# Correlations between quiescent ports in network flows

Joshua McNutt
Markus De Shon
*CERT Network Situational Awareness Group,*
*Carnegie Mellon University, Pittsburgh, PA 15213, USA*
{*jmcnutt,mdeshon*}*@cert.org*

## Abstract

TCP/IP ports which are not in regular use (quiescent ports) can show surges in activity for several reasons. Two examples include the discovery of a vulnerability in an unused (but still present) network service or a new backdoor which runs on an unassigned or obsolete port. Identifying this anomalous activity can be a challenge, however, due to the ever-present background of vertical scanning, which can show substantial peak activity. It is, however, possible to separate port-specific activity from this background by recognizing that the activity due to vertical scanning results in strong correlations between port-specific flow counts. We introduce a method for detecting onset of anomalous port-specific activity by recognizing deviation from correlated activity.

## 1 Introduction

The CERT Network Situational Awareness Group is using SiLKtools [1] to analyze Cisco NetFlow data collected for a very large network. Details on the functionality of SiLKtools can be found in other publications from our center. [1]

The analysis techniques in this paper can be used on any flow-based data source. In our case, the analysis is performed on hourly summaries on the inbound number of flows, packets and bytes on each port, where "inbound" simply refers to traffic coming into the monitored network from the rest of the Internet.

Analysis of network flows for anomalous traffic can be challenging due to fluctuations in traffic that are resistant to variance-based statistical analysis. [2] We have discovered that, for a restricted set of network phenomena, correlations between flow counts on different ports can be a useful way of filtering out "background" activity.

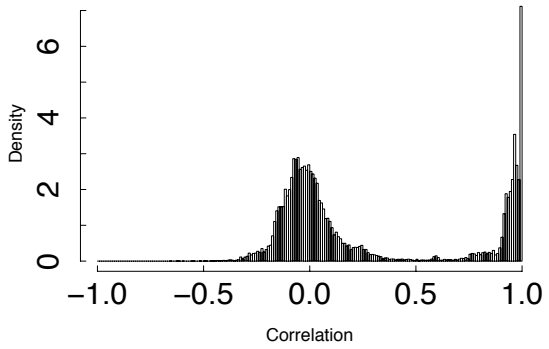


Figure 1: Histogram of robust pairwise correlation values (not including self-correlations or duplicates, since the correlation matrix is symmetric) for hourly flow counts on server ports 0-1024. Note that a significant proportion of the ports are well-correlated.

## 2 Correlations

Our data shows extremely high correlation (frequently $> 0.99$) between flow counts on many ports which do not have active services running on them (see Fig. 1). Because of the lack of "normal" traffic on these ports, any activity which is present is very likely to be due to vertical port scanning. As long as the port scanning proceeds quickly enough, then a vertical scan deposits the same number of flow records for each port within the time period over which flow counts are summed. Thus, the number of flows to each port will be highly correlated with the number of flows to other quiescent ports. This characteristic is indeed observed on the very large network that we are monitoring.

Given that a set of ports are normally correlated, then by calculating the median of the number of flows on each

port in an hour, and then subtracting that median value from the number of flows observed on each port in that hour, we can remove the correlated background activity from analysis. This background-subtracted time series can then be analyzed for port-specific behavior through normal peak-finding algorithms.

A useful (though untested) method for detecting the remaining peaks might include using a "trimmed mean" (mean calculated from the data points remaining after removing outlier data points) and "trimmed standard deviation." The "trimmed" mean and standard deviation would be used similarly to the ordinary mean and standard deviation to identify outliers (flow counts which lie above the mean by some multiple of the standard deviation).

The use of the median instead of the mean, and the use of "trimmed" means and standard deviations in our method is for the same reason we used a "robust" correlation method as described below–to prevent outliers (which would be the things we are trying to detect) from attenuating the sensitivity of the detection algorithm inappropriately.

## 2.1 Correlation clusters

If traffic on port A is correlated with port B, and port B is correlated with port C, then port A is also correlated with port C. Thus, ports A, B and C will form a cluster of correlated ports. We processed our correlation matrices to discover such clusters of ports which are all mutually correlated. These clusters are surprisingly large, and could be even larger in situations (unlike our own) where traffic is not filtered (a darknet, for example).

To prevent isolated anomalies during the learning period from interfering with identifying the true clusters, we used a "robust correlation." The robust correlation measure is calculated using the minimum volume ellipse approach. This method was discussed in [3] in the context of calculating robust statistical distances. Since correlation is a measure which is highly sensitive to even one or two outliers, we wish to exclude extreme observations. Therefore, the data used for the correlation calculation consist of all points enclosed by the 95 percent minimum volume ellipse. This is the smallest possible ellipse which covers 95 percent of our data.

