Global Information Assurance Certification Paper

Enhancing Information Security in a Complex Environment: a Case Study

GSEC Practical Version 1.4b

Author: Marcia Goetsch
Date: April 11, 2004

Abstract

Enhancing the information security posture of an organization can be complicated by the structure and culture of the organization itself. This paper will explore designing and implementing security in a laboratory embedded within a larger organization. The security posture of the laboratory is open while the security posture of the larger organization is closed. This leads to a tension where the two infrastructures intersect, and information security design and implementation will be constrained by this tension.

This case study describes how information security was enhanced in the laboratory and how the complex environment shaped both the design and the implementation. A risk assessment is undertaken and the recommendations are shaped by this structure. During the implementation phase the design is further modified due to infrastructure tensions.

Background

The Laboratory has 800 users, an IT department supporting most of the desktops and all of the servers used by these users. I am the manager and lead system administrator of this IT department. I report to a Computer Board with representatives from each of the Laboratory's major groups, the director and other officers. The Organization has an IT department supporting 20,000 desktops, servers and the network infrastructure. There are another 20,000 desktops controlled by the Nonstandard groups embedded within the Organization. Communication between the IT department of the Laboratory and the IT department of the Organization is minimal. I want to enhance the security of the computer systems of the Laboratory within the network of the Organization.

Before

Physical Layout

The Laboratory is housed in a primary location consisting of half of an eight floor building, three laboratories in an adjoining building, part of a floor in a building located a mile away and a floor in a building located 20 miles away. The primary location is entered by swiping an ID card through a card reader or by buzzer to the receptionist desk. The other locations are entered by swiping an ID card or by asking the security card to call an employee. The datacenter is located at the primary location in two adjacent locked air conditioned rooms with UPS and temperature monitoring. The building at the primary location has emergency electrical generators in the event of a power outage.

Network Layout

The computer systems of the Laboratory are on a network shared with the Nonstandard groups operated by the IT department of the Organization. The Organization has implemented router based packet filters (ACLs) on this network with a default deny stance and permit exceptions to this stance as needed by the Nonstandard groups including the Laboratory. All machines at the primary location and laboratories in adjacent building share a VLAN (HOME) with the Nonstandard groups. The machines in the building located one mile away have a network (NEAR) connected to the Organization via microwave. The machines at the location 20 miles away share a network (FAR) with the Nonstandard groups in their building and are connected to the Organization network.

Host Layout

Desktops are Windows 2000 and Solaris 8. There are a handful of Macintosh OSX and Linux Redhat 9 desktops. Servers are Solaris 8 and 9 and Linux Redhat 7.2 and 8. All servers are located in the datacenter at HOME except the two in the datacenter at NEAR. Desktops use Windows Automatic Update for

patching. Solaris machines are patched quarterly with critical security patches applied as soon as they become available. Linux machines have critical security patches applied as they become available. Windows 2000 desktops use TCP filters to block incoming TCP. Solaris machines use TCP wrappers to restrict inetd services to HOME, NEAR and FAR. Host based intrusion detection is in place using the commercial version of Tripwire for the Solaris machines. Network intrusion detection is simulated by using snort with 7 sensors on 7 machines distributed on HOME. The network is switched, so this is not a comprehensive look at the network. The Solaris and Linux servers and services are monitored for responsiveness using nagios. Email and pages are sent to appropriate staff if a machine or service doesn't respond.

Application Layout

Internet facing services from anywhere are remote shell access and file transfer via SSH to an internal login server, web access via Apache on a dozen virtual interfaces on a bastion host webserver, incoming email via Postfix on two bastion hosts (one at HOME and one at NEAR), and webmail via OpenSSL on Apache on a bastion host. Disaster Recovery is limited to offsite copies of all tape backups. OpenSSH uses password authentication. There is no password policy.

Internal services are filesharing via NFS and Samba from two fileservers, Sendmail and IMAP for email, LPD print serving, Oracle, Sun Grid Engine for scheduling batch computational jobs, an intranet with Apache, a Legato backup server, NIS, LDAP, BOOTP, TFTP, NTP and DNS (BIND).

Info Layout

All data is on one of two fileservers one of which is also the Oracle server and is shared out via NFS. Users login to the login server locally and remotely. Access to data is controlled by ACLs on the files or Oracle permissions. The data is backed up with incremental backups nightly to tape and with weekly incremental and monthly full backups kept indefinitely. These backups are cloned and the second copy is sent offsite to a professionally managed tape storage facility.

Risk Assessment

I performed a risk assessment[1] of the Laboratory computer facility. I presented this assessment to the Computer Board of the Laboratory overseeing the computing facility so that I had the permission and support I needed.

Mission Statement

> "Ensure the confidentiality, integrity and availability of The Laboratory computer system."

What are the assets?

- Data

- Intellectual property
- Computer system availability
- Reputation

What are the risks?

I used the risk definition: Risk=Threat x Vulnerability(TO THAT THREAT) x Impact.
Initially, I classified the risks into these groups:

- Physical threats
  - Fire
  - Water damage
  - Electrical outage
  - Heat
  - Theft
  - Vandalism
  - Terrorism
- Accidents and mistakes
  - Misconfigurations
  - Unpatched vulnerabilities
  - Hardware failures
- Insider attacks
  - Attacks within network
  - Disgruntled employee
- Attacks from external adversaries
  - Script kiddies
  - Hackers
  - Criminals
  - People with intent to discredit the Laboratory
  - Vandals
  - Terrorists

I prioritized these risks into groups based on a risk analysis matrix of probability of likelihood versus severity of consequence, focusing on those with highly or somewhat likelihoods and severe or more moderate consequences:

- Highly likely risks with severe consequences. I concluded that there were none in this category.
- Somewhat likely risks with severe consequences. These are those with potential total loss of the entire computer system.
  - Fire
  - Water damage
  - Terrorism
  - Disgruntled employee
- Highly likely risks with more moderate consequences. These are those with loss of some but not all hardware, software or integrity of host.
  - Electrical outage
  - Misconfigurations

    - Unpatched vulnerabilities
    - Hardware failures
    - Script kiddies
    - Hackers
    - People with intent to discredit
    - Attacks within network
- Somewhat likely risks with more moderate consequences.
    - Theft
    - Vandalism
    - Criminals

