source of an infection, a new alert has been generated because the correlation engine has related this new activity to the previous events. The security analyst is now alerted to the system compromise that has taken place.

Note too how quickly this attack unfolded. The first event arrived at 10:02:53, infecting the vulnerable system. Less than two minutes later the compromised system starts attempting to spread the infection. Any correlation or log analysis system with any latency would clearly have been ineffective at flagging the attack in time to prevent the compromise, allowing this compromised system to infect others undetected.

Correlation Conclusion

Vulnerability correlation increases the effectiveness and value of NIDS. While it isn't possible to filter out 100% of the false positives due to partial VA vendor references, matches that are made are indeed true positives and deserve the prompt attention of the security analysts.

It is important to understand that vulnerability and compromise correlation techniques as described here are only one form of correlation—in fact, they may be considered the “correlation of last resort” as a threat that triggers these rules will have successfully evaded detection (or at least effective control). Correlation technologies exist⁹ that can identify threats in the reconnaissance stage, identifying attacks before they become successful compromises. By correlating threats earlier, threats can be managed and contained before they find a host vulnerable to the exploit. 

Joe Minieri, CISSP, is Director of Application Engineering at OpenService, an enterprise security information management company based in Westborough, MA.

¹ “2003 CSI/FBI Computer Crime and Security Survey”, www.gocsi.com

² See http://www.cert.org/stats/cert_stats.html

³ Koetzle, Laura, “No More IT Security Widgets, Please”, Forrester, December 2003.

⁴ The recent list of CVE and CAN ID's downloadable from <http://cve.mitre.org/cve/index.html> shows a combined vulnerability count of 7291 vulnerabilities. While SecurityFocus shows over 10,450 BugTraq ID's (<http://www.securityfocus.com/bid>). ISS's XForce database (<http://xforce.iss.net>) holds over 16,000 vulnerabilities.

⁵ See http://www.open.com/products/threatmanager/STM2_Vulnerabilities.shtml for details.

⁶ See http://www.cisco.com/cgi-bin/front.x/csec/getIDSinfo.pl?SIG_ID=3171&SIG_SUB_ID=0 Note, you will need a CCO user id and password to view this signature description.

⁷ See <http://www.open.com/products/threatmanager> for information on Security Threat Manager

⁸ See <http://cgi.nessus.org/plugins/dump.php?id=11214> for the Nessus details on this particular vulnerability.

⁹ Visit <http://www.open.com/products/whitepapers/whitepaperindex.shtml> for white paper on Real-Time Enterprise Risk and Vulnerability Management.