GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Case Study: Improving Solaris System and Security Monitoring

D Bick
SANS GSEC Paper
22 Oct 2002

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Table of Contents

© SANS Institute 2000 - 2002, Author retains full rights.

B. Sample Alert Files

1. Watch Process (Restart BIND named Daemon)

2. Monitor SNMP (SNMP Hack Attempt)

3. Monitor Filesystem (/usr Filesystem)

4. Watch Logfiles (fatal Error)

VII. References

© SANS Institute 2000 - 2002, Author retains full rights.

I. Abstract

Best practice of system administration requires being able to monitor systems 24/7 for their performance and security condition. While unix provides powerful capabilities for logging information, alerting based on the logged information is often a weak link for the system and security administrator.

I will detail in this paper an alerting system based on a commercial product with additional programming and system enhancements I have implemented in completion of the GSEC Certification which provides for an increased level of performance and security monitoring/alerting for Sun Solaris servers at our location.

While implementing the enhancements I detail in this paper provide for an improved security posture, another goal was to improve the system administration environment.

II. Existing System and Security Administration Environment

A. Description of System Environment

We maintain numerous Sun servers which are used for a variety of network centric tasks. These include database, Domain Name System (DNS), Network File System (NFS), print and web servers. These are primarily Sun 420R server systems with 256MB to 1GB of RAM memory, depending on their specific application. The amount of disk storage again is application dependent. Some systems have basic SCSI attached disk while some have additional hardware RAID storage; the aggregate storage ranges from 9GB to 90GB per server.

These servers, while protected by enterprise firewalls, have considerable exposure to the Internet which is required due to their functionality. They run the Solaris version 8.0 operating system with additional application software. This application software includes the DNS server software (BIND), Apache web server software and Oracle database software, depending on the purpose of the specific system. All of our servers run the Solaris logging software syslog.

Syslog is the unix utility which logs system and user messages and for our purposes of security monitoring contains important security information on the host (Sun, syslog, p.1). As is typical with a well defined security policy syslogging is defined for each of these hosts in our environment.

B. Existing Security Administration Environment

Our former way of dealing with security issues required responding to security events as they were discovered. This was the de-facto, non written security policy. In practice, this meant that as a system administrator happened to be working on a system, he would discover problems. Problems were discovered either by running commands or by examining the syslog for problem conditions.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Problems typically may have gone undiscovered for a considerable period of time before being corrected with this method. In many cases they became a critical problem which adversely impacted a service or downed the system, when they should have been caught earlier.

III. Risk Analysis of the Existing Environment

A. Threats to the Systems

Threats to these servers come from hosts on the Internet as well as potentially from internal users with access to the systems exceeding their authority. In this regard they typify internal/external threats common to many networked systems.

1. External Threats

External threats are those that originate from the outside of our network (the Internet).

One example would be attacks against our systems which run SNMP software. A hacker could attempt to guess the SNMP community strings through multiple access attempts (CERT, SNMP, p.1).

Another example would be application specific attacks that could occur, such as exploits against an Apache web server. A hacker could attempt to login to a restricted area of a web server by guessing usernames and passwords.

Lastly buffer overflow attacks may be done against operating system or application software which is vulnerable to such a type of attack. This is a type of network attack which results in the hacker gaining privileged access to the target system (CERT, "Buffer Overflows, p. 1).

A. Internal Threats

Internal threats are those that are perpetrated by someone within the organization.

Internal threats could be any of the above network centric threats with the addition of host based attacks, such as attempting to gain root access on a system while logged in as a non-privileged user.

B. Risk Model

In reviewing the options available to improve system security, I used the risk management model to characterize risk. In this model there are three options in risk management, accepting the risk, transferring the risk, or mitigating the risk (SANS GSEC, chapter 3-6, p. 6-3.)

© SANS Institute 2000 - 2002, Author retains full rights.

It was unacceptable to accept the current level of risk as it existed, this was a decided lack of due diligence (i.e. management's expectation was to improve the level of security thereby reducing the risk).

Transferring the risk was also unacceptable, for two reasons. First, buying insurance was not possible. Second, all systems and network management was performed in house and transferring the risk was not possible.

Mitigation was the clear choice for our environment. I undertook steps to reduce the risk in our environment through systems enhancements which I detail in this paper.

C. Calculated Risk Levels

I choose to use the qualitative method for determining the risk level to these systems.

"Qualitative Risk Analysis does not attempt to assign numeric values to the components. In this method, scenarios are created that outline the potential threats to the business and rank these threats according to their seriousness. " (Miller, p.1)

As my goal was to mitigate major risk areas, the qualitative method accurately identified the risks which required remedy.

Our old environment, while logging system and user information in syslog, was very inadequate in terms of alerting the system/security administrators when problems arouse.

Given the inherently hostile network environment (connected to the Internet), as described above, immediate reporting of conditions was imperative to insure the continued operation of our servers. Without adequate timely alerting, our vulnerability was greatly increased to security incidents.

By examining the network exposure and factoring in internal/external threats I made the following risk assessment:

Risk Level	Source	Type of Server
High	External	Well known servers with Internet connectivity (DNS)
Medium	External	Other servers with Internet connectivity
Medium	Internal	Internal personnel host or network attacks

The assigned risk levels were derived by combing the environment the systems were exposed to combined with the system administration environment.

As a provider of network service, there would be a negative affect if an attack succeeded against a critical server, such as a DNS server. This would be at worst a potential lost of income but certainly an erosion of customer confidence, which is in itself an intangible

© SANS Institute 2000 - 2002, Author retains full rights.

important asset for a business. The qualitative risk analysis provides us a measure of risk to our servers and allows us to tailor measures to mitigate the risk based on overall need.