Vulnerabilities and attacks are a highly likely concern. We have ample evidence from our network intrusion detection system of scans of our network. So we know that the threats are there. Additionally, looking at the twenty most critical internet security vulnerabilities (http://www.sans.org/top20/)[2], although we don't run most of the identified Windows vulnerable services, we do share HOME with the Nonstandard groups which do run these services within our network perimeter. And we run almost all of the identified UNIX vulnerable services. There is also concern due to the nature of the vulnerability exploit cycle itself. Kevin O'Shea in his paper "Examining the RPC DCOM Vulnerability: Developing a Vulnerability-Exploit Cycle,"[3] outlined the cycle of a vulnerability and exploit through vulnerability: birth, discovery, disclosure, fix and exploit: publication, use, automation and finally mitigation and exploit death. But with this cycle repeated for different vulnerabilities, on any given day there is some vulnerability at the height of its virulence. Finally, at least twice in the last two months I have found that the router ACLs were not correctly implemented and our internal machines were accessible from outside our network, thus making the vulnerabilities we have that much more exposed.

Existing countermeasures

Now that I had identified and prioritized the risks, I turned to the countermeasures to these risks that were already in place:

- Physical threats
    - Offsite tape backups of all data: weekly backups are cloned and a copy sent offsite to a professionally managed tape storage facility 20 miles away.
    - Locked computer rooms: locks are mechanical locks.
    - UPS and emergency generators.
    - Air conditioners: two in each room, one is running the other a backup.
    - Temperature monitoring: alarm goes to alarm company and that company calls our staff.
- Accidents, mistakes and failures
    - System administration training.
    - Regular patching schedule.
    - Hardware replacement schedule.

    - 24/7 Software and hardware support contracts.
    - Computer monitoring: using nagios and orca.
    - Tape backups: incremental backups nightly, full backups monthly, weekly snapshots are retained indefinitely.
- Attacks
    - Network packet filters: ACLs on the routers.
    - Central logging: both logging to local files and to a central syslog server.
    - Network intrusion detection system: using snort and acid.
    - Host based network intrusion detection: using tripwire.
    - Anti-virus software: on Windows desktops using Norton Anti-Virus.
    - Email sanitizing of dangerous attachments: using anomy sanitizer.
    - Encryption: using OpenSSL, and NT LAN Manager.
    - Tape backups.

Recommendations

My recommendations for reducing risk were:

- Strong Password policy. Since password authentication is used for remote access, a strong password policy is crucial.
- Vulnerability scanning. We are being scanned, we are running many exploited services, and our network perimeter defense is inadequate. Remediation of vulnerabilities as quickly as possible is a good defense.
- Host based packet filters as a form of distributed firewall.[4] Our network perimeter defenses may not be robust and some threats are internal to that perimeter. Host based packet filters can mitigate both types of situations and provide defense in depth.
- Security and benchmark scoring of configurations. As with vulnerability scanning a proactive stance to securing our services and hosts is a good defense.
- Bastion login server. Moving remote access to a bastion host rather than an internal server will make securing that host easier, since it will be running fewer services and can be placed on a DMZ.
- Disaster recovery plan. This will mitigate many of the risks with severe consequences. Cost is a significant factor.
- Firewall with NAT for internal servers, DMZ for internet facing servers. This, as with host based packet filtering, will provide another layer of defense.

Implementation of Recommendations

Password policy

I began with the Password Protection Policy available at http://www.sans.org/resources/policies/[5] and modified it for the Laboratory. I emailed this policy to all users and added it to the computer system FAQ on the

intranet. This policy is now given to new users as part of the new account process. I modified the default password configuration to require a minimum password length of 8 characters on all UNIX systems. I downloaded the password cracker John the Ripper from http://www.openwall.com/john/[6] and configured it to enforce the policy. I sent an email to all users informing them that I would be running a password cracker and that the passwords of all guessed passwords would be expired. John cracked 5% of the passwords and I expired the passwords. Several weeks later I ran John again.