For incoming traffic on TCP ports 0-1023, using the "robust" correlation measure, and requiring a correlation $\geq 0.96$ for one port to be considered correlated with another port, we found a single cluster of 133 ports, and a second cluster of 3 ports. More careful analysis may reveal the clusters of mutually correlated ports to be larger, if some ports had sustained anomalous activity, but are otherwise well-correlated.

# 3 Server ports

The ports numbered 0-1023 are by convention reserved for use by server programs owned by the "superuser" or system user. For this reason, the traffic patterns on these ports are quite different than for the higher-numbered "ephemeral" ports. The traffic on our network is consistent with this generalization.

Since a number of ports in the server port range are in active use by common services (most notably "web" traffic on ports 80/tcp and 443/tcp, and "email" on port 25/tcp), and others are usually filtered on real-world networks, not all ports would be expected to correlate well. However, unused ports, whether unassigned or obsolete, would be expected to have very little active traffic; we call such ports "quiescent," i.e. mostly quiet.

Correlations on quiescent server ports arise from the presence of vertical scanning activity (where a source host is scanning through all port numbers, or at least the server port numbers, on a target host). Deviations from correlated activity would be expected in the case of horizontal scanning (scanning for a particular port across hosts). An onset of horizontal scanning on a particular port might be expected if a new vulnerability is announced in an obsolete, but still present, service.

Onset of sustained activity which deviates from prior correlation could indicate the presence of a worm (self-propagating exploit program) on that port.

# 4 Ephemeral ports

The ports numbered 1024 and above are used by user space programs, primarily as temporary ports for outgoing connections by client programs such as web browsers. While this convention has been blurred somewhat by peer-to-peer programs and other unprivileged servers, the model still holds for most ports.

In our data set, nearly all the ephemeral ports show strong correlations, with daily and weekly seasonal patterns (see Figs. 2 and 3). This is consistent with the rhythms of user-driven traffic, meaning that the data comes from user space client programs connecting out to servers. Because such a connection creates two flow records (one for the outbound connection, but another for the return traffic within the same TCP session), we see return traffic flows in our "incoming" data set. An analysis of ephemeral port traffic verifies this hypothesis, with most of the data (where the definition of "most" depends on the day and time of day) consists of traffic from source ports 80/tcp, 443/tcp and 25/tcp, in that order.

Future improvements to our flow record collection software will allow easy differentiation of true incoming flows vs. return traffic from outbound connections. For

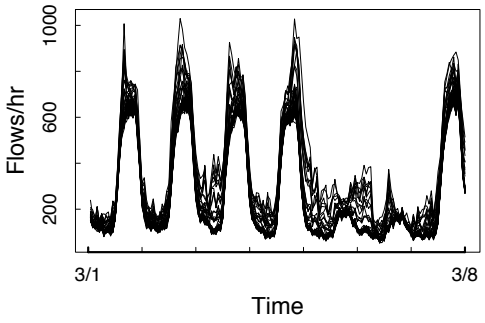


Figure 2: Example of incoming flow counts for 50 ephemeral ports for a one week time period in 2005. Note the daily and weekly seasonality consistent with user-generated activity.

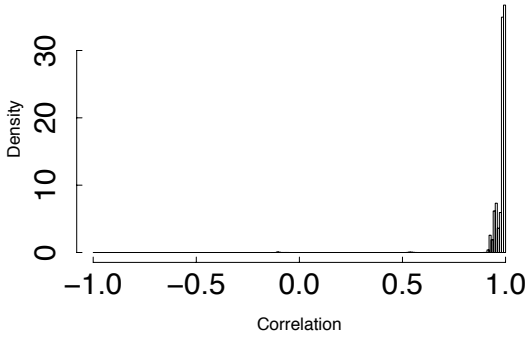


Figure 3: Histogram of correlation values for pairwise correlations between 1024 ephemeral ports (specifically, 50000-51024), excluding self-correlations and duplicates (since the correlation matrix is symmetric). Note the high concentration of highly correlated pairs.

the purposes of the analysis method described in this paper, however, the distinction is unimportant, as the return traffic patterns are highly correlated, as are any vertical scans taking place (though analysis has revealed that vertical scans of ephemeral ports are rare in our data). Deviation from correlated activity will have already removed the background of return traffic flows and vertical scanning, at least approximately. Any remaining significant peak activity will be due to special attention to a particular port or set of ports, just as with server ports.

Possible explanations for the onset of persistent deviant activity on an ephemeral port include: widespread scanning for a particular backdoor, port activity due to a new peer-to-peer protocol, the onset of activity for a worm that uses a particular ephemeral port to spread or perform other tasks, or scanning or exploit of a vulnerability in a mostly quiet server running on a high-numbered port.