IV. Security Improvements Implemented

A. Installation of SystemEdge

I deployed host based monitoring agents which were a product of Concord Communications (http://www.concord.com). Being a commercial product it saved us from extensive development time, and it was also suitably priced. It met our technical requirements, which was to provide for monitoring of logfiles, system parameters and the capability of alerting and restarting processes. SNMP agent functionality was also required. In these ways it fulfilled our technical and business requirements.

Other solutions either provided not enough functionality, or provided additional capabilities (at a price beyond our budget) which were unneeded.

These agents run continually on the system. In unix terminology the term for such a persistent process is know as a "daemon" (Nemeth et al, p. 705). This daemon monitors standard and extended (proprietary) SNMP MIBs, logfiles, and system parameters. Our requirements dictated functionality beyond what a program such as swatch (Atkins, p.1), which is a freeware unix log monitor, would provide. Specifically we required SNMP agents resident on each server, system process monitoring/restarting and system parameter monitoring.

As the SystemEdge agent is a commercial licensed product, it must first be installed. The product is distributed on CDROM and it has standard Solaris package format.

The following is the command line syntax to install the product on a server (Concord, chapter 2).

Note that the conditions needed to perform this installation are that the root user is logged into the unix system that the software is to be installed on. The SystemEdge software CDROM has already been mounted on the system. My comments in the following section appear in parenthesis.

First I change directories and list the files present on the CDROM for the sysedge product. The Solaris package is in the sol2 subdirectory:

```
#
# cd /cdrom/empire/sysedge/sol2
# ls
readme      sysedge.pkg
#
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Next I use the Solaris package add utility "pkgadd" to install this software. The "-d" option specifies that the next argument is the path and filename of the package to install. In our case it is "sysedge.pkg". The package installation then begins:

```
# pkgadd -d sysedge.pkg

The following packages are available:
  1  EMPsysedg     Empire SystemEdge Agent
                   (sparc) 4.0

(1 is entered to indicate that I want to install the item listed above)

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1

Processing package instance <EMPsysedg> from
</cdrom/empire/sysedge/sol2/sysedge.pkg>

Empire SystemEdge Agent
(sparc) 4.0

                    Concord SystemEdge Agent
                        Version 4.0

                    Concord Communications, Inc.

                COPYRIGHT 2000 Concord Communications, Inc.
                      ALL RIGHTS RESERVED.

                    Concord Communications, Inc
                        600 Nickerson Rd
                       Marlboro, MA 01752
                        +1 800.851.8725
                        info@concord.com

THIS PROGRAM CONTAINS CONFIDENTIAL AND TRADE SECRET
INFORMATION OF THE Concord Communications, Inc.  COPYRIGHT NOTICE IS
PRECAUTIONARY ONLY AND DOES NOT IMPLY PUBLICATION.

