Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

The Use of Measures in Security Analysis

James McCurley

Sponsored by the U.S. Department of Defense

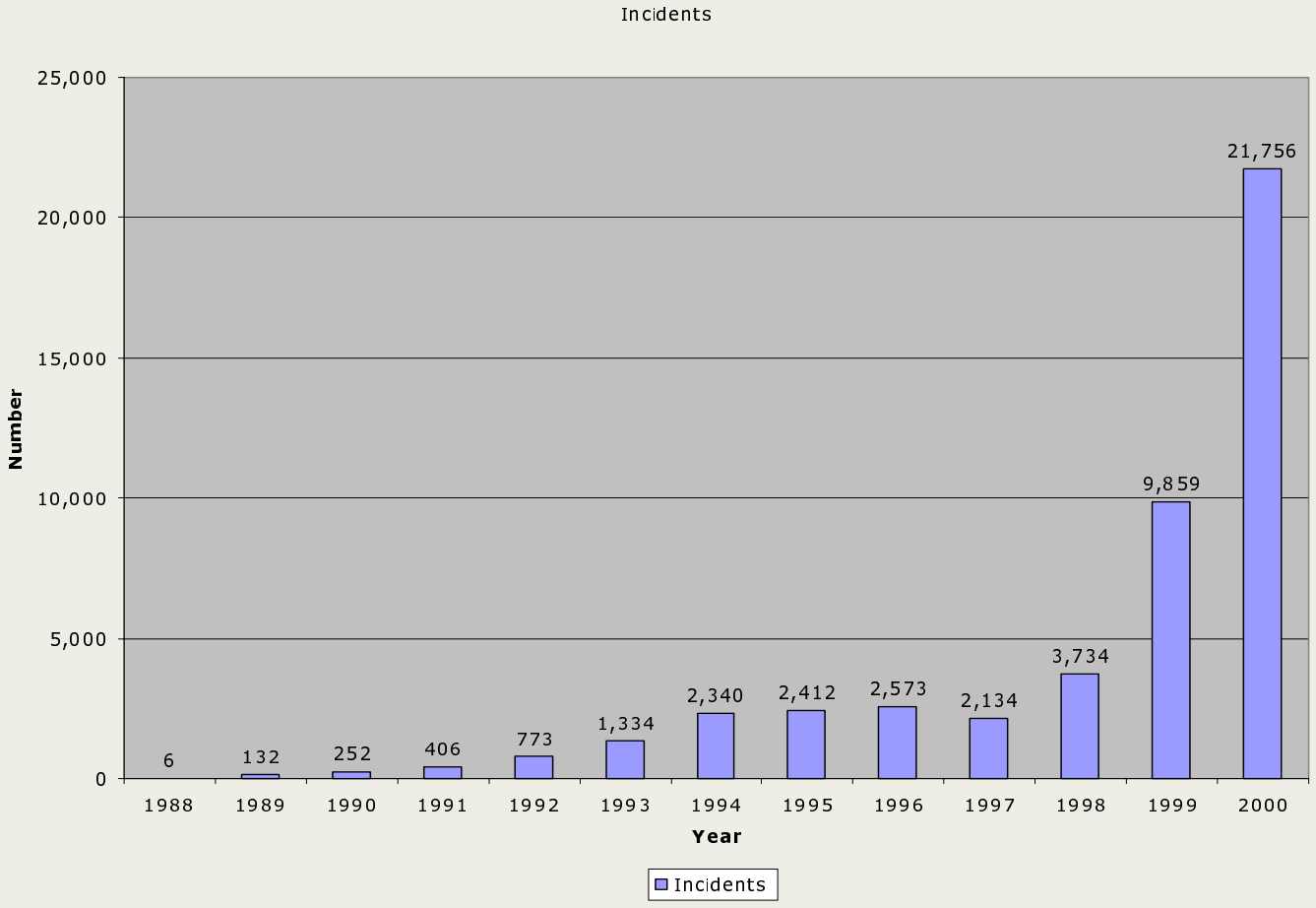
Carnegie Mellon
Software Engineering Institute

News from Nov 2001

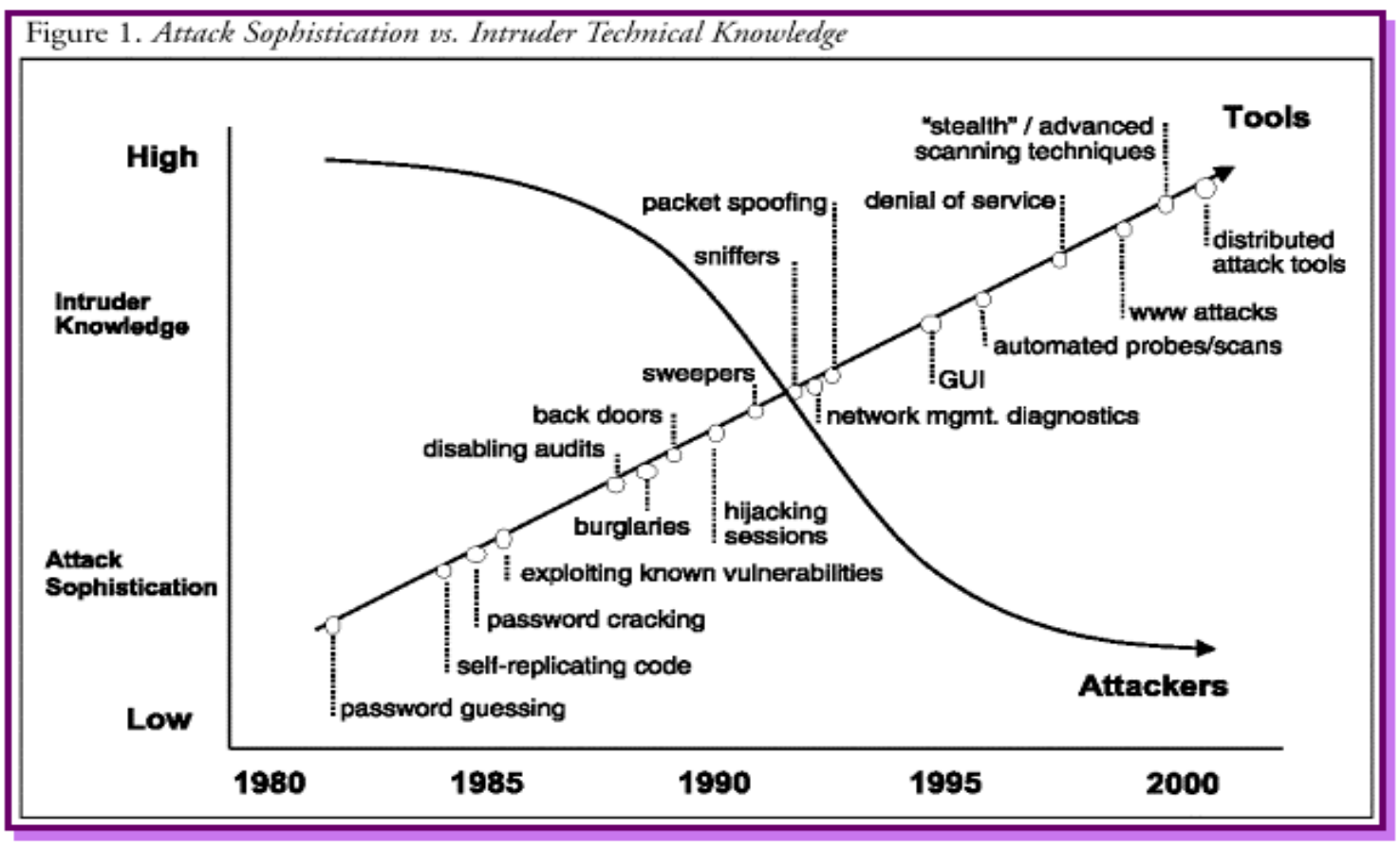
- Experts Say Key Internet Servers Vulnerable to Attack
- Printers could be security risk
- Hacker: Don't Bank on IBM Security System
- White House: Prepare for Super-Hackers
- Report: Net threat looms for global firms
- Playboy Says Hacker Stole Customer Info
- Record-breaking year for security incidents expected
- Hybrid viruses set to become bigger threat