Vulnerability Scanning

For vulnerability scanning I chose the open source package nessus available from http://nessus.org/.[7] This package provides basic vulnerability scanning, updated often and for the initial implementation is good value for cost, since it is free. We may want to investigate commercial products which offer administrative advantages at a later date, but for now this will get us started. I compiled and installed it, updated the plugins and ran scans on a representative UNIX desktop node. The initial scan was:

```
Nessus Scan Report
------------------

SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 5
 - Number of security warnings found : 12
 - Number of security notes found : 23

TESTED HOSTS

 200.200.3.66 (Security holes found)

DETAILS

+ 200.200.3.66 :
 . List of open ports :
   o time (37/tcp) (Security notes found)
   o daytime (13/tcp) (Security warnings found)
   o dtspc (6112/tcp) (Security hole found)
   o unknown (111/tcp) (Security notes found)
   o general/tcp (Security notes found)
   o unknown (32788/tcp) (Security hole found)
   o unknown (4045/tcp) (Security notes found)
   o unknown (32775/tcp) (Security notes found)
   o unknown (32777/tcp) (Security notes found)
   o unknown (38550/tcp) (Security notes found)
   o sunrpc (111/udp) (Security notes found)
   o lockd (4045/udp) (Security warnings found)
   o unknown (32779/udp) (Security warnings found)
   o unknown (32785/udp) (Security warnings found)
   o unknown (32790/udp) (Security warnings found)
   o unknown (32795/udp) (Security warnings found)
```

```
  o unknown (32798/udp) (Security hole found)
  o unknown (22/tcp) (Security warnings found)
  o unknown (79/tcp) (Security warnings found)
  o unknown (514/tcp) (Security warnings found)
  o daytime (13/udp) (Security warnings found)
  o unknown (177/udp) (Security warnings found)
  o general/udp (Security notes found)
  o unknown (32798/tcp) (Security hole found)

. Information found on port time (37/tcp)

   A time server seems to be running on this port

. Warning found on port daytime (13/tcp)

   The remote host is running a 'daytime' service. This service
   is designed to give the local time of the day of this host
   to whoever connects to this port.

   The date format issued by this service may sometimes help an attacker
   to guess the operating system type of this host, or to set up
   timed authentication attacks against the remote host.

   In addition to that, the UDP version of daytime is running, an attacker
   may link it to the echo port of a third party host using spoofing, thus
   creating a possible denial of service condition between this host and
   a third party.

   Solution :

   - Under Unix systems, comment out the 'daytime' line in /etc/inetd.conf
     and restart the inetd process

   - Under Windows systems, set the following registry keys to 0 :
     HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpDaytime
     HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpDaytime

   Then launch cmd.exe and type :

      net stop simptcp
      net start simptcp

   To restart the service.

   Risk factor : Low
   CVE : CVE-1999-0103

. Vulnerability found on port dtspc (6112/tcp) :

   The 'dtspcd' service is running. This service deals with
   the CDE interface for the X11 system.

   Some versions of this daemon are vulnerable to a buffer overflow
   attack which may allow an attacker to gain root privileges on
   this host.

   *** This warning might be a false positive,
```

```
    *** as no real overflow was performed

    Solution : See http://www.cert.org/advisories/CA-2001-31.html
    to determine if you are vulnerable or deactivate this service
    (comment out the line 'dtspc' in /etc/inetd.conf and restart the inetd
     process)

    Risk factor : High
    CVE : CVE-2001-0803
    BID : 3517
    Other references : IAVA:2002-A-0001

. Information found on port unknown (111/tcp)

    The RPC portmapper is running on this port.

    An attacker may use it to enumerate your list
    of RPC services. We recommend you filter traffic
    going to this port.

    Risk factor : Low
    CVE : CAN-1999-0632, CVE-1999-0189
    BID : 205

. Information found on port unknown (111/tcp)


    RPC program #100000 version 4 'portmapper' (portmap sunrpc rpcbind) is
     running on this port
    RPC program #100000 version 3 'portmapper' (portmap sunrpc rpcbind) is
     running on this port
    RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is
     running on this port

. Information found on port general/tcp

    Nessus was not able to reliably identify the remote operating system. It
     might be:
    Sun Solaris 8
    The fingerprint differs from these known signatures on 1 points.
    If you know what operating system this host is running, please send this
     signature to
    os-signatures@nessus.org :

:1:1:1:1:255:1:255:1:1:255:1:0:255:1:64:255:0:1:1:1:1:3:1:1:0:1:1:64:24616:NNTNWNNS
M:0:1:1

. Information found on port general/tcp

    Remote OS guess : Sun Solaris 8 early access beta through actual release

    CVE : CAN-1999-0454

. Vulnerability found on port unknown (32788/tcp) :

    The tooltalk RPC service is running.
```

```
   A possible implementation fault in the ToolTalk object database server may
    allow an
   attacker to execute arbitrary commands as root.

   *** This warning may be a false positive since the presence of this
    vulnerability is only
   **** accurately identified with local access.

   Solution : Disable this service.
   See also : CERT Advisory CA-98.11
   Risk factor : High
   CVE : CVE-1999-0003, CVE-1999-0693
   BID : 122
   Other references : CERT:CA-98.11

. Vulnerability found on port unknown (32788/tcp) :

   The tooltalk RPC service is running.

   There is a format string bug in many versions
   of this service, which allow an attacker to gain
   root remotely.

   In addition to this, several versions of this service
   allow remote attackers to overwrite abitrary memory
   locations with a zero and possibly gain privileges
   via a file descriptor argument in an AUTH_UNIX
   procedure call which is used as a table index by the
   _TT_ISCLOSE procedure.

   *** This warning may be a false positive since the presence
   *** of the bug was not verified locally.

   Solution : Disable this service or patch it
   See also : CERT Advisories CA-2001-27 and CA-2002-20

   Risk factor : High
   CVE : CAN-2002-0677, CVE-2001-0717, CVE-2002-0679
   BID : 3382

. Information found on port unknown (32788/tcp)

   RPC program #100083 version 1 is running on this port

. Information found on port unknown (4045/tcp)

   RPC program #100021 version 1 'nlockmgr' is running on this port
   RPC program #100021 version 2 'nlockmgr' is running on this port
   RPC program #100021 version 3 'nlockmgr' is running on this port
   RPC program #100021 version 4 'nlockmgr' is running on this port

. Information found on port unknown (32775/tcp)

   RPC program #100007 version 3 'ypbind' is running on this port
   RPC program #100007 version 2 'ypbind' is running on this port
   RPC program #100007 version 1 'ypbind' is running on this port
```

```
. Information found on port unknown (32777/tcp)

   RPC program #100024 version 1 'status' is running on this port
   RPC program #100133 version 1 is running on this port

. Information found on port unknown (38550/tcp)

   RPC program #100068 version 2 is running on this port
   RPC program #100068 version 3 is running on this port
   RPC program #100068 version 4 is running on this port
   RPC program #100068 version 5 is running on this port

. Information found on port sunrpc (111/udp)

   RPC program #100000 version 4 'portmapper' (portmap sunrpc rpcbind) is
    running on this port
   RPC program #100000 version 3 'portmapper' (portmap sunrpc rpcbind) is
    running on this port
   RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is
    running on this port

. Warning found on port lockd (4045/udp)

