

# **NISCC Technical Note 03/03: Protective Monitoring - Introduction to Audit and Accounting Log Analysis**

## **Background**

The purpose of this technical note is to help organisational security officers and system administrators analyse accounting and audit logs generated by IT systems for signs of suspicious or malicious behaviour.

This note does not discuss host based intrusion detection systems that perform analysis of logs on a particular host for signs of intrusions. The reason for this is that the logs typically available for a networked system will be from a variety of different sources and may not include sufficient information to enable positive identification of an intrusion. For details on host based intrusion detection systems see NISCC [Technical Note 05/02](#).

## **Summary**

The note provides advice on the sources of logs within a network, definition of a common format for logs, attack patterns and their correlation. It is based on experience gained by CESG during the development and piloting of a proof-of-concept log analysis capability.

The main conclusion is that analysis of logs is a practical proposition and can be implemented in both the public and private sectors using the techniques set out in the paper. However, it needs to be borne in mind that log analysis remains a research area and that the range of commercial log analysis products available is currently small.

## **Sources of Logs**

IT systems and networks contain a variety of components that are capable of generating logs, including:

- Domain Controllers
- Workstations
- Firewalls
- Proxy servers
- Web Servers
- Application servers

Each will generate records in a defined format, which may conform to a common standard (eg the extended logging format put forward

by the World Wide Web Consortium, see <http://www.w3.org>) or may be proprietary (as is common with firewalls). It is normally possible to configure the component to select the amount of logging required.

There are commercial products on the market that will process these logs, but most will not cover the whole range of network components, for example some may concentrate on logs from web servers and domain controllers, while others may handle web server logs and firewall logs. Thus, if analysis of a single source is required, then an appropriate commercial product is likely to be available. However, if an overall picture is required, then it will probably be necessary to develop appropriate tools on a commercial database.

### **Sources of Threat**

There are two sources of threat to the security of a network:

- External entities, such as the large pool of potential attackers on the Internet
- Internal entities, such as network users that may compromise the integrity, confidentiality or availability of the network either accidentally or through deliberate action

A typical network will provide more services to its internal users than it offers to outsiders (if it offers any). The network will usually be separated from the outside world by a firewall, which may perhaps allow only limited access across it by outsiders, eg to access a public web server. The services accessible to the outside world will normally be segregated from the internal network and reside in what is commonly called a "demilitarised zone" or DMZ.

In most cases, an external entity will not have authorised access to a network and will first have to penetrate the network, for example by exploiting a vulnerability in the boundary protection device (eg a firewall) or by extending their privileges on an internal component such as a web server to which they have legitimate, but limited, access. An internal user will, however, already have access to the network and some level of privilege. Consequently, it may be necessary to collect logs from different network components to track the activities of the two types of attacker, and depending on the facilities available to each.

Malicious activity by internal users may take a number of forms, for example attempting to access data for which they are not authorised, attacking network components or using the network as a host to attack other (external) networks. In such cases, the logs

from domain controllers, application servers and file servers would be relevant. In the final example, the logs from the boundary devices would be important.

The data needed to identify an external attacker will come from the logs of components that can be accessed from outside the network, such as a firewall, router, web proxy and mail server. Such logs will give the first indication of suspicious activity as an attacker will need to compromise one or more of these components before they can advance further into the internal network. However, once an external attacker has penetrated the internal network they effectively become "internal" and the same log sources that apply for a malicious internal user must be used.

A successful attack will probably result in the loss of confidentiality, integrity or availability (or potentially all three). In the case of a loss of confidentiality, an attacker may move data to an accessible location. An external attacker will then need to exfiltrate the data through the boundary protection components. In this case logs from components such as mail relays and web proxies will need to be examined.

## **Log Analysis Issues**

**Log Configuration** – Log data gives an indication of the activity undertaken by an attacker, but it cannot give a complete picture as it does not record the contents of the IP packets. If records are generated only in respect of denied activities, then the picture is reduced further, for example this might show an attacker performing a port scan but miss the data for the attacker returning to a port that is shown as open by the scan. It may therefore be as important to have a record of permitted activity as of that which is denied.

A record of outbound traffic may also be important, as this may provide evidence that a network has been compromised and that (a) data is being exfiltrated by the attacker or (b) the system is being used to attack a third party system.

It is recognised that such an approach would lead to an increased volume of data being recorded, but this may be necessary to allow analysis of an attack. Once the attack has finished and the perpetrator has moved on to their next target, it will be too late to enable logging to obtain further information.

It is recommended that system administrators should not allow the overwriting a log file as a means of reducing storage capacity requirements, and should not disable logging when a particular

percentage of storage capacity is reached. Doing so may seriously limit the capability to investigate an attack, and may mean that a successful attack could occur without being detected.

Ultimately, the configuration of systems to generate logs will be a matter for local decision. It is recommended that a clear policy for the system or network should be agreed, documented and enforced, specifying the events to be logged, the level of detail required and archive and retention periods for logs.

**Know your Network** – It is important that the log analyst should have a good understanding of what constitutes “normal” or “allowed” behaviour on the system under examination. This will reduce the number of “false positives”, i.e. activity incorrectly determined to be malicious. The capability to investigate some of the issues raised by log analysis will develop further understanding and allow the analyst to develop a method for determining which activities are important and require further investigation.

**Log Formats** –All network components will, in general, produce logs in their own specific formats. They may also produce logs that are, at least initially, in a format that can only be read by their own proprietary log viewers (a good example of which is Checkpoint’s Firewall-1 product).

To be able to analyse logs from multiple systems simultaneously, it is important to be able to manipulate the data into a common structure. This is made easier if all the data is in a single format, such as ASCII text, as this allows the data to be manipulated as desired. Most systems that produce logs in a proprietary format often have a mechanism for exporting them as text. For example, for Firewall-1 the web page [www.enteract.com/~lspitz/logger.html](http://www.enteract.com/~lspitz/logger.html) describes how the logs can be exported as ASCII text.

**Common Log Format** – Once the log sources are in an appropriate output format, they should be manipulated into an appropriate data structure, such as a table within a database, which can then be analysed for signs of malicious activity. The exact structure used will depend on the types of systems from which log data is being collected.

The structure used in the proof-of-concept system developed at CESG is shown in full in Table A.1 of Annex A. Table A.2 shows how the logs from some typical components used within networks map onto this format. Other fields are used in Table A.1 which are specific to particular applications, but these are not shown in Table A.2 as these are not currently used in the analysis and are merely

placeholders for data that may be used at a later date when specific queries are developed for them.

The structure shown in Annex A reflects the type of data that has been available during the development of the CESG database. It may not completely match the needs of every system, but can be extended if required to accommodate other components that generate logs, for example database servers.

Typical fields that are required are:

- Date and time of activity
- Source and destination information – typically these will be IP addresses but could also be machine names.
- Protocol – TCP, UDP or ICMP
- Service information – such as source and destination port information, or ICMP codes and types
- Action – In the case of a firewall, this may be permit, deny or drop for example
- HTTP information if analysing web server logs – such as Method, URL, Status Code
- User Information – such as authenticated user information or client information

**Correlations** – If evidence of malicious activity is found in the logs produced by a component, the incident should be investigated. To do this, logs from other components should be examined and the information they contain correlated to build a full picture of activities at the time of the incident. Correlation can be achieved either by tracking an identifier related to the source (such as IP address) or by looking at the activity throughout the system in time order (temporal correlation).

The simplest method is to track the activity based on the IP address of the attacker (this assumes that the attacker is not using a spoofed address, but, even in this case, it does provide an identifier that can be searched for). This will enable the activity to be tracked from the front door (firewall) through to any internal system that records source IP address. But the use of this type of identifier for tracking is complicated by the widespread use of proxy servers, which mask the true source and usually result in the proxy IP address being recorded. Unless the proxy server is under your own control, it is unlikely that it will be possible to track back to the proxy to determine the true source identity.

Where proxies are being used, the best chance of tracking an attacker through a system is by temporal correlation, i.e. using the timestamps of the log records. This involves attempting to locate

sequential events across potentially multiple log sources. However, for this to be successful the network components must be synchronised (for example, using an NTP server). Correlating activity in this way is extremely difficult if every log source is operating its own independent clock.

**IP Address Resolution** – Ultimately, the source IP address needs to be converted to its owner’s name, especially if it is the intention to file abuse reports with the source’s ISP. This can be quite an overhead, especially if the analysis system cannot be connected to the Internet to perform the necessary “whois” lookups. Some of the Internet registries (such as RIPE) offer the ability to download their databases, enabling the resolution to be performed offline.

**Limitations of Log Analysis** – In general, log analysis is an “after the fact” process, though some commercial products do offer real-time capability (making them similar in function to host-based Intrusion Detection systems).

Logs alone can never provide the complete picture nor fully describe the intentions of the attacker. For example, a firewall may record an attempt to connect to a port but it is unlikely to record what was in the IP packets, apart from possibly the TCP flag information. Therefore, it is only possible to surmise that an exploit was intended but it is not possible to identify the specific exploit, especially in the case of ports associated with services such as FTP which have a number of publicly known vulnerabilities.

Logs alone are not usually enough to identify the skill level of an attacker, although skilled attackers are not likely to leave obvious traces of their activities. A combination of the logs together with an idea of the contents of the packet would enable a detailed assessment of the skill level of the attacker and therefore the risk posed by the source. (A paper by Toby Miller at <http://www.incidents.org/detect/rating.html> describes a metric for making such an assessment which, although currently immature, may develop into a useful tool for future use).

## **Attack Patterns**

There are many useful resources on the Internet that describe the structure of exploits and their potential effects. These sources can also be used as an indication of the type of data within a log that indicates an attempt at exploiting the vulnerability. Annex B provides a non-exhaustive list of useful resources.

An attack against a network or system typically falls into 3 phases, though not all attackers will necessarily fall into this pattern:

- a. Reconnaissance
- b. Compromise
- c. Consolidation

In the reconnaissance stage, the attacker will attempt to gain knowledge of the topology of a network and the systems within a network. This may include attempting to map the network, port scanning/probing, Operating System fingerprinting or password guessing.

These activities may manifest themselves in a variety of guises, depending on the preferred mechanism of the attacker. There are many ways to obtain this kind of information from a network, some of which involve an element of stealth. The kind of activities to look for include:

- DNS (Domain Name Service) interrogation, such as zone transfer attempts;
- Traceroute packets;
- Ping sweeps attempting to determine "live" IP addresses on a network;
- Port scanning;
- Active OS fingerprinting activities;
- Attempted connections to ports 139 and 445, used for file and print sharing in the Microsoft Windows operating systems;
- Attempted connection to port 389 and 3268 used by Active Directory in Win2k;
- Attempted connection to SNMP ports;
- Suggestion of banner grabbing against web, FTP or e-mail servers, e.g. use of HEAD command against a web server;
- Attempted connection to port 79 (finger);
- Attempted connection to port 111 to enumerate RPC services on Unix systems or port 135 on Microsoft systems;
- Attempted use of default accounts
- Multiple password failures or account lock-outs;

DNS zone transfers are primarily intended to be used to synchronise the data held by secondary DNS servers with their primary. However, it is often possible for any individual to get access to this information. Typically, performing a zone transfer will yield the IP addresses of the machines within a network and may identify particular types of servers (web servers, mail relays) and give information as to the operating systems of the machine (via the optional HINFO records).

Ports 139 (NT) and 445 (Win2k) are used for file and print sharing. If the attacker can create a session via these ports then it is possible to enumerate the target network for machines, shares and user accounts.

Two options for enumerating Unix systems are through ports 79 (*finger*) and 111 (*portmapper*). The first of these, *finger*, gives information regarding users of a system, including those currently logged on and their "idle" times. Traffic to port 111, such as by the use of *rpcinfo* can yield information as to the ports different RPC services are mapped to, against which other commands may be run (such as "showmount") to enumerate other network properties.

In the vulnerability exploit stage, the attacker will look to exploit a potential weakness in a system to gain escalation of privilege, cause an unexpected behaviour on the remote system or perform an action that will result in a denial of service.

In general, the activities in this stage will target particular systems within the network that are vulnerable to a particular exploit. The type of exploit used will obviously depend on the information gained during the reconnaissance phase. For example, if a port scan of the target revealed that the FTP port was open, then an exploit against this service may be employed.

Due to the wide range of exploits that could be attempted, it is not possible here to provide an exhaustive list of the types of activity to look for. Various sources exist that give information regarding vulnerabilities and their exploitation and these are a valuable source of information as to what signatures these exploits may have in a log.

Alternative sources of vulnerability exploit information are the rule sets from Intrusion Detection systems (IDS). An IDS works by comparing the contents of network traffic to a set of defined signatures, and generates alerts when a match is found. Some IDS systems, such as the open-source application *Snort*, have their rules files readily available, and these can be used to determine potential patterns that may be of interest if seen in a log file.

In the final stage, having gained access to the target platform, the attacker will usually attempt to exploit their success, potentially facilitating their future access to the now compromised platform by installing a backdoor and possibly covering their tracks by deleting any audit records pertaining to their activities.

Possible activities to consider during this phase are:

- Attempted outbound connections using FTP, SSH or telnet;
- Attempted outbound connections via IRC channels;
- Creation of new user accounts with administrator privileges;
- Attempted use of default accounts;
- Contents of security event logs showing escalation of privileges or successful/failed attempts to access files or directories.
- Unusually large amounts of outbound web traffic.

It is possible that the attacker has compromised a host to act as an intermediary (or “zombie”) in an attack against a third party, thus masking the true source of an attack. For this reason, the presence of unusual connection attempts to remote systems, high levels of outbound traffic or unusual traffic profiles should be investigated.

The majority of these can be mitigated by good security practices, such as using a well-configured firewall, that only allows business critical services across the boundary and applying the relevant security updates to fix relevant vulnerabilities. The use of an IDS tuned to the vulnerabilities of the components within the network and the services allowed across the boundary will also identify potential exploitations. Knowledge of their occurrence will give an indication of the level of threat to a system. Ultimately, log analysis is another technique that can be brought to bear to gain assurance to the security of a computer system.

## **Data Queries**

This section aims to provide a flavour of what can sensibly be extracted from log data to enable a System Administrator or IT Security Officer to determine whether malicious activity has been attempted against a system under their control.

The mechanism by which the queries are performed will depend on the structure in which the data is stored. For example, if the data is stored in flat text files it may be possible to define batch files that process the data and output text files containing data of interest. Alternatively, if a database is employed, then more powerful search techniques may be brought to bear against the data set.

The “proof-of-concept” system developed at CESG is based around an Oracle database with the queries developed using the PL/SQL programming language. Such a database improves the ability to manipulate the data and produce outputs in a more accessible format. The following paragraphs describe some of the areas for which queries can be developed in more detail.

**Port Scanning/Probing** – Port scanning or probing is perhaps the most common activity that is observed in the audit logs from Internet-connected networks. The probing may be targeted at a particular organisation, but often is a result of the affected network residing in a large network block that the attacker is systematically scanning for opportunities.

Scanning activities can be characterised in two ways, “vertical scanning” and “horizontal scanning”. Vertical scanning features a single target being probed for a range of ports, whereas horizontal scanning activity covers a range of IP addresses probing for a single (common) port on each host.

Activities of interest with regard to scanning will depend on local conditions, for example the ports that are open on a network’s firewall and the services offered by the network to external entities.

Perhaps of prime importance within scanning activities is the response of an attacker when a port is noted as “open”, either as part of a vertical or horizontal scan. If the attacker then revisits the open port, then it is important that the logs of any back-end server that may be affected are examined for evidence of suspicious activity.

Other areas to be aware of include:

- TCP connections to port 53 assigned to DNS. Name resolution uses UDP on port 53, whereas TCP connects may imply attempts at performing a zone transfer;
- Trojan horse activity – These usually operate on well-known ports, e.g. the *SubSeven* Trojan uses TCP port 27374. Be aware of these activities both inbound and outbound of the network.
- DoS or DDoS tools – Similarly to Trojans, these tools, such as *Trinoo*, usually operate on well defined ports. Again, be aware of these activities both inbound and outbound of the network.

**ICMP Traffic** – ICMP (Internet Control Message protocol) is the mechanism by which IP stacks can send simple messages containing information or errors. However, it can also be used for more nefarious purposes such as network scanning, OS fingerprinting and denial of service attacks.

ICMP ping (more correctly described as an ICMP Echo Request) traffic is familiar to most users as it provides a mechanism for determining whether a machine is “live” on a network. Consequently, it can also be used to scan IP blocks to determine which are being used within a network.

A slight variation of this enables "ping" traffic to be used to perform a denial of service attack against a third party system, the so-called 'smurf' attack. This is described later in this document.

ICMP traffic can also be used to fingerprint systems, for example Address Mask Requests (ICMP type 16) should only be responded to by routers and can therefore be used to identify these within a network. Similarly, an ICMP Timestamp Request (type 13) will only elicit a response from a Unix system. Other techniques include the use of ICMP subtypes or codes and the method in which these are handled by the target system that potentially differentiates between Microsoft and Unix platforms. In general, only "destination unreachable" and "echo responses" (to generated "echo requests" 0 should be allowed across a boundary into a network.

ICMP traffic can also be potentially used to perform Denial of Service attacks against a system. An obvious example is the sending of large numbers of oversized ping packets, the maximum size allowed being 65k. Sending these packets across an Ethernet segment will generate large numbers of fragments that will arrive at the target destination. Less obvious means are the following:

- ICMP Redirects (type 5) – These are used to adjust routing tables, and have subtypes for Hosts and networks. If an attacker can get a router to accept such a packet, they can effectively remove their target from a network
- Source Quench (type 4) – These packets are generated to throttle the transfer rate of the sender to the receiver. A malicious entity could attempt to send large quantities of these packets to their target in an attempt to slow the target's data transfer rate to such a level that service is effectively denied.

**Web Server/Proxy Traffic** – Exploitation of holes in web server software tend to be the most visible types of attack, with defacement of web sites a common occurrence and the higher profile events receiving wide publicity.

The mechanisms for exploiting these applications are widespread, whether they exploit some native behaviour, result from a buffer overflow or just bad design. There are many useful resources that can be used to determine which attacks are relevant to local topology, such as [cve.mitre.org](http://cve.mitre.org) or *Bugtraq*. Similarly, the rules for web traffic for the *Snort* IDS can also be used to determine relevant strings that may appear in log files.

Often, an attacker will attempt to fingerprint a server before launching an attack in order to determine whether a server is vulnerable to a particular attack or so as to tailor their attack to the server in question, for example so as to avoid aiming an IIS-specific exploit at an Apache web server. Signs of such fingerprinting may be

- Use of the HEAD command, grabbing banners from the web site;
- GET requests with bad request data.

It is important to correlate the presence of the "attack signature" in the web logs with the HTTP response code that the request generated. A success indication (HTTP status code 200) for the "attack signature" should result in a fuller investigation of the target server.

**Denial of Service** – The concept behind Denial of Service attacks is the desire of an attacker to remove their target's ability to connect to a network. There are many methods by which this can be achieved, from direct attacks against the target, use of distributed "zombies" or the modification of DNS records such that the target domain is no longer reachable.

Several examples of such attacks that may be evident in log records are:

- **Smurf attack** – This attack involves 3 entities, the attacker, their target and an amplifier network. The attack works by the attacker sending ICMP Echo requests to the network broadcast address of the amplifier domain, having first spoofed the source address to be that of the target domain. Any address within the amplifier network that is able to respond will return ICMP echo responses packets to the target domain.

Therefore, two types activity should be searched for within logs, ie:

- a) Large number of ICMP echo responses from a domain indicating that network is being targeted for attack; or
  - b) Large number of ICMP Echo request packets to the network's broadcast address suggesting an attempt to use the network as an amplifier in the attack.
- **Fraggle attack** – This attack is similar to the smurf attack except in its use of UDP traffic rather than ICMP. The traffic

will typically be directed to UDP port 7, assigned to echo or port 13, chargen.

- **SYN Flood** – This attack exploits the three-way handshake used to establish TCP connections. The attacker sends a SYN packet with the source address spoofed, which elicits a SYN-ACK from the target to the 'source'. As the 'source' has been spoofed and is effectively unreachable, the expected "ACK" is never received by the target in response. The target holds the connection open until a time-out occurs, thus using system resources. A mechanism for counting the number of connections from a source and alerting those that exceed a set threshold limit within a defined time period may give an indication that this form of attack has been attempted.

**Internet Worms** – In recent years, there has been a large increase in the number of instances of malicious code causing widespread damage across the Internet, exploiting vulnerabilities in popular web server software to compromise huge numbers of Internet-connected computers. Perhaps the most widely known, and still very active examples of these, are the Code Red and Nimda worms.

The ability of these worms to spread rapidly across the Internet means that very quickly details of their actions becomes well publicised. It is therefore possible to develop queries that look for the particular signature of these worms, for example with the Code Red worm, there are entries in web server logs featuring a "GET" request incorporating default.ida followed by a very long string of the letter "N"). Similarly, Nimda has a well defined pattern of 16 requests attempting to either exploit the backdoors left by infection with Code Red or to gain a command shell using Unicode to traverse directories.

Particular attention in these instances should be paid to the status code returned by the web server, a value of "200" indicating that the request was successful.

**NT Event Logs** – A full description of NT Event logging is beyond the scope of this paper, a good source of information being listed in the reference section. A selection of security-relevant audit events are listed below by Event ID and description.

- Event ID 512 – System Restart
- Event ID 517 – Audit Log Cleared
- Event ID 529 – Unknown Username or Bad Password (if multiple occurrences within a short time period)
- Event ID 533 – User not allowed to log on

- Event ID 539 – Account locked out
- Event ID 608 – User right assigned
- Event ID 612 – Audit policy changed
- Event ID 624 – User account created
- Event ID 633 – Global group member added (involving administrator groups)
- Event ID 636 – Local group member added (involving administrator groups)

Similarly, any events indicating the starting or stopping of security relevant services, such as the event logging service, should also be noted.

Whilst a number of these events will commonly occur within a network, it is important to determine whether they are due to authorised activities, user error or malicious attack. For example, the shutdown/restart of a server is a common event, but this is also a typical requirement following the installation of a software package so it is important to determine whether the event occurs outside of normal operating patterns.

**Correlation** – There are several aspects to this topic, such as the ability to track the activities of a single entity through a system through to the ability to spot reoccurrence of either an entity within a log file or a pattern of activities within a log file.

Assuming that all the logs are aggregated into a common structure, then the main requirement is for all major components within a network to be synchronised to a common time source to allow temporal correlation across the various log sources. Otherwise the requirement to know the time differences between the various components and to adjust accordingly will incur processing overhead. Queries can then be developed that can select time-slices around significant events so that the impact on different components can be detected. This will also allow examination of time periods where there are significant levels of activity that are out of pattern.

Another simple correlation is to group activity by IP address (which may be either the source or destination address). This will quickly aggregate all records for a particular entity, providing the 'source IP' is a reliable identifier through the system (with the caveat described earlier where proxies are used). This correlation may only hold for a short time period however, many ISPs now allocate their client IP addresses dynamically, rather than provide a static address, on a short "lease", which may change during a long session. An alternative method may be to group the data by the IP Root or by the domain owning the IP address in question.

Perhaps the most powerful correlation is the ability to detect when an entity revisits a system, perhaps many weeks after their previous attempt. This type of query can aid the detection of entities (with the same qualification regarding dynamically assigned IP addresses as above) showing more than a passing interest in a network, which may show changes in techniques with different vulnerabilities being attempted on different occasions. Again, this can be improved if systems are configured to log both "permitted" as well as "denied" activities.

The logging of both permitted and denied activities does however have an impact on data storage. Care should be taken as to what levels of "history" are recorded and maintained, for example just a simple summary or full log records.

### **Log Gaps**

A skilful attacker may try and cover their tracks, which could range from the unsubtle option of disabling logging on the attacked box to selectively wiping any relevant log entries. Conversely, the attacker could flood the target with activity generating a large number of log entries in an attempt to hide the one or two important activities.

Over time, it may be possible to build up a typical profile of activity on a system that can be used to detect levels of activity that do not conform to the norm. An alternative is to compare the activity within a time period to the average activity calculated over several such time periods. Any periods which greatly exceed the average, or which show a much smaller degree of activity than the average then become suspect.

At this point, the only option for determining whether the discrepancy is real or a "false positive" is to examine the log entries for a given period before and after the suspect time slice for signs of malicious activity.

### **Trends**

The Internet is a dynamic environment, with new attacks being generated constantly in response to newly publicised vulnerabilities. Whilst attackers will generally favour a subset of exploits for gaining initial access to a system, usually on standard ports such as FTP, the backdoors that are often left in place can be associated with any port.

Over time, it is possible to build up a profile of activity against a system based on the amount of activity against a particular port in

a given time period. A good example of this is the graphical representations of activity against an individual port (across multiple systems) at [www.incidents.org](http://www.incidents.org). These graphs visibly display changing trends, showing increasing or decreasing interest in particular ports by the hacker community at large.

The drawback with using audit log data for trend analysis is that a reasonable data set, which must be continuous, is required. Perhaps the best data set to use for this type of analysis is firewall logs, though logging all traffic ("permit" as well as "deny", "reject" and "drop") will provide information on probes and successful exploits.

With over 65,000 possible ports on a computer, for both TCP and UDP protocols, it is difficult to review all of these graphically to determine changing trends. It is necessary to select a subset of ports that are of interest (which could be selected to match those services that are allowed across the network boundary) or to develop a metric that can be applied to the data set that alerts the System Administrator (or other person responsible for reviewing the log data) to those ports which are demonstrating a significant change in the levels of activity.

The potentially random nature of the data sets makes the use of standard mathematical functions, such as significance tests, problematical. In light of this it is not obvious that a change in activity levels over a period of only one day can be considered absolutely significant.

It is here that having a baseline of data becomes useful as it allows the calculation of an average level of activity over a period significantly greater than that which the "delta" is being calculated over. Searching for activity where the amount of activity is greater than a multiple of the average (or less than a particular fraction of the average) over 2 or more periods can, for example, be used to give a measure of significance.

A change in the trend of activity on a particular port may be due to:

- the publication of a new vulnerability and associated exploit if the trend change is positive;
- If the trend change is positive but no new exploit has been publicised, it may be due to activity using a non-public domain vulnerability. In this instance, it may be worth checking the number of sources responsible for the activity and checking these against the data held at [www.incidents.org](http://www.incidents.org).

- If trend is positive on port with known associations with Trojan or DoS agents, then it is possible that the attacker is attempting to build up an array of “zombies”;
- If the trend is negative, this may be due to the hacker community losing interest in a particular exploit.

### **Log Analysis – Follow Ups**

Activities that are highlighted by log analysis as potentially malicious should be investigated, especially if the data indicates that the attack may have been successful.

The means or methods of investigation will in general be a matter of local policy and will reflect the perceived seriousness of the incident involved. Options for this activity may include:

- Seizure and forensic examination of the hard disks from the affected machine
- Monitoring of network traffic on the system for unusual activity
- Examination of server logs for signs of unauthorised activity
- Making contact with the appropriate CERT organisations or ISPs to report malicious activity.

Once the activity has been investigated, there may be a need for a response, which again will depend on the seriousness of the activity and local policy. Options may include

- Making contact with the appropriate CERT organisations or ISPs to report malicious activity
- Modifying firewall or router configurations to block access from particular sources
- Modifying logging/alerting configuration or procedures to provide greater levels of information on future events.

### **Conclusions**

CESG has analysed audit log data from a variety of governmental and critical national infrastructure organisations for more than a year, initially by hand and later using an automated analysis tool that was developed.

The findings of the research activities to date can be summarised as:

- It is possible to detect evidence of attempted hacking activity through log analysis. The techniques employed are applicable

at a departmental network level and can be used to correlate activity across multiple systems. Numerous examples of probing of firewalls, scanning of networks and attempted exploits against public facing web servers were detected.

- Activity observed in the logs that were analysed generally mirrored the patterns of activity observed on the wider Internet, with the probing of common ports, network sweeps, web exploits and worm activities reflected in the logs.
- Log analysis is an ongoing process with greatest benefit being gained from a continuous data set that enables the establishment of a baseline activity level, allowing sensible trend analysis to be performed.

The research has also brought up several issues, a number of which have been mentioned previously in this document. Two of these are worth re-stating, namely:

- Without a form of time synchronisation, it is difficult to track an attacker's activities through a network, primarily due to the widespread use of proxies that tend to anonymise the identity of the source.
- A definitive picture of the intentions of an attacker is not possible through log analysis alone, often the data shows an attempted connection to a system, but does not give an indication of what data was in the connection attempt. Complementary techniques, such as intrusion detection can be used to extend the available information to form a more complete picture. Log analysis can, however, aid the development of the assessment of the level of threat to a network, from the number and range of probing attempts observed within the logs.

Work is continuing at CESG to extend the range of activities that the automated tool can detect. Further research is required to determine whether it is possible to detect the kind of activities that may be brought to bear by a more sophisticated attacker.