Carnegie Mellon
Software Engineering Institute

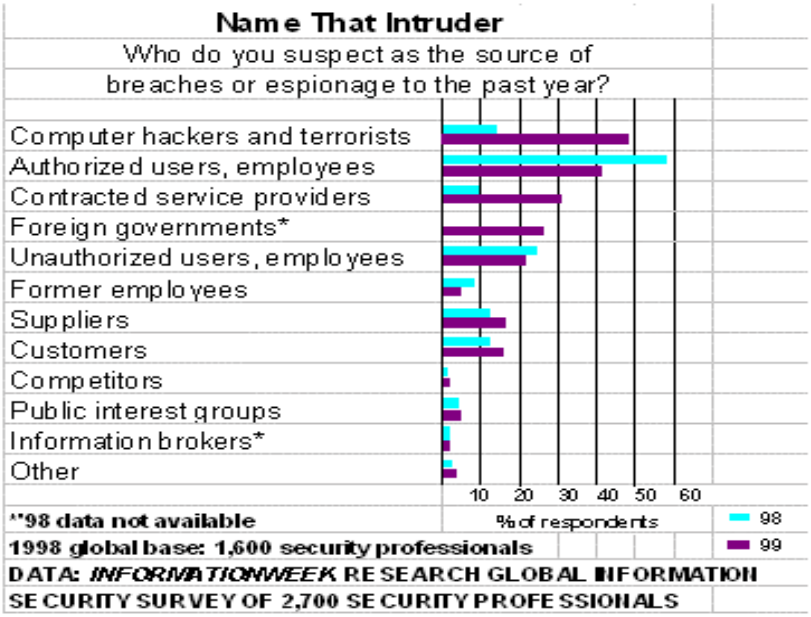
Incidents Reported to CERT



Attack evolution



Who is causing this?



Carnegie Mellon
Software Engineering Institute

Incident Impact: Report to Law Enforcement

Document all losses your organization suffered as a result of the incident. These could include the:

- estimated number of hours spent in response and recovery. (Multiply the number of participating staff by their hourly rates.)
- cost of temporary help
- cost of damaged equipment
- value of data lost
- amount of credit given to customers because of the inconvenience
- loss of revenue
- value of any "trade secret" information

Carnegie Mellon
Software Engineering Institute

Defensive Strategy & Tactics

The “Lockdown” approach:

- Inventory
- Certification/accreditation
- Common Criteria
- BS 7799/ISO 17799
- Audit Standards
- SSE-CMM

Deployment of firewalls, authentication technologies, intrusion detection systems, patch vulnerabilities.

- provides a starting point for security
- personnel often become overwhelmed
- relies on automation
- hackable

Carnegie Mellon
Software Engineering Institute

CERT Incident Handling: Analyses

Analyses reports:

- - determine attack method
- - correlate with other reports
- - determine scope and magnitude
- - what can be learned from this attack
  - determine if new type of attack
  - identify a change in frequency of attack method
  - identify need for new defences or countermeasures

- provides feedback to reporting sites involved

Carnegie Mellon
Software Engineering Institute

Empirical Baselines

Purpose: Build tools to establish traffic baselines between netblocks and hosts. Detect suspicious activity as deviations from these baselines.

Approach: Build and estimate models for time and service based traffic between netblocks. Extend to selected hosts.

Status:

- Approach developed. Adequate volumes of data being collected.
- Preliminary results identify non-routable addresses being passed by border routers.

Carnegie Mellon
Software Engineering Institute

Baseline Approaches

Build time series models (ARIMA, Fourier series, Filters) of dependent variables:

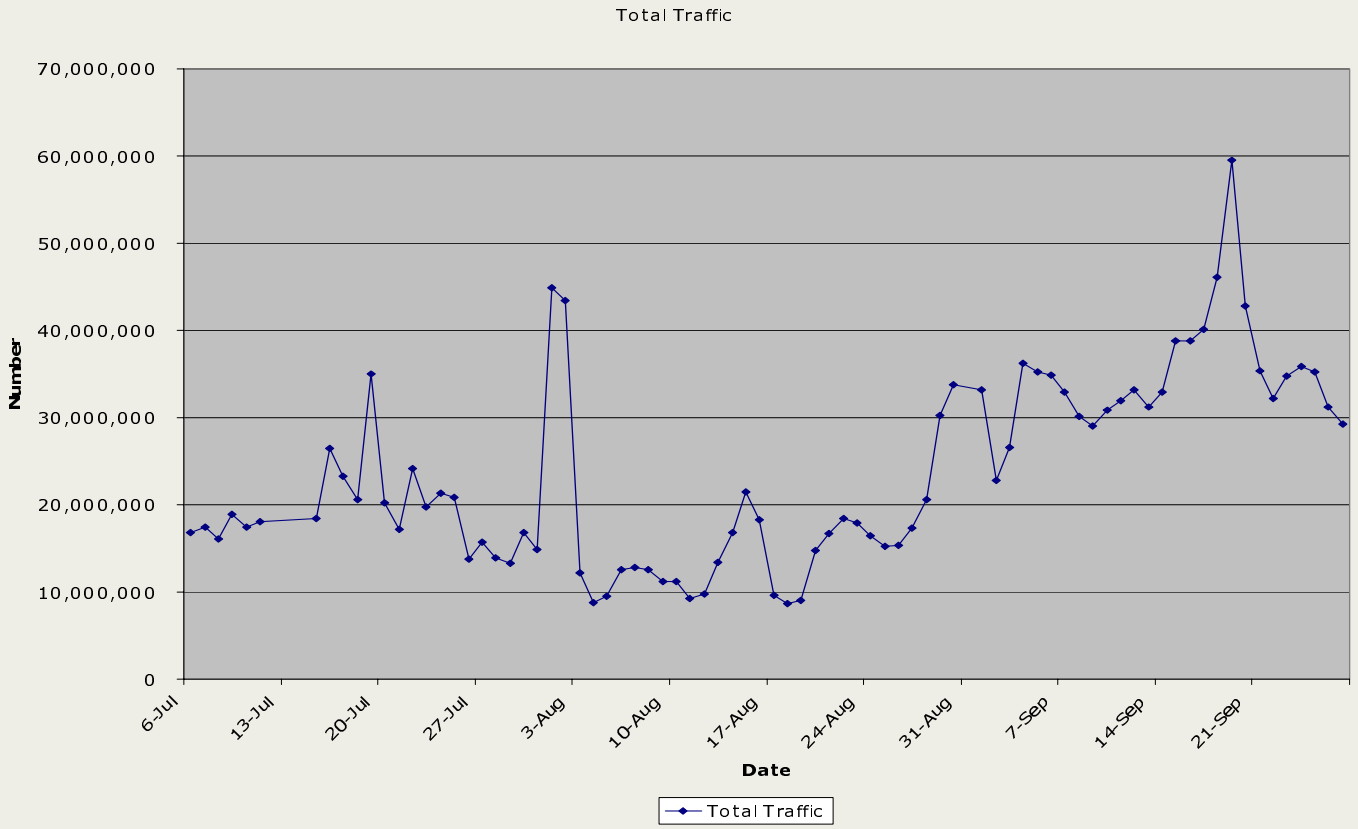
- volumes (bytes, packets, flows per unit time)

on independent variables:

- time of day
- day of week
- service (port/protocol)
- source netblock/host
- destination netblock/host

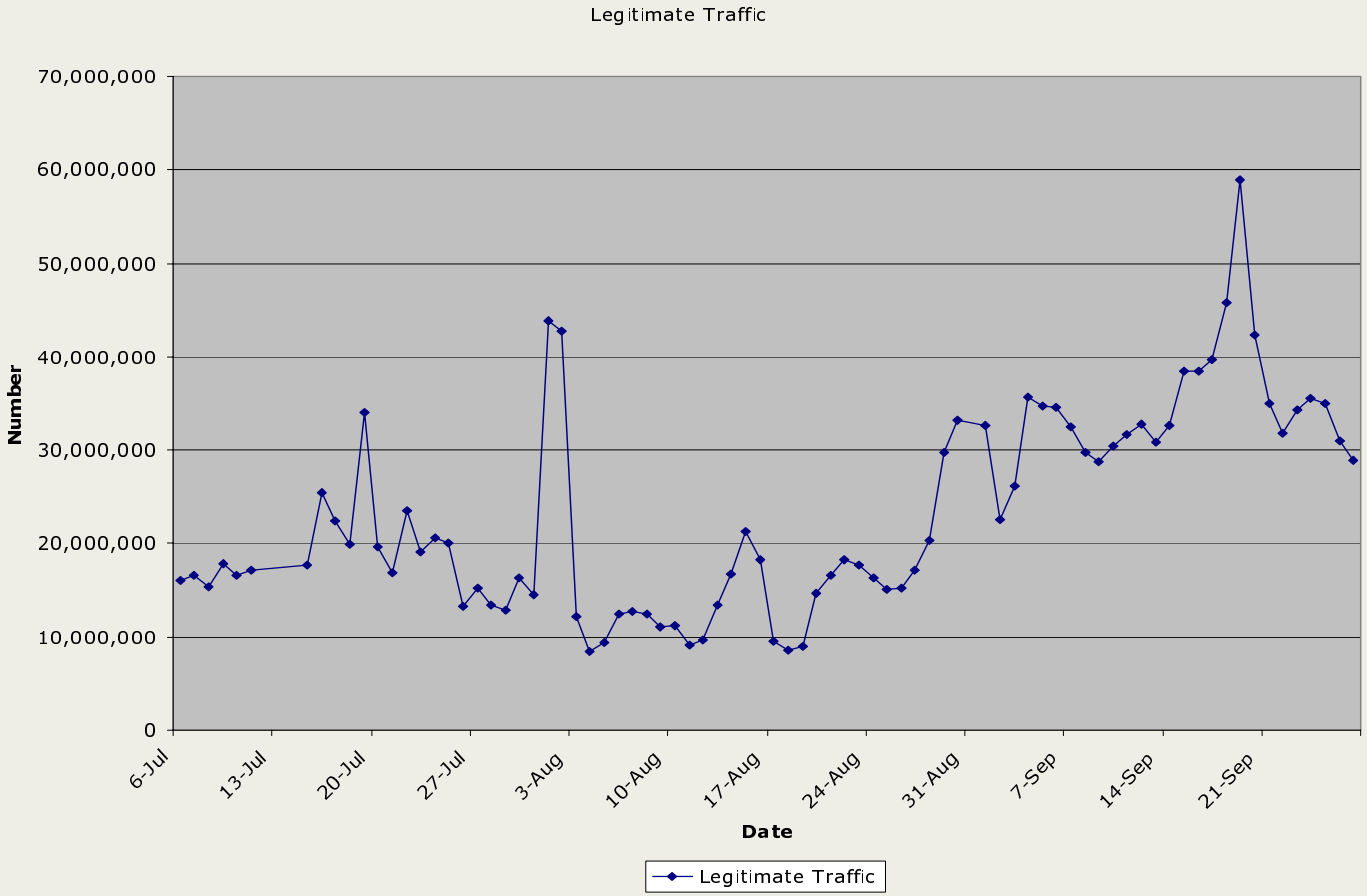
Carnegie Mellon
Software Engineering Institute