BY USING THIS SOFTWARE, YOU ACCEPT AND AGREE TO BECOME BOUND
BY THE TERMS OF THIS LICENSE.  IF YOU DO NOT ACCEPT THE TERMS OF
THIS LICENSE, DO NOT USE THE SOFTWARE.  EMPIRE TECHNOLOGIES, INC.
RETAINS THE OWNERSHIP OF THIS COPY OF THE SOFTWARE AND
DOCUMENTATION, WHICH IS LICENSED TO YOU FOR USE UNDER THE
FOLLOWING CONDITIONS.  EMPIRE TECHNOLOGIES, INC. MAKES NO
WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH
```

© SANS Institute 2000 - 2002, Author retains full rights.

RESPECT TO THIS PRODUCT, INCLUDING QUALITY, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THE RIGHTS OF THIRD PARTIES. IN NO EVENT WILL EMPIRE TECHNOLOGIES, INC. BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SPECIFICALLY, EMPIRE TECHNOLOGIES, INC. IS NOT RESPONSIBLE FOR ANY COSTS INCLUDING, BUT NOT LIMITED TO, THOSE INCURRED AS A RESULT OF LOST PROFITS OR REVENUE, LOSS OF USE OF THE PRODUCT, LOSS OF DATA, THE COSTS OF RECOVERING SUCH SOFTWARE OR DATA, THE COSTS OF SUBSTITUTE SOFTWARE OR DATA, CLAIMS BY THIRD PARTIES, OR FOR OTHER SIMILAR COSTS. IN NO CASE SHALL THE LIABILITY OF EMPIRE TECHNOLOGIES, INC. EXCEED THE AMOUNT OF THE LICENSE FEE.

Using </opt> as the package base directory.
Processing package information.
Processing system information.
Verifying disk space requirements.
Checking for conflicts with packages already installed.
Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

(I now enter “y” to indicate that I agree that the installation will execute with super-user permissions during this installation. This is necessary and as the package is from a known trusted source (vendor CDROM) acceptable).

Do you want to continue with the installation of <EMPsysedg> [y,n,?] y

Installing Empire SystemEdge Agent as <EMPsysedg>

Installing part 1 of 1.
/opt/EMPsysedge/bin/edgemon
/opt/EMPsysedge/bin/edgewatch
/opt/EMPsysedge/bin/emphistory
/opt/EMPsysedge/bin/licenseme
/opt/EMPsysedge/bin/sendtrap
/opt/EMPsysedge/bin/snmpget
/opt/EMPsysedge/bin/snmpset
/opt/EMPsysedge/bin/sysedge.sol25
/opt/EMPsysedge/bin/sysedge.sol26
/opt/EMPsysedge/bin/sysedge.sol27
/opt/EMPsysedge/bin/sysedge.sol27-sparcv9
/opt/EMPsysedge/bin/sysedge.sol28

© SANS Institute 2000 - 2002, Author retains full rights.

/opt/EMPsysedge/bin/sysedge.sol28-sparcv9
/opt/EMPsysedge/bin/sysvariable
/opt/EMPsysedge/bin/version
/opt/EMPsysedge/bin/walktree
/opt/EMPsysedge/bin/xtrapmon
/opt/EMPsysedge/config/S99sysedge
/opt/EMPsysedge/config/license.txt
/opt/EMPsysedge/config/sysedge.cf
/opt/EMPsysedge/config/sysedge.lic
/opt/EMPsysedge/config/sysedge.mon
/opt/EMPsysedge/config/sysedge.reg
/opt/EMPsysedge/config/sysedge.rsrc
/opt/EMPsysedge/contrib/README
/opt/EMPsysedge/contrib/auto.install/README
/opt/EMPsysedge/contrib/auto.install/guess-config
/opt/EMPsysedge/contrib/auto.install/install-all
/opt/EMPsysedge/contrib/auto.install/install-hpux10
/opt/EMPsysedge/contrib/auto.install/install-hpux9
/opt/EMPsysedge/contrib/auto.install/install-solaris
/opt/EMPsysedge/contrib/auto.install/install-sunos
/opt/EMPsysedge/contrib/auto.install/licenses/send-for-keys
/opt/EMPsysedge/contrib/auto.install/packages/yes
/opt/EMPsysedge/contrib/auto.install/select-script
/opt/EMPsysedge/contrib/call-sendtrap.c
/opt/EMPsysedge/contrib/getextension.sh
/opt/EMPsysedge/contrib/mrtg/berkelium.shtml
/opt/EMPsysedge/contrib/mrtg/calcium.shtml
/opt/EMPsysedge/contrib/mrtg/info-30.shtml
/opt/EMPsysedge/contrib/mrtg/info-60.shtml
/opt/EMPsysedge/contrib/mrtg/krypton.shtml
/opt/EMPsysedge/contrib/mrtg/mrtg.cfg
/opt/EMPsysedge/contrib/mrtg/systemheader.html
/opt/EMPsysedge/contrib/mrtg/tin.shtml
/opt/EMPsysedge/contrib/ping.bat
/opt/EMPsysedge/contrib/ping.sh
/opt/EMPsysedge/contrib/restartproc.sh
/opt/EMPsysedge/doc/edgemon.1
/opt/EMPsysedge/doc/edgewatch.1
/opt/EMPsysedge/doc/emphistory.1
/opt/EMPsysedge/doc/empire.asn1
/opt/EMPsysedge/doc/hostmib.asn1
/opt/EMPsysedge/doc/manual.pdf
/opt/EMPsysedge/doc/relnotes.pdf
/opt/EMPsysedge/doc/rowstatus.ps
/opt/EMPsysedge/doc/sendtrap.1
/opt/EMPsysedge/doc/snmpset.1

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
/opt/EMPsysedge/doc/sysedge.1
/opt/EMPsysedge/doc/sysedge.cf.5
/opt/EMPsysedge/doc/sysedge.lic.5
/opt/EMPsysedge/doc/sysedge.mon.5
/opt/EMPsysedge/doc/sysvariable.1
/opt/EMPsysedge/doc/topprocs.asn1
/opt/EMPsysedge/doc/walktree.1
/opt/EMPsysedge/doc/xtrapmon.1
/opt/EMPsysedge/plugins/topprocs/topprocs-sol25.so
/opt/EMPsysedge/plugins/topprocs/topprocs-sol26.so
/opt/EMPsysedge/plugins/topprocs/topprocs-sol27-sparcv9.so
/opt/EMPsysedge/plugins/topprocs/topprocs-sol27.so
/opt/EMPsysedge/plugins/topprocs/topprocs-sol28-sparcv9.so
/opt/EMPsysedge/plugins/topprocs/topprocs-sol28.so
/opt/EMPsysedge/plugins/topprocs/topprocs.asn1
[ verifying class <none> ]
## Executing postinstall script.

Empire SystemEdge Agent Installation complete.
#
```

(The software has now been installed in the /opt/EMPsysedge directory on this system. I change to this directory on the system.)

```
#
# cd /opt/EMPsysedge
# ls
bin     config  contrib  doc     plugins
#
```

(Next copy the default license key to /etc):

```
# cp config/sysedge.lic /etc
#
```

(And copy the default configuration file to /etc):

```
# cp config/sysedge.cf /etc
#
```

(Now the program is started. This will generate an error message requesting the user to submit the license string to the software vendor. After this message, I abort the program with control c):

```
# bin/sysedge -f /etc/sysedge.cf
SystemEdge Version 4.0 Patchlevel 2
Copyright 2000 by Concord Communications, Inc

Could not find a valid license for machine 'metro'
```

© SANS Institute 2000 - 2002 Author retains full rights.

WWW: http://www.empire.com/licensing/
Provide the following:
 sysedge metro SunOS 5.8 80f0f25800007784 4.0 Patchlevel 2
sysedge: running in restricted mode
sysedge: unable to open monitor file /etc/sysedge.mon
sysedge: unable to open monitor file /etc/snmpd.empire.monitor
sysedge: using port 161, config file /etc/sysedge.cf
^C
#
#
Now I go online to the vendor's web site and register this host with the vendor. A license key is generated by this site, and it is put into the /etc/sysedge.lic file using a unix text editor such as "vi". I now have a active license key installed on the system.

At this point the SystemEdge software has been installed and has been licensed. The remaining task is to define the configuration file which contains the conditions to be monitored and the alerts to be generated per alert. In addition the Author's program to invoke the alerts and the alert descriptions must be installed.

B. Installation of Perl "sysedgeaction" Program and Supporting Files

The SystemEdge product monitors specific conditions on a system and calls a user defined program to respond to the given alert condition. This is a perl program I wrote called 'sysedgeaction'.

This program interprets the configuration file syntax, which has been supplemented to encode specific actions to perform. The reason I created this program was to have one generalized program which SystemEdge would invoke, and embed any specific actions to perform in the Sysedge configuration file /etc/sysedge.cf. This makes manageability much easier than having multiple programs for each monitored server.

The possible actions to perform are: send an email alert, send a page alert, or start a program.