   The nlockmgr RPC service is running.
   If you do not use this service, then
   disable it as it may become a security
   threat in the future, if a vulnerability
   is discovered.

   Risk factor : Low
   CVE : CVE-2000-0508
   BID : 1372

. Information found on port lockd (4045/udp)

   RPC program #100021 version 1 'nlockmgr' is running on this port
   RPC program #100021 version 2 'nlockmgr' is running on this port
   RPC program #100021 version 3 'nlockmgr' is running on this port
   RPC program #100021 version 4 'nlockmgr' is running on this port

. Warning found on port unknown (32779/udp)

   The ypbind RPC service is running.  If you do not use this service, then
   disable it as it may become a security threat in the future, if a
    vulnerability
   is discovered.

   Risk factor : Low
   CVE : CVE-1999-0312
   BID : 52

. Information found on port unknown (32779/udp)

   RPC program #100007 version 3 'ypbind' is running on this port
   RPC program #100007 version 2 'ypbind' is running on this port
   RPC program #100007 version 1 'ypbind' is running on this port
```

```
. Warning found on port unknown (32785/udp)

   The statd RPC service is running.  This service has a long history of
   security holes, so you should really know what you are doing if you decide
   to let it run.

   *** No security hole regarding this program have been tested, so
   *** this might be a false positive.

   Solution : We suggest that you disable this service.
   Risk factor : High
   CVE : CVE-1999-0018, CVE-1999-0019, CVE-1999-0493
   BID : 127, 450

. Information found on port unknown (32785/udp)

   RPC program #100024 version 1 'status' is running on this port
   RPC program #100133 version 1 is running on this port

. Warning found on port unknown (32790/udp)

   The rquotad RPC service is running.  If you do not use this service, then
   disable it as it may become a security threat in the future, if a
    vulnerability
   is discovered.

   Risk factor : Low
   CVE : CAN-1999-0625

. Information found on port unknown (32790/udp)

   RPC program #100011 version 1 'rquotad' (rquotaprog quota rquota) is running
    on this port


. Warning found on port unknown (32795/udp)

   The rstatd RPC service is running.
   It provides an attacker interesting
   information such as :

    - the CPU usage
    - the system uptime
    - its network usage
    - and more

   Usually, it is not a good idea to let this
   service open

   Risk factor : Low
   CVE : CAN-1999-0624

. Information found on port unknown (32795/udp)

   RPC program #100001 version 2 'rstatd' (rstat rup perfmeter rstat_svc) is
    running on this port
   RPC program #100001 version 3 'rstatd' (rstat rup perfmeter rstat_svc) is
```

```
    running on this port
   RPC program #100001 version 4 'rstatd' (rstat rup perfmeter rstat_svc) is
    running on this port

. Vulnerability found on port unknown (32798/udp) :

   The cmsd RPC service is running.
   This service has a long history of
   security holes, so you should really
   know what you are doing if you decide
   to let it run.

   *** No security hole regarding this program has been tested, so
   *** this might be a false positive

   Solution : We suggest that you disable this service.
   Risk factor : High
   CVE : CVE-1999-0320, CVE-1999-0696, CVE-2002-0391
   BID : 428, 5356

. Information found on port unknown (32798/udp)

   RPC program #100068 version 2 is running on this port
   RPC program #100068 version 3 is running on this port
   RPC program #100068 version 4 is running on this port
   RPC program #100068 version 5 is running on this port

. Warning found on port unknown (22/tcp)

   The remote SSH daemon supports connections made
   using the version 1.33 and/or 1.5 of the SSH protocol.

   These protocols are not completely cryptographically
   safe so they should not be used.

   Solution :
    If you use OpenSSH, set the option 'Protocol' to '2'
    If you use SSH.com's set the option 'Ssh1Compatibility' to 'no'

   Risk factor : Low

. Information found on port unknown (22/tcp)

   The remote SSH daemon supports the following versions of the
   SSH protocol :

     . 1.33
     . 1.5
     . 1.99
     . 2.0

. Information found on port unknown (22/tcp)

   Remote SSH version : SSH-1.99-OpenSSH_3.7.1p2

. Warning found on port unknown (79/tcp)
```

```
   The 'finger' service provides useful information to attackers, since it
    allows
   them to gain usernames, check if a machine is being used, and so on...

   Here is the output we obtained for 'root' :

   Login       Name               TTY         Idle    When    Where

   root     Super-User            pts/3        <Dec 12 09:09> coffee
   root     Operator              pts/3        <Dec 12 09:09> coffee

   Solution : comment out the 'finger' line in /etc/inetd.conf
   Risk factor : Low
   CVE : CVE-1999-0612

. Warning found on port unknown (79/tcp)

   The remote finger service accepts to redirect requests. That is, users can
   perform requests like :

     finger user@host@victim

   This allows an attacker to use this computer as a relay to gather
    information
   on a third party network.

   Solution: Disable the remote finger daemon (comment out the 'finger' line
   in /etc/inetd.conf and restart the inetd process) or upgrade it to a more
   secure one.

   Risk factor : Low
   CVE : CAN-1999-0105, CVE-1999-0106

. Warning found on port unknown (514/tcp)


   The rsh service is running.
   This service is dangerous in the sense that
   it is not ciphered - that is, everyone can sniff
   the data that passes between the rsh client
   and the rsh server. This includes logins
   and passwords.

   You should disable this service and use ssh instead.

   Solution : Comment out the 'rsh' line in /etc/inetd.conf.

   Risk factor : Low
   CVE : CAN-1999-0651

. Warning found on port daytime (13/udp)

   The remote host is running a 'daytime' service. This service
   is designed to give the local time of the day of this host
   to whoever connects to this port.

   The date format issued by this service may sometimes help an attacker
```

```
   to guess the operating system type of this host, or to set up
   timed authentication attacks against the remote host.

   In addition to that, the UDP version of daytime is running, an attacker
   may link it to the echo port of a third party host using spoofing, thus
   creating a possible denial of service condition between this host and
   a third party.

   Solution :

   - Under Unix systems, comment out the 'daytime' line in /etc/inetd.conf
     and restart the inetd process

   - Under Windows systems, set the following registry keys to 0 :
     HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableTcpDaytime
     HKLM\System\CurrentControlSet\Services\SimpTCP\Parameters\EnableUdpDaytime

   Then launch cmd.exe and type :

      net stop simptcp
      net start simptcp

   To restart the service.

   Risk factor : Low
   CVE : CVE-1999-0103

. Warning found on port unknown (177/udp)

   The remote host is running XDMCP.

   This protocol is used to provide X display connections for
   X terminals. XDMCP is completely insecure, since the traffic and
   passwords are not encrypted.

   An attacker may use this flaw to capture all the keystrokes of
   the users using this host through their X terminal, including
   passwords.

   Risk factor : Medium
   Solution : Disable XDMCP

. Vulnerability found on port unknown (32798/tcp) :

   Your Sun rpc.cmsd has integer overflow problem in
   xdr_array. An attacker may use this flaw to execute
   arbitrary code on this host with the privileges
   rpc.cmsd is running as (typically, root), by
   sending a specially crafted request to this service.

   We suggest that you disable this
   service and apply a new patch.

   Risk factor : High
   CVE : CVE-2002-0391
   BID : 5356
```

```
------------------------------------------------------------
This file was generated by the Nessus Security Scanner
```

