



गृह मंत्रालय  
MINISTRY OF  
HOME AFFAIRS

राष्ट्रीय न्यायिक विज्ञान विश्वविद्यालय  
National Forensic Sciences University



# Unit 5 Network Forensics



**Dr. Lokesh Chouhan**  
Associate Professor



गृह मंत्रालय  
MINISTRY OF  
HOME AFFAIRS

**राष्ट्रीय न्यायालयिक विज्ञान विश्वविद्यालय**  
(राष्ट्रीय महत्त्व का संस्थान, गृह मंत्रालय, भारत सरकार)  
**National Forensic Sciences University**  
(An Institution of National Importance under Ministry of Home Affairs,  
Government of India)



E-Mail: [Lokeshchouhan@gmail.com](mailto:Lokeshchouhan@gmail.com), [Lokesh.chouhan\\_goa@nfsu.ac.in](mailto:Lokesh.chouhan_goa@nfsu.ac.in)

Mob: +91-898924399, 9827235155

# What is Computer Forensics?

**Definition:** Involves obtaining and analyzing digital information, often as evidence in civil, criminal, or administrative cases

## Computer forensics:

- Investigates data that can be retrieved from a computer's hard disk or other storage media
- Task of recovering data that users have hidden or deleted and using it as evidence
- Evidence can be **inculpatory** (“incriminating”) or **exculpatory**

# Computer Forensics Versus Other Related Disciplines

- Network forensics
  - Yields information about how a perpetrator or an attacker gained access to a network
- Data recovery
  - Recovering information that was deleted by mistake, or lost during a power surge or server crash
  - Typically you know what you're looking for

# Computer Forensics Versus Other Related Disciplines (continued)

- Disaster recovery

- Uses computer forensics techniques to retrieve information their clients have lost

Investigators often work as a team to make computers and networks secure in an organization



# Digital Evidence

- Locard's principle: "every contact leaves a trace"
- any information, stored or transmitted in digital form, that a party to a court case may use at a trial

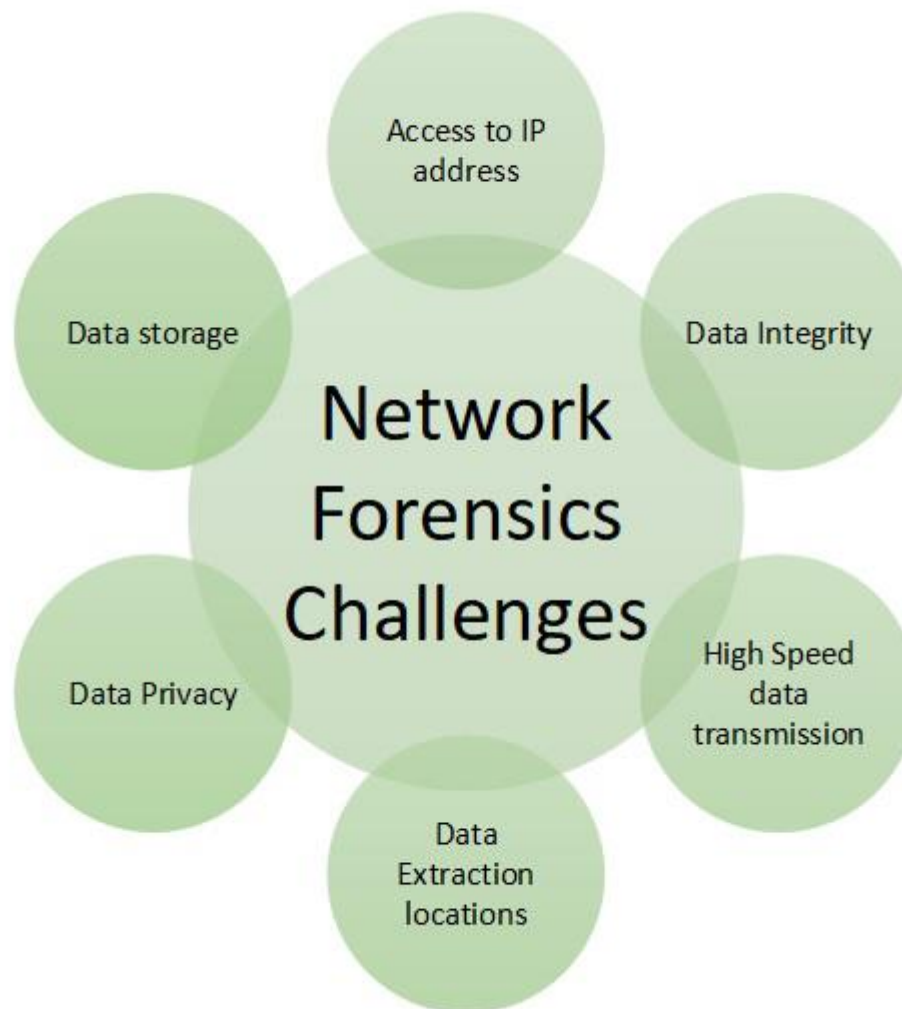
To be accepted in court, digital evidence must meet certain criteria ...

- Admissibility
- Authenticity

# Network Forensics

- Definition
  - The study of network traffic to search for truth in civil, criminal, and administrative matters to protect users and resources from exploitation, invasion of privacy, and any other crime fostered by the continual expansion of network connectivity. (Source: Kevin Mandia & Chris Prosise, Incident response, Osborne/McGraw-Hill, 2001. )

# Network Forensics





# Category of Digital Evidence

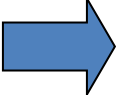
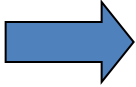
- Hardware
- Software
  - Data
  - Programs



# Digital Evidence

- Definition
  - Digital data that can establish that a crime has been committed or can provide a link between a crime and its victim or a crime and its perpetrator.(source: Casey, Eoghan, *Digital Evidence and Computer Crime: Forensic Science, Computer and the Internet*, Academic Press, 2000.)
  - Categories
    - Text
    - Audio
    - Image
    - Video

# Where Evidence Resides

- Computer systems
  - Logical file system
    - File system
      - Files, directories and folders, FAT, Clusters, Partitions, Sectors
    - Random Access memory 
    - Physical storage media
      - magnetic force microscopy can be used to recover data from overwritten area.
  - Slack space 
    - space allocated to file but not actually used due to internal fragmentation.
  - Unallocated space

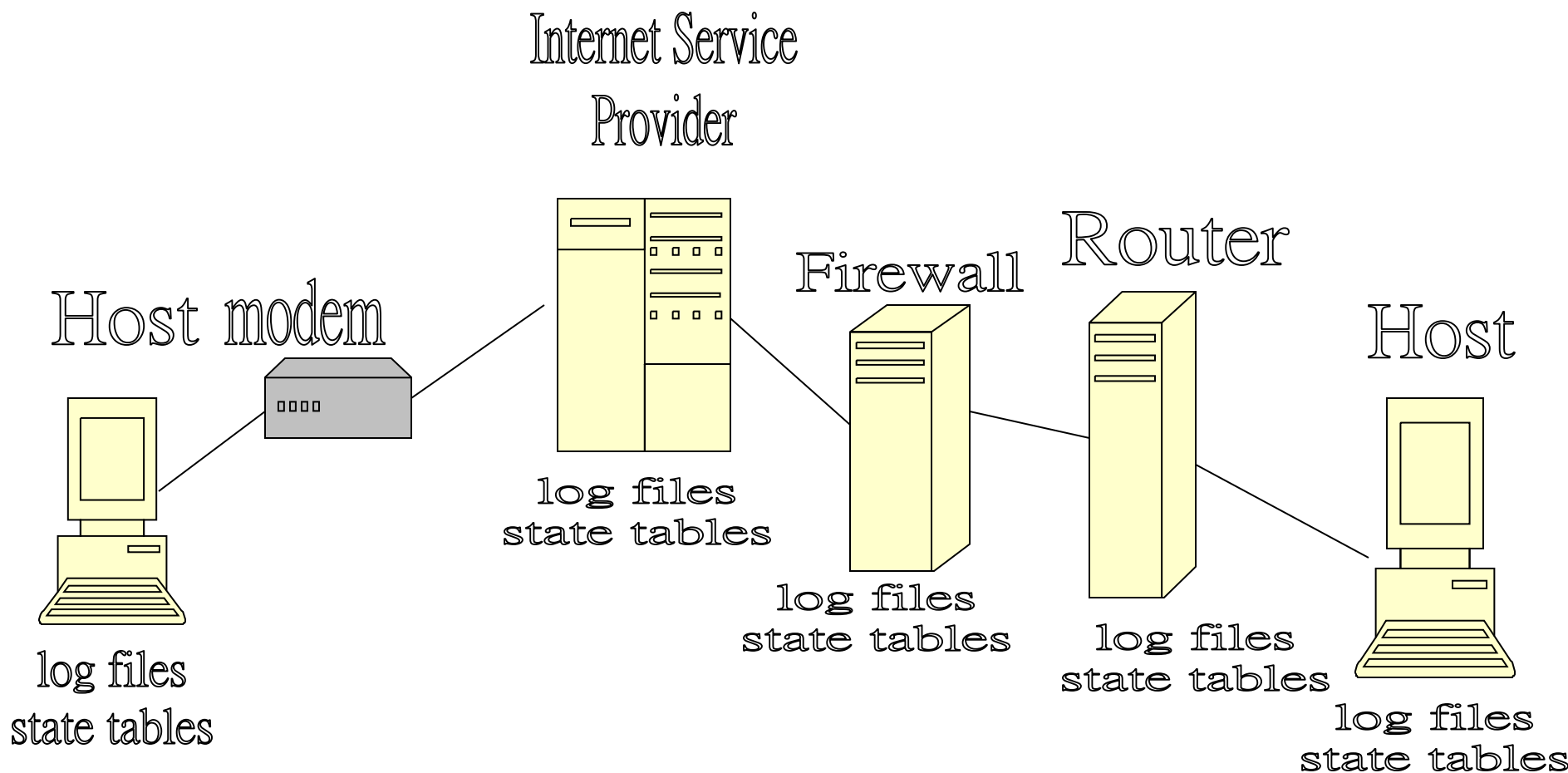
## Where Evidence Resides (continued)