The program is installed from a central server using the scp secure copy utility:

scp –p /usr/local/bin/sysedgeaction metro:/usr/local/bin
password:
#

In addition to the sysedgeaction program, there are text files which were created which describe a specific alert condition. These are emailed to the designated user when an alert occurs so that they can understand the problem. These are copied to each host with this command:

scp –p /usr/local/etc/* metro:/usr/local/etc

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

password:
#

The remaining item of configuration for our site is to use the existing capabilities of the unix mail aliases file to expand email and paging addresses. This facilitates entering multiple email or paging addresses, for instance having a 'sysadmin' alias with 4 actual email addresses. This keeps the SystemEdge configuration file shorter and allows for easier manageability of adding and deleting users, in that only the alias file needs to be changed, not the SystemEdge configuration file. The unix 'vi' text editor is used to modify this file, and the 'newaliases' program is run to rebuild the mail database (Sun, newaliases, p.1).

C. General Description of SystemEdge Monitored Categories

The SystemEdge product monitors 4 general categories on the unix system. These are: watch process, monitor SNMP, monitor filesystems, and watch logfiles.

1. Watch Process

The first category is watch process. This category monitors a given process in the process table (Concord, p. 10-1). Should the process end for any reason, an alert is generated. In most cases I configure the system to automatically restart the process.

2. Monitor SNMP

The second category is monitor SNMP. The SystemEdge product can monitor any Solaris OID and has and extended set of OIDs which cover extensive system performance and security variables (Concord, p. 6-1). For example, average CPU utilization can be monitored.

3. Monitor Filesystem

The third category is monitor filesystem. This category monitors the size of a filesystem and is useful to determine if a given filesystem has exceeded a safe threshold, for example 90% of its total size (Concord, 7-4 and 9-2).

4. Watch Logfiles

The last category, and one of major importance for security monitoring, is watching logfiles. The agent monitors using unix regular expression syntax any logfile on the system. This includes common unix syslog files as well as any application specific logfiles that are desired (Concord, 12-1).

D. Syntax of Configuration Files

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The SystemEdge configuration file /etc/sysedge.cf has one line (row) per monitored entry. Examples of these will be presented in following sections. Here is a field description of each of these lines, presented field by field. Each row has a fixed, space delimited field syntax. Lines are wrapped in the below configuration file due to long line length. Continuations of the same line are indented.

For all of the below conditions, an important field is the alert field. What is to the right of the right hand tilde defines who will be paged, emailed, or the program which will be started. "E" indicates email, "P" indicates page, and "S" indicates start a program.

```
#WATCH LOGFILE
watch logfile 35 0x0 /var/adm/sulog 'root' 'User has used su to root'
    ~WL~Ewebmasters:Eoncall:' '/usr/local/bin/sysedgeaction /local/etc/SystemEdge-
    suroot.txt'
```

Fields description: watch, logfile, index, flags, logfile-to-watch, regular-expression-to-match, alert field, file-to-pass-to-program (Concord, p. 12-10, 12-11).

Alert field description: Error-message, delimiter-tilde, action-code, delimiter-tilde, list of one or more to email or page, program-to-invoke(sysedgeaction).

In this case, "webmasters" and "oncall" will be emailed the alert.

```
# WATCH PROCESS
watch process procAlive 'cron' 51 0x00100402 60 'The cron process died; restarted
    by SystemEdge~WP~S/etc/rc2.d/S75cron start:' '/usr/local/bin/sysedgeaction
    /usr/local/etc/SystemEdge-cron.txt'
```

Fields description: watch, process, proctype, process-name, index, flags, check-every-seconds, alert field, file-to-pass-to-program (Concord, p. 10-19 to 10-22).

Alert field description: Error-message, delimiter-tilde, action-code, delimiter-tilde, list of one or more to email or page, program-to-invoke(sysedgeaction)

In this case, noone will be emailed or paged, but the cron process will be automatically restarted.

```
# MONITOR SNMP
monitor oid snmpInBadCommunityNames.0 90 0x00100400 600 delta > 5 '5 SNMP
    Hack Attempts~MO~Eunixadmins:Eoncall:' '/usr/local/bin/sysedgeaction
    /usr/local/etc/SystemEdge-snmphack.txt'
```

Fields description: monitor, oid, snmpOID, index, flags, check-every-seconds, delta, operator, value, logfile-to-monitor, regular-expression-to-match, alert field, file-to-pass-to-program (Concord, p. 9-14 to 9-16).

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Alert field description: Error-message, delimiter-tilde, action-code, delimiter-tilde, list of one or more to email or page, program-to-invoke(sysedgeaction)

In this case the “unixadmins” and “oncall” will be emailed.

```
# MONITOR FILESYSTEM
monitor filesystem '/' devCapacity 130 0x00100402 120 absolute >= 80 'Notice: Root
    filesystem is >= 80% full~MF~Eoncall:Eunixadmins:' '/usr/local/bin/sysedgeaction
    /usr/local/etc/SystemEdge-root.txt'
```

Fields description: monitor, filesystem, filesystem-name, parameter, index, flags, seconds, absolute, operator, value, alert field, file-to-pass-to-program (Concord, p. 9-3 to 9-5).

Alert field description: Error-message, delimiter-tilde, action-code, delimiter-tilde, list of one or more to email or page, program-to-invoke(sysedgeaction)

In this example, “oncall” will be emailed and “unixadmins” will also be emailed.

E. Specific Examples of Monitored Conditions in Our Environment

Every Solaris server that is monitored by SystemEdge has monitored conditions that are the same on every server. In addition, each server has specific conditions that need to be monitored, based on the function/application of the server.