Using the results I modified the desktop:

- Disabled time
- Disabled daytime
- Disabled dtspc
- Disabled rquotad
- Disabled rstatd
- Disabled cmsd
- Set OpenSSH protocol to 2

and rescanned until I was satisfied with the results. One of the UNIX system administrators made the same modifications on all the identical UNIX desktops. We created accounts on our Meetingmaker server for users who had been using Solaris Calendar Manager (cmsd) and had them switch. Now I focused on the servers. I scanned each one and the administrators and I remedied the issues I found. One of the UNIX administrators updated and reconfigured our bind service. He changed the configuration of our Openssh to allow only SSH version 2 protocol not SSH version 1. The Windows administrators upgraded the SSH and scp client software used by about 100 users since the version they were using did not support version 2. And we disabled unneeded services. The webserver administrator upgraded apache and zope and modified the apache configuration to remedy vulnerabilities in the development and production web servers. The Oracle data base administrator modified the Oracle configuration to remedy Oracle vulnerabilities. I rescanned the modified systems to verify that the issues had indeed been resolved. I also compiled and installed nessus on my home computer so that I could verify that any vulnerabilities that were open internally but closed to hosts outside our network were indeed closed. For instance, we allow relaying on our smtp servers internally but not externally.

Host based packet filters

All the Windows desktops already had TCP filters configured to block all incoming TCP packets except those established sessions initiated by the client desktop, so I focused on the UNIX machines. For host based packet filtering I chose IP filter available from http://coombs.anu.edu.au/ipfilter/.[8] It is free. The Solaris implementation has been extensively used and tested and it provides dynamic or stateful packet filtering at the host level. I compiled and installed it on a representative UNIX desktop node. I set up configuration on our configuration management system and modified it until I had enforced the policy I wanted. The desktop rules were:

```
# block malformed packets
block in     quick proto tcp/udp all with short
block in     quick proto icmp all with frag
#
```

```
# filter ingress spoofing
block in     quick from 192.168.0.0/16 to any
block in     quick from 172.16.0.0/12 to any
block in     quick from 10.0.0.0/8 to any
block in     quick from 127.0.0.0/8 to any
block in     quick from 0.0.0.0/8 to any
block in     quick from 169.254.0.0/16 to any
block in     quick from 192.0.2.0/24 to any
block in     quick from 204.152.64.0/23 to any
block in     quick from 224.0.0.0/3 to any
#
# filter egress spoofing
block out     quick from any to 192.168.0.0/16
block out     quick from any to 172.16.0.0/12
block out     quick from any to 10.0.0.0/8
block out     quick from any to 127.0.0.0/8
block out     quick from any to 0.0.0.0/8
block out     quick from any to 169.254.0.0/16
block out     quick from any to 192.0.2.0/24
block out     quick from any to 204.152.64.0/23
block out     quick from any to 224.0.0.0/3
#
# loopback interface
pass in quick on lo0 all
pass out quick on lo0 all
#
# allow all traffic from our static IP Class C subnets
# using ficticious 200.200.3 and 200.200.4 for example
pass in quick from 200.200.3.0/24 to any
pass in quick from 200.200.4.0/24 to any
pass out quick from any to 200.200.3.0/24
pass out quick from any to 200.200.4.0/24
#
#allow outbound and established
pass out quick all keep state
#
# block all icmp
block in     quick proto icmp from any to any
#
# default deny
block in     all
block out    all
```

The desktop machines offer no services and I will not log dropped packets. One of the UNIX system administrators installed it on all the UNIX desktops. Now I focused on the UNIX servers. I configured the rules for each server modifying it for the services offered by the server and installed it.
Our mail gateway server rules, for example, open smtp and I log dropped packets on all servers:

```
# block malformed packets
block in log quick proto tcp/udp all with short
block in log quick proto icmp all with frag
#
# filter ingress spoofing
```

```
block in log quick from 192.168.0.0/16 to any
block in log quick from 172.16.0.0/12 to any
block in log quick from 10.0.0.0/8 to any
block in log quick from 127.0.0.0/8 to any
block in log quick from 0.0.0.0/8 to any
block in log quick from 169.254.0.0/16 to any
block in log quick from 192.0.2.0/24 to any
block in log quick from 204.152.64.0/23 to any
block in log quick from 224.0.0.0/3 to any
#
# filter egress spoofing
block out log quick from any to 192.168.0.0/16
block out log quick from any to 172.16.0.0/12
block out log quick from any to 10.0.0.0/8
block out log quick from any to 127.0.0.0/8
block out log quick from any to 0.0.0.0/8
block out log quick from any to 169.254.0.0/16
block out log quick from any to 192.0.2.0/24
block out log quick from any to 204.152.64.0/23
block out log quick from any to 224.0.0.0/3
#
# loopback interface
pass in quick on lo0 all
pass out quick on lo0 all
#
# allow all traffic from our static IP Class C subnets
# using ficticious 200.200.3 and 200.200.4 for example
pass in quick from 200.200.3.0/24 to any
pass in quick from 200.200.4.0/24 to any
pass out quick from any to 200.200.3.0/24
pass out quick from any to 200.200.4.0/24
#
#allow outbound and established
pass out quick all keep state
#
#bind(dns)
pass in quick proto udp from any to any port = 53 flags S keep state
#
#our services
#smtp(email)
pass in quick proto tcp from any to any port = 25 flags S keep state
#
# default deny
block in log all
block out log all
```

I installed similar rules for the servers serving NFS, Samba, LPD, NIS, Oracle, HTTPD, SSH, IMAP, BOOTP, NTP.