Total Network Traffic



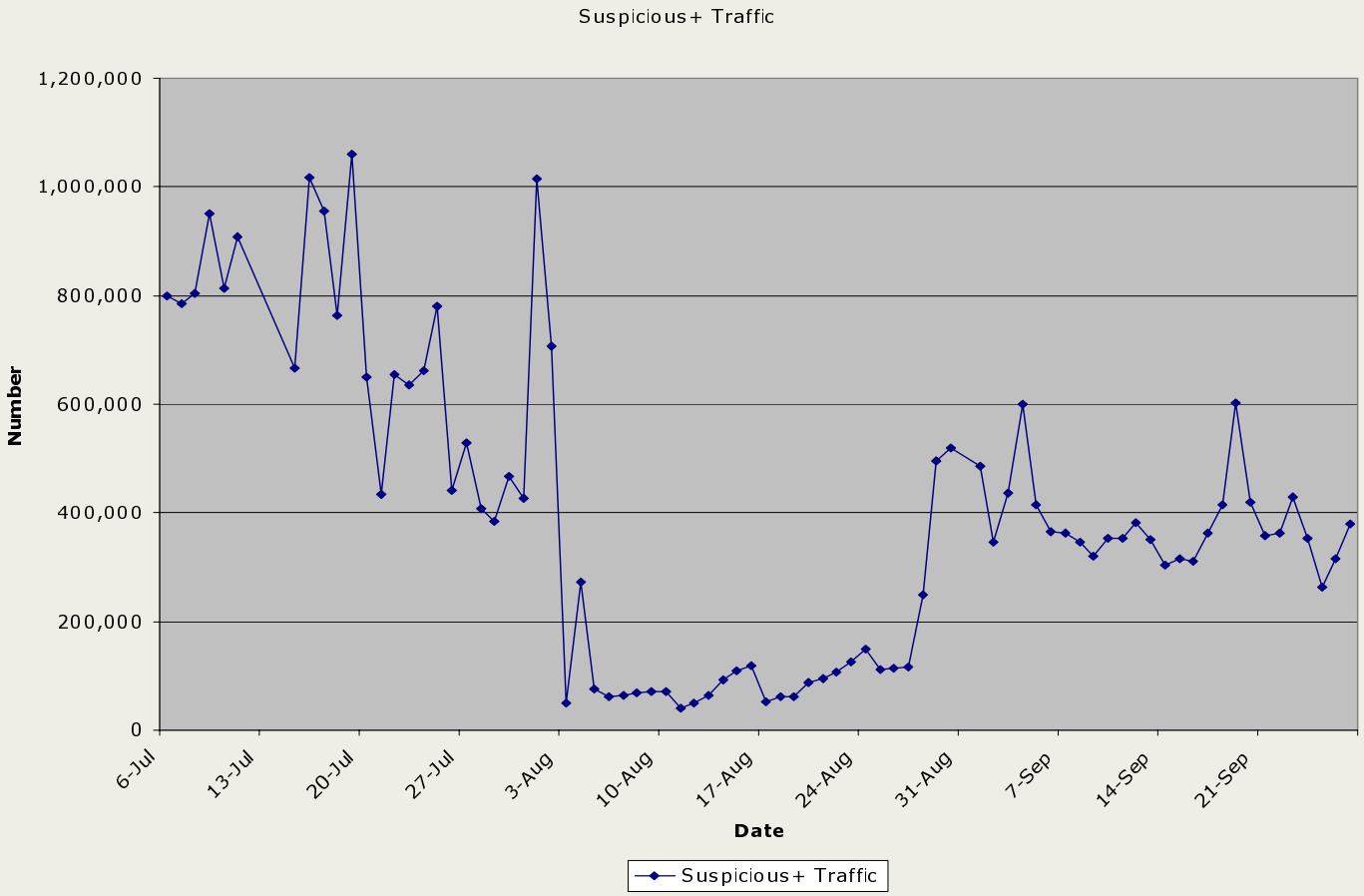
Carnegie Mellon
Software Engineering Institute

Known Legitimate Traffic



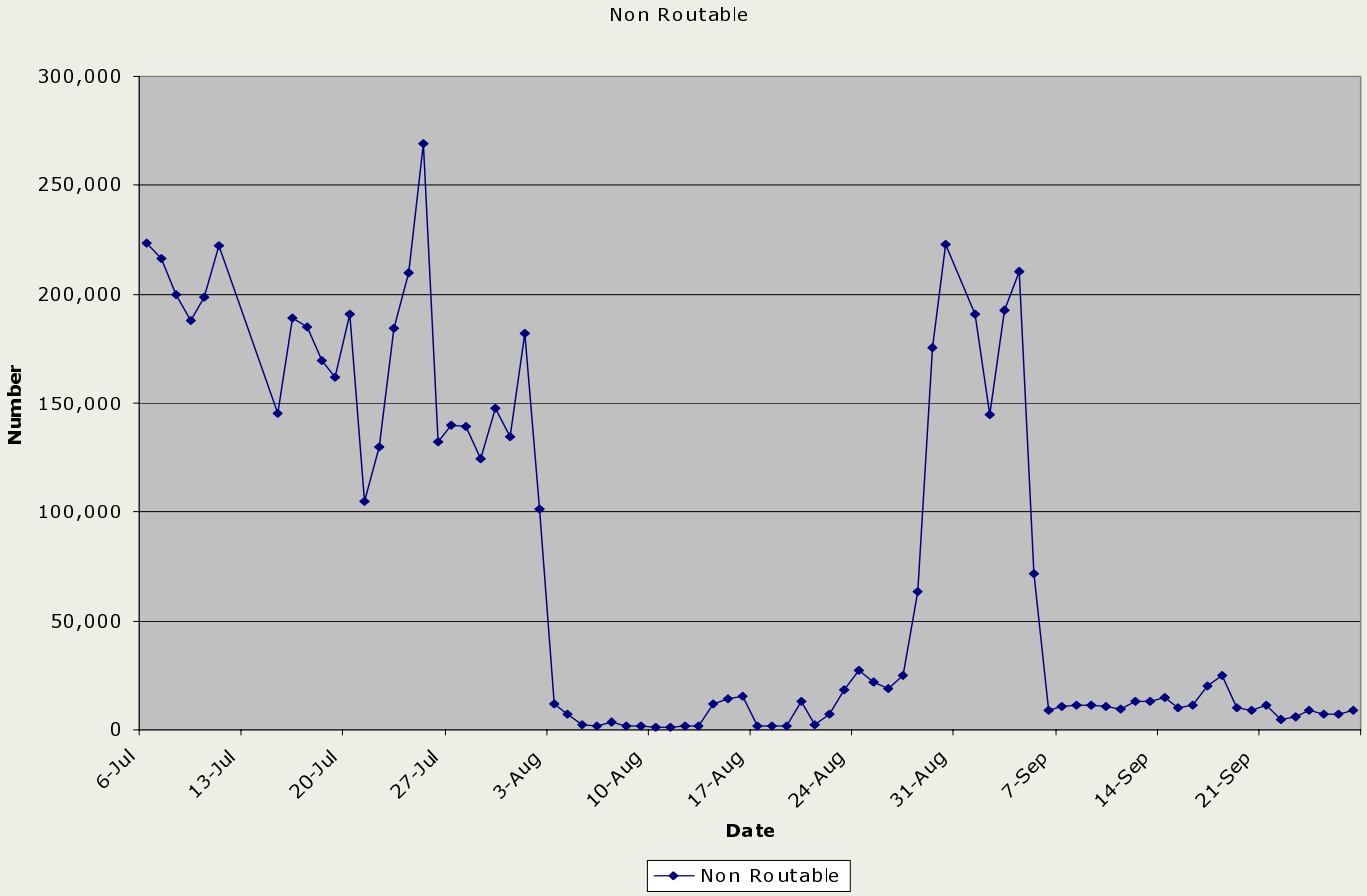
Carnegie Mellon
Software Engineering Institute