- Computer networks.
  - Application Layer
  - Transportation Layer
  - Network Layer
  - Data Link Layer

## Evidence on Application Layer

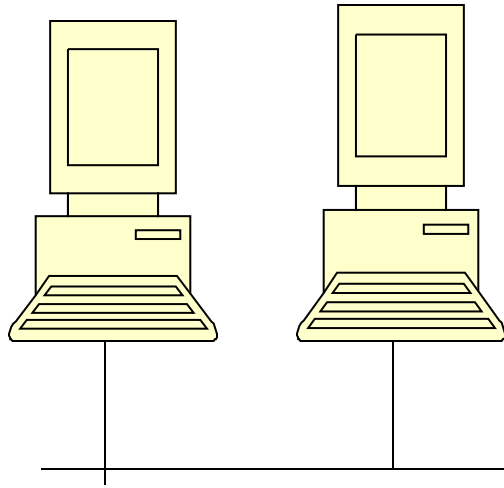
- Web pages, Online documents.
- E-Mail messages.
- News group archives.
- Archive files.
- Chat room archives.
- ...

# Evidence on Transport and Network Layers

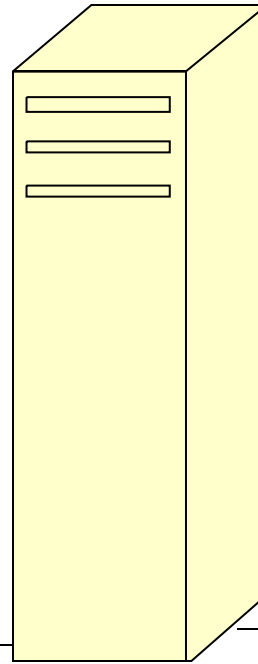


# Evidence on the Data-link and Physical Layers

Computer A

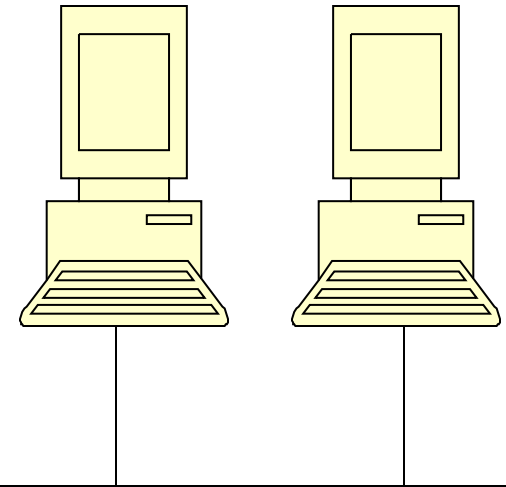


Ethernet Network



Router

Computer Z



ATM Network

MAC --> IP

MAC <-- IP



- Traditional Forensics
  - Analyzing a “dead” system that has had its power cord pulled
  - Least chance of modifying data on disk, but “live” data is lost forever
- Live Forensics (Often Incident Response)
  - Methodology which advocates extracting “live” system data before pulling the cord to preserve memory, process, and network information
  - Goal is to minimize impacts to the integrity of the system while capturing volatile forensic data



- Volatile Data includes :
  - current network configuration,
  - current system date and time,
  - current network connections,
  - currently open sockets(ports),
  - The applications listening on the open sockets
  - processes currently running,
  - users currently logged on – remote and local,
  - applications currently running on the machine.





## ➤ Includes information from

- IDS logs
- Monitoring logs
- Router logs
- Firewall logs
- Authentication servers
- Remote logs from Centralized Host
- Network Packets



## Securing Crime Scene

### ➤ Upon entry:

- Secure premises
- Secure occupant(s)
- Prevent destruction of evidence
- Prevent unauthorized access to premises-  
Physical and Virtual
- Connected to a Network?
  - Wireless
  - Wi-Fi/Wi-max
  - Cat 5 cable
  - Modem

## Documenting Crime Scene

- Once you have secured the electronic crime scene, you should document the surroundings as initially found:
  - Photograph
  - Sketch
- Photograph every item before it is seized
- Photograph documents on the monitor
- Sketch- overall – “Not to Scale”



# Live Forensics





# Live Forensics - Challenges

- ☐ Need access to the system
- ☐ Minimize impact on system.
- ☐ Some tools leave footprint and hence proper audit/notes to be made.
- ☐ Timely evidence acquisition and analysis.



1. **Retrieval of volatile data**
2. **Forensic imaging of live system**
3. **Analysis of evidence collected**



# Scenario : Ongoing Crime

- ☐ Want to catch them “in the act”
- ☐ See how things change (web pages, file access times, registry, memory, etc.)
- ☐ Want to understand:
  - ☐ How they got in
  - ☐ What they compromised
  - ☐ Where they are
  - ☐ Who they are



## Live Data

- System time
- Logged-on user(s)
- Open files
- Network information
- Network connections
- Process information
- Process-to-port mapping
- Process memory
- Network status
- Clipboard contents
- Service/driver information
- Command history
- Mapped drives
- Shares

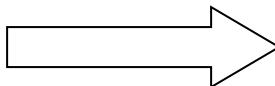




# Non-volatile Data

- Event logs
- Registry
- Disks

## Memory Acquisition



**Perform Ram Dump to a file** [Using Tools like  
DumpIt, mdd etc]

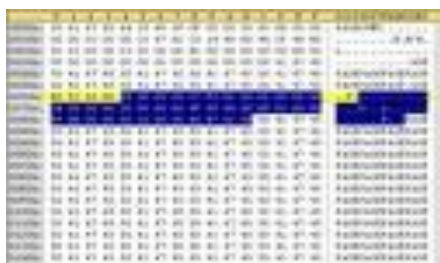
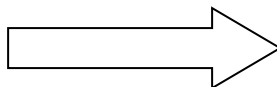


