



## Allens Temporal Algebra Used Between Login –Logout Event Generation Probability

S.Murugan\*

MCA.,Mphil.,CGT.,MISTE.,(MS).,  
ACTS, CDAC Knowledge Park, No 1 Old  
Madras Road, Bangalore, India  
[s.murugan@teacher.com](mailto:s.murugan@teacher.com)

Dr.K.Kuppusamy

Director i/c  
Computer Centre,Alagappa  
University,Karaikudi, india  
[kkdikamy@yahoo.com](mailto:kkdikamy@yahoo.com)

**Abstract:** In any system like small or large that normal processes consists of login and logout as a main process .In between that two main process lot of sub process like system wise and user wise will be generated according to user to user or os to os .Let we find the probability of usage with unknown attack solutions. As usual system process are called Internal and induce or connected processes are external.By using allen algebra we can analysis the temporal event. Auditing provides a way for an administrator to detect an attack that has already occurred or is in progress. In addition, auditing can help a developer to debug security-related problems. For example, if an error in the configuration of the authorization or checking policy accidentally denies access to an authorized user, a developer can quickly discover and isolate the cause of this error by examining the event log. Good system and network security starts with a good understanding of an organization's operating environment. Organizations that have a good understanding of their operating environment and that environment's limitations and vulnerabilities – should be able to secure their system relatively easily. Maintaining a high level of system security, however, is an on-going process that requires continued vigilance and solid organizational policies and procedures. Pro-active system administrators not only keep their systems patched, but also continuously monitor system and network logs and system resource usage reports for interesting events.

**Keywords:**Allens Temporal Algebra, Probablity, Login, Logout

### I. INTRODUCTION

All systems can log interesting system events, although sometimes the event types and depth of logging information can vary from system to system.

On LINUX systems, for example, there are several facilities that could generate messages using the syslog facility. The information collected by syslog is a valuable resource in determining the health of the system, and when reviewed regularly can provide an advance warning for some types of attacks.

On Windows systems, the event log can record various types of application and security events that can be useful when analyzing system errors or when tracing possible intrusions or security compromises.

Syslog-type software is also available for Windows systems, thus allowing for central reporting of interesting system events across all platforms.

To improve security on LINUX systems, the syslog UDP port (514) should be blocked at the firewall in order to reduce the likelihood of a buffer overflow attack or other vulnerability being remotely exploited, and remote logging should be disabled unless the host acts as a central log server.

Traditionally, resource accounting and chargeback products have been used to track shared resources on central servers, and for utilization reporting on server consolidation projects. Development and production environments that concurrently work on multiple projects using common resources/ computers have also been traditional users of chargeback products.

Analysis of system usage data can be very useful in improving system performance by helping detect performance bottlenecks and in the detection of intrusions,

since anomalies in chargeback data can sometimes reveal inefficient applications and/or misuse of computer resources. An IT department that finds unusual usage patterns or excessively high usage during review of chargeback records should consult with the user organization to determine whether an inefficient application can be improved. This would reduce its resource utilization in order to save money for the user organization, and spare the IT organization from planning for a system upgrade to meet bogus system demands.

Likewise, a spike in user activity, or worse, a spike in activity of previously dormant accounts and projects may indicate that security has been compromised or that the systems are misused.

On LINUX systems, the standard system accounting files can provide a wealth of system usage information when analyzed on a regular basis, and can be used for both chargeback and capacity planning purposes. 'wtmp' or 'wtmptx' and 'pacct' are the standard LINUX system accounting files, containing login information and resource usage information by processes respectively.

The 'last' and 'acctcom' programs can be used to view detailed usage data while the 'acctcon' and 'acctprc'/'acctcms' programs can be used to view summarized data.

Windows systems can record login, logout, application start and application stop events, although resource usage information is not recorded in the event log. This type of auditing, however, is turned off by default – so these changes should be applied sitewide through the group or enterprise audit policy by turning on auditing of login and logoff events and process tracking events.

The 'dumpe' utility, available on the Resource Kit, can be used to report on the various event logs or to format

event log entries for export to spreadsheets (or other applications) for further review.

Commercial products can simplify the presentation of system usage data for chargeback. UNISOL JobAcct from UniSolutions Associates, for example, can generate system usage reports by user, group, project or cost-center, for one or more computers on the network, collecting the same type of data on both Windows and LINUX systems. JobAcct can collect application resource usage information on Windows systems without relying on the limited data available through the event log, thus providing a consistent report across various operating system types.

There are several free scripts and tools available on the web for LINUX systems that can be used by system administrators to summarize and monitor the syslog and login accounting files. One syslog summary tool is newlogcheck, which enhances security by reducing the amount of log entries administrators have to examine, and categorizing the log entries. Sentryd is a Perl script that monitors the syslog and wtmp files for unusual events and bad login attempts, and notifies users (by broadcast) of selected events.

There are also tools for more specialized log analysis that can parse log entries in real time and correlate system and network events, such as SEC (Simple Event Correlation tool), Swatch, Logsurfer, and Logwatch. Several vendors also provide managed security (e.g. analyzing firewall data recorded in syslog files), while several vendors provide managed security services typically including firewall log analysis, intrusion detection, virus protection, gateway services, and vulnerability assessment and policy compliance services.

Organizations considering outsourcing certain security services should consult the paper Outsourcing Managed Security Services from CERT in order to better understand the benefits and risks involved with hiring a Managed Security Service Provider (MSSP) and getting maximum value for their security budget without compromising system security – or giving up too much control of their IT environment.

Finally, checking for login errors should also be performed regularly, perhaps by incorporating the process together with an “Analysis of system usage data can be very useful in improving system performance by helping detect performance bottlenecks and in the detection of intrusions” automated syslog analysis procedure. The location of the logged login errors differs from machine to machine, but all systems log some type of login errors. When auditing is turned on for logon success and failures on Windows, the Security event log will contain these events.

Pulling together all of the logs and analyzing them regularly (preferably in an automated process) is the first step in establishing an incident response policy. An automated log analysis process that notifies administrative personnel promptly can be an invaluable tool to providing a quick response to system attacks or other significant security events, that can stop a security breach early enough before irreparable damage can take place.

## II. DIFFERENCE BETWEEN ATTACK AND INTRUSION

So far, the term “intrusion” has been used without providing any precise definition of its meaning. The

problem is that also in the security community there is no common agreement on a technical definition of the term intrusion detection, defined an intrusion as a successful unauthorized attempt to access information, manipulate information, or render a system unreliable or unusable. Similar definitions have been proposed which defines an intrusion as an action that compromises a resource’s integrity, confidentiality or availability, where an intrusion is simply any unauthorized use, misuse, or abuse of computer systems.

“A security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so.” and a security incident is described as a “security-relevant system event in which the system’s security policy is disobeyed or otherwise breached”.

The difference is clear: any intrusion is a consequence of an attack, but not all attacks lead to an intrusion. An attack may fail for many reasons: because the target system has been patched, because the installed version is not vulnerable, or because a network device (e.g., a firewall or a reverse proxy) blocks or normalizes the malicious trace before it can reach the target service. This is not just a terminology problem. From an intrusion detection point of view, the distinction between attacks and intrusions is very important.

As the name says, the purpose of intrusion detection should be to detect intrusions. Unfortunately, this task can be very hard since the fact to be an intrusion is not just a property of the network stream, but it also depends on the effect that the stream produces on the target system.

The result is that most of the network intrusion detection systems do not even try to distinguish between attacks and intrusions and just let the user decide which was the result of the malicious events by carefully analyze the alerts reported by the analysis.

Intrusion detection is *a process of intelligently monitoring the events occurring in a computing resource*, the purpose of which is to analyze these events for successful and unsuccessful attempts to abuse the computing resource.

An organization’s operating environment. Organizations that have a good understanding of their operating environment and that environment’s limitations and vulnerabilities – should be able to secure their system relatively easily. Maintaining a high level of system security, however, is an on-going process that requires continued vigilance and solid organizational policies and procedures. Pro-active system administrators not only keep their systems patched, but also continuously monitor system and network logs and system resource usage reports for interesting events. All systems can log interesting system events, although sometimes the event types and depth of logging information can vary from system to system.

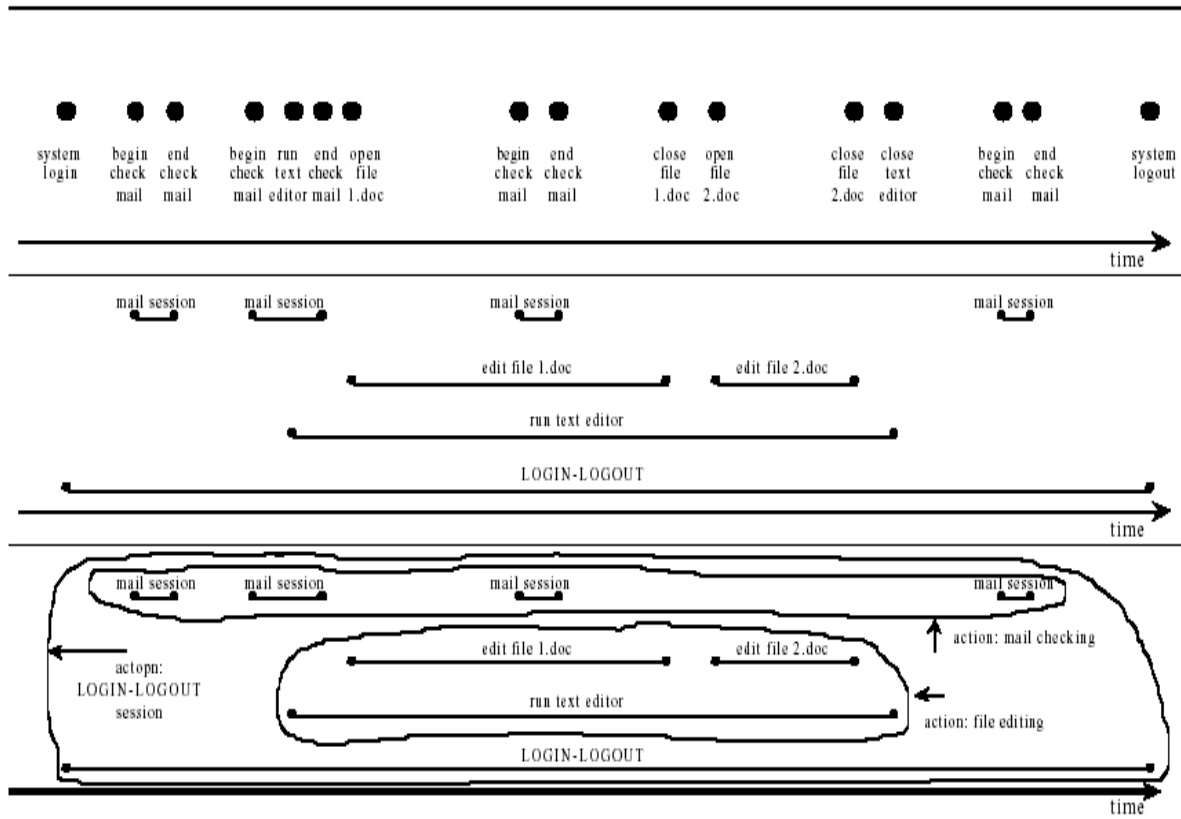


Figure: 1

### III. AUDIT LEVEL AND BEHAVIOR

Two levels of security audits exist:

- [a] Service authorization level, in which a caller is authorized.
- [b] Message level, in which WCF checks for message validity and authenticates the caller.

You can check both audit levels for success or failure, which is known as the *audit behavior*.

#### A. Audit Log Location

Once you determine an audit level and behavior, you (or an administrator) can specify a location for the audit log. The three choices include: Default, Application, and Security. When you specify Default, the actual log depends on which system you are using and whether the system supports writing to the security log. For more information, see the "Operating System" section later in this topic.

To write to the Security log requires the **Se Audit Privilege**. By default, only Local System and Network Service accounts have this privilege. To manage the

Security log functions **read** and **delete** requires the **Se Security Privilege**. By default, only administrators have this privilege.

In contrast, authenticated users can read and write to the Application log. Windows XP writes audit events to the Application log by default. The log can also contain personal information that is visible to all authenticated users.

#### B. Suppressing Audit Failures

Another option during auditing is whether to suppress any audit failure. By default, an audit failure does not affect an application. If required, however, you can set the option to **false**, which causes an exception to be thrown.

#### C. Programming Auditing

You can specify auditing behavior either programmatically or through configuration.

#### D. Auditing Classes

The following table describes the classes and properties used to program auditing behavior.

Table I

Class	Description
Service Security Audit Behavior	Enables setting options for auditing as a service behavior.
Audit Log Location	Enumeration to specify which log to write to. The possible values are Default, Application, and Security. When you select Default, the operating system determines the actual log location. See the "Application or Security Event Log Choice" section later in this topic.

Message Authentication Audit Level	Specifies which types of message authentication events are audited at the message level. The choices are <b>None</b> , <b>Failure</b> , <b>Success</b> , and <b>Success Or Failure</b> .
Service Authorization Audit Level	Specifies which types of service authorization events are audited at the service level. The choices are <b>None</b> , <b>Failure</b> , <b>Success</b> , and <b>Success Or Failure</b> .
Suppress Audit Failure	Specifies what happens to the client request when auditing fails. For example, when the service attempts to write to the security log, but does not have <b>Se Audit Privilege</b> . The default value of <b>true</b> indicates that failures are ignored, and the client request is processed normally.

For an example of setting up an application to log audit events, see How to: Audit Windows Communication Foundation Security Events.

**E. Configuration**

You can also use configuration to specify auditing behavior by adding a serviceSecurityAudit element under the Behaviors element. You must add the element under a Behavior element as shown in the following code.

**Copy Code**

```
<configuration>
<system.serviceModel>
<behaviors>
<behavior>
<!-- auditLogLocation="Application" or "Security" -->
<serviceSecurityAudit
auditLogLocation="Application"
suppressAuditFailure="true"
serviceAuthorizationAuditLevel="Failure"
messageAuthenticationAuditLevel="SuccessOrFailure" />
</behavior>
</behaviors>
</system.serviceModel>
</configuration>
```

If auditing is enabled and an **auditLogLocation** is not specified, the default log name is "Security" log for the platform supporting writing to the Security log; otherwise, it is "Application" log. Only the Windows Server 2003 and Windows Vista operating systems support writing to the

Security log. For more information, see the "Operating System" section later in this topic.

**F. Security Considerations**

If a malicious user knows that auditing is enabled, that attacker can send invalid messages that cause audit entries to be written. If the audit log is filled in this manner, the auditing system fails. To mitigate this, set the **SuppressAuditFailure** property to **true** and use the properties of the Event Viewer to control the auditing behavior. For more information, see the Microsoft Support article on viewing and managing event logs by using the Event Viewer in Windows XP available at <http://go.microsoft.com/fwlink/?LinkId=89150>.

Audit events that are written to the Application Log on Windows XP are visible to any authenticated user.

**G. Choosing Between Application and Security Event Logs**

The following tables provide information to help you choose whether to log into the Application or the Security event log.

**H. Operating System**

Table II

System	Application log	Security log
Windows XP SP2 or later	Supported	Not supported
Windows Server 2003 SP1 and Windows Vista	Supported	Thread context must possess Se Audit Privilege

**I. Other Factors**

In addition to the operating system, the following table describes other settings that

control the enablement of logging.

Table III


Factor	Application log	Security log
Audit policy management	Not applicable.	Along with configuration, the Security log is also controlled by the local security authority (LSA) policy. The "Audit object access" category must also be enabled.
Default user experience	All authenticated users can write to the Application log, so no additional permission step is needed for application processes.	The application process (context) must have <b>Se Audit Privilege</b> .

The calculus defines possible relations between time intervals and provides a composition table that can be used as a basis for reasoning about temporal descriptions of events.

**J. Relations**

The following 13 base relations capture the possible relations between two intervals.

Table IV

Relation	Illustration	Interpretation
$X < Y$		X takes place before Y

$XmY$ $YmiX$		X meets Y (i stands for inverse)
$XoY$ $YoiX$		X overlaps with Y
$XsY$ $YsiX$		X starts Y
$XdY$ $YdiX$		X during Y
$XfY$ $YfiX$		X finishes Y
$X=Y$		X is equal to Y

X- Login

Y-Logout

Using this calculus, given facts can be formalized and then used for automatic reasoning. Relations between intervals are formalized as sets of base relations.

**K. Composition of relations between intervals**

For reasoning about the relations between temporal intervals, Allen's Interval Algebra provides a composition table. Given the relation between X and Y and the relation between Y and Z, the composition table allows for concluding about the relation between X and Z. Together with a converse operation, this turns Allen's Interval Algebra into a relation algebra.

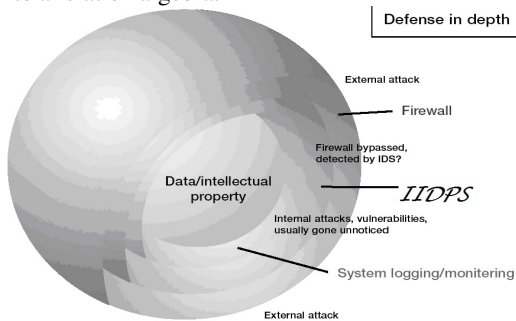


Figure. 2

When monitoring all layer with a proposed tool will provide solution but the stored data might need more storage that is uses datamining technique to reduce data size by encryption /decryption after that it will filter dataset passes through iidps engine that provide feasible solution ,new attack /unknown attack will be stored in database like dictionary / encyclopedia.

When event log abnormal condition alarm signature comes iidps take immediate action to stop the access .user interaction to reduce damages or avoid file corruption.

**IV. CONCLUSION**

Maintaining secure systems and networks is an ongoing process weighed down with difficulties, which are further exacerbated in heterogeneous, multi-Operating System environments by the multitude of differences between the various operating systems and the procedures that must be followed in order to maintain a high level of system and network security.

To keep systems and networks secure, an organization must adapt the defense-indepth mindset and work on as many security layers as possible, from the external facing firewalls or IIDPS to the internal development servers. Firewalls are an essential first line of defense and IIDPS systems are successful in tracking and preventing some types of Unknown internal threats or attacks that have penetrated the firewall.

However, organizations that consistently analyze, report, and use the collected system utilization data and logged interesting events can respond quicker to security threats and thus avoid or minimize the damage of these threats.

**V. REFERENCE**

- [1] Anomaly Intrusion Detection Systems: Handling Temporal Relations between Events, Alexandr Seleznyov, Seppo Puuronen.
- [2] Algebra for Capability Based Attack Correlation Navneet Kumar Pandey, S. K. Gupta, and Shaveta Leekha.
- [3] Fuzzifying Allen’s Temporal Interval Relations Steven Schockaert, Martine De Cock, and Etienne E. Kerre.