## 5 Port 42/TCP, a case study

Port 42/TCP hosts the Microsoft Windows Internet Naming Service (WINS) service on Microsoft Windows hosts, an obsolete directory service which nevertheless was present in some versions of Windows as recent as Windows Server 2003, for backwards compatibility. On November 25, 2004, a remote exploit vulnerability in the WINS service was first announced by CORE Security Technologies (www.coresecurity.com) to their CORE Impact customers, in their exploits update for that day. The next morning, a more public announcement was made by Dave Aitel of Immunity, Inc. (www.immunitysec.com) on his "Daily Dave" email list. This vulnerability was later assigned CVE number CAN-2004-1080, and is discussed in CERT Vulnerability Note VU#145134.

In Fig. 4 we compare the data for incoming flow counts, destination port 42/TCP, to the median of incoming flow counts to several other ports. Fig. 5 shows the difference between the 42/TCP data and the median data. These two plots cover approximately a two month time period in 2004 preceding the announcement of the WINS vulnerability. The median value from a correlation cluster is used (rather than the mean) because a large deviation in one of the time series could significantly affect the mean, but not the median. The difference between a flow count and the median flow count for the cluster, therefore, would be a better indicator of the deviation from the expected value.

There are two significant periods of deviation in 42/TCP in the two month period before the announcement of the WINS vulnerability, which are explained below. The important thing to note is that the deviant peaks in the two week time period around October 15th, and the peak at October 28th, are well within the normal variability of the data. The correlation technique separates the background (due to vertical scanning) from the signal we are looking for (due to special attention to port 42, or port 42 and some list of other ports).

In early October, two IP's scanned a set of 18 mostly non-contiguous ports (e.g. 22, 25, 53, 1080) on the monitored network, including port 42. Because of the larger number of ports targeted, these scans probably do not indicate foreknowledge of the WINS vulnerability to be announced the next month. Instead, the set of ports could have been used to determine active hosts, and a simple OS identification (port 42 indicating Microsoft Win-

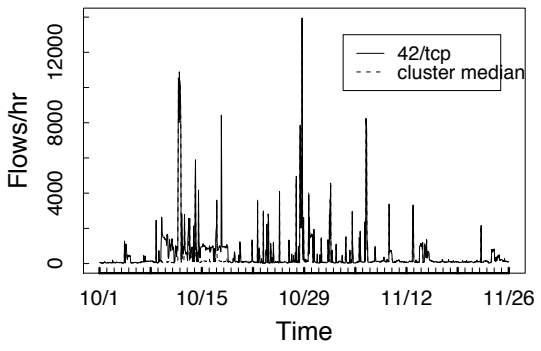


Figure 4: Incoming flow counts to destination port 42/TCP from our data and median incoming flow counts to several other destination ports. The dashed line showing the median flow counts is mostly obscured by the solid line because of the high correlation. An exception is the uncorrelated 42/TCP peaks in the two week period centered on October 15th (which are pictured more clearly in Fig. 5). Note that the uncorrelated peaks are within the normal variation of the activity.

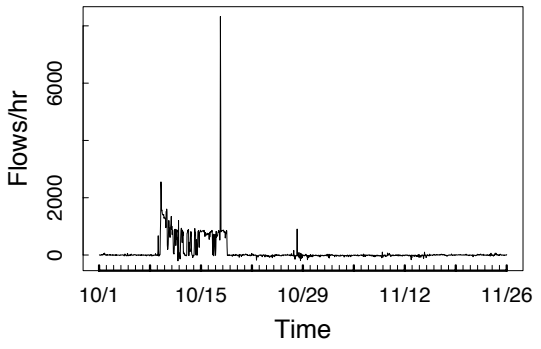


Figure 5: The difference between the two time series in Fig. 4. The two-week period of deviation, and the smaller isolated peak, are explained in the text.

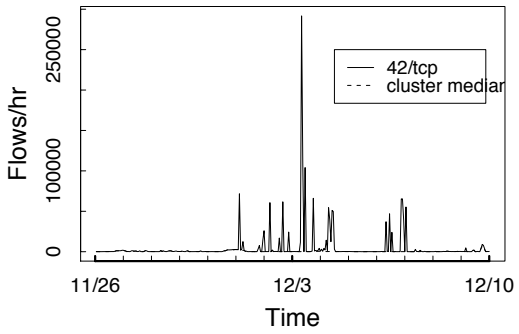


Figure 6: Incoming flow counts to port 42/TCP after the announcement of the WINS vulnerability, compared to median incoming flow counts to several other ports.