**Dump file**

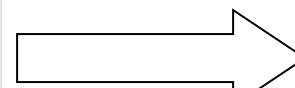
## Dump File Analysis



**Dump file**



**Analyzed using tools**  
(eg. Volatility, Win-LiFT)



**Report  
Generation**





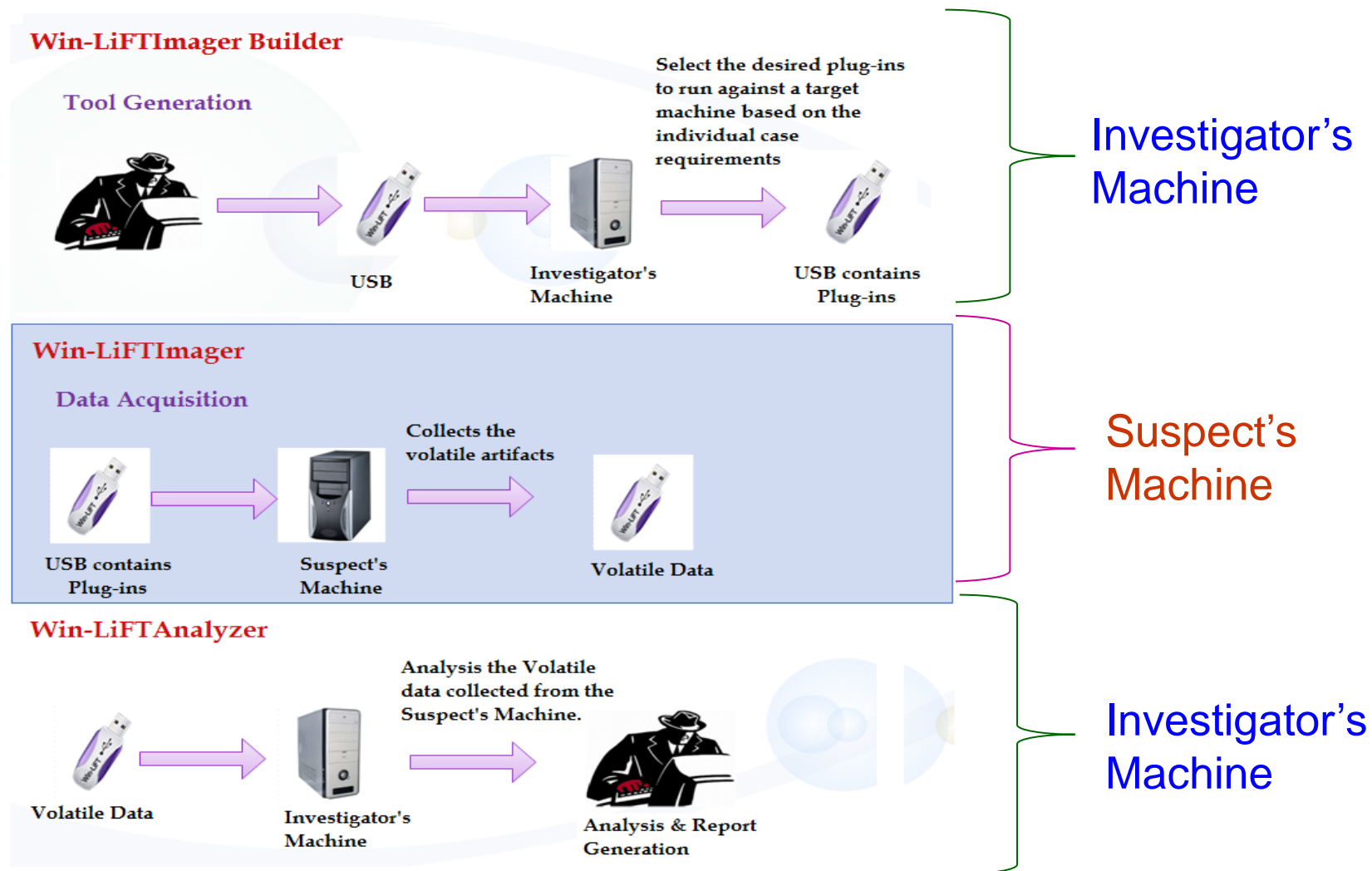
# Live Forensics Tools

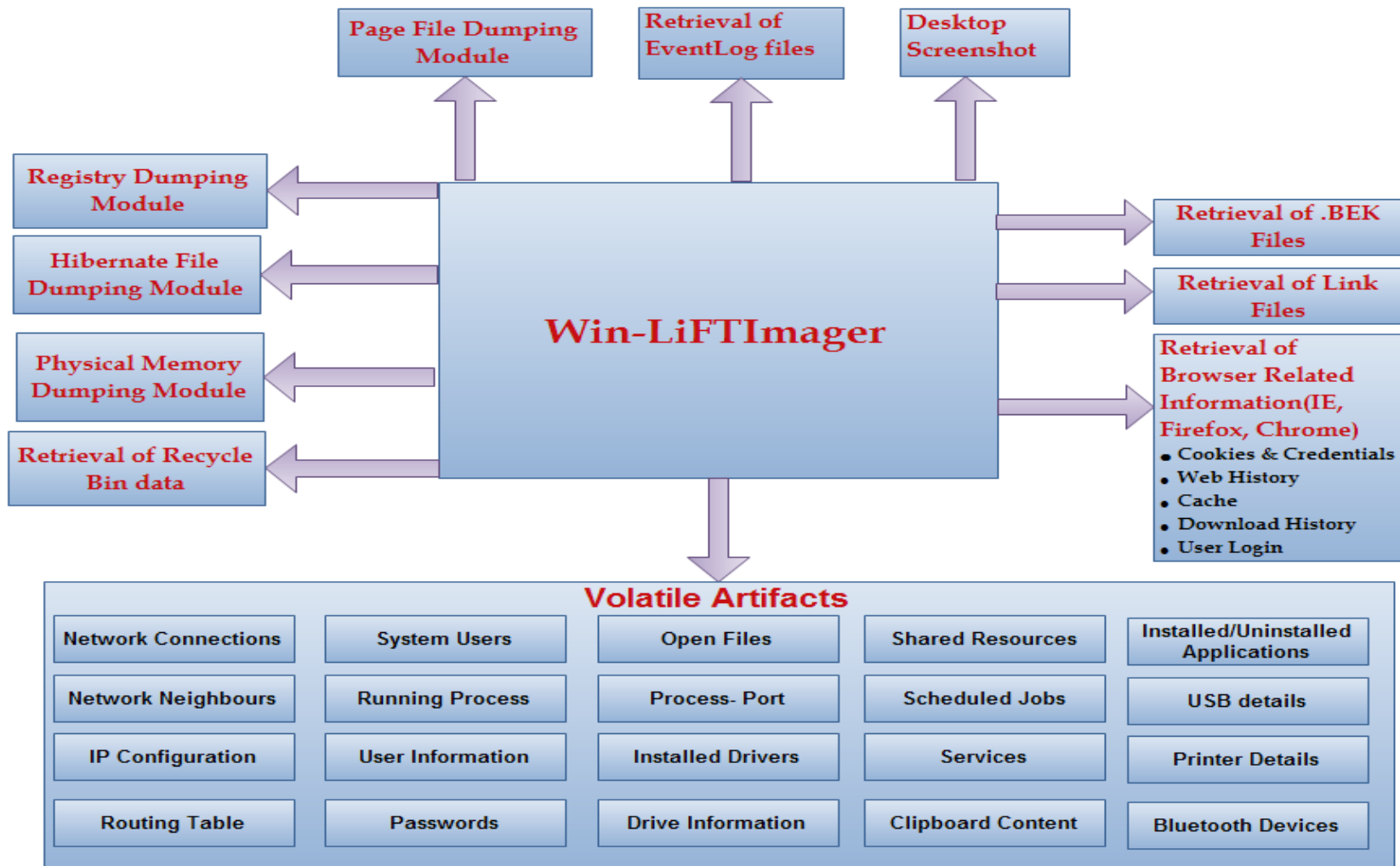


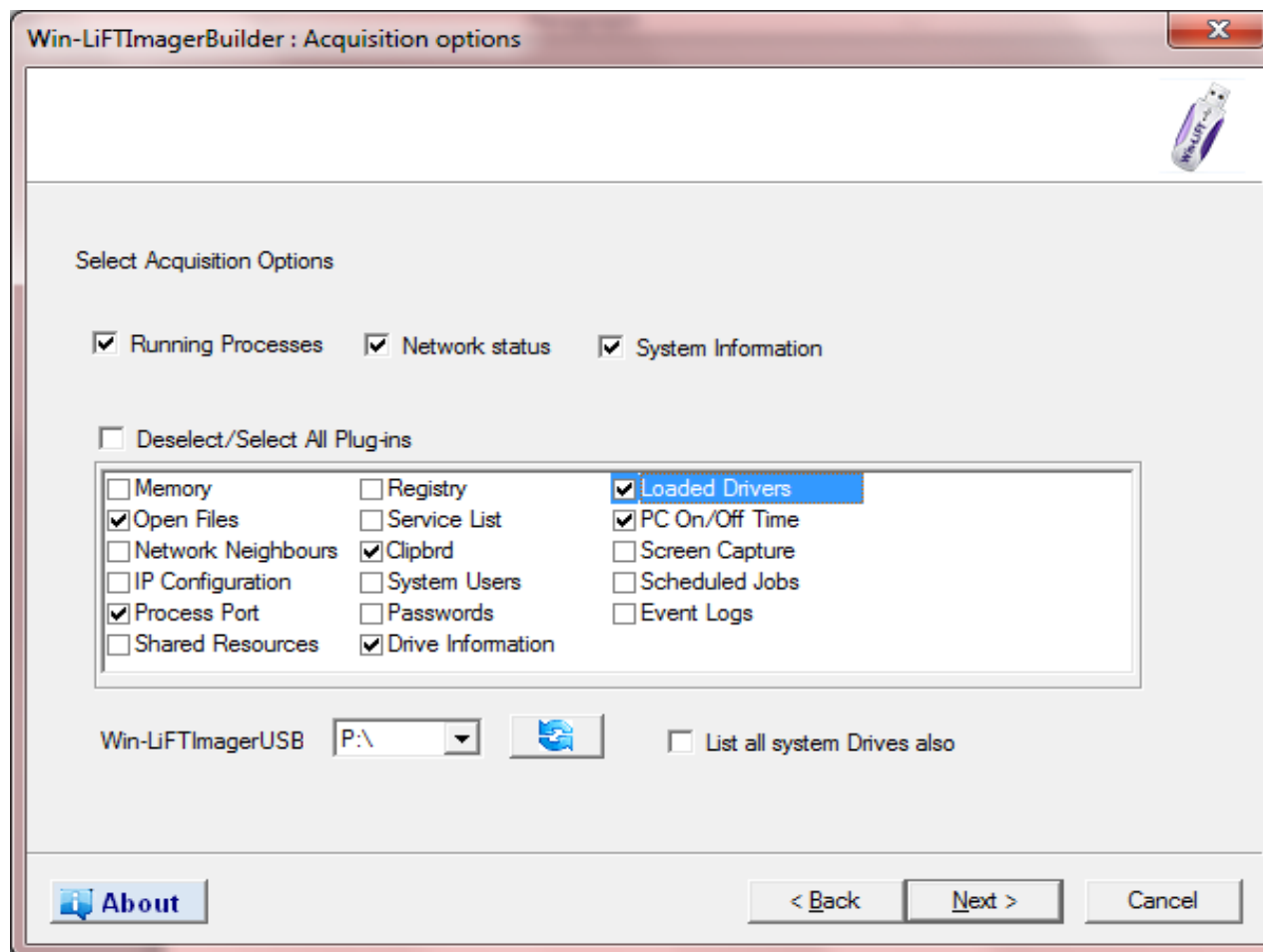
- ❑ C-DAC's Win-LiFT
- ❑ COFEE(Computer Online Forensic Evidence Extractor )
- ❑ EnCase Portable