Security and Benchmark Scoring of Configurations

In order to harden all the machines using consensus best practice recommendations I turned to the Center for Internet Security CIS Benchmark/Security Tools available at http://www.cisecurity.org/. I started with Solaris.[9] I downloaded CISscan and installed it on a representative UNIX

desktop node. I scanned the node with the tool and it had a score of 3.72 on a scale of 0 (less secure) to 10 (more secure). Using this node as my template I set up a configuration for CIS on our configuration server adding the hardening driver script and the supporting scripts to harden the desktop to where I wanted:

- Disabled logins from serial ports other than /dev/console.
- Started logging connections to inetd services.
- Stopped listening on port 514 except for central login server.
- Renamed 23 startup scripts to prevent services from starting at boot.
- Disabled NFS server startup script except for NFS file servers.
- Modified kernel parameters to not save core dumps, limit and log user stack size and require NFS clients to use privileged ports.
- Set TCP and IP network parameters.
- Use good TCP sequence numbers.
- Log failed login attempts.
- Set logging on the root filesystem.
- Set nosuid mount option on nonroot filesystems.
- Set perms on passwd, group and shadow files.
- Remove empty crontab files and set permissions on crontab entries.
- Set authorization required banners.
- Set login retries.
- Change root shell to /dev/null for nonroot system accounts.
- Set default umask.
- Set mesg to no.
- Fix group writeable home dirs on system accounts

This brought the score up to 6.85. One of the UNIX system administrators installed it, running the hardening scripts and ran the scoring tool on all UNIX desktops. I then turned to the servers and created individual configurations for some servers depending on the services they were running. For example, on the NFS server I did not disable the NFS startup script. One of the UNIX administrators installed it on all the servers during their next patching cycle since a reboot was required. I asked one of the Windows administrators to download the Level-2 Windows 2000 Professional Benchmark[10] and harden a representative Windows 2000 desktop using the Win2kProGold security template. He began with a score of 1.7 and brought the score up to 8.7 after hardening.

Bastion Login Server

For the bastion login server I chose to create a bastion host with The S/Key One Time Password System introduced by Neil Haller in 1994 (http://citeseer.nj.nec.com/haller94skey.html)[11] acting as a gateway to the internal login server. It is available at no cost, can be compiled into OpenSSH, can move with the users as they use different machines, and the one time password lists can be faxed to our users in remote locations. The one time passwords have the benefits of minimizing the risks of users storing their passwords in their client software which could subsequently be stolen or

compromised and mitigates the risk associated with keyloggers being installed on client machines via trojans. I started with a new host using our standard build. One of the UNIX administrators downloaded The OpenPKG S/Key source package from http://www.openpkg.org/[12] built it and built an OpenSSH package with S/Key support. I installed the package on the host. I created /etc/passwd and /etc/shadow files of all our users with a * in the shadow entry hash field so that the S/Keys would be required for login. I wrote a gateway script which will SSH to our internal server and set this as the shell script for all users in the passwd file. I added support for adding new users to bastion login server passwd and shadow files and the S/Key database to the add new user protocol. I then turned the host into a bastion host by removing support for NIS, NFS, modifying DNS, NTP to use servers outside our network.

Despite the extensive anaylsis which led to choosing the S/key implementation, I ran into user resistance from several members of the Computer Board to the S/Key process as being too cumbersome for people who connect a lot and too complicated for people who connect infrequently. They hated actually using it and wanted to spend some money for something a bit nicer to use. I looked at biometrics and at commercial hardware token based authentication as alternatives.[13] I rejected biometrics as an inadequate solution on an untrusted network due to replay attacks and it would also require sensors on every client. This was not feasible for us since users would login from other universities and institutions. For commercial token based authentication I required that the token move with the user and that it support SSH port forwarding. I contacted the three vendors RSA Security, Cryptocard and Secure Computing and settled on RSA Security SecurID key-fobs. This solution was a stronger form of authentication than my original S/Key solution since it was two factor authentication with the PIN in addition to the hardware token. Also the encryption used 256 bit AES rather than the 128 bit MD5 used by the S/Keys. It is far easier to use than the S/Keys from the user perspective. The enhanced security and the scalability justified the not inconsiderable expense.

In an unrelated conversation with members of the Organization's IT department, I mentioned my plans to create a bastion host using RSA SecurID key-fobs and was told that the Organization planned to setup a bastion login host using RSA SecurID for use by the Nonstandard groups such as the Laboratory and that the hardware would be purchased within a month. These plans had not been communicated to the NonStandard groups including the Laboratory. The cost would be significantly less for us since much of the cost we have already paid for through the money we give to the Organization for infrastructure. I presented my findings to my Computer Board with my recommendation that we pursue the Organization's intentions to implement, but be prepared to implement RSA SecurID ourselves if it became clear that the Organization could not implement its solution within a few months.

Firewall and DMZ

My initial design[14] (see Figure 1) was to segment the network into our internal servers and a DMZ by adding a switch and firewall in the datacenter of the Laboratory and connecting this to the Organization switch.

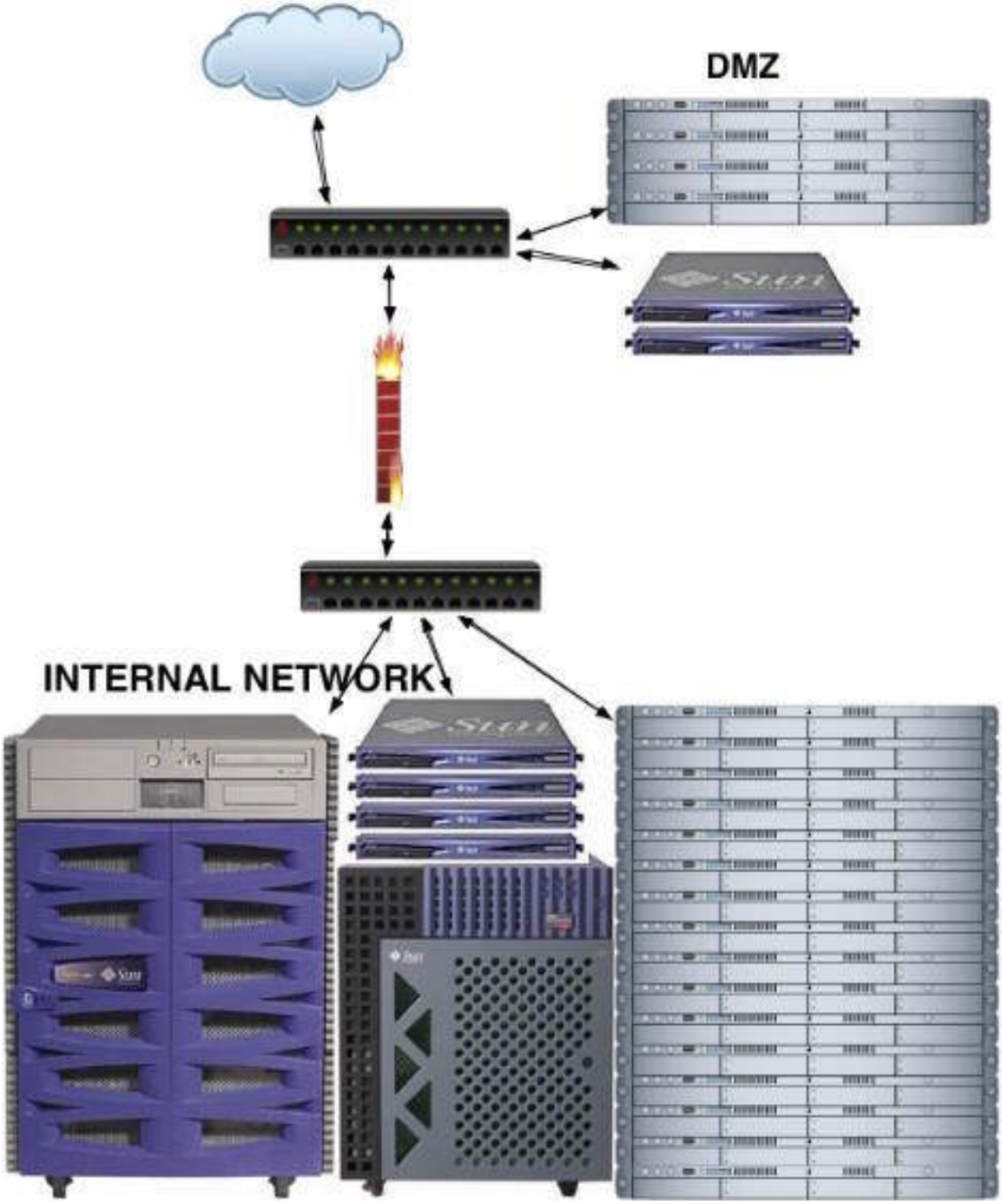

Figure 1

Our internal servers would connect to our internal switch, the firewall would NAT our servers to the private address space. The DMZ machines (internet facing machines) would connect directly the Organization switch.

The Organization objected to this design on the grounds that allowing each group to setup its own firewall would not scale well for the Organization as a whole and would cause troubleshooting difficulties when a problem occurred. They proposed setting up a DMZ for the Organization as a whole and a firewall for the Organization as a whole which the Organization would control. Their proposal was the culmination of a two year planning process but they had not begun to implement the DMZ. The Laboratory firewall would have to be shelved.

I revisited the ipfilter rules and used this to block incoming packets from the machines on the DMZ which will now remain on our network and be in a DMZ in name only. The desktop machines rules became:

```
# block malformed packets
block in     quick proto tcp/udp all with short
block in     quick proto icmp all with frag
#
# filter ingress spoofing
block in     quick from 192.168.0.0/16 to any
block in     quick from 172.16.0.0/12 to any
block in     quick from 10.0.0.0/8 to any
block in     quick from 127.0.0.0/8 to any
block in     quick from 0.0.0.0/8 to any
block in     quick from 169.254.0.0/16 to any
block in     quick from 192.0.2.0/24 to any
block in     quick from 204.152.64.0/23 to any
block in     quick from 224.0.0.0/3 to any
#
# filter egress spoofing
block out     quick from any to 192.168.0.0/16
block out     quick from any to 172.16.0.0/12
block out     quick from any to 10.0.0.0/8
block out     quick from any to 127.0.0.0/8
block out     quick from any to 0.0.0.0/8
block out     quick from any to 169.254.0.0/16
block out     quick from any to 192.0.2.0/24
block out     quick from any to 204.152.64.0/23
block out     quick from any to 224.0.0.0/3
#
# loopback interface
pass in quick on lo0 all
pass out quick on lo0 all
#
#allow outbound and established
pass out quick all keep state
#
# block traffic from DMZ
#webmail server
block in     quick from 200.200.3.24/32 to any
#mail gateway
```

```
block in     quick from 200.200.3.49/32 to any
#web server
block in     quick from 200.200.3.57/32 to any
#ssh gateway
block in     quick from 200.200.3.68/32 to any
#
# allow all traffic from our static IP Class C subnets
# using ficticious 200.200.3 and 200.200.4 for this example
pass in quick from 200.200.3.0/24 to any
pass in quick from 200.200.4.0/24 to any
pass out quick from any to 200.200.3.0/24
pass out quick from any to 200.200.4.0/24
#
# block all icmp
block in     quick proto icmp from any to any
#
# default deny
block in     all
block out     all
```

And the internal smtp server rules must allow access from the mail gateway on the DMZ:

```
# block malformed packets
block in log quick proto tcp/udp all with short
block in log quick proto icmp all with frag
#
# filter ingress spoofing
block in log quick from 192.168.0.0/16 to any
block in log quick from 172.16.0.0/12 to any
block in log quick from 10.0.0.0/8 to any
block in log quick from 127.0.0.0/8 to any
block in log quick from 0.0.0.0/8 to any
block in log quick from 169.254.0.0/16 to any
block in log quick from 192.0.2.0/24 to any
block in log quick from 204.152.64.0/23 to any
block in log quick from 224.0.0.0/3 to any
#
# filter egress spoofing
block out log quick from any to 192.168.0.0/16
block out log quick from any to 172.16.0.0/12
block out log quick from any to 10.0.0.0/8
block out log quick from any to 127.0.0.0/8
block out log quick from any to 0.0.0.0/8
block out log quick from any to 169.254.0.0/16
block out log quick from any to 192.0.2.0/24
block out log quick from any to 204.152.64.0/23
block out log quick from any to 224.0.0.0/3
#
# loopback interface
pass in quick on lo0 all
pass out quick on lo0 all
#
#allow outbound and established
pass out quick all keep state
#
```

```
# block traffic from DMZ
#our services to DMZ
#smtp
pass in quick proto tcp from 200.200.3.0/24 to any port = 25 keep
state
pass in quick proto tcp from 200.200.4.0/24 to any port = 25 keep
state
pass in quick proto tcp from 200.200.3.0/24 to any port = 587 keep
state
pass in quick proto tcp from 200.200.4.0/24 to any port = 587 keep
state
#webmail server
#we are imap server for webmail
pass in quick proto tcp from 200.200.3.24/32 to any port = 993 keep
state
block in log quick from 200.200.3.24/32 to any
#mail gateways
block in log quick from 200.200.3.49/32 to any
block in log quick from 200.200.4.36/32 to any
#web server
block in log quick from 200.200.3.57/32 to any
#
# allow all traffic from our static IP Class C subnets
# using ficticious 200.200.3 and 200.200.4 for this example
pass in quick from 200.200.3.0/24 to any
pass in quick from 200.200.4.0/24 to any
pass out quick from any to 200.200.3.0/24
pass out quick from any to 200.200.4.0/24
#
# block all icmp
block in log quick proto icmp from any to any
#
# drop all netbios traffic on any interface
block in quick proto tcp/udp from any port 136 >< 140 to any
block in quick proto tcp/udp from any to any port 136 >< 140
#
# default deny
block in log all
block out log all
```

