

# White Paper



Got Correlation? Not Without Normalization

---

# Got Correlation? Not Without Normalization

Colby DeRodeff, Member of Technical Staff  
ArcSight, Inc.

## Introduction

---

There have been many attempts by various groups to develop a standard that will allow the capture of a security event from any source and the conversion into a common format. Why would an analyst want something like this? Lets look at what steps an analyst might use to determine if the network may have been compromised.

A network Intrusion Detection System (IDS) can detect a web exploit targeted at a web server. As a first step, the analyst would review the perimeter router logs to see if the router passed the packet that triggered the alert. Based on the nature of this exploit, the probability that the packet was forwarded through the router is high. This is due to the fact that the exploit uses a standard TCP port (80).

Second the analyst would want to review the firewall logs to see if this was blocked by any of the filters that are in place. Since the firewall is "statefull" it could have blocked something that the router may have passed as acceptable traffic. Unless shown otherwise, at this point the analyst is sure that the packets reached the webserver so further investigation is necessary and the integrity of that box must be checked.

Third, to check the integrity of the webserver and look at all traffic that originated from the compromised box the analyst would run an application such as Tripwire, which is a file integrity checker using MD5 checksums, to see which files if any have been accessed or modified.

Fourth, the analyst would look at the Syslog output or the EventLog from that server, as well as pull the tcpdump data off the dedicated tcpdump host for the segment of time surrounding the attack to see what actually happened. At this point the analyst has accessed four different systems and looked at five different types of logs. That's a lot of work, and it takes time away from the vitally important tasks of securing the network and cleaning the compromised server to make sure that no other systems will be affected. **Based on common scenarios such as this, it is much more efficient to have all the relevant information located in one logging facility allowing the analyst to look at the data in whatever sequence or depth required.**

In the realm of intrusion detection there are many sources of information that can lead to an explanation for, or the confirmation of an exploit targeted at a network system. Confirming that the network has been compromised is like putting together a puzzle; the problem is that each piece is from a different jigsaw. When investigating an incident an analyst is dealing with a heterogeneous environment, where each device has a different logging format and reporting mechanism. There will

also be logs from remote sites where security policies and procedures may be different, with different types of network devices, security devices, operating systems and application logs in place. Hence, there is an urgent requirement for normalization and correlation.

## Correlation

---

What is correlation? Correlation is derived from the word correlate that means to be in or bring into mutual relation. That's the dictionary definition, but the "information security world" interprets correlation as the ability to access, analyze, and relate different attributes of events from multiple sources to bring something to the attention of an analyst that would have otherwise gone unnoticed. Referring to the earlier example, there are multiple pieces of information that can be "correlated":

- ❖ the accepted packet on the perimeter router,
- ❖ the accepted packet on the firewall,
- ❖ the IDS alert that detected a web exploit headed for the webserver,

all coming from the same source IP address. Along with the results of the integrity check, the collection of these alarms and status make it easier to confirm and reinforce the determination that the web server was indeed compromised and further action is necessary. In analysis it is ideal to access all the logs from the entire enterprise from a single console, and have them stored in one common database. A relational database is the most logical central storage facility because it supports querying and reporting. Without a commercial solution, an analyst would first need to get the logs from all these devices, normalize them, and insert them into the database in a common format. In order to have real correlation an absolute prerequisite is normalization.

## Normalization

---

How does normalization meaning, conforming to an accepted standard or norm, apply to hunting down hackers and examining log files? Picture a typical enterprise environment which consists of many different types of network devices ranging from border routers, VPN devices, to firewalls, to authentication servers, along with an even wider range of application servers like web servers, email servers, and always-critical database servers. These devices generate logs that are critical to an analyst who is responsible for the security of the site. It is seldom if ever the case that two manufacturers will use the same logging mechanism or format their logs identically. For example a Cisco PIX will not report an accepted packet in the same way as a Check Point firewall or even the same as a Cisco Router. The fact that the formats are all different makes it virtually impossible to store the log data in a common location such as a database without normalizing the events first.

The following are logs from different network devices are all reporting on the exact same packet traveling across the network. These logs represent a remote printer buffer overflow that connects to IIS servers over port 80.