Suspicious Traffic



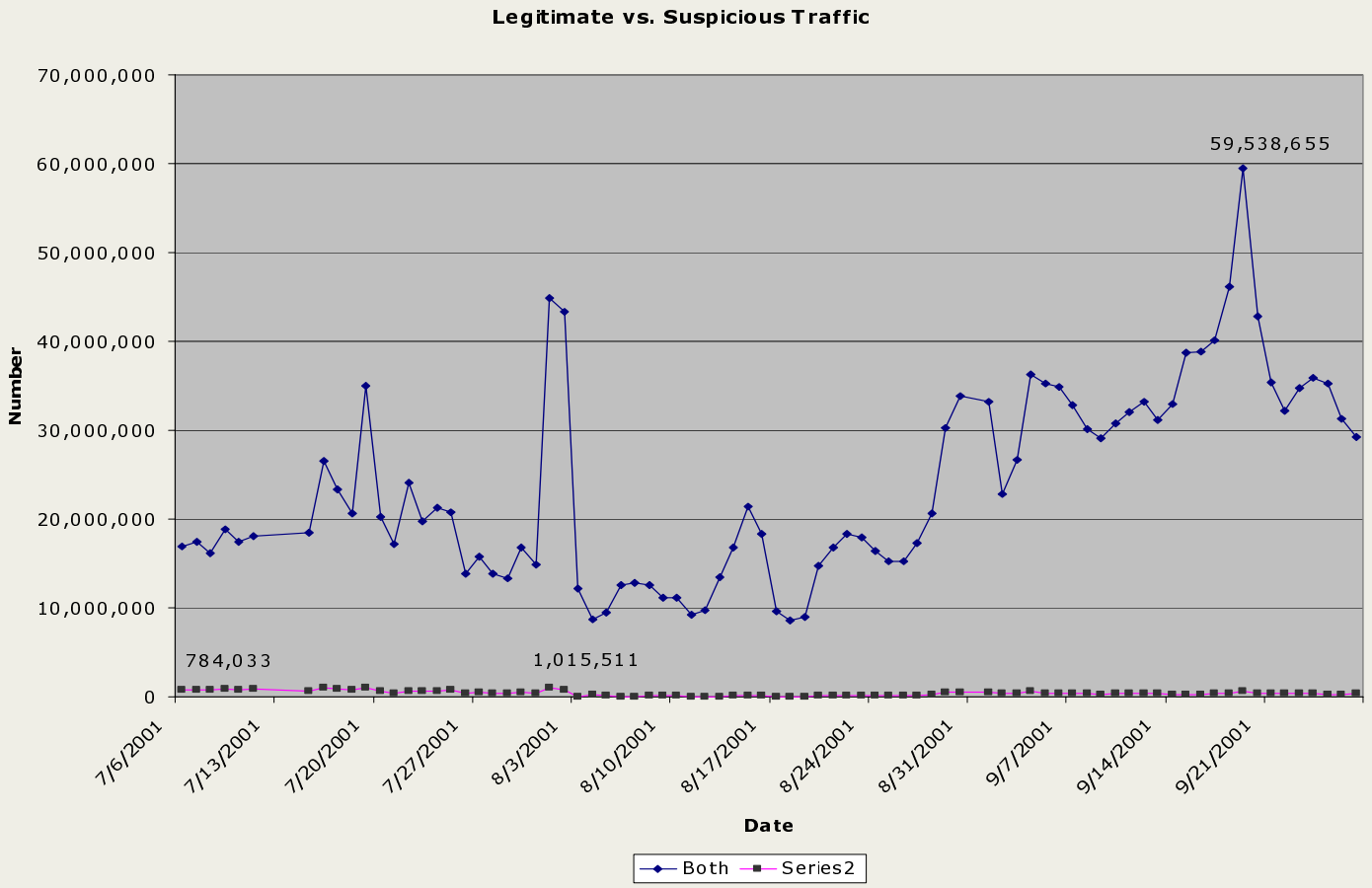
Carnegie Mellon
Software Engineering Institute