1. General Conditions Monitored on All Solaris Servers

The following section has the common SystemEdge conditions that are monitored on each server. An explanation of these conditions follows this section.

a. General Conditions Configuration File

```
# WATCH LOGFILE
watch logfile 10 0x0 /var/adm/messages '.*fail ' 'Syslog logged a fail
    message~WL~Eunixadmins:Eoncall:' '/usr/local/bin/sysedgeaction /tmp/SystemEdge-
    msg.txt'
watch logfile 12 0x0 /var/adm/messages '.*SCSI' 'Syslog logged a SCSI
    error~WL~Eunixadmins:Eoncall:' '/usr/local/bin/sysedgeaction /tmp/SystemEdge-
    msg.txt'
watch logfile 13 0x0 /var/adm/messages '.*fatal' 'Syslog logged a fatal
    message~WL~Punixadmins-pg:Eunixadmins:Eoncall:Poncall-pg:'
    '/usr/local/bin/sysedgeaction /tmp/SystemEdge-msg.txt'
watch logfile 14 0x0 /var/adm/messages '.*panic' 'Syslog logged a panic
    message~WL~Punixadmins-pg:Eunixadmins:Eoncall:Poncall-pg:'
    '/usr/local/bin/sysedgeaction /tmp/SystemEdge-msg.txt'
watch logfile 35 0x0 /var/adm/sulog '.*su root.*failed.*’ 'A failed su attempt was logged’
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
    ~WL~Ewebmasters:Eoncall:' '/usr/local/bin/sysedgeaction /local/etc/SystemEdge-
    suroot.txt

# WATCH PROCESS
watch process procAlive 'cron' 51 0x00100402 60 'The cron process died; restarted
    by SystemEdge~WP~S/etc/rc2.d/S75cron start:' '/usr/local/bin/sysedgeaction
    /usr/local/etc/SystemEdge-cron.txt'

# MONITOR SNMP
monitor oid snmpInBadCommunityNames.0 90 0x00100400 600 delta > 5 '5 SNMP
    Hack Attempts~MO~Eunixadmins:Eoncall:' '/usr/local/bin/sysedgeaction
    /usr/local/etc/SystemEdge-snmphack.txt'

# MONITOR FILESYSTEM
monitor filesystem '/' devCapacity 130 0x00100402 120 absolute >= 80 'Notice: Root
    filesystem is >= 80% full~MF~Eoncall:Eunixadmins:' '/usr/local/bin/sysedgeaction
    /usr/local/etc/SystemEdge-root.txt'
monitor filesystem '/usr' devCapacity 131 0x00100402 120 absolute >= 80 'Notice:
    /usr filesystem is >= 80% full~MF~Eoncall:Eunixadmins:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-usr.txt'
monitor filesystem '/var' devCapacity 132 0x00100402 120 absolute >= 80 'Notice:
    /var filesystem is >= 80% full~MF~Eoncall:Eunixadmins:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-var.txt'
monitor filesystem '/local' devCapacity 134 0x00100402 120 absolute >= 80 'Notice:
    /local filesystem is >= 80% full~MF~Eoncall:Eunixadmins:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-local.txt'
monitor filesystem '/opt' devCapacity 135 0x00100402 120 absolute >= 80 'Notice:
    /opt filesystem is >= 80% full~MF~Eoncall:Eunixadmins:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-opt.txt'

monitor filesystem '/' devCapacity 136 0x00100400 120 absolute >= 95 'Warning:
    Root filesystem is >= 95% full~MF~Poncall-pg:Punixadmins-
    pg:Eoncall:Eunixadmins:' '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-
    root.txt'
monitor filesystem '/usr' devCapacity 137 0x00100400 120 absolute >= 95 'Warning: /usr
    filesystem is >= 95% full~MF~Poncall-pg:Punixadmins-pg:Eoncall:Eunixadmins:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-usr.txt'
monitor filesystem '/var' devCapacity 138 0x00100400 120 absolute >= 95 '
    Warning: /var filesystem is >= 95% full~MF~Poncall-pg:Punixadmins-
    pg:Eoncall:Eunixadmins:' '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-
    var.txt'
monitor filesystem '/local' devCapacity 140 0x00100400 120 absolute >= 95 'Warning:
    /local filesystem is >= 95% full~MF~Poncall-pg:Punixadmins-
    pg:Eoncall:Eunixadmins:' '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-
    local.txt'
monitor filesystem '/opt' devCapacity 141 0x00100400 120 absolute >= 95 'Warning: /opt
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

filesystem is >= 95% full~MF~Poncall-pg:Punixadmins-pg:Eoncall:Eunixadmins:'
'/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-opt.txt'

b. Explanation of Monitored General Conditions

The configuration file above has one line (row) for each alert condition to be monitored. Each row has a fixed, space delimited field syntax. Lines are wrapped in the below configuration file due to long line length; continuations of the same line are indented.

It was necessary to pass multiple arguments to the program. These included the name of a text file describing the alert condition as well as the email addresses to contact for a given alert. To accomplish this the arguments were encoded within the standard SystemEdge configuration file syntax, which was then passed to my perl program. This was an extension of the commercial product's configuration file.

Many of these entries in the configuration file provide security monitoring for the host or are security related. Starting from the top of the file, I will explain the entries:

Under "WATCH LOGFILE" the standard syslog logfile (/var/adm/messages) will be continually monitored. If any of the following: fail, fatal, SCSI or panic appear in the file, an alert will be generated. There are wide varieties of system conditions, which may contain these strings, and they are always serious system warnings. When such a message is received, it will need to be evaluated by the system administrator as in some cases it may be security related.

The last line under "WATCH LOGFILE" monitors the syslog file for failed attempts to use the unix su program. This program is used to become another user, most frequently the "root" or privileged user (Sun, su, p.1). Should an internal user on the system be attempting to exceed his authority by guessing passwords using this command, the failures will be logged and the system administrators will be emailed.

The next section is "WATCH PROCESS". This section contains one entry that is used to monitor the "cron" process. The cron process serves as the scheduler for batch jobs that start in the future (Sun, cron, p.1). As this is a critical service, SystemEdge will monitor the process table and should it detect that this process has died, it will automatically restart it.

The next section is "MONITOR SNMP". The SystemEdge daemon has an embedded SNMP agent that allows for read access by specific network management hosts (Concord, 1-1). However, I want to insure that hackers are not attempting to access our system by brute force SNMP string attacks (CERT, SNMP, p.1). This entry will email the system administrator and page the oncall person should such an attempt occur.

The last section is "MONITOR FILESYSTEM". The monitor filesystem entries are used for both routine system administration and security administration. System administrators want to know the percentage of used capacity on their filesystems to

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

prevent running out of space. For the security administrator, a rapidly increasing filesystem could indicate various problems. First, it could be indicative of a large amount of logging, for example if a systematic attempt to access prohibited areas on a web server. Another example would be if the system was compromised by a hacker, file storage might have been allocated for illicit purposes like a warez software server. Should the set threshold of 80% be exceeded, emails are sent whereas if 95% of the disk storage is exceeded, both and email and page are sent due to the urgency of the situation.

2. DNS Server Monitored Conditions

We maintain DNS servers that are a critical service to our network and therefore require specific monitoring. Following is the configuration file; an explanation follows this section.

a. DNS Server Specific Conditions Configuration File