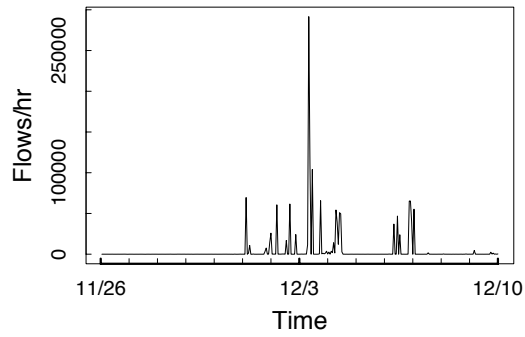


Figure 7: Difference between flow counts to port 42/TCP and the median flow count to other correlated ports, after the WINS vulnerability announcement.

dows, for example).

The smaller peak in late October appears on closer analysis to be benign activity, possibly due to some legacy systems attempting to use the WINS service.

While these these port 42-specific activities do not represent important security events, the fact that they were found easily using this method indicates that port-specific activity of a more malicious nature, which would otherwise be obscured by the background noise of vertical scanning in the server port range, could be discovered easily using the methods described in this paper.

The data after the WINS vulnerability announcement shows a significant peak in the number of incoming flows starting on December 1st at 2:00am GMT, but the number of hosts involved was still small. By midnight GMT of that same day, however, the number of hosts had surged considerably, and it would have been clear that there was new, widespread interest in port 42/TCP.

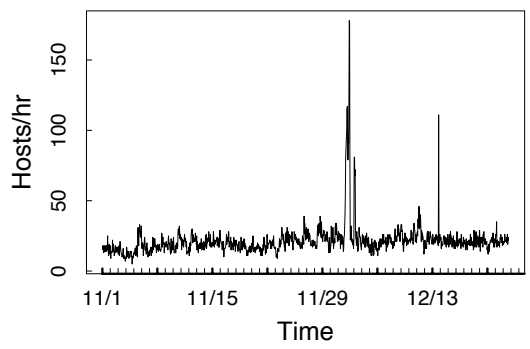


Figure 8: Number of unique hosts per hour attempting connections to 42/TCP from the Internet. By 12:00am GMT December 2nd the number of hosts was clearly much higher than had previously been seen.

The first public announcement we were able to find of widespread scanning on port 42/TCP was on December 13, in an email message by James Lay to the Full Disclosure email list—11+ days after significant scanning was clearly visible in our data using our correlation technique. If we had been using our correlation technique operationally at that time, an earlier announcement of widespread scanning would have been possible.

## 6 Port 2100/TCP

Port 2100/TCP lies in the "ephemeral" port range, but is actually also used for the Oracle FTP service. An exploit was released on March 18, 2005 for a vulnerability announced in August of 2003.[2] Fig. 9 clearly shows that scanning of port 2100/TCP commenced at that time.

## 7 Conclusions

Our analysis of port 42/TCP traffic shows a clear onset of scanning activity specific to port 42 after the announcement of the remote exploit vulnerability in the WINS service announced in late November of 2004. The scanning activity was clearly detectable well before any public announcement of such scanning.

The usefulness of subtracting the correlated background from per-port traffic summaries to detect port-specific behavior lies in the simplicity of the method, and in its ability to ignore vertical scanning as well as the background of web/email activity.

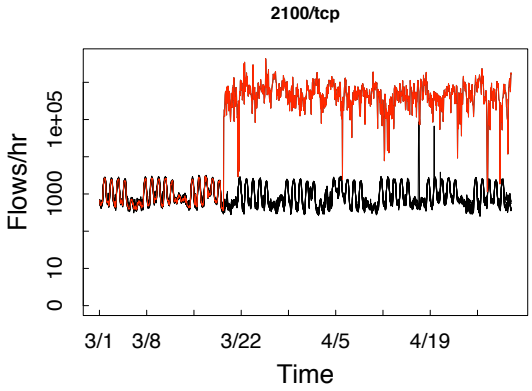


Figure 9: Flows per hour incoming to port 2100/TCP in March–April 2005 (red, or upper, line) as compared to nine ports which were correlated to 2100/TCP at the beginning of that time period.

## References

[1] CARRIE GATES, MICHAEL COLLINS, E. A. More netflow tools: For performance and security. In *LISA XVIII* (2004), pp. 121–131.

[2] LELAND, W. E., TAQQ, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of Ethernet traffic. In *ACM SIGCOMM* (San Francisco, California, 1993), D. P. Sidhu, Ed., pp. 183–193.

[3] ROUSSEEUW, P. J., AND VAN ZOMEREN, B. C. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association 85* (1990), 633–639/648–651.

## Notes

[1] http://silktools.sourceforge.net/

[2] http://www.oracle.com/technology/deploy/security/pdf/2003Alert58.pdf