<https://cyberforensics.in/>

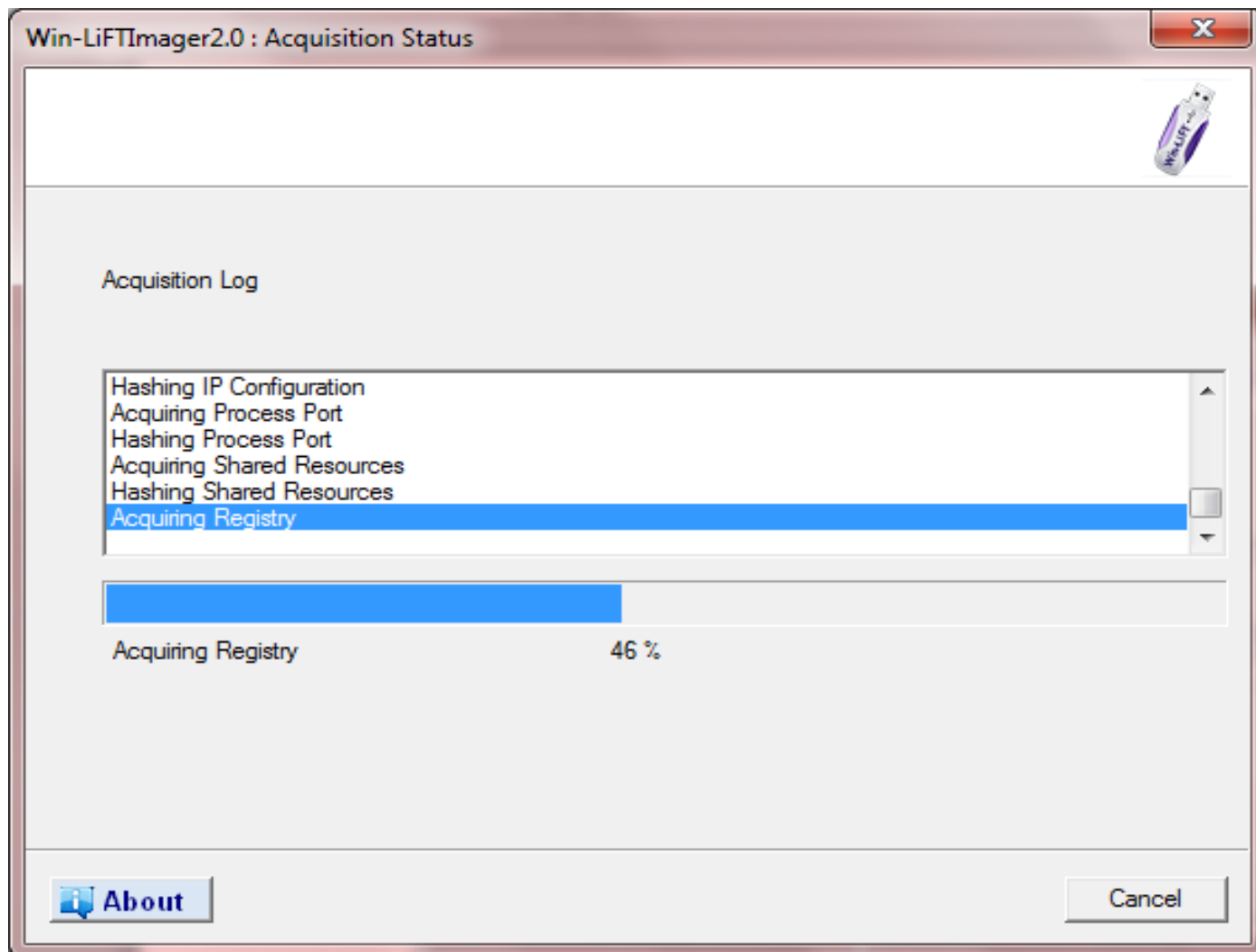












LiveData Analysis

- Root
  - System Users
  - Running Processes
  - PC On/Off Time
  - Network Connections
  - Network Neighbours
  - Installed Drivers
  - Drive Information
  - Stored Passwords
- Registry
  - System Information
  - Computer Name
  - Shutdown Time
  - Timezone Information
  - OS Details
  - Logon Details
- Recently Used
  - Recent Documents
  - Opened/Saved Files
  - Last Visited ExE's
  - Winzip Archives
  - Media Player Files
  - Excel Files
  - Powerpoint Files
  - Wordpad Files
  - Paint Files
  - Run Commands
  - Searched Terms
- AutoRun Softwares
- Services
- Startup Programs
- Removable Devices
- USB Devices
- User/ Group Details
  - User Details
  - Group Details
- User Activities
- Network
  - Network Cards
  - Machines in Network
  - Softwares Installed
- Internet Explorer
  - Typed URLs
  - IE Settings
  - Url History-Days Saved
  - Download Directory
- Memory
  - Processes Running
  - List of DLLs
  - Network Connections
  - Open Sockets
  - Loaded Modules
  - Open Files
  - Command Line Information
- Event Logs
  - Application
  - Security
  - System

ListView Registry Event Logs Bookmarks Search Results Report

Protocol	Source IP	Destination IP	State
TCP	0.0.0.0:135 ...	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445 ...	0.0.0.0:0	LISTENING
TCP	0.0.0.0:912 ...	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1033 ...	127.0.0.1:27015...	ESTABLISHED
TCP	127.0.0.1:1036 ...	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1051 ...	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1073 ...	127.0.0.1:1074 ...	ESTABLISHED
TCP	127.0.0.1:1074 ...	127.0.0.1:1073 ...	ESTABLISHED
TCP	127.0.0.1:1078 ...	127.0.0.1:1079 ...	ESTABLISHED
TCP	127.0.0.1:1079 ...	127.0.0.1:1078 ...	ESTABLISHED
TCP	127.0.0.1:1118 ...	127.0.0.1:1119 ...	ESTABLISHED
TCP	127.0.0.1:1119 ...	127.0.0.1:1118 ...	ESTABLISHED
TCP	127.0.0.1:1120 ...	127.0.0.1:1121 ...	ESTABLISHED
TCP	127.0.0.1:1121 ...	127.0.0.1:1120 ...	ESTABLISHED
TCP	127.0.0.1:5152 ...	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5152 ...	127.0.0.1:1233 ...	CLOSE_WAIT
TCP	127.0.0.1:5354 ...	0.0.0.0:0	LISTENING
TCP	127.0.0.1:27015...	0.0.0.0:0	LISTENING
TCP	127.0.0.1:27015...	127.0.0.1:1033 ...	ESTABLISHED
TCP	172.16.29.18:13...	0.0.0.0:0	LISTENING
TCP	172.16.29.18:12...	172.16.60.200:8...	ESTABLISHED
TCP	172.16.29.18:12...	172.16.50.74:31...	ESTABLISHED
TCP	192.168.115.1:1...	0.0.0.0:0	LISTENING
TCP	192.168.152.1:1...	0.0.0.0:0	LISTENING
UDP	0.0.0.0:445 ...	...	...
UDP	0.0.0.0:500 ...	...	...
UDP	0.0.0.0:1025 ...	...	...
UDP	0.0.0.0:1028 ...	...	...
UDP	0.0.0.0:4500 ...	...	...
UDP	0.0.0.0:53478 ...	...	...
UDP	127.0.0.1:123 ...	...	...
UDP	127.0.0.1:1040 ...	...	...
UDP	127.0.0.1:1270 ...	...	...
UDP	127.0.0.1:1900 ...	...	...
UDP	172.16.29.18:12...	...	...
UDP	172.16.29.18:13...	...	...
UDP	172.16.29.18:13...	...	...
UDP	172.16.29.18:19...	...	...
UDP	172.16.29.18:53...	...	...
UDP	192.168.115.1:1...	...	...

Summary HexView Gallery

1 Item(s) selected.

Item No : 1

Protocol	TCP
Source IP	0.0.0.0:135
Destination IP	0.0.0.0:0
State	LISTENING



# Live Forensic Tools - COFEE

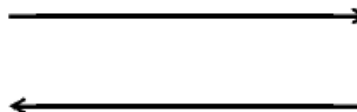


- ☐ Easy to use
- ☐ Capture important "live" computer evidence
- ☐ Special forensics expertise not needed.

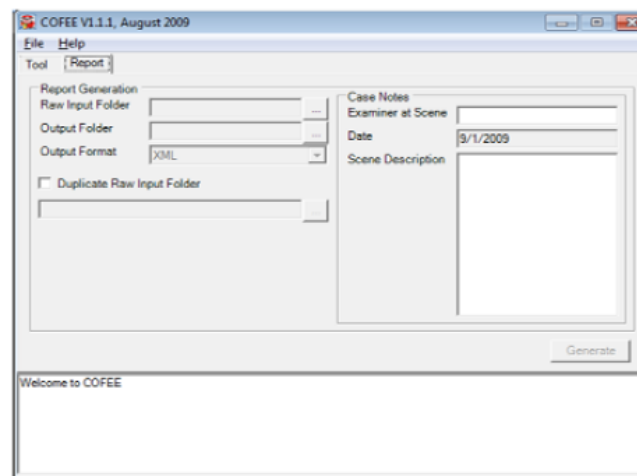
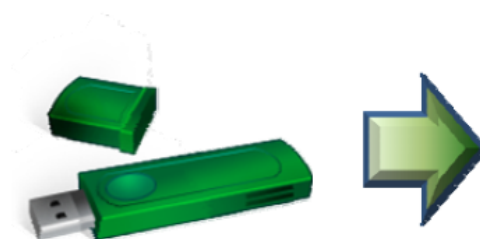
Computer Online Forensics Evidence Extractor



Investigator's Machine



Target Machine



## Computer Online Forensics Evidence Extractor

# Live Forensic Tools - Encase Portable



- ☐ Easy to Use
- ☐ Forensically Sound
- ☐ Ultra-Portable
- ☐ Stealth



# Forensic Tools in Kali Linux

- More than 200 penetration testing tools are packaged in Kali Linux.
- More than 20 tools for forensics packaged inside Kali Linux.