```
#WATCH LOGFILE
watch logfile 200 0x0 /var/adm/messages '.*rejected due to' 'Syslog logged a
    nameserver zone load failure~WL~Eunixadmins@orator.state.mn.us:Eoncall:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-syslog-reject.txt'
watch logfile 201 0x0 /var/adm/messages '.*data error' 'Syslog logged a nameserver
    load error message~WL~Eunixadmins@orator.state.mn.us:Eoncall:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-syslog-dataerror.txt'
watch logfile 202 0x0 /var/adm/messages '.*empty label' 'Syslog logged a nameserver
    load error message~WL~Eunixadmins@orator.state.mn.us:Eoncall:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-syslog-dataerror.txt'

#WATCH PROCESS
watch process procAlive 'named' 52 0x00100402 30 'The named (DNS) process died;
    restarted by SystemEdge~WP~S/usr/local/sbin/named -u 60001:'
    '/usr/local/bin/sysedgeaction /usr/local/etc/SystemEdge-named.txt'
```

c. Explanation of DNS Server Monitored Specific Conditions

SystemEdge is used on our DNS Servers to monitor for specific DNS application related problems.

The first three entries above all refer to the same problem, namely that the DNS server detected a problem with the DNS zone configuration files that needs to be corrected. These conditions always need to be corrected, as DNS will not be correct until they are. While typically they are caused by administrator errors in editing zone files, it could be possible that a hacker has gained access to the DNS zone files and has entered an invalid entry.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The next entry indicates that someone is attempting to do a zone transfer without permission of the server. As we prohibit zone transfers except to specific hosts, these requests are denied.

The last entry monitors the process "named" which is the DNS daemon process that runs DNS. As this is a critical process, it is necessary to restart it if it fails. The administrators are also emailed if this occurs. This is useful as named has historically had security problems which caused it to die (CERT, BIND, p.1). This can serve as advance warning for the administrator that hacking attempts are being made against DNS and that a newer release of the daemon may be indicated to fix the instability.

3. Apache Web Server Monitored Conditions

We maintain Apache web servers which that require specific monitoring. These servers require a user to authenticate to access material stored within them

a. Apache Server Specific Conditions Configuration File

I have SystemEdge entries that monitor the Apache logfiles for failed logins and improper access attempts. Following are the specific entries that I use for Apache web servers. An explanation follows these lines:

```
#WATCH LOGFILE
watch logfile 31 0x0 /local/httpd/logs 'password mismatch' 'Apache Web Server
    logged an authentication failure~WL~Ewebmasters:' '/usr/local/bin/sysedgeaction
    /local/etc/SystemEdge-authfail.txt'
watch logfile 32 0x0 /local/httpd/logs/error_log 'user.*not found' 'Apache Web
    Server logged an authentication failure~WL~Ewebmasters:Eoncall:'
    '/usr/local/bin/sysedgeaction /SystemEdge-authfail.txt'
watch logfile 33 0x0 /local/httpd/logs/error_log 'Permission denied' 'Apache Web
    Server logged an authentication failure~WL~Ewebmasters:Eoncall:'
    '/usr/local/bin/sysedgeaction /SystemEdge-authfail.txt'
watch logfile 34 0x0 /local/httpd/logs/error_log 'client denied' 'Apache Web Server
    logged an authentication failure ~WL~Ewebmasters:Eoncall:'
    '/usr/local/bin/sysedgeaction /SystemEdge-authfail.txt'
```

b. Explanation of Monitored Apache Server Specific Conditions

In the first case, the user entered a bad password. In the second case, a non-existent username was used to gain access. The remaining two lines are attempts by the user to access forbidden areas of the web server filesystem. If any of these four conditions are logged, both the system administrators and the webmasters will be emailed. In some cases, the problem will be user mistakes, and in others, it may indicate a hacking attempt.

V. Results of Security Enhancements

© SANS Institute 2000 - 2002, Author Retains Full Rights

The first step I undertook in this security project was to identify the risks, which were present for the systems. These were determined to be medium to high depending on the function of the system in question.

The next step was to automate the monitoring and alerting of the system by using a commercial product and writing a perl program that performs the actual alerting.

The results that I and other system administrators have received have been considerable.

All filesystems are continuously monitored such that they do not exceed capacity thresholds. The systems are monitored for SNMP probing attacks. Critical daemons are restarted if they fail, and the admin is notified of this event, which could be a security attack against the application, such as a port overflow. System logfiles are monitored for users exceeding their authority as well as for other critical syslog events. Lastly, application logfiles, such as for DNS or web servers, are monitored for security events.

Timely notification of potential security problems has proven to put us in a more prepared, proactive position, whereas before we were in a reactive/after the fact mode of dealing with problems.

Alerts from the abovementioned conditions have prevented more serious security incidents from occurring in that we had timely notice of problems. For example, restarting daemons are indicative of possible port overflow problems that we can remedy with system patches. Authentication problems that arise we have found are typically internal staff matters that can be dealt with by the administrator.

Lastly, automating of monitoring and alerting has saved considerable staff time. This is not only a savings of the time needed to perform these security tasks manually but from a management point of view an opportunity to utilize the time for other important work.

Coupled with best practice system and security administration, these improvements have greatly improved our security readiness and response.

VI. Author's Perl Program and Supporting Files

A. Sysedgeaction Perl Code