### Check Point:

```
"14" "21Dec2001" "12:10:29" "eth-slp4c0" "ip.of.firewall" "log"  
"accept" "www-http" "65.65.65.65" "10.10.10.10" "tcp" "4" "1355" "  
"" "" "" "" "" "" "" "" "" "firewall" " len 68"
```

### Cisco Router:

```
Dec 21 12:10:27: %SEC-6-IPACCESSLOGP: list 102 permitted tcp  
65.65.65.65(1355) -> 10.10.10.10(80), 1 packet
```

### Cisco PIX:

```
Dec 21 2001 12:10:28: %PIX-6-302001: Built inbound TCP connection 125891  
for faddr 65.65.65.65/1355 gaddr 10.10.10.10/80 laddr 10.0.111.22/80
```

### Snort:

```
[**] [1:971:1] WEB-IIS ISAPI .printer access [**]  
[Classification: Attempted Information Leak] [Priority: 3]  
12/21-12:10:29.100000 65.65.65.65:1355 -> 10.10.10.10:80  
TCP TTL:63 TOS:0x0 ID:5752 IpLen:20 DgmLen:1234 DF  
***AP*** Seq: 0xB13810DC Ack: 0xC5D2E066 Win: 0x7D78 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 493412860 0  
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0241]  
[Xref => http://www.whitehats.com/info/IDS533]
```

All these formats are different and would be practically useless to store in a database with out normalizing them first.

Looking at the Check Point record it contains the following fields: event id, date, time, firewall interface, IP address of the firewall interface, logging facility, action, service, source IP, target IP, protocol, source port, some Check Point specific fields and then the size of the datagram. This is the most obscure format and it is especially hard to read with all the empty fields that are represented by double quotes.

---

**Got Correlation? Not Without Normalization:**  
**Normalization**

---

Now the Cisco router has a different format. The fields it populates are date, time, logging facility, event name, source IP, source port, target address, target port, and number of packets. The Cisco PIX, which one would expect to have the same format as the Cisco router since the same company makes them both, does not. It uses date, time, event name, source IP, source port, translated address or target address, target port, local address, and local port.

The final record is the Snort alert that claims this traffic was malicious. The fields Snort populates are exploit or event name, classification, priority, date, time, source IP, source port, target IP, target port, protocol, TTL (Time to Live), type of service, ID, IP length, datagram length, tcp flags, sequence number, acknowledgement number, window size, and tcp length. Snort also includes additional data such as references to investigate the exploit.

So how could these events possibly be productively stored in a database? It must first be decided which fields are interesting and develop a schema to accommodate the different fields that are populated by these devices. Choosing the fields must be content driven not based on semantic differences between what Check Point may call target address and what Cisco calls destination address. To accomplish this normalization, a parser must be coded to pull out those values from the event and populate the corresponding fields in the database. Here is an example of a database containing these alerts after they have been normalized.

Date	Time	Event_Name	Src_IP	Src_Port	Tgt_IP	Tgt_Port	Device_Type	Additional_data
21-Dec-01	12:10:29	accept	65.65.65.65	1355	10.10.10.10	80	CheckPoint	
21-Dec-01	12:10:27	list 102 permitted tcp	65.65.65.65	1355	10.10.10.10	80	Cisco Router	
21-Dec-01	12:10:28	Built inbound TCP connection	65.65.65.65	1355	10.10.10.10	80	Cisco PIX	
21-Dec-01	12:10:29	WEB-INS ISAPI_printer access	65.65.65.65	1355	10.10.10.10	80	Snort	TCP TTL:63 TOS:0x0 ID:5752 IpLen:20 DgmLen:1234 DF ***AF*** Seq: 0xB13810DC Ack: 0xC5D2E066 Win: 0x7D78 TcpLen: 32 TCP Options (3) => NOP NOP TS: 493412860 0

These are the same four events described earlier, except they have been normalized. This would be ideal for an analyst investigating an incident. With the data organized like this one could pull all records containing a value that is of interest or sort by any field that may be relevant. The problem is that entering this data into a spreadsheet manually is relatively easy in low volume but to get a program to do it is much more difficult. For instance the Check Point firewall reports target port as www-http--not 80 like most devices. Therefore there must be a lookup mechanism to ensure that www-http gets translated into port 80 otherwise this value would be useless during correlation. Another complication would be converting the date/timestamps. Since the devices all use a different format the program

cannot simply parse out the time stamp reported by the device. It would also need to convert it to a common format such as GMT.

## Conclusion

---

As large organizations move to manage enterprise security as a critical business process, correlation of events from disparate devices is absolutely essential for success. Without the efficiency and effectiveness that correlation introduces into the security process workflow, organizations will never catch up to the level of threat that they current face.

A foundation technology for correlation is data normalization. A key attribute that should be focused on when evaluating enterprise management solutions is the scope and deployment of the normalization mechanisms. Many products claim normalization but in reality capture only a portion of the relevant security information. Others only capture the data via a single mechanism such as SNMP traps. Effective correlation will only be available if the normalization process can capture and organize 100% of the relevant security information for every device and source in the network.

Looking at logs in twenty different formats and on four different consoles, as well as trying to find all the events across the network that may pertain to the attack being investigated is one of the hardest parts of any analyst's job. There is no way to visualize the sequence of events when they are stored in different locations, and visualization is one of the keys to deciphering a network attack. The ability to relate and analyze events from a multitude of vendors, from a variety of intrusion detection devices, and from all the event generating devices that make up the common enterprise network makes every analyst's puzzle a little easier to solve.