### 1. **Binwalk tool:**

- ❖ searches a specified binary image for executable code and files.

### 2. **Bulk extractor tool:**

- ❖ Extracts credit card numbers, URL links, email addresses etc..
- ❖ Works on compressed data and incomplete or damaged data.

### 3. **HashDeep tool:**

- ❖ For hashing of files.

### 4. **Magic rescue tool:**

- ❖ Performs scanning operations on a blocked device.
- ❖ Recovers files deleted or from corrupted partition.

### 5. **Guymager tool:**

- ❖ Used to acquire media for forensic imagery



## 6. Pdftid tool:

- ❖ Scans pdf files for specific keywords.

## 7. Pdf-parser tool:

## 8. Peepdf tool:

## 9. Autopsy tool:

- ❖ An autopsy is all in one forensic utility for fast data recovery and hash filtering.
- ❖ This tool carves deleted files and media from unallocated space.

## 10. img\_cat tool:

# Case study

- In this case, American Express (Amex) claimed that Mr. Vinhnee had failed to pay his credit card debts, and took legal action to recover the money. But the trial judge determined that Amex failed to authenticate its electronic records, and therefore Amex could not admit its own business records into evidence. Among other problems, the court said that Amex failed to provide adequate information about its computer policy & system control procedures, control of access to relevant databases & programs, how changes to data were recorded or logged, what backup practices were in place, and how **Amex** could provide assurance of continuing integrity of their records.
- The judge pointed out that, "... the focus is not on the circumstances of the creation of the record, but rather on the **circumstances of the preservation of the record** so as to assure that the document being proffered is the same as the document that originally was created ..."
- <http://www.proofspace.com/technology/discovery.php>

# Lessons

- **Document** your access control and backup procedures and policies and test effectiveness of your controls.
- Have the changes to your databases and content/record management system **routinely recorded and logged**.
- **Protect your electronic record** from post-archival tampering with modern data integrity and trusted time-stamping technologies.
- Document the audit procedures you use to provide assurance of the continuing authenticity of the records.
- <http://www.proofspace.com/technology/discovery.php>

## An Incident

- Received a report of a compromised system from a sysadmin
  - Well... of ten systems.
  - Systems had a file which listed the other machines in the same BotNet – 35 on our network
  - Each system we looked at had a different list of systems in that file.
- The Challenge: find all compromised machines.

## Incident, continued

- We chose a system that wasn't doing much (network-wise).
  - Why?
- Pulled all the NetFlows for that system.
- By reading the Flows, we were able to find the signs of a compromise (attack against an RPC service).
- Also saw another host (38.31.107.236) connecting to a backdoor on a port (1524)
- Then connections out from the machine to download a rootkit (to 152.3.127.99).
- Then the BotNet traffic starts up.
- Note: I'm skipping some of the analysis of how we figure out that it was downloading a rootkit, etc.

## Incident, continued

- Looked for all connections from our network to 152.3.127.99
- Sorted the output and got the list of our IP addresses in it.
- We were able to easily pull together the list of compromised machines by looking at the network trail of the one machine.
- In practice, we hand verified a lot of this.
  - It's always good to double check.
  - Compared the machines to those listed in the file on the infected machines.
  - Did other analysis of the systems to ensure that they were compromised. (Forensics in depth?)



## NetForensics: “when?”

- When was the first traffic to a back door?
- When did traffic first seem to successfully connect to ports that previous had no traffic?
- When did BotNet traffic start up?
- Did the machine start transferring lots more traffic at a certain point in time?

## NetForensics: “how?”

- What type of machine is it?
- When you look at it, what vulnerable services, if any, do you see?
- Did unexpected traffic start up shortly after a connection to a service running on the machine?
- Connections to a port's service that isn't used.
  - ftp/http out to a third site
  - Particularly if it is grabbing a rootkit.



## NetForensics: “who?”

- Who is connecting to it that you don't expect to connect?
- Which IP caused the compromise?
  - If you know when/how it was compromised.
- Who connected to the backdoors?
- Look for weird connections.
  - May be a backdoor or a covert channel.

## NetForensics: “what?”

- How much data was transferred?
  - Useful if you fear that information was stolen.
  - If they only transferred 500k of data, they didn't take the 20 Gig master plans for the new virtual-hyper-cyber-widget of doom.
- What was on the machine in the first place?



## NetForensics: “why?”

- In my experience, NetFlows aren't so useful for determining motive.
- Sorry.



## Hackers: Who are they?

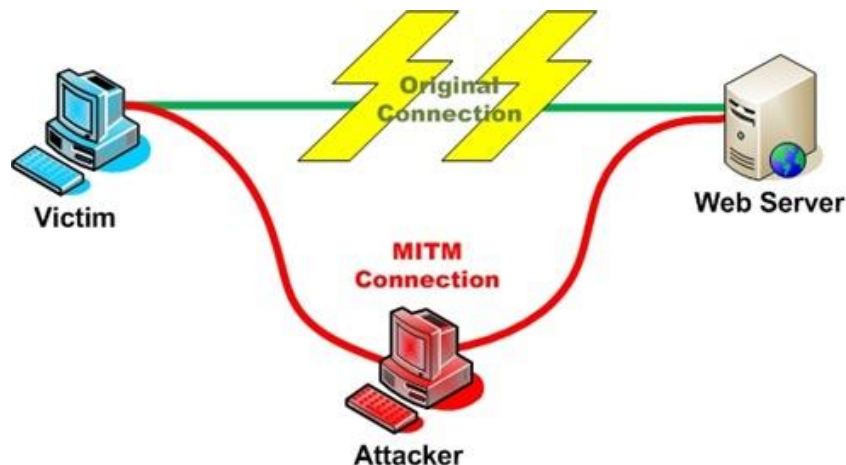
Hackers are people who attack computers or intercept data from servers for malicious purposes

Why?

- Financial Gain
- Blackmail
- Vengeance

# SSL-Authentication Attack

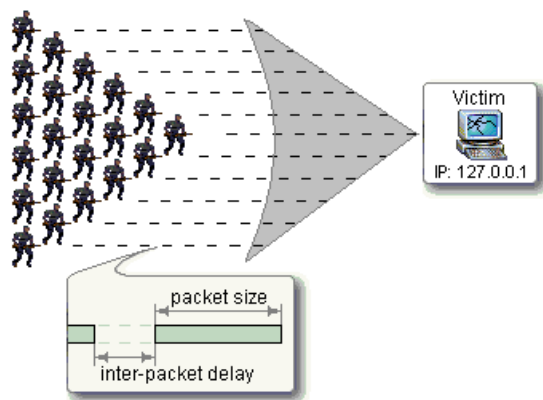
- “Man-in-the-Middle” Attack



- Hacker hijacks User Data
- Bypasses nearly all site security

# Ping Flood

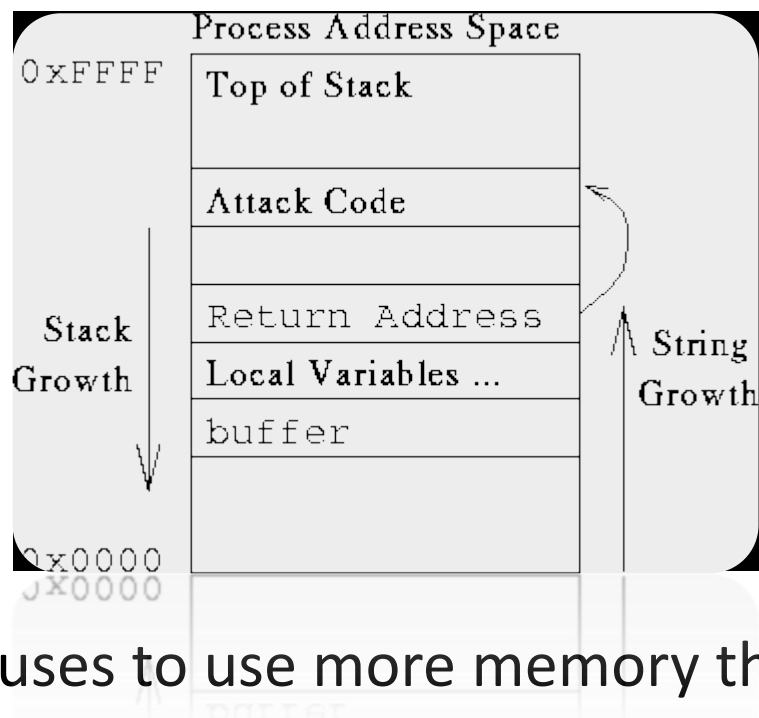
- Often from many computers
- Ping used for testing latency



- Huge amounts of pinging can slow a server and make it unusable

# Buffer Overflow

- Buffer is the window of space allotted for use in a computer



- Hackers use viruses to use more memory than the buffer can handle
- Often causes computer or server to crash

# SQL-Injection

- Executes malicious code with purposeful errors

SQL Injection.