```
#!/bin/perl
#
# Program: sysedgeaction
# Purpose: program to perform alerting for SystemEdge
# By:      Dan Bick
#
#
#
#------------------------------------------------------------
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
#
# NOTES
#
# ---------------
#
# Description_flag field is limited to 130 characters
#
# Format example:
# 'sendmail dead~WP~S/usr/lib/sendmail:Eadmins^sendmaildead:'
#
# WP  Watch Process
# WL  Watch logfile
# MO  Monitor SNMP oid
# MF  Monitor filesystem
#
# Action codes
#
# S  start process
# E  email alert
# P  page
#
# ---------------
#


    $logfile="/var/log/se.log";

    open(LOG, ">>$logfile") || die "sysedgeaction: can't open log file\n";

if ($#ARGV < 1)
    {
    exit;
    print LOG "SYSEDGE: failed -- no arguments passwd to me\n";
    }

#
# define initial variables and check for arguments
#
    @cmds="";

    $command_row="";
    $current_value=0;
    $curr_action=0;
    $description_flag="";
    $i=0;
    $length_arg=0;
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
    $oid="";
    $operator=0;
    $program_arg1="";
    $row=0;
    $row_status=0;
    $message_file="";
    $this_action_code="";
    $this_action_arg="";
    $threshold_value=0;
    $trap_type=0;
    $type="";
    $hostname=`hostname`;


$_=$ARGV[2]; if (/\~/) { $args=2; }
$_=$ARGV[3]; if (/\~/) { $args=3; }
$_=$ARGV[6]; if (/\~/) { $args=6; }

$command_row=$ARGV[$args];

#
# get type code
#

    @cmds = split(/\~/, $command_row);
    $type=$cmds[1];

if ($type == "WL") {

    $program_arg1=$ARGV[0];
      $trap_type=$ARGV[1];
      $log_monitored=$ARGV[2];
      $match_regex=$ARGV[3];
      $matched_line=$ARGV[5];
      $description_flag=$ARGV[6];
      $row=$ARGV[7];
      $hexflags=$ARGV[8];

    system ("rm /tmp/sys-tmp.txt");
    system ("cat /usr/local/etc/sysedge-syslog-generic.txt > /tmp/sys-tmp.txt");
    open(TMP, ">> /tmp/sys-tmp.txt") || die "sysedgeaction: can't open tmp
    file\n";
    print TMP $matched_line;
    print TMP "\n\n";
    close (TMP)
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
                }

elsif ($type == "WP") {

    $program_arg1=$ARGV[0];
    $trap_type=$ARGV[1];
    $description_flag=$ARGV[3];
    $hexflags=$ARGV[8];
                }

elsif ($type == "MO") {

    $program_arg1=$ARGV[0];
    $trap_type=$ARGV[1];
    $description_flag=$ARGV[2];
    $oid=$ARGV[3];
    $current_value=$ARGV[4];
    $threshold_value=$ARGV[5];
    $row_status=$ARGV[6];
    $operator=$ARGV[7];
    $row=$ARGV[8];
                }

elsif ($type == "MF") {

    $program_arg1=$ARGV[0];
    $trap_type=$ARGV[1];
    $description_flag=$ARGV[2];
    $oid=$ARGV[3];
    $current_value=$ARGV[4];
    $threshold_value=$ARGV[5];
    $row_status=$ARGV[6];
    $operator=$ARGV[7];
    $row=$ARGV[8];
                }
else {
    print LOG "No valid type code, exiting\n";
    exit 1;
}

    @action = split(/\:/, $cmds[2]);

    $message_file=$ARGV[0];
    $this_action_code=substr($action[0],0,1);
    $length_arg=length($action[0]); $length_arg--;
    $this_action_arg=substr($action[0],1,$length_arg);
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
    $curr_action=0;


while ($length_arg > 0) {

 for ($this_action_code) {
                /P/ && do { &Page; last; };
                /S/ && do { &Start; last; };
                /E/ && do { &Email; last; };
                                }

    $curr_action++;
    $this_action_code = substr($action[$curr_action],0,1);
    $length_arg = length($action[$curr_action]); $length_arg--;
    $this_action_arg = substr($action[$curr_action],1,$length_arg);

}

#
#================ S U B R O U T I N E S =========
#


sub Page
{

if ($type eq "WP" )
 {

if ($trap_type==10) {

    system ("mailx -s 'SYSEDGE Alert: host: $hostname $cmds[0]'
    $this_action_arg  < /dev/null");
     print LOG "SYSEDGE(E): page sent to ($this_action_arg) because:
    [host: $hostname] $cmds[0]\n";