Allow rules for the DMZ had to be added for the central syslog/rdist machine and the ntp servers, and the internal SSH server.

After

A risk assessment has been performed for the Laboratory providing a snapshot of the security posture and a blueprint for enhancing this security. By implementing many of the recommendations, the security of the Laboratory has been enhanced in the host and application layers. The implementation of the bastion login host was modified due to user resistance and to efforts by the Organization. The recommendations for the network layer ran into difficulties due to Organizational considerations further emphasizing the need for a defense in depth approach.

Host Layout

The OS of the hosts have had vulnerabilities remedied: services turned off, applications upgraded, configurations hardened. The OS has been hardened modifying kernel parameters and configurations in line with consensus security guidelines. The internal SSH login host will only allow SSH from the bastion login host and from our internal network rather than from the internet. IP packet filters on the hosts have been added to enforce the desired network access control list and to simulate the firewall that we could not implement.

Application Layout

Authentication has been augmented in two ways. A strong password policy has been implemented on the internal servers and enforced using a password cracker to discover weak passwords. SSH now uses protocol 2. The bastion login host allows us to use a hardened server for our remote login access rather than an internal server and two factor hardware token based strong authentication with one time passwords rather than 8 character reusable passwords further strengthening the password policy and preventing users from sharing or reusing passwords or having passwords stolen with keyloggers. BIND and apache have been upgraded.

References

[1] Cole, Eric, Fossen, Jason, Northcutt, Stephen, Pomeranz, Hal, SANS Security Essentials with CISSP CBK, Version 2.1, SANS Press, 2003.

[2] SANS Institute. "The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts Consensus." Version 4.0. 8 Oct. 2003. URL: http://www.sans.org/top20/ (8 Feb. 2004).

[3] O'Shea, Kevin. "Examining the RPC DCOM Vulnerability: Developing a Vulnerability-Exploit Cycle." 3 Sep. 2003. URL: http://www.sans.org/rr/papers/index.php?id=1220. (8 Feb. 2004).

[4] Cheswick, William R, Bellovin, Steven M, Rubin, Aviel D, Firewalls and Internet Security: Repelling the Wily Hacker, Second Edition, Addison Wesley, 24 Feb 2003.

[5] SANS Institute. "The SANS Security Policy Project." URL: http://www.sans.org/resources/policies/. (8 Feb 2004).

[6] Openwall Project. "John the Ripper password cracker." Version 1.6. URL: http://www.openwall.com/john/. (8 Feb. 2004).

[7] Nessus Project. "Nessus Security Scanner." Version 2.0.9. URL: http://nessus.org/. (8 Feb. 2004).

[8] Reed, Darren. "IP Filter." Version 3.4.33pre2. URL: http://coombs.anu.edu.au/ipfilter/. (8 Feb. 2004).

[9] CIS Security. “CIS Level 1 Benchmark and Scoring Tool for Solaris.” Version 1.4.0. Oct. 2003. URL: http://www.cisecurity.org/bench_solaris.html. (8 Feb. 2004).
[10] CIS Security. “CIS Level-2 Windows 2000 Professional Operating System Benchmark.” Version Win2kProGold_R1.2.4. URL: http://www.cisecurity.org/bench_win2000.html. (8 Feb. 2004).
[11] Haller, Neil. “The S/KEY one-time password system.” Proceedings of the ISOC Symposium on Network and Distributed System Security, pages 151--157, San Diego, CA, Feb. 1994. URL: http://citeseer.nj.nec.com/haller94skey.html. (8 Feb 2004).
[12] OpenPKG. Cross-platform RPM-based Unix software packaging. URL: http://www.openpkg.org/. (10 Apr 2004).
[13] Smith, Richard E., Authentication: From Passwords to Public Keys, Addison-Wesley Pub Co, 1 Oct. 2001.
[14] Chapman, D Brent, Cooper, Simon, Zwicky, Elizabeth D, Building Internet Firewalls, 2nd Edition, O’Reilly, Jun 2000.

Last Updated: October 22nd, 2017

Upcoming Training

SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201710,	Oct 23, 2017 - Nov 29, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS Vancouver SEC401^	Vancouver, BC	Nov 06, 2017 - Nov 11, 2017	Community SANS
Community SANS Colorado Springs SEC401~	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS St. Louis SEC401	St Louis, MO	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS Khobar 2017	Khobar, Saudi Arabia	Dec 02, 2017 - Dec 07, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Dec 04, 2017 - Dec 09, 2017	Community SANS
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201712,	Dec 11, 2017 - Jan 24, 2018	vLive
SANS Bangalore 2017	Bangalore, India	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Cyber Defense Initiative 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Dec 14, 2017 - Dec 19, 2017	vLive
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Community SANS Hawaii SEC401	Honolulu, HI	Jan 08, 2018 - Jan 13, 2018	Community SANS
Mentor Session - SEC401	Memphis, TN	Jan 09, 2018 - Mar 13, 2018	Mentor
SANS Amsterdam January 2018	Amsterdam, Netherlands	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VA	Jan 15, 2018 - Jan 20, 2018	Live Event
Mentor Session - SEC401	Minneapolis, MN	Jan 16, 2018 - Feb 27, 2018	Mentor
Las Vegas 2018 - SEC401: Security Essentials Bootcamp Style	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event