User-Id:   
Password:

`select * from Users where user_id= ' srinivas '  
and password = ' mypassword '`

User-Id:   
Password:

`select * from Users where user_id= ' ' OR 1 = 1; /* '  
and password = ' */-- '`

9lessons.blogspot.com

- Allows access into the computer's or server's databases



# Network Security Tools



- Wireshark - monitors all information entering and leaving the computer

- Snort - detects intrusions into the system and logs them for further examination on servers AND networks



<http://www.wireshark.org/>

<http://www.snort.org/>



## Network Security Tools (cont.)

- Perl - programming language used for web development and interfacing with servers, files and databases
- MySQL - engine used for managing databases on a server or computer



# Attack Log Generation



## Goals

- Generate attack log identical to real log file
- Randomly select IP address from list
- Randomly select network protocol
- Randomly generate timestamps in chronological order

```
1  #!/usr/bin/perl
2
3  $delimiter = "[**]";
4  @attackList = ("SQL_Injection", "Buffer_Overflow", "SSL_Auth", "Ping_Flood");
5  @IPList      = ("212.77.12.48", "88.82.91.86", "116.119.121.63");
6
7  $timestamp = 20110608000000;
8  $destIP = "131.187.117.93";
9
10 sub decideSource {
11     my $random = rand();
12     $sourceIP = $IPList[0] if $random le .18;
13     $sourceIP = $IPList[1] if $random > .18 && $random le .80;
14     $sourceIP = $IPList[2] if $random > .80;
15 }
16
17 %country = (
18     "212.77.12.48" => "Italy",
19     "88.82.91.86" => "Russia",
20     "116.119.121.63" => "China"
21 );
22 $hashRef = \%country;
23
24 sub decideAttack {
25
26 sub decideProtocol {
27     my $random = rand();
28     $protocol = "{ICMP}" if $random le .33;
29     $protocol = "{TCP}"  if $random > .33 && $random le .66;
30     $protocol = "{UDP}"  if $random > .66;
31 }
32
33 sub accumTimestamp {
34
35 sub alterTime {
36     $x = 1;
37
38 sub decideFrequency {
39
40 sub threatLevel {
41
42 sub writeToFile {
43     writeToFile;
```

```
sub decideSource {  
  my $random = rand();  
  $sourceIP = $IPList[0] if $random le .18;  
  $sourceIP = $IPList[1] if $random > .18 && $random le .80;  
  $sourceIP = $IPList[2] if $random > .80;  
}
```

Select an  
IP  
address

Select  
corresponding  
country

```
sub decideProtocol {  
  my $random = rand();  
  $protocol = "{ICMP}" if $random le .33;  
  $protocol = "{TCP}" if $random > .33 && $random le .66;  
  $protocol = "{UDP}" if $random > .66;  
}
```

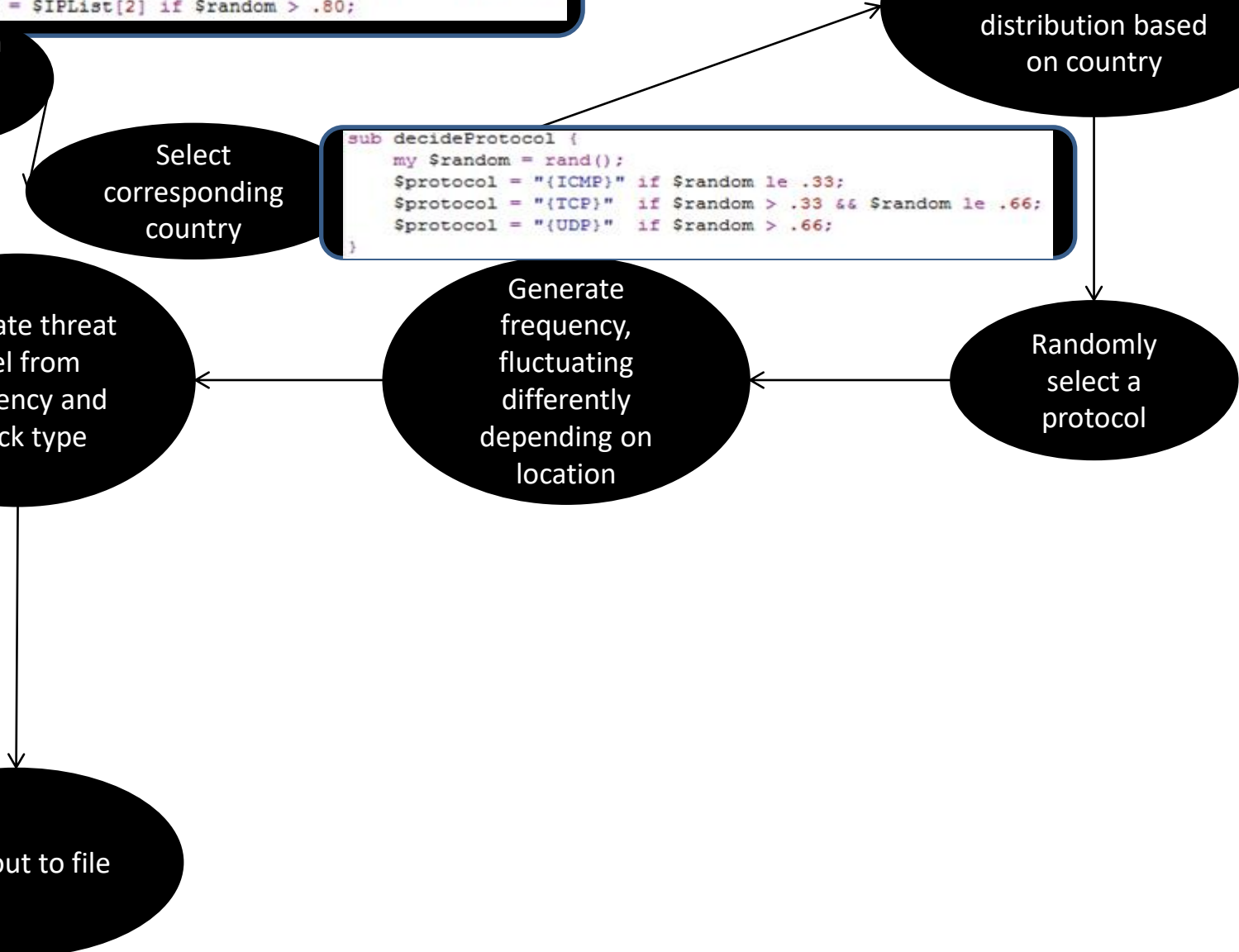
Select attack type  
from uneven  
distribution based  
on country

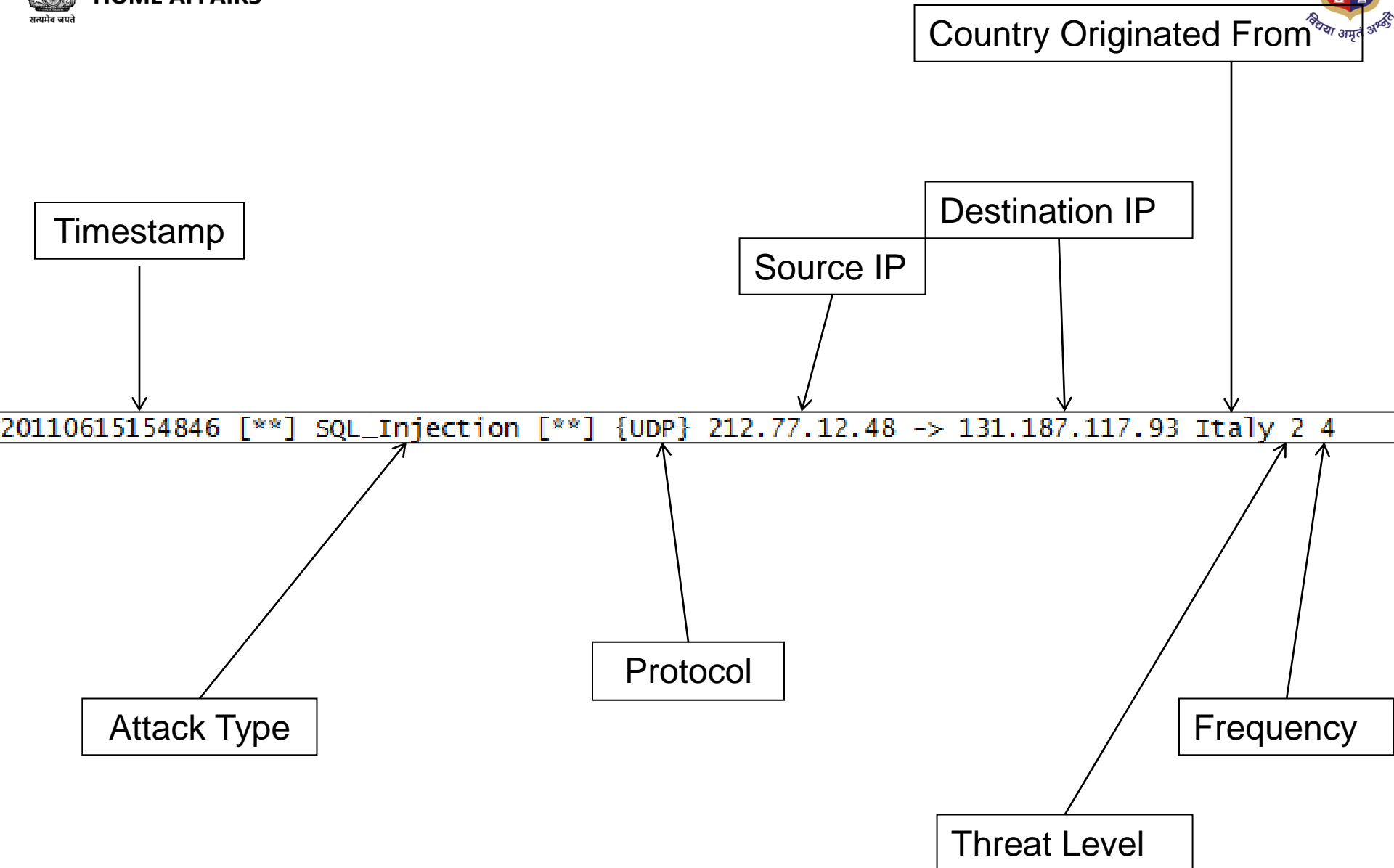
Randomly  
select a  
protocol

Generate  
frequency,  
fluctuating  
differently  
depending on  
location

Calculate threat  
level from  
frequency and  
attack type

Print out to file







```
20110615032918 [**] Ping_Flood [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 1 1
20110615035424 [**] Ping_Flood [**] {UDP} 116.119.121.63 -> 131.187.117.93 China 2 3
20110615045026 [**] SSL_Auth [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 3 2
20110615060325 [**] SQL_Injection [**] {UDP} 88.82.91.86 -> 131.187.117.93 Russia 2 2
20110615070803 [**] SQL_Injection [**] {TCP} 212.77.12.48 -> 131.187.117.93 Italy 2 2
20110615080900 [**] SSL_Auth [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 3 4
20110615090921 [**] SSL_Auth [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 3 4
20110615101145 [**] SSL_Auth [**] {UDP} 88.82.91.86 -> 131.187.117.93 Russia 5 7
20110615103704 [**] Ping_Flood [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 2 4
20110615110727 [**] Ping_Flood [**] {UDP} 116.119.121.63 -> 131.187.117.93 China 2 3
20110615120639 [**] SSL_Auth [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 5 7
20110615131124 [**] SSL_Auth [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 3 2
20110615141230 [**] SSL_Auth [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 3 4
20110615144003 [**] Ping_Flood [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 2 3
20110615154846 [**] SQL_Injection [**] {UDP} 212.77.12.48 -> 131.187.117.93 Italy 2 4
20110615161539 [**] Ping_Flood [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 2 3
20110615172233 [**] SQL_Injection [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 2 4
20110615181955 [**] SSL_Auth [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 4 5
20110615191924 [**] Buffer_Overflow [**] {TCP} 116.119.121.63 -> 131.187.117.93 China 3
20110615203327 [**] SQL_Injection [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 4 7
20110615210427 [**] Ping_Flood [**] {UDP} 116.119.121.63 -> 131.187.117.93 China 2 4
20110615220246 [**] SQL_Injection [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 2 3
20110615230128 [**] SSL_Auth [**] {UDP} 88.82.91.86 -> 131.187.117.93 Russia 3 2
20110615235852 [**] Buffer_Overflow [**] {TCP} 212.77.12.48 -> 131.187.117.93 Italy 3 3
20110616005532 [**] Buffer_Overflow [**] {ICMP} 212.77.12.48 -> 131.187.117.93 Italy 3 4
20110616020022 [**] Buffer_Overflow [**] {UDP} 88.82.91.86 -> 131.187.117.93 Russia 3 4
20110616030039 [**] SSL_Auth [**] {TCP} 88.82.91.86 -> 131.187.117.93 Russia 4 6
20110616035407 [**] Buffer_Overflow [**] {UDP} 88.82.91.86 -> 131.187.117.93 Russia 4 7
20110616042141 [**] Ping_Flood [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 2 3
20110616052310 [**] SSL_Auth [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 3 3
20110616055002 [**] Ping_Flood [**] {TCP} 116.119.121.63 -> 131.187.117.93 China 2 4
20110616065435 [**] SSL_Auth [**] {UDP} 88.82.91.86 -> 131.187.117.93 Russia 5 7
20110616080452 [**] Buffer_Overflow [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 2 1
20110616091008 [**] Buffer_Overflow [**] {ICMP} 212.77.12.48 -> 131.187.117.93 Italy 3 3
```





गृह मंत्रालय  
MINISTRY OF  
**HOME AFFAIRS**

राष्ट्रीय न्यायिक विज्ञान विश्वविद्यालय  
National Forensic Sciences University



# PARSING THE LOG



## Parsing the Attack Log: Goals

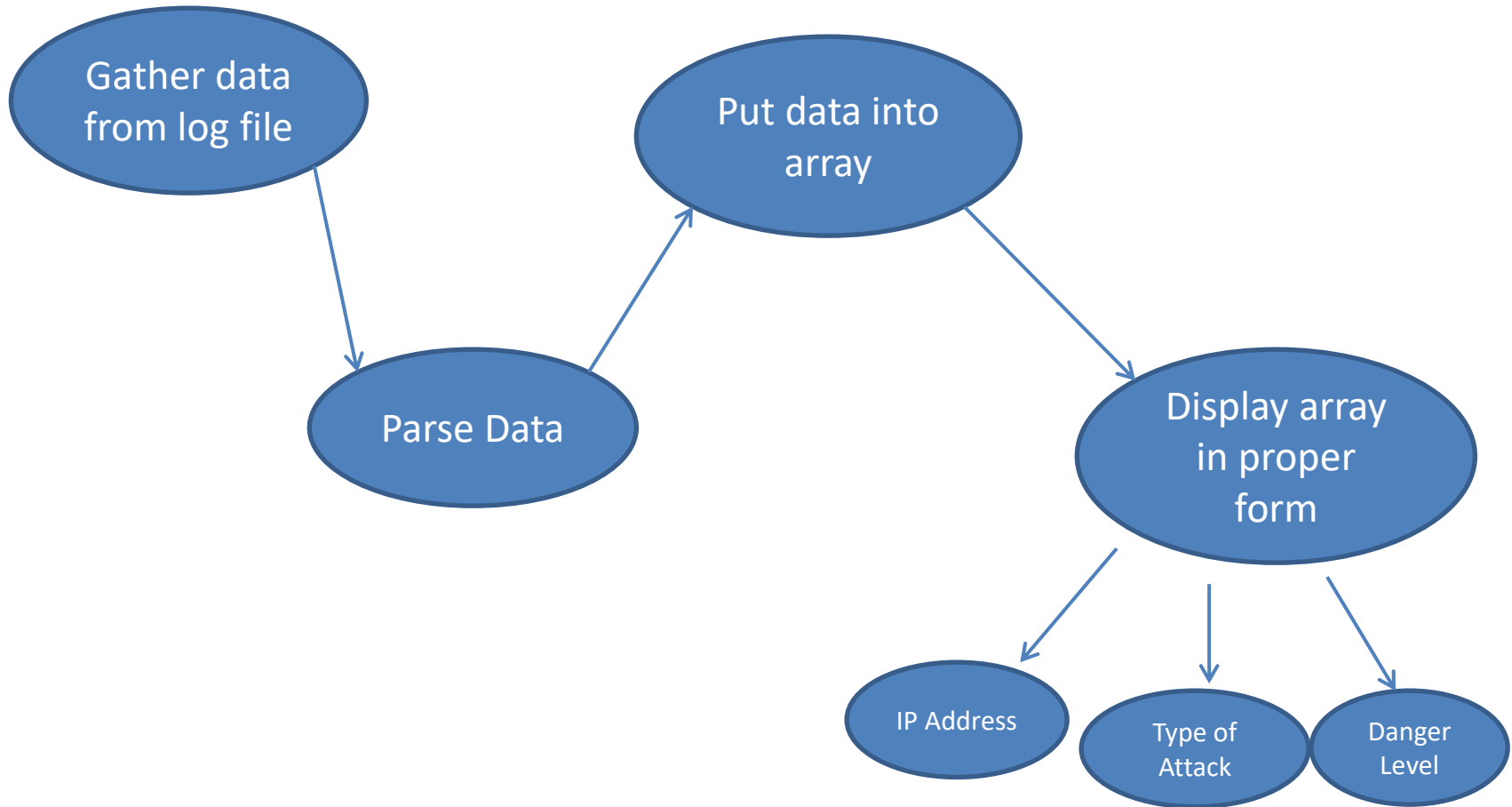
- Compile the attack log into an organized list
- Calculate “Danger Level” of each entry
- Calculate the Threshold level
- Compile threats into a table
- Import tables into MySQL



# Plan

- Load log file into our Perl code
- Organize anomalies into a list
- Modify Danger Level based upon certain parameters
- Organize data into table
- Upload table to MySQL database

# Part 1



# Explanation of the Danger Level

- Wanted Danger Level to represent frequency, time, and logged Threat Level
- $$\text{Danger Level} = \frac{((\text{Frequency/Time}) + \text{Threat Level})}{2}$$
- Allowed us to represent all “Danger” factors in one variable

SEVERE

HIGH

ELEVATED

GUARDED

LOW



## Part 2

1. Gather data from log file

```
20110615141230 [**] SSL_Auth [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 3 4
20110615144003 [**] Ping_Flood [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 2 3
20110615154846 [**] SQL_Injection [**] {UDP} 212.77.12.48 -> 131.187.117.93 Italy 2 4
```

2. Parse Data with Threshold

2.5 Get IP's and look for repeats

3. Apply to remove false alarms

```
my $Dangerlevel = (($FRQ/1)+$lv1)/2;
$SumDanger+=$Dangerlevel;
$AveDanger=($SumDanger/300);
$Diff=($Dangerlevel-$AveDanger)**2;
$TotalDanger+=$Diff;
$StanDev=sqrt($TotalDanger/299);

$Thresh = $AveDanger+((1.96*($StanDev))/(sqrt(300)));
print $Thresh;
```

4. Put data into array

5. Display array in proper form

6. Upload forms to database

```
20110615141230 [**] SSL_Auth [**] {ICMP} 88.82.91.86 -> 131.187.117.93 Russia 3 4
20110615144003 [**] Ping_Flood [**] {ICMP} 116.119.121.63 -> 131.187.117.93 China 2 3
20110615154846 [**] SQL_Injection [**] {UDP} 212.77.12.48 -> 131.187.117.93 Italy 2 4
```

## Explanation of the Threshold

- Wanted to capture all entries that were above the average Danger Level
- Used a confidence interval
- Interval gave a range for the mean
- Upper limit of the interval became our threshold

```
my $Dangerlevel = (($FRQ/1)+$lvl)/2;
```

```
$SumDanger+=$Dangerlevel;
```

```
$AveDanger=($SumDanger/300);
```

```
$Diff= (($Dangerlevel-$AveDanger)**2);
```

```
$TotalDanger+=$Diff;
```

```
$StanDev=sqrt (($TotalDanger/299));
```

```
$Thresh = $AveDanger+((1.96*($StanDev))/(sqrt(300)));
```

```
print $Thresh;
```

Average Danger Level

Standard Deviation

Confidence Interval



116.119.121.63	Ping_Flood	3
116.119.121.63	Buffer_Overflow	3
116.119.121.63	Ping_Flood	2
212.77.12.48	Buffer_Overflow	3
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	6
116.119.121.63	Ping_Flood	3
212.77.12.48	Buffer_Overflow	2
116.119.121.63	Buffer_Overflow	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	3
88.82.91.86	SSL_Auth	3
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
212.77.12.48	Buffer_Overflow	3
212.77.12.48	Buffer_Overflow	3
88.82.91.86	SSL_Auth	5
88.82.91.86	Ping_Flood	5
212.77.12.48	Buffer_Overflow	3
88.82.91.86	SQL_Injection	3
88.82.91.86	SSL_Auth	3
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	4
212.77.12.48	Buffer_Overflow	5
88.82.91.86	SSL_Auth	2
88.82.91.86	SSL_Auth	3
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	4
116.119.121.63	Ping_Flood	3
88.82.91.86	SSL_Auth	5
212.77.12.48	Buffer_Overflow	2
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	6
116.119.121.63	Ping_Flood	3
88.82.91.86	SQL_Injection	3
116.119.121.63	Ping_Flood	2
88.82.91.86	SQL_Injection	3
212.77.12.48	SSL_Auth	4
88.82.91.86	SSL_Auth	4
212.77.12.48	Buffer_Overflow	2
212.77.12.48	Buffer_Overflow	3
88.82.91.86	SSL_Auth	5
116.119.121.63	Ping_Flood	3
88.82.91.86	SSL_Auth	6
116.119.121.63	Ping_Flood	3
116.119.121.63	Ping_Flood	2
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	3
88.82.91.86	SSL_Auth	3
116.119.121.63	Buffer_Overflow	4
88.82.91.86	SSL_Auth	2
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	5
88.82.91.86	SQL_Injection	5
212.77.12.48	Buffer_Overflow	3
88.82.91.86	Ping_Flood	5
212.77.12.48	Buffer_Overflow	3
88.82.91.86	SSL_Auth	3
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	4
212.77.12.48	Buffer_Overflow	4
88.82.91.86	SSL_Auth	5
88.82.91.86	Ping_Flood	5
212.77.12.48	SSL_Auth	3
88.82.91.86	SSL_Auth	3
116.119.121.63	Ping_Flood	2
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	4
88.82.91.86	SSL_Auth	6

## Output of our Code:

- We imported certain data from the log file to MySQL and into this table
- Listed IP, Type, and Danger Level
- IP addresses differed among the processes
- Contained 4 Types of attacks; SQL Injection, Buffer Overflow, SSL Authorization, and Ping Overflow



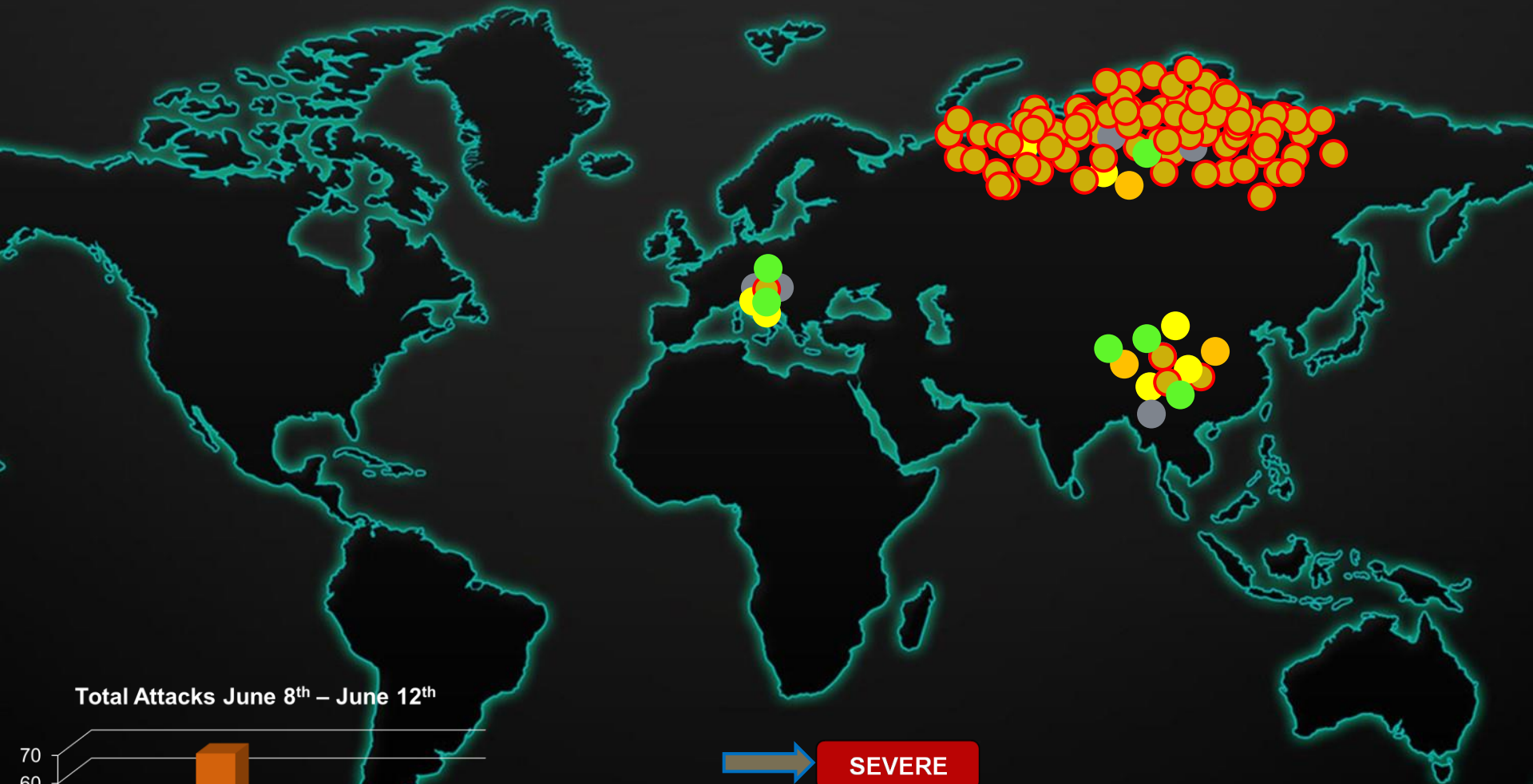


IP	Type	Danger_Level
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	Ping_Flood	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	6
88.82.91.86	SQL_Injection	6
88.82.91.86	Ping_Flood	5
88.82.91.86	SSL_Auth	5
88.82.91.86	Ping_Flood	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	Ping_Flood	5
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	Buffer_Overflow	5
88.82.91.86	SQL_Injection	4
88.82.91.86	SQL_Injection	6
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	6
88.82.91.86	Buffer_Overflow	4
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SQL_Injection	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	5
88.82.91.86	SSL_Auth	6
88.82.91.86	SSL_Auth	6

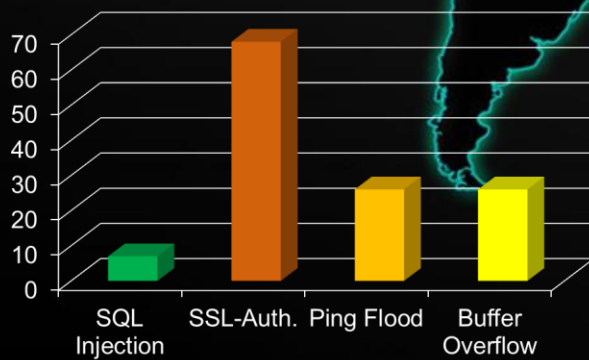
- We listed the anomalies in the attack log
- These entries all had a Danger Level that was more than our Threshold
- We registered 170 anomalies/threats to the system
- We tracked through a series of CPU's being used as botnets to find the IP address of the controller

We traced the source IP addresses of the attacks to certain locations. We will now show you a demonstration illustrating where the attacks came from:

# Map of all Attacks Throughout the 5 Days



Total Attacks June 8<sup>th</sup> – June 12<sup>th</sup>



Legend:

- Red circle = SSL Authentication
- Orange circle = Ping Flood
- Yellow circle = Buffer Overflow
- Green circle = SQL Injection