Security in Computing & Information Technology (COSC2536/COSC2537)

Lecture 10: Digital Authorization and Intrusion Detection

RMIT UNIVERSITY

Lecture Overview

- During this lecture, we will learn
  - What is Authorization
  - How to enforce Authorization
  - Inference Control
  - Packet Filtering
  - Signature Based Intrusion Detection
  - Anomaly Based Intrusion Detection

Authentication vs Autherization

- Two parts to access control…
- **Authorization:** Are you allowed to do that?
  - Once you have access, what can you do?
  - Enforces limits on actions
  - To enforce actions we also need intrusion detection (to be covered in this lecture)

  In the last lecture we learned
- **Authentication:** Are you who you say you are?
  - Determine whether access is allowