Code tables. A code table is a data table that can be stored in the form of a picture or printout and contains XxY number of cells. Generally anything from 10x10 to 16x16 is used. A table is generated by the server and a copy is sent to the user for printing (or a copy is sent via registered post). When carrying out the authorisation process, the user is prompted to enter the contents of certain cells selected at random by the server. This process can either take the place of a password or can be in addition to it.

The interception of a specific combination will make it possible to learn the value of two or three cells from the possible 100-plus and that during subsequent authorisation sessions, other cells will be requested. However, the keylogger is unable to fix the requested cells’ positions – and so the criminals will not know which cells were involved without resorting to taking a screenshot. The benefit of this method lies in its simplicity. Similar technology is used by the Russian payment transfer system Yandex Money.

One-time passwords. This method is similar to the previous one, but in this case the user receives a table of one-time passwords. Once used, each password is then crossed out. The method can be used in reverse, with the user removing the opaque protective coating from a sequential password list printed on a card. In this method the danger of password interception with the aid of a keylogger is completely avoided. However, the quantity of passwords is limited and it will be necessary to obtain a new list at some point.

Password generator. An electronic token is used for the generation of passwords, the generated passwords are not repeated and they are produced according to a specific algorithm. It is considered that the algorithm and the secret key within cannot be deciphered based on just a few intercepted passwords.

Two-factor authorisation, for example, with the use of an eToken. In this case the theft of passwords is not dangerous, since it is useless without the accompanying token. The reverse also applies – the eToken is useless without the password.

"Two or three years ago, hardware keyloggers were some sort of techno-marvel – now they are produced commercially”

Conclusion

We have examined the basics of software and hardware keyboard spies. In summing up, it is worth mentioning that the situation in this area of technology is changing radically day-by-day. Two or three years ago, hardware keyloggers were some sort of techno-marvel – now they are produced commercially with many different models available, beginning with the basic 32KB types, right up to devices with several GB of memory and wireless control. It should be assumed that the development of the hardware keylogger market will continue and possibly in the very near future we will see an entirely new generation of software-hardware spies that will not require access to the victim’s computer at all.

About the author

Oleg Zaitsev is chief technology expert at Kaspersky Lab. He joined in 2007 as a developer in the Complex Threat Analysis Group. He is now responsible for carrying out research into new detection and disinfection technologies, investigating and disinfecting remote systems and analysing the behaviour of malware.

References

1. For an example of a ‘catalogue’ site, see: <http://www.keylogger.org>.

Investigating digital fingerprints: advanced log analysis

Eric Knight, LogRhythm

Perhaps your organisation has recently suffered a data breach, which would not be uncommon as last year set records for information theft. For example, in 2009 the Identity Theft Resource Center reported 498 breaches that exposed over 222 million protected records. The unreported numbers traditionally have been three or more times higher. More still go undetected or ignored.

When a breach takes place, the first and most obvious step in finding out where, when and how it happened would be to check the logs, as they are the computer’s version of the official record of events. When you open the log files, whether on Windows via the Event Viewer or from *NIX’s message logs, you might stop immediately to consider one pressing question: what am I looking for?

There is a direct connection between the need for log analysis and log management – the two concepts go hand in hand. The core problem is that there are many types of log sources, high volumes...
of logs, and more logs are generated by updated systems each year – as explained in Donal Casey’s article ‘Turning log files into a security asset’.2

“The process of finding the sequence of logs that describes a breach, incident or system failure can be challenging”

Log management traditionally consists of collection, searching and storage. Analysis of log entries is optional in many products, but it is the area where the most value is generated for an organisation’s IT team. The process of finding the sequence of logs that describes a breach, incident or system failure can be challenging.

Data contained in logs
Here, we’ll look at which elements may be extracted from a log, the enrichment of log information to apply context that adds value to log data, and a method of classification of logs that can be used to reduce analysis time. The log holds most of the important details that need to be extracted before additional determinations about the log can be made.

In the example log in Figure 1, there are several elements present, such as origin and destination IP address, port numbers, a protocol designation, an identification number describing an attack, a priority rating and a severity rating. At a glance, this log may be quickly understood by the reader, but if over 10,000 of these are generated daily, analysing this log in comparison to the others – without mistakes or bias – is nearly impossible without automation.

One of the fields worth mentioning is the ID number of the event, often referred to as the Vendor Message Identification (VMID). The VMID is a unique value that corresponds to an event, and many platforms support some form of VMID. These are handy for processing but don’t really aid for human readability. Systems that use VMIDs may either provide translations as part of the log message or expect that some form of event management system will be used to interpret the meaning.

The extraction of this information can be done by a number of methods, including pattern matching (generally employing regular expressions), tokenising and hard coding. Each has its advantages and disadvantages, but always of primary consideration are accuracy, compatibility and performance.

One of the interesting limitations of most log and event management systems is the number of fields they parse and retain. The more fields they allow, the more space each log requires, and this reduces performance for all other actions, such as investigations and reporting. Each vendor will have its own mixture of important fields.

XML is an emerging standard for logs, which is an attempt to break free of fixed traditional ASCII text strings perpetuated by the *NIX syslog system and Simple Network Transfer Protocol (SNTP). XML allows for a schema with many elements, conditions and attributes represented. Users of Windows Vista, 2008 and 7 might have noticed that logs are now stored in XML rather than text strings (Figure 2).

Once the data from the log is extracted, it can be used to apply additional meaning to the log. The information collection is still raw and, although useful, it does not contain everything that we can determine about the log and the event that it describes.

Enrichment
There are many pieces of information that can be determined from the contents
or origin of a log, such as how critical the affected systems are, where the network connections originate from, if the event is occurring from the Internet or intranet, if the event is about a blacklisted or whitelisted system, and more. The context that log enrichment provides is invaluable to the investigator because, although applying this context takes time, it often provides rapid clarification.

Consider a non-routable address space such as 192.168.x.x/16 or 10.x.x.x/8 – this is likely to refer to an event taking place on a Local Area Network (LAN), protected by a firewall or edge device performing Network Address Translation (NAT). Administrative activity or authentication on an internal server may be considered suspicious and alarm-worthy where the originating address is an externally routable IP address, indicating that a connection/attack has been made from the external public Internet.

Domain Name Service (DNS) lookups are capable of providing some additional information about the origin of a system. For example, if a host resolves to ‘some.system-in.ru’, the ‘.ru’ immediately indicates that the origin resides in Russia. If a large amount of information is sent to a foreign country that does not have a company base of operations, this action may signal that a breach has occurred and must be investigated. However, a gigabyte transfer may appear typical for local LAN activity.

Location resolution is helpful for cases when a domain name isn’t assigned to an IP address and when trying to identify if a user is local or foreign. Routable Internet space sometimes (but not always) can be traced back to a geographic location by latitude and longitude. Many companies produce databases with this information – an example is MaxMind which provides both commercial and open source databases that can assist in resolving an IP address to a location.\(^3\) This step is best performed while processing a log rather than afterwards.

The process of enrichment can extend to other databases and resources as well. In general, the better defined a network, its components and its users, the better the enrichment process.

**Classification**

The *NIX syslog system provided the original classification scheme for logs, dividing them into two dimensions: by purpose (kernel, email, etc); and by severity (panic, critical, error, warning, information, debug). These allow for the routing of events to their proper locations, such as different log files or being sent to log servers elsewhere in the network.

For Microsoft Windows, logs have been classified into typed categories such as Startup and Shutdown, Authentication Success, Authentication Failure, Access Allow, Access Deny, Audit Success, Audit Failure, Audit Other, Critical, Error, Warning and Information. These provide additional robustness over the *NIX logs and help define the components needed for an auditor to verify against regulatory requirements.

Neither of these addresses a complete range of issues, such as firewall, router and switch events for standard packet-level traffic, intrusion detection systems, anti-malware systems, proxy systems, vulnerability detection tools, user abuse detection, and so on. The merging of these logs classifications forms a taxonomy of logs, such as the one in Figure 3.

In addition, the use of one or more additional layers can assign greater context to a log. For example, take Authentication Success. A log that fits into this classification could include Administrator Login, User Login, Monitoring Account Login, Service Account Login, Anonymous Login and so on. If these seem like they might be common to many different types of network equipment, they are. These subsections of classification are called ‘common events’.

Privileged user monitoring can be performed after assigning a log the proper common event, or alternatively comparing a classification to a known administrator account that was extracted from the log (such as ‘administrator’, ‘root’, ‘sa’, etc) or by group (‘domain admins’, ‘root’, ‘backup admins’, etc).

Classification into common events also provides a manner of assigning risk or severity to a log entry. For example, a security event can be assigned a threat value. The threat value might not appear in the initial log, but if a particular setting is changed the threat value can be assigned to highlight it as an immediate concern.

Classification allows for the assignment and grouping of logs, and offers an opportunity to redefine a log that normally would appear uninteresting, such as in the case of privileged user monitoring. Once the log has been properly identified, it can then be analysed using advanced techniques.
Events and correlation

After a log has been properly parsed, had context applied and classified, it may be escalated to an ‘event’. An event is an important log; or alternatively, it can be considered as a ‘condition’ that has been created through the cross-correlation of more than one log. Events are used to highlight issues that may need additional investigation, such as the changing of the system configuration, policies, user accounts, privileges, etc. Security events such as malware, or operational events such as a system crash, are also worthy of promotion to event status.

The term ‘correlation’ means different approaches to different vendors. The basic definition is establishing additional meaning from two or more conditions. If an NMAP port scan is detected coming from the VPN concentrator’s IP address pool, then it could be assumed that an attacker might have penetrated the perimeter, and a Perimeter Breach event of high priority can be created as a result.

Correlation can come in several forms, ranging from trending, behavioural analysis, programmed expertise, multiple occurrences within a time frame, from two or more events are seen in close proximity, or even advanced datamining. Each vendor will have its own recipe for correlation and this may be geared to the needs of the industries for which the product was first developed.

Conclusion

Logs contain a considerable amount of valuable information, but the information may be slow to extract and comprehend by manual review. Additional information can be determined by applying context, such as directionality of traffic flow, assigning proper network grouping and assigning geographic location. The logs can then be further classified to assign meaning and cross-referenced to similar events from other types of devices to assist in reporting and investigation. When the log is fully processed, it should be easy to identify as being a critical event or normal occurrence, helping to reduce the workload on the operators assigned to monitor network operations and security.

About the author

Eric Knight is senior knowledge engineer at LogRhythm where he is responsible for building tailored compliance intelligence packages for regulations such as NERC CIP, SOX and PCI that are shipped as part of LogRhythm’s log management and SIEM 2.0 solutions. He is responsible for the strategic integration of network & security products to enable LogRhythm to collect, analyse, alert and report on log data. Knight has over 15 years’ experience in the field of network security with an emphasis on vulnerability management and enterprise security architectures.

References