## Annex A Common Logs Format

A.1 The table below shows the generic fields that should be present in the Common Logs table. IP root is an internal technical term and its use does not imply that the IP address is a Class C address.

<b>Field Name</b>	<b>Description</b>
Datestamp	Date/time on which event occurred
Source_IP	Source IP address
Src IP Root	First three sections of IP address
Dest_IP	Destination IP address
Dest IP root	First three sections of IP address
Source_port	Originating port on source address
Dest_port	Destination port on target address
Transport_protocol	Define what transport mechanism used eg TCP
Dest_service	If used separately to port, this will be things like HTTP, SMTP, etc
Servename	Name of server where activity occurred
Server_IP	IP address of server where activity occurred
Time_taken	Time taken for transaction
Bytes_recvd	No. of bytes received by server from client
Bytes_sent	No. of bytes sent by server to client
http_method	Method used as part of HTTP request
Url	URL requested by client
HTTP_code	HTTP status code returned
Action	Firewall rule response
ICMP_type	As named
ICMP_code	As named
Username	Any authenticated username recorded in the logs
Useragent	Any user agent (such as web browser) details recorded in the log
Log_source	System details where logs came from
Log_type	Details of application producing log
Src Interface	Interface name on which source is attached
Dest Interface	Interface name on which destination is attached

A.2 This table maps data from the log files of network components to the Common Logs format shown above. (Sample components from the networks analysed by CESG).

<b>Generic Field</b>	<b>FW-1</b>	<b>ISA Firewall</b>	<b>ISA Proxy</b>	<b>Web</b>	<b>IIS</b>
Date	Date	Date	Date		Date
Time	Time	Time	Time		Time
Source_IP	Src	c-ip	c-ip		c-ip
Dest_IP	Dst	r-ip	r-ip		s-ip
Source_port	S_port		-		-
Dest_port	Service	r-port	r-port		s-port
Transport_protocol	Proto	Cs-transport	Not mapped but will be TCP		This is not mapped but will be TCP
Dest_service	Service	r-port/cs-protocol	Cs-protocol		s-port
Servername	-	s-computername	s-computername		s-computername
Server_IP	Orig	-			s-ip
Time_taken	-	Time-taken	Time-taken		Time-taken
Bytes_recvd	-	Cs-bytes	Cs-bytes		Cs-bytes
Bytes_sent	-	Sc-bytes	Sc-bytes		Sc-bytes
http_method	-	-	s-operation		Cs-method
Url	-	-	Cs-uri		Cs-uri
HTTP_code	-	-	Sc-status		Sc-status
Action	Action	-	-		-
ICMP_type	Icmp-type	-	-		-
ICMP_code	Icmp-code	-	-		-
Username	-	Cs-username	Cs-username		c-username
Useragent	-	c-agent	c-agent		Cs(useragent)

## Annex B References

B.1 This annex provides a list of useful resources to aid the development of log analysis. The list is not exhaustive - the reader will undoubtedly note that some favourite sources of information are not present - but it aims to give a selection that covers the basics.

Hacking Exposed Third Edition, S. McClure, J. Scambray & G. Kurtz, published by McGraw-Hill, ISBN 0-07-219381-6

[www.incidents.org](http://www.incidents.org) - useful site for learning about current trends in Internet activity. Also many useful articles and analyses.

[www.counterpane.com/log-analysis.html](http://www.counterpane.com/log-analysis.html) - Long list of Internet resources on log analysis

ICMP stands for trouble, [www.networkmagazine.com/article/NMG20000829S0003](http://www.networkmagazine.com/article/NMG20000829S0003) - description of means in which ICMP traffic can be used to scan a network

[www.simovits.com](http://www.simovits.com) - Good information on Trojans

[www.somarsoft.com](http://www.somarsoft.com) - Useful tools, such as dumpevt that can be used to aid log analysis

Windows NT Event Logging, James D. Murray, Published by O'Reilly Press, ISBN 1-56592-514-9

[www.robertgraham.com](http://www.robertgraham.com)

[www.washington.edu/People/dad/](http://www.washington.edu/People/dad/) Homepage of David Dittrich, covering a number of useful areas.

Network Intrusion Detection 3<sup>rd</sup> Edition, Stephen Northcutt & Judy Novak, ISBN 0735712654

The Hacker's Challenge series of publications by Mike Schiffman *et al*