            }
}


else
 {
    system ("mailx -s 'SYSEDGE Alert: host: $hostname $cmds[0]'
     $this_action_arg  < /dev/null");
      print LOG "SYSEDGE(E): page sent to ($this_action_arg) because:
      [host: $hostname] $cmds[0]\n";
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
 }
}

sub Start {

if ( $trap_type == 10 )
{
print LOG $trap_type;
     system ("$this_action_arg &");
       print LOG "SYSEDGE(S): $this_action_arg was automatically restarted
       because the process died\n";
}
}

sub Email
{

if ($type eq "WP" )
 {

if ($trap_type==10) {

  if ( -r "$ARGV[0]" )      {
    system ("mailx -s 'SYSEDGE Alert: host: $hostname $cmds[0]'
    $this_action_arg  < $ARGV[0]");
                    }
    else                {
    system ("mailx -s '[host: $hostname] $cmds[0]'  < /dev/null");
                    }

      print LOG "SYSEDGE(E): email sent to ($this_action_arg) because:
    [host: $hostname] $cmds[0]\n";
                }
 }

else
 {

    if ( -r "$ARGV[0]" ) {
    system ("mailx -s 'SYSEDGE Alert: host: $hostname $cmds[0]'
   $this_action_arg  < $ARGV[0]");
    }
    else {
    system ("mailx -s '[host: $hostname] $cmds[0]'  < /dev/null");
    }
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
        print LOG "SYSEDGE(E): email sent to ($this_action_arg) because:
       [host: $hostname] $cmds[0]\n";
  }
 }
```

B. Sample Alert Files

The examples below are text messages which are stored on each server. They are emailed to the designated address and contain a description of the alert condition.

1. Watch Process (Restart BIND named Daemon)

This is an automated description of an event
condition detected by the sysedge monitor.

Condition:

The named daemon, which is a system process which always runs on the system is no longer in the process table (i.e. it died). This process does DNS Mnet wide.

Suggested Resolution:

Sysedge restarted this process automatically. If you want to verify this, log into the system and enter:

ps -ef | grep named

If the process has been restarted, you will see a line similar to this:

root 227 1 0 Apr 13 ? 0:04 named

2. Monitor SNMP (SNMP Hack Attempt)

This is an automated description of an event
condition detected by the sysedge monitor.

Condition:

Authentication has failed too many times for SNMP queries

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

to this system. This could indicated that someone/something is attempting to gain access to this system.

Suggested Resolution:

Examine /var/adm/messages and Openview and attempt to identify the source of the SNMP queries to determine their legitimacy.

3. Monitor Filesystem (/usr Filesystem)

This is an automated description of an event condition detected by the sysedge monitor.

Condition:

The /usr filesystem has exceeded a predetermined limit. If this filesystem fills up system problems may occur.

Suggested Resolution:

Examine the files on this filesystem in order to free up space by removing or trimming files.

The command to list which files have changed recently is:

find /usr -mtime -3 -ls

This will give you a listing of the files which have changed within the last 3 days.

Note: DO NOT delete files unless you are sure of their function.

4. Watch Logfiles (fatal Error)

This is an automated description of an event condition detected by the sysedge monitor.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Condition:

The system has submitted an entry to syslog with
indicating a failed attempt to "su" in the message. This means
you may have had a possible attempt to compromise security.

Suggested Resolution:

If possible log into the system and look at the /var/adm/messages
file to find recent messages with "su" in them.
Isolate the cause to the user and determine the
corrective action.

VII. References

The Apache Software Foundation. "Apache HTTPD Server Project". 1999-2002. URL: http://httpd.apache.org (2 Oct 2002).

Cisco Systems. "Network Security Policy: Best Practices White Paper". 9 April 2002. URL: http://www.cisco.com/warp/public/126/secpol.html (3 Oct 2002).

Cert Coordination Center, Rogers, Larry. "Buffer Overflows – What Are They and What Can I Do About Them?". 3 Dec 2001.
URL: http://www.cert.org/homeusers/buffer_overflow.html (2 Oct 2002).

Cert Coordination Center, "Simple Network Management Protocol (SNMP) Vulnerabilities Frequently Asked Questions (FAQ)". 13 Feb 2002. URL: http://www.cert.org/tech_tips/snmp_faq.html (22 Oct 2002).

Cert Coordination Center. "Understanding system log files on a Solaris 2.x operating system".
URL: http://www.cert.org/security-improvement/implementations/i041.12.html
(2 Oct 2002).

Todd Atkins. "Swatch: the Simple WATCHdog". 14 June 2000. URL: ftp://coast.cs.purdue.edu/pub/tools/unix/logutils/swatch/swatch-3.0b5.tar.Z/README (2 Oct 2002).

Miller, Jean C. "Risk Management for Your Web Site". September 2000. URL: http://www.irmi.com/expert/articles/schoenfeld003.asp (2 Oct 2002).

US Department of Energy Computer Incident Advisory Capability (CIAC). "K-036: Continuing Compromises of DNS Servers". 28 April 2000. URL:

© SANS Institute 2000 - 2002, Author retains full rights.

http://www.ciac.org/ciac/bulletins/k-036.shtml.

Sans Institute. Security Essentials Risk Management and Auditing. 2002. Pages 6-1 to 6-12, 6-27 to 6-55.

Sans Institute. Security Essentials Unix Security Step-by-Step. 2002. Pages 3-1 to 3-14, 4-29 to 4-30.

Sun Microsystems. newaliases(1) man page, SunOS 5.8. 11 Aug 1998.

Sun Microsystems. syslogd(1M) man page, SunOS 5.8. 11 May 1999.

Sun Microsystems. su(1M) man page, SunOS 5.8. 27 Jan 2000.

Sun Microsystems. cron(1M) man page, SunOS 5.8. 1 Mar 1994.

Nemeth, Evi; Synder, Garth; Seebass, Scott; Hein, Trent R. Unix System Administration Handbook. Upper Saddle River: Prentice Hall PTR, 1995.

Concord Communications, SystemEdge Agent User Guide 4.0 Patchlevel 2. Marlboro: Concord Communications, 2000.

Wall, Larry and Schwartz, Randal L. Programming Perl. Sebastopol: O'Reilly & Associates, Inc.

Pomeranz, Hal. Unix Security: Step-by-Step. Oakland: Deer Run Associates.

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.