Non Routable Traffic



Carnegie Mellon
Software Engineering Institute

Legitimate vs. Suspicious Traffic



Carnegie Mellon
Software Engineering Institute

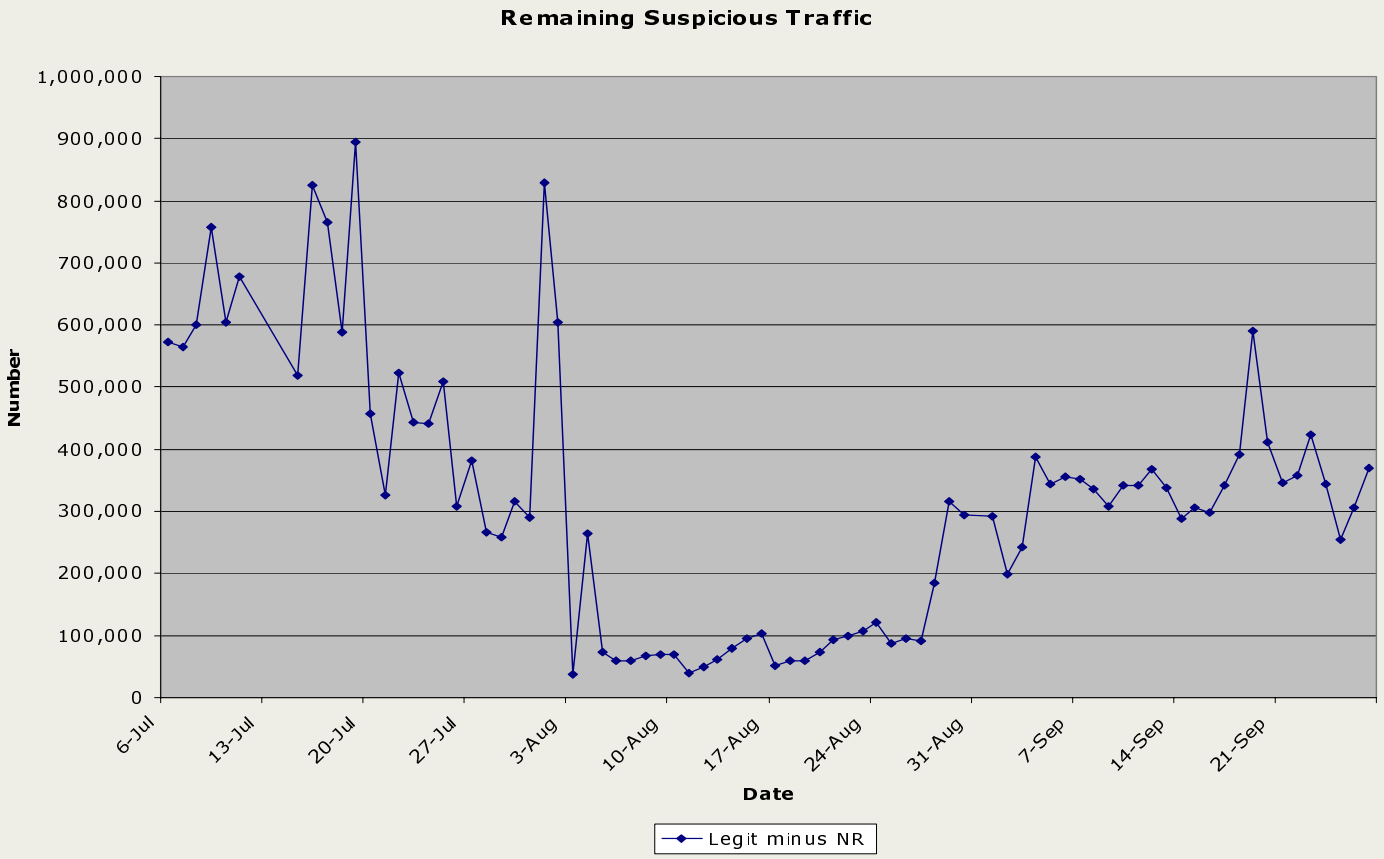
NonRoutable Source Addresses

Deny Private and Reserved Source IP Addresses.
These source addresses should be filtered:

- 0.0.0.0/8 - Historical Broadcast
- 10.0.0.0/8 - RFC 1918 Private Network
- 127.0.0.0/8 - Loopback
- 169.254.0.0/16 - Link Local Networks
- 172.16.0.0/12 - RFC 1918 Private Network
- 192.0.2.0/24 - TEST-NET
- 192.168.0.0/16 - RFC 1918 Private Network
- 224.0.0.0/4 - Class D Multicast
- 240.0.0.0/5 - Class E Reserved
- 248.0.0.0/5 - Unallocated
- 255.255.255.255/32 - Broadcast

Carnegie Mellon
Software Engineering Institute

minus the Non Routable Traffic



Carnegie Mellon
Software Engineering Institute

Scanning Activity

Scanning/Probing/Reconnaissance/Surveillance can be done in innumerable ways – how to characterize?

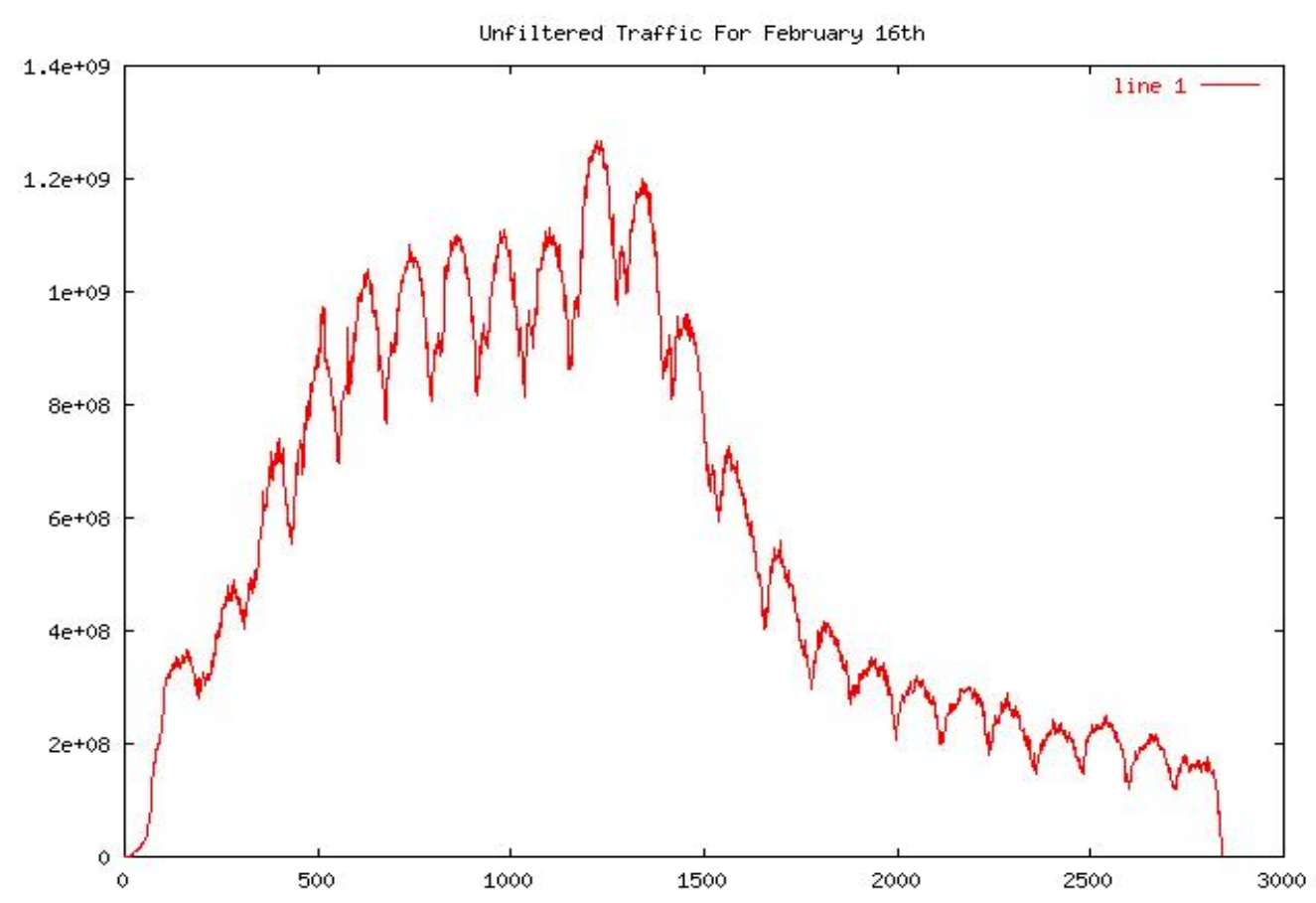
Two methods of particular interest include:

"Stealth" scanning:

- probes which fall below thresholds for alerts
- long and slow
- handcrafted packets
- <4 packets per flow
- few per day transmitted

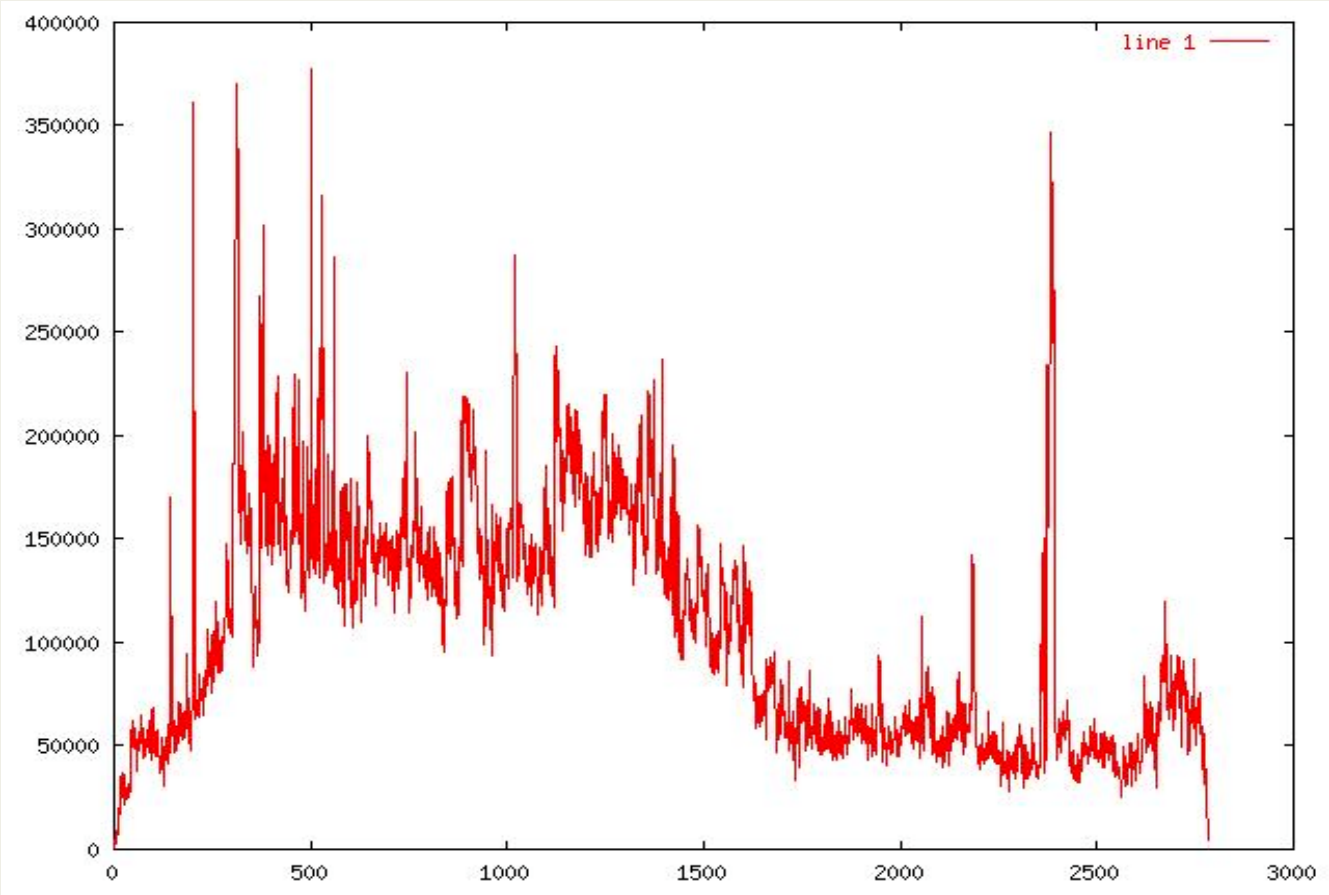
Use of ICMP

Low-Packet Filtering - Normal Traffic



Carnegie Mellon
Software Engineering Institute

Low-Packet Traffic



Carnegie Mellon
Software Engineering Institute

ICMP Analysis

Purpose: Detect ICMP based attacks, scans, tool probes, and covert channels

Approach: ICMP is a very mechanical protocol

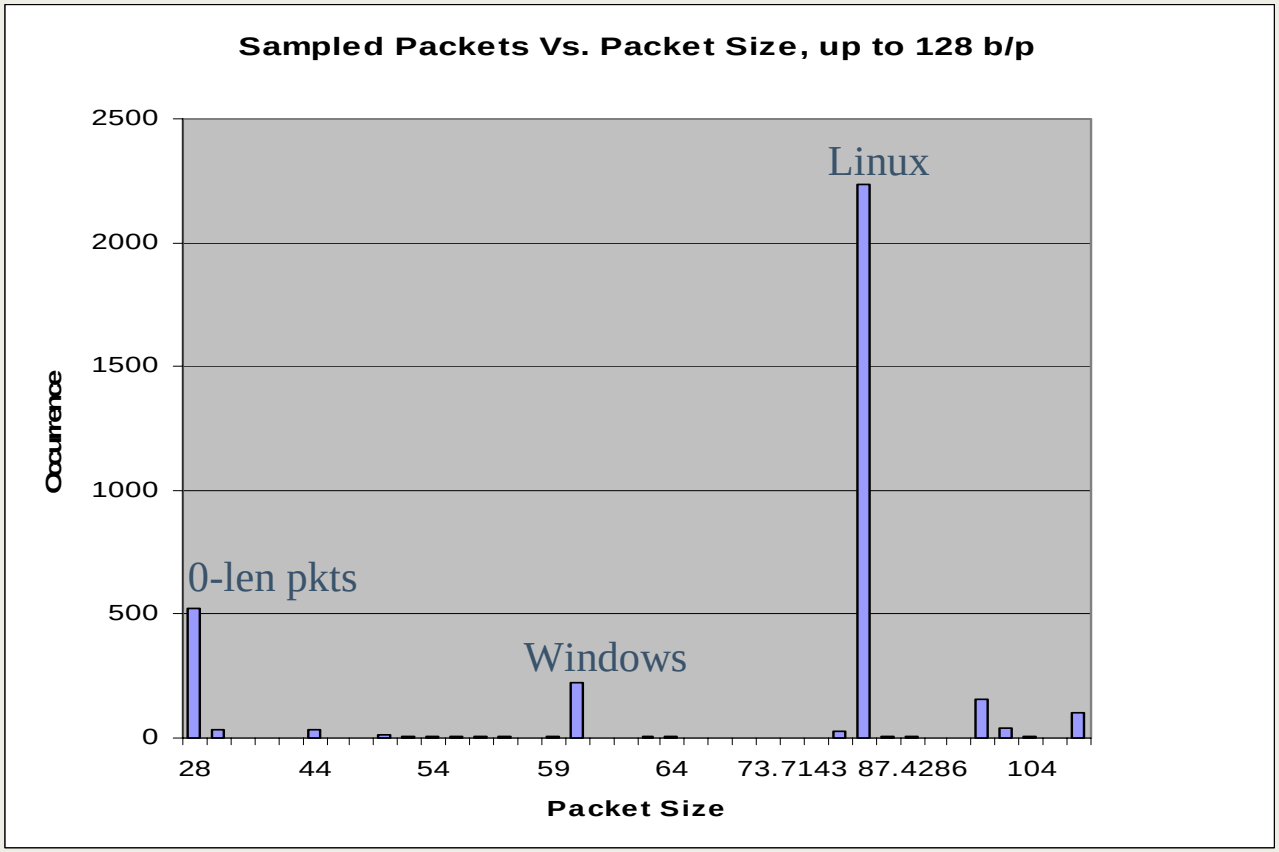
- 1 message per packet
- Type of message defined by packet header
- Some attacks are very obvious; e.g., fragmented packets or MTU sized packets

Status:

- Taxonomy of ICMP sizes/types completed.
- In progress:
  - Identifying normal ICMP traffic profile
  - Characterizing ICMP exploit signatures
  - Detecting ICMP exploits

Carnegie Mellon
Software Engineering Institute

ICMP Sampled Packets Vs. Packet Size



Carnegie Mellon
Software Engineering Institute

Predicting Exploit lifetime

Used incident data for vuls in phf, imap, and bind.

$$C = I + S \times \sqrt{M}$$

where C = cumulative count of reported incidents
M = time since start of exploit
I, S = regression coeffeicients
(intercept , slope)

IMAP and phf data spanned 30 months.
Model applied to mountd and statd (15 months).

Carnegie Mellon
Software Engineering Institute

Predicting Exploit lifetime 2

R^2 Results:

- Non-comparable intercepts and slopes (I,S)
- Square Root transformation best fit

	Sq Root	Log	Raw
bind	.908	.903	.884
phf	.939	.910	.881
IMAP	.981	.952	.971
mountd	.839	.868	.761
statd	.857	.935	.707

Carnegie Mellon
Software Engineering Institute

Survivable Network Analysis

Focus on essential services and preservation of essential assets that are critical to fulfilling mission objectives.

The Three Rs: Resistance, Recognition, and Recovery

Four main activities:

- System Definition
- Essential Capability Definition
- Compromisable Capability Definition
- Survivability Analysis

Carnegie Mellon
Software Engineering Institute

Survivability 1

Life-Cycle Activities	Key Survivability Elements	Examples
Mission Definition	Analysis of mission criticality and consequences of failure	Estimation of cost impact of denial-of service attacks
Concept of operations	Definition of system capabilities in adverse environments	Enumeration of critical mission functions that must withstand attacks
Project planning	Integration of survivability into lifecycle activities	Identification of defensive coding techniques for implementation
Requirements definition	Definition of survivability requirements from mission perspective	Definition of access requirements for critical system assets during attacks
System specification	Specification of essential service and intrusion scenarios	Definition of steps that compose critical system transactions

Carnegie Mellon
Software Engineering Institute

Survivability 2

Life-Cycle Activities	Key Survivability Elements	Examples
System architecture	Integration of survivability strategies into architecture definition	Creation of network facilities for replication of critical data assets
System design	Development and verification of survivability strategies	Correctness verification of data encryption algorithms
System implementation	Application of survivability coding and implementation techniques	Definition of methods to avoid buffer overflow vulnerabilities
System testing	Treatment of intruders as users in testing and certification	Addition of intrusion usage to usage models for statistical testing
System evolution	Improvement of survivability to prevent degradation over time	Redefinition of architecture in response to changing threat environment

Carnegie Mellon
Software Engineering Institute

OCTAVE Method (CERT)

Phase 1: Build Asset-Based Threat Profiles

- Process 1: Identify Senior Management Knowledge
- Process 2: Identify Operational Area Knowledge
- Process 3: Identify Staff Knowledge
- Process 4: Create Threat Profiles

Phase 2: Identify Infrastructure Vulnerabilities

- Process 5: Identify Key Components
- Process 6: Evaluate Selected Components

Phase 3: Develop Security Strategy and Plans

- Process 7: Conduct Risk Analysis
- Process 8: Develop Protection Strategy

Carnegie Mellon
Software Engineering Institute

Moral: Pay Attention

Collect and look at your data.

Know your network/system.

Accommodate training needs.

Develop in-house capabilities.

Relying on automated procedures and technologies without analytical insight can get you into trouble.