Interested in learning
more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

# Host Based Intrusion Detection: An Overview of Tripwire and Intruder Alert

Intrusion detection has been defined by Peter Loshin of Computerworld magazine as "the art and science of sensing when a system or network is being used inappropriately or without authorization".

Copyright SANS Institute
Author Retains Full Rights

AD

**Host Based Intrusion Detection: An Overview of Tripwire and Intruder Alert**

Allison Hrivnak

January 29, 2002

Intrusion detection has been defined by Peter Loshin of Computerworld magazine as "the art and science of sensing when a system or network is being used inappropriately or without authorization". [1] Intrusion detection systems monitor system and network resources to detect unusual activity or changes. There are two types of intrusion detection systems: host and network based. A network based IDS is placed on the network near the system or systems being monitored and analyzes network traffic for attack patterns and suspicious behavior. A host based IDS resides on the system being monitored and tracks changes made to important files and directories. While both are part of a good defense-in-depth strategy to prevent attackers from being able to enter networks and alter or compromise critical information, only a host based intrusion detection system with a well written policy will provide a strong foundation to good system security.

One of the main benefits of a host-based IDS is that it does not have to look for patterns, only changes within a specified set of rules. Most intrusion detection systems include default policies for specific operating systems. An administrator can use this information upon initial installation to learn the behaviors of files and directories under normal system activity and enable he or she to fine-tune the policy through trial and error. Choosing the right software for an intrusion detection system can be a challenging task that often requires extensive research. While there are many different products available, Tripwire from Tripwire Inc. and Symantec's Intruder Alert offer two possible solutions for a host-based intrusion detection system.

Dr. Eugene Spafford and Gene Kim developed Tripwire in 1992 at Purdue University. It quickly became one of the most used freeware tools among security professionals. Tripwire Inc. was formed in 1997 and began to release its software as a fully supported product in 1999. Tripwire is now available for multiple platforms including Windows NT and 2000, Solaris, AIX, HP-UX, Free BSD, and Linux.

The basic function of Tripwire is to check the integrity of important files and directories against a baseline database and raise an alert when any changes occur within the preset policy. It is considered best practice to ensure that the system on which Tripwire is being installed is in a known secure state, that is, the operating system and any application software has not already been compromised.

The installation and setup process overview covered here refers to a Solaris command line install of Tripwire for Servers, using the default locations for the files and directories. Root or superuser access is required to install Tripwire software. To begin, run the install script by typing ./install.sh. After accepting the license agreement, the screen will display the directories into which Tripwire will be installed. Once this has been acknowledged, the installer will begin copying files to the system. The next important step is to choose a site passphrase to encrypt the configuration and policy files to prevent unauthorized modification. Any changes to these files will require the use of this passphrase. Next, a local passphrase must be chosen which cryptographically signs the Tripwire database and report files. Guidelines for choosing a strong passphrase are included during the installation process.

The policy file contains the set of rules describing the files and directories that Tripwire will monitor. Below is the default policy file.

The first section lists the location of the Tripwire files and directories as well as the name of the workstation the software is installed on.

```
@@section GLOBAL
TWROOT=; /opt/TSS
TWBIN=; /opt/TSS/bin
TWPOL=; /opt/TSS/policy
TWDB=; /opt/TSS/db
```

```
TWSKEY=; /opt/TSS/key
TWLKEY=; /opt/TSS/key
TWREPORT=; /opt/TSS/report
HOSTNAME=; somehost
```

The next section defines the rules and parameters for files changes as well as assigning a number value to differentiate between low, medium, and high levels of criticality. Variables are preset in this section to use throughout the policy. Comments are also included to provide detailed explanations of each rule.

```
@@section FS
SEC_CRIT      = $(IgnoreNone)-SHa;  # Critical files - we can't afford to
                                      miss any changes.
SEC_SUID      = $(IgnoreNone)-SHa;  # Binaries with the SUID or SGID flags
                                      set.
SEC_BIN       = $(ReadOnly);        # Binaries that shouldn't change
SEC_CONFIG    = $(Dynamic);         # Config files that are changed
                                      infrequently but accessed often.
SEC_LOG       = $(Growing);         # Files that grow, but that should never
                                      change ownership.
SEC_INVARIANT = +pug;               # Directories that should never change
                                      permission or ownership.
SIG_LOW       = 33;                 # Non-critical files that are of minimal
                                      security impact
SIG_MED       = 66;                 # Non-critical files that are of
                                      significant security impact
SIG_HI        = 100;                # Critical files that are significant
                                      points of vulnerability
```

Tripwire binary, data, configuration, and policy files can be commented out or removed but it is recommended to leave them in the policy to prevent changes to Tripwire program files from being undetected and compromise the integrity of the reports. Before each list of files, the rule name is defined along with a severity level. In the default policy, the variables have been pre-assigned with either an action or numeric value.

```
# Tripwire Binaries
(rulename = "Tripwire Binaries", severity = $(SIG_HI))
{
  $(TWBIN)/siggen   -> $(ReadOnly);
  $(TWBIN)/tripwire -> $(ReadOnly);
  $(TWBIN)/twadmin  -> $(ReadOnly);
  $(TWBIN)/twprint  -> $(ReadOnly);
}

# Tripwire Data Files - Configuration Files, Policy Files,
  Keys, Reports, Databases
(rulename = "Tripwire Data Files", severity = $(SIG_HI))
{
  # NOTE: Removing the inode attribute because when Tripwire creates a backup
  # it does so by renaming the old file and creating a new one (which will
  # have a new inode number).  Leaving inode turned on for keys, which
  # shouldn't ever change.

  # NOTE: this rule will trigger on the first integrity check after database
  # initialization, and each integrity check afterward until a database update
  # is run, since the database file will not exist before that point.
  $(TWDB)                           -> $(Dynamic) -i;

  $(TWPOL)/tw.pol                   -> $(SEC_BIN) -i;
  $(TWBIN)/tw.cfg                   -> $(SEC_BIN) -i;
  $(TWLKEY)/$(HOSTNAME)-local.key   -> $(SEC_BIN) ;
  $(TWSKEY)/site.key                -> $(SEC_BIN) ;

  #don't scan the individual reports
  $(TWREPORT)                       -> $(Dynamic) (recurse=0);
}
```

This is a brief list where important changes might be occurring such as the root and home directories.

Actual lists can be several pages long depending upon how many files and directories are being monitored. The recurse setting tells Tripwire how many levels below the main directory it should look for changes. In this case, the ‘0’ setting tells it to only search the main directory.

```
# Commonly accessed directories that should remain
    static with regards to owner and group
(rulename = "Invariant Directories", severity = $(SIG_MED))
{
  /      -> $(SEC_INVARIANT) (recurse = 0);
  /home  -> $(SEC_INVARIANT) (recurse = 0);
  /etc   -> $(SEC_INVARIANT) (recurse = 0);
  /opt   -> $(SEC_INVARIANT) (recurse = 0);

}
```

While the default policy is customized for the platform on which the software will be installed, it is important to evaluate the system on a case-by-case basis to ensure the configuration is suited for that particular workstation.

Once the plain text policy file has been copied to the /usr/TSS/policy directory it must be encrypted as Tripwire software can only read binary, encrypted policy files. To encrypt the policy file, run the following command from the from the Tripwire bin directory: /usr/TSS/bin:

```
.twadmin -create-polfile ../policy/twpol.txt
```

The site passphrase must be entered to create the new encrypted policy file. Now the baseline database that Tripwire uses to compare for changes to files listed in the policy must be generated. To initialize the database, enter the following command from the /usr/TSS/bin directory:

```
./tripwire -init
```

The local passphrase must be entered to generate the database. Once the database is created, an integrity check can be run using the rules in the policy against the current system state and compare the data in the database. The output of the integrity check command tripwire -m c, which can be run in an interactive mode or sent to a file, is the Tripwire report. Interactive mode automatically displays the report using the default text editor, vi.

The first part of the report displays the date, the user that ran the report, host name and IP address as well as the policy, database, and configuration files used to run the integrity check.

```
Tripwire(R) 2.2.1a Integrity Check Report

Report generated by:          root
Report created on:            Mon Jan 14 13:52:01 2002
Database last updated on:     Never

===============================================================================
Report Summary:
===============================================================================

Host name:                    somehost
Host IP address:              127.0.0.1
Host ID:                      10c0d020
Policy file used:             /opt/TSS/policy/tw.pol
Configuration file used:      /opt/TSS/bin/tw.cfg
Database file used:           /opt/TSS/db/somehost.twd
Command line used:            tripwire -m c
```

The Rule Summary lists all of the rules and their severity levels that were defined in beginning of the

policy file. The Added, Removed, or Modified columns display how many files changed within a particular rule. In this case, one violation was found under the Invariant Directories rule.

```
Rule Summary:
===============================================================================
-------------------------------------------------------------------------------
  Section: Unix File System
-------------------------------------------------------------------------------

  Rule Name                       Severity Level   Added    Removed  Modified
  ---------                       --------------   -----    -------  --------
  Invariant Directories           66               0        0        1
  Tripwire Data Files             100              0        0        0
  Configuration Files             0                0        0        0
  Critical devices                100              0        0        0
  Home directory permissions      50               0        0        0
  Root config files               100              0        0        0
  Tripwire Binaries               100              0        0        0
  OS executables and libraries    100              0        0        0
  System boot changes             100              0        0        0
  setuid/setgid                   100              0        0        0
  Libraries                       66               0        0        0
  Low security impact directories 10               0        0        0
  User binaries                   66               0        0        0
  (/usr/local)
  Critical system boot files      100              0        0        0
  (/kernel)

Total objects scanned:  4543
Total violations found:  1
```

This section offers detailed information of any violations listed in the Rule Summary. In the default policy, the rule name ‘Invariant Directories’ was created to detect any changes in ownership to significant directories such as the "/opt" directory. As this is a directory that third party application software is often installed into, a violation of this rule should warrant an investigation to find out why changes were made to owner or group permissions.

```
===============================================================================
Object Summary:
===============================================================================
-------------------------------------------------------------------------------
# Section: Unix File System
-------------------------------------------------------------------------------
Rule Name: Invariant Directories (/opt)
Severity Level: 66
-------------------------------------------------------------------------------
Modified:
"/opt"
```

If the Tripwire policy has been configured to monitor files that do not exist on the system, they will appear in the error report. Missing files should be reviewed and removed from the policy if they are not part of the operating system or any applications.

```
===============================================================================
Error Report:
===============================================================================
-------------------------------------------------------------------------------
  Section: Unix File System
-------------------------------------------------------------------------------

1.   File system error.
     Filename: /usr/home
     No such file or directory
2.   File system error.
     Filename: /usr/aset/tmp
     No such file or directory
```

When a violation occurs, it will continue to appear in the Tripwire report until it has been acknowledged. To acknowledge changes, the report must be viewed using the integrity check command with the interactive flag: ./tripwire –m c –I. The output displayed in the vi text editor will appear similar to the above report with the exception of checkboxes:

```
Modified:

[x] "/opt"
```

To prevent an update to the database, the 'x' can be removed otherwise the database will automatically be refreshed once the report has been read. The report does not have to be saved in vi to acknowledge the changes. Tripwire does not automatically run though it can be incorporated with regular auditing procedures and scheduled as a cron job.

After Tripwire has been run for several days and the report output has been reviewed, patterns for file changes should emerge and may even show that the default policy does not detect files changes in the best manner. Errors for files not on the system should be removed and any files for programs that are not appearing should be edited into the policy. The default policy defines rule names and actions along with severity levels at the beginning of the file, and then assigns files to each rule. While this might work with a system running many services and software programs installed, another option for a smaller workstation running few services and software would be to define the rule name as a severity level and specify individual actions for each file under the rule. This is demonstrated in the following example:

```
(rulename = "High Significance", severity = 100 )
{
  /                 ->        $(IgnoreNone) -SHa;
  /etc              ->        $(IgnoreNone) -SHacm ;
  /etc/shadow       ->        $(Dynamic) -i ;
  /opt              ->        $(IgnoreNone) -SHa;
}
```

And the resulting report for a policy file configured in this manner:

```
Rule Summary:
===============================================================================

-------------------------------------------------------------------------------
  Section: Unix File System
-------------------------------------------------------------------------------

  Rule Name                     Severity Level   Added    Removed  Modified
  ---------                      --------------   -----    -------  --------
  High Significance              100              0        0        0
  Tripwire Data Files            100              0        0        0
  Low Significance               50               0        0        0
  Almost No Significance         25               0        0        0
  Moderate Significance          75               0        0        0
  Tripwire Binaries              100              0        0        0
```

The commercial version of Tripwire software for Unix or Windows NT can be found at http://www.tripwire.com/. The open source version of Tripwire for Linux can be downloaded at http://www.tripwire.org/downloads/index.php. The original academic source release or ASR for Unix is available as a free download at http://www.tripwire.com/products/tripwire_ASR/.

Intruder Alert is Symantec's host-based intrusion software. Similar to Tripwire, it monitors files and directories based on a set of policies. In contrast to Tripwire's text output, it offers a graphical user interface or GUI and real time updates. Intruder Alert can run on either a Windows, Unix, or NetWare platform though is also possible to integrate systems that run different operating systems. One possible configuration could have the Agents and Managers installed on Unix servers and the Administrator and

Event Viewer installed on Windows NT machines.

There are four components to Intruder Alert:

- **Agent** - a Unix daemon or Windows NT service that monitors events and can perform actions within the parameters of the predefined security policies. They can also receive policy updates from the Manager and establish secure connections to transmit encrypted data across the network. Agents run 24 hours a day, 7 days a week.
- **Manager** - a Unix daemon or Windows NT service that facilitates communication between the Agents, the Event Viewer, and the Administrator. Also stores event data from the Agents and their applied policies and domain information.
- **Event Viewer** - the GUI that displays events captured by the Agents. It can also be used to send commands to the Agents and generate reports.
- **Administrator** –the GUI main control for ITA. Creates and administers the security policies, creates and manages domains and connects to the Managers. Also used to manage ITA users and privileges.

The first step in setting up ITA is to determine which systems will be the Agents, what Manager(s) they will report to and what systems will have the Event Viewer and Administrator tools. The Agents can be systems with important data files or machines that perform a service such an ftp or domain name server that should be closely monitored for possible security breaches. The Manager, which is capable of maintaining up to100 Agents, does not have to be on a dedicated machine but it should reside on a fast, stable system that is able to connect to all of the Agents from which it is collecting data. The Administrator and Event Viewer can be installed onto any workstation or PC. An Administrator can support an unlimited number of Managers so even in a large company, only one or two consoles may be necessary to perform administrative tasks.

Organization is one of the key features of Intruder Alert. Agents can be organized into domains with a common element such as location, workgroup, access restriction or operating system. Domains are used so that there is a systematic method of administering policies. All Agents within a specified domain will have the same policies.

During the installation of the ITA Manager, the option to install Symantec's ready-made policies is available. All of the pre-defined polices fall into two categories: Drop & Detect and Configure to Detect. The Drop & Detect policies do not need any tweaking or customization. Configure to Detect polices require a few changes before they can be applied to a domain and cannot be selected during the initial installation. All of the policies, including any new policies created by the administrator, reside in the Policy Library database.

Each policy has its own set of rules and rule criteria to detect changes. A rule is composed of three parts: Select, Ignore, and specified Actions, however all three are not necessarily required for a rule to be considered valid. Select defines the event to detect, Ignore defines the exceptions to the rule, and Action defines the process to be performed when the Select criteria is met. The most common action is to record to the Event Viewer, which will log the event into the Manager's event database. Other actions include raising flags to mark events or to notify other Agents, send an email or page, disconnect the session or disable a user.

Rules are also given a severity level with values between 0-100. The standard values that are assigned to the pre-defined policies are:

1. Administrative

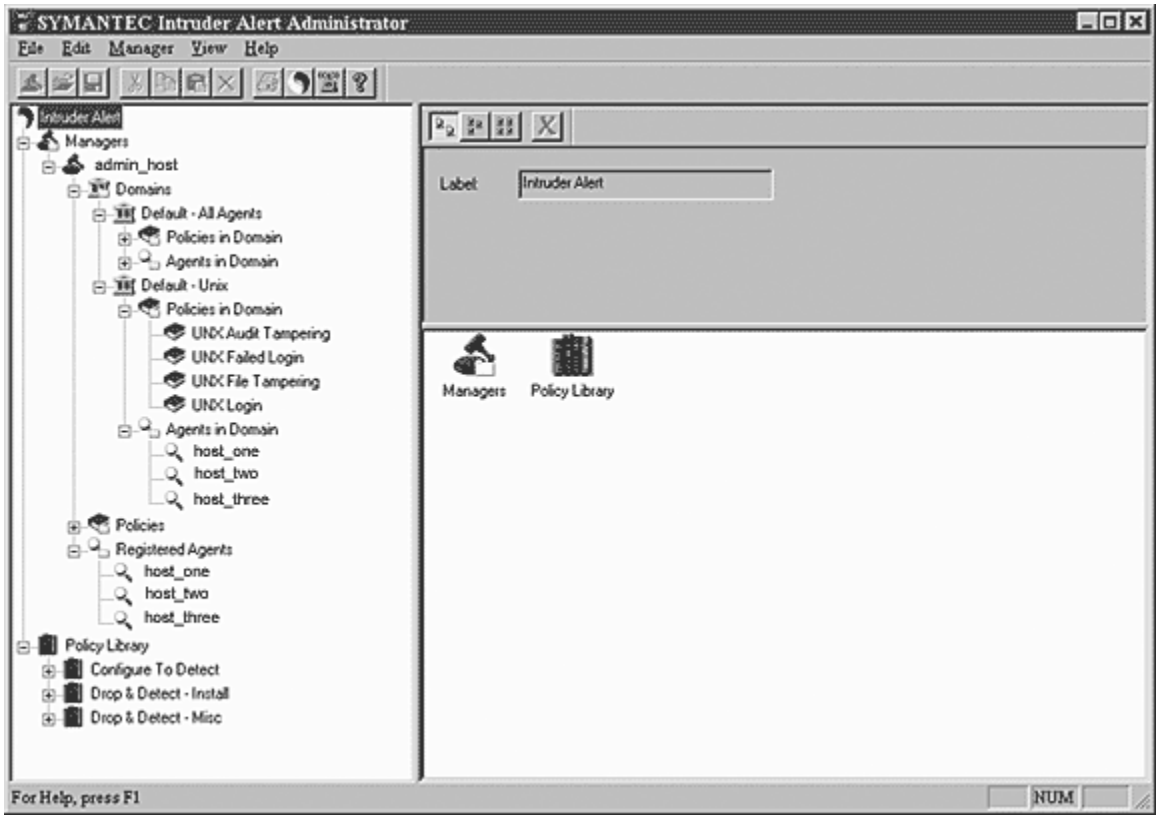
1. FYI

1. Alert

1. Emergency

These levels can be used as guidelines when creating new rules.

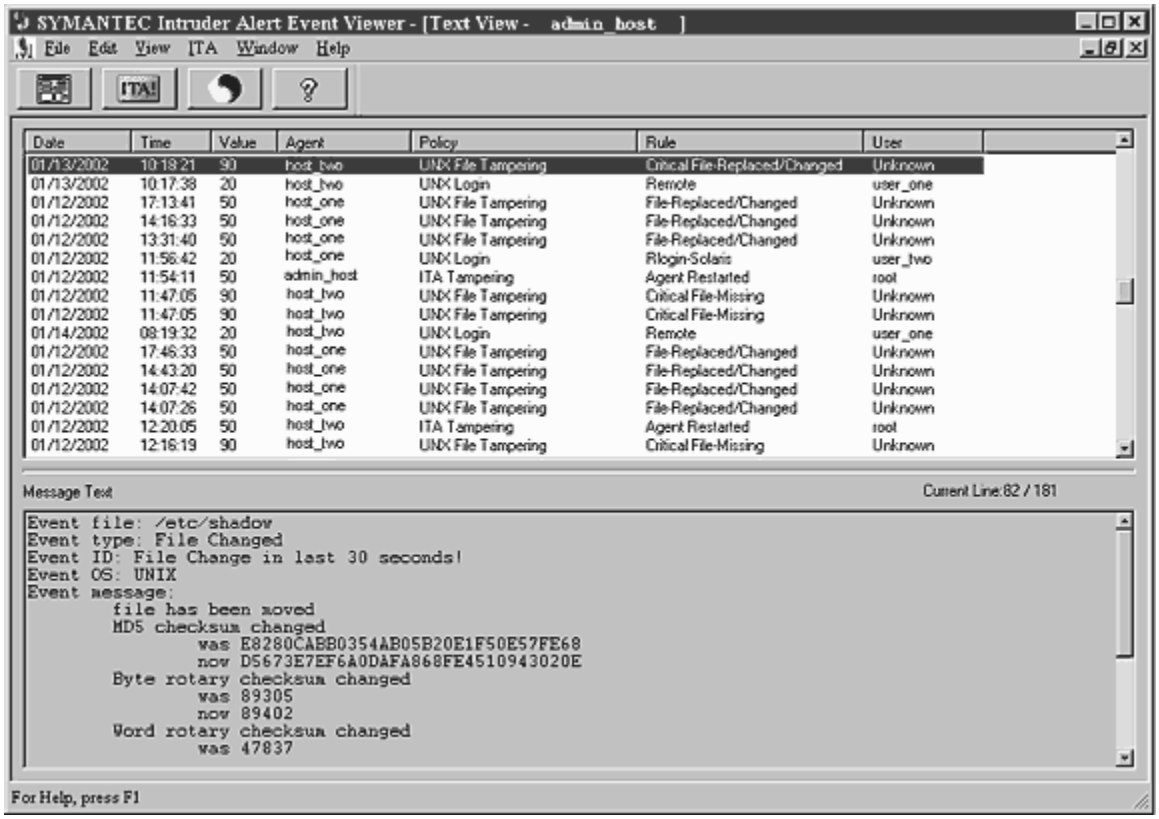
A closer look at the ITA Administrator can display how a Manager with three Agents has been set up. The Manager in this example has been named ‘admin_host’ and the three registered Agents are hosts one, two, and three. Admin_host has two domains, Default All Agents and Default Unix. The Default All Agents is always created during the initial installation and will include all of the registered Agents. The Default Unix domain was created because Drop & Detect Unix policies were selected when Intruder Alert was installed on the Manager. Any Agents with Unix polices will be listed in the Default Unix Domain.



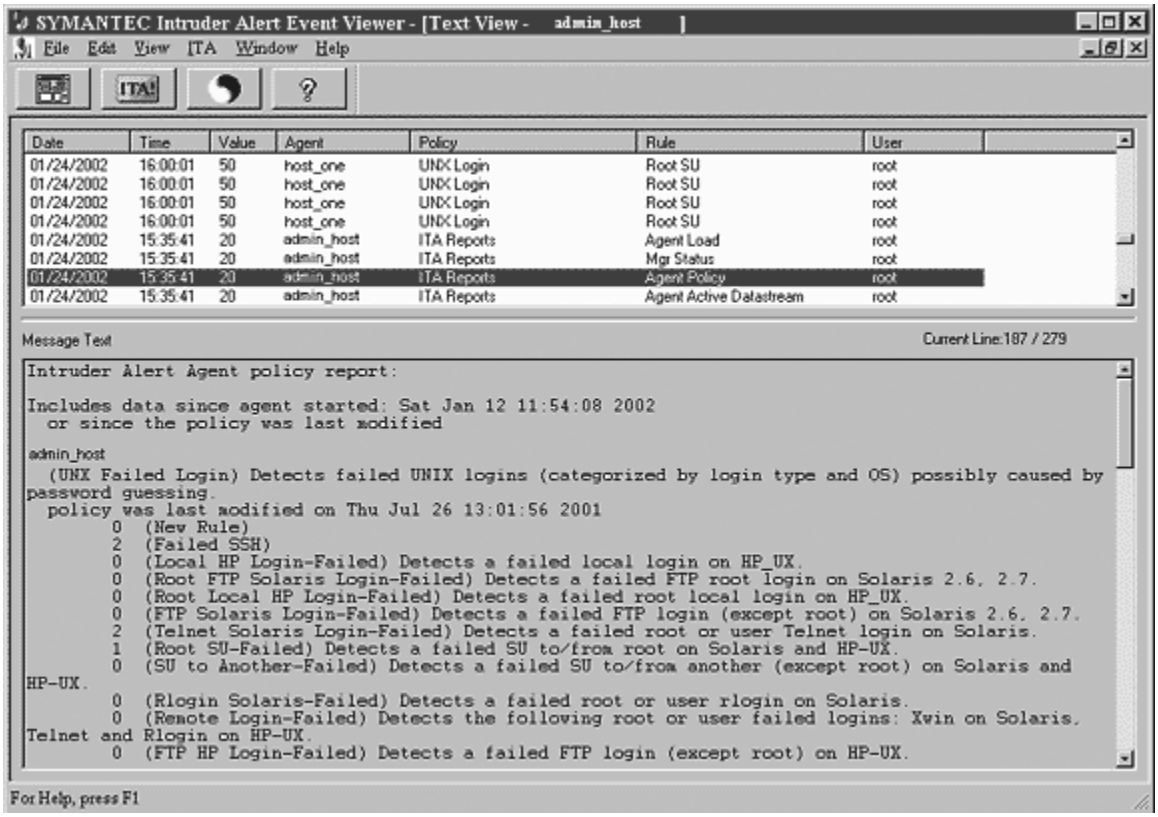

Once a Manager has been set up, new accounts can be created with the User Manager to allow other users to access the Administrator and Event Viewer. Access levels can vary from full administrative access with the ability to create, update, and apply policies, add new users or change Agent, Manager, or ITA configurations to only allowing access to the Event Viewer.

Each time the rule criteria are met within a policy on an Agent, an event is recorded to the event database on its Manager. The events captured by the Agents are then communicated from the Manager to the Event Viewer and displayed as they occur. An event can be highlighted to find out further information. The Message Text will display the event file, type, and ID as well as a message describing the change to the file.

Below is an example of daily events that might be captured by Agents.




The Event Viewer can be used to send a report command to an Agent to view its current policy. The report command will be logged as an event and the output will be displayed in the Message Text section of the Event Viewer. Each policy applied to the Agent will be listed in detail along with the rules that fall under the policy. The number beside each rule indicates how many violations have occurred for that particular rule since the Agent was started or the policy modified. In the example below, the Unix Failed Login policy is shown with violations for three of its rules.



Multiple reporting capabilities are another feature of the Event Viewer. Reports can be displayed in a graphic format such bar or line graphs or in a text report that can be customized to a specific audience such as a management or technical team. This can be useful to track which policies, rules, or Agents generate the most incidents.

Information on Intruder Alert can be found on Symantec's web site at http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=48&PID=10012335&EID=0.

Deciding which of these two possible solutions would fit the needs of a particular company should involve careful consideration of the systems and the level of monitoring required. Tripwire might be recommended for a smaller environment consisting mainly of Unix workstations or servers that do not need constant review or live updates where it could be easily incorporated with regular administrative logging. Symantec's Intruder Alert could better serve larger network architecture with systems in multiple locations that should have a consistent level of auditing.

Depending upon the environment, Tripwire can have both advantages and disadvantages. A potential drawback could be a higher learning curve to install, edit, and maintain the software. For the Unix version of Tripwire, there should be a familiarity with a command line interface and vi text editor. Creating and editing a policy can take time and requires a solid understanding of how and why the operating system and applications files change to become effective in detecting unusual activity. Tripwire Manager, the central management console that can minimize administration, is not included with Tripwire for Servers. Without the Manager, maintenance and updates need to be performed on individual machines. Tripwire can also generate many log files that can make managing multiple machines a cumbersome task.

Running Tripwire on a Unix platform can be a daunting task if one is not familiar with system administration however that challenge could be outweighed by the benefit of a high level of customization. Policy files can be edited from the default or completely rewritten to best fit the system.

Intruder Alert also has positive and negative aspects. One advantage of ITA is that it is designed to integrate with NetProwler, the network-based IDS that is part of Symantec's intrusion detection suite, and it is possible to manage both programs from one central console. Intruder Alert Agents can monitor a system 24 hours a day, 7 days a week with real-time updates. The use of a database stored on the Manager allows events from previous days or weeks to be viewed. Policies can be created or updated and quickly extended to all Agents.

Potential issues might include the purchase of additional hardware if there are not enough system resources to dedicate to running a Manager with an event database. If a company has individual teams that administer each platform, coordinating efforts with each to monitor multiple operating systems could be difficult.

Viewing intrusion detection as an "art and a science" [1] can enable a security analyst to find, deploy, and maintain a host-based IDS that will detect to the best of its ability. The scientific aspect of intrusion detection could be considered the research and installation of the appropriate software while the artistic side might encompass adapting the software and policies to closely monitor critical systems in the most efficient manner.

**Bibliography**

1. Loshin, Peter of Computerworld. "Intrusion Detection" April 16, 2001
http://www.computerworld.com/cwi/story/0,1199,NAV47_STO059611,00.html

2. Van Wyk, Kenneth & Forno, Richard: excerpt from Incident Response Chapter 7 Tools of the Trade, O'Reilly 2001 http://www.onlamp.com/lpt/a//onlamp/excerpt/incidentres_07/index2.html

3. Symantec Intruder Alert 3.5
http://www.nss.co.uk/ids/symantec_intruderalert/symantec_intruder_alert.htm

4. Arnold, Edward of SecurityFocus. "The Trouble With Tripwire: Making a Valuable Security Tool More Efficient. June 6, 2001 http://www1.securityfocus.com/frames/?focus=sun&content=/focus/sun/articles/tripwire.html

5. Walder, Bob & Jayne Parkhouse. "Unearthing the Invaders – July 2001 Test Center"
http://www.scmagazine.com/scmagazine/2001_07/testc/prod1.html

6. Zirkle, Laurie. "Intrusion Detection FAQ – What is host-based intrusion detection?"
http://www.sans.org/newlook/resources/IDFAQ/host_based.htm

7. Symantec Intruder Alert 3.6 "Installation and Getting Started Guide"

8. Symantec Intruder Alert 3.6 "Users Guide"

9. Tripwire Evaluation Guide for Unix

to top of page | to Reading Room Home

Last Updated: September 3rd, 2015

# Upcoming SANS Training

**Click Here for a full list of all Upcoming SANS Events by Location**

| | | | |
|---|---|---|---|
| SANS Crystal City 2015 | Crystal City, VAUS | Sep 08, 2015 - Sep 13, 2015 | Live Event |
| SANS Network Security 2015 | Las Vegas, NVUS | Sep 12, 2015 - Sep 21, 2015 | Live Event |
| SANS Seoul 2015 | Seoul, KR | Sep 14, 2015 - Sep 19, 2015 | Live Event |
| Data Breach Investigation Summit & Training | Dallas, TXUS | Sep 21, 2015 - Sep 26, 2015 | Live Event |
| SANS Baltimore 2015 | Baltimore, MDUS | Sep 21, 2015 - Sep 26, 2015 | Live Event |
| SANS Perth 2015 | Perth, AU | Sep 21, 2015 - Sep 26, 2015 | Live Event |
| SANS Tallinn 2015 | Tallinn, EE | Sep 21, 2015 - Sep 26, 2015 | Live Event |
| SANS ICS Amsterdam 2015 | Amsterdam, NL | Sep 22, 2015 - Sep 28, 2015 | Live Event |
| SANS Bangalore 2015 | Bangalore, IN | Sep 28, 2015 - Oct 17, 2015 | Live Event |
| SANS DFIR Prague 2015 | Prague, CZ | Oct 05, 2015 - Oct 17, 2015 | Live Event |
| SANS Seattle 2015 | Seattle, WAUS | Oct 05, 2015 - Oct 10, 2015 | Live Event |
| SANS Tysons Corner 2015 | Tysons Corner, VAUS | Oct 12, 2015 - Oct 17, 2015 | Live Event |
| SOS: SANS October Singapore 2015 | Singapore, SG | Oct 12, 2015 - Oct 24, 2015 | Live Event |
| SANS Gulf Region 2015 | Dubai, AE | Oct 17, 2015 - Oct 29, 2015 | Live Event |
| SANS Cyber Defense San Diego 2015 | San Diego, CAUS | Oct 19, 2015 - Oct 24, 2015 | Live Event |
| SANS Tokyo Autumn 2015 | Tokyo, JP | Oct 19, 2015 - Oct 31, 2015 | Live Event |
| SANS South Florida 2015 | Fort Lauderdale, FLUS | Nov 09, 2015 - Nov 14, 2015 | Live Event |
| SANS Sydney 2015 | Sydney, AU | Nov 09, 2015 - Nov 21, 2015 | Live Event |
| SANS London 2015 | London, GB | Nov 14, 2015 - Nov 23, 2015 | Live Event |
| Pen Test Hackfest Summit & Training | Alexandria, VAUS | Nov 16, 2015 - Nov 23, 2015 | Live Event |
| SANS Hyderabad 2015 | Hyderabad, IN | Nov 24, 2015 - Dec 04, 2015 | Live Event |
| SANS Cape Town 2015 | Cape Town, ZA | Nov 30, 2015 - Dec 05, 2015 | Live Event |
| SANS San Francisco 2015 | San Francisco, CAUS | Nov 30, 2015 - Dec 05, 2015 | Live Event |
| SANS Milan 2015 | OnlineIT | Sep 07, 2015 - Sep 12, 2015 | Live Event |
| SANS OnDemand | Books & MP3s OnlyUS | Anytime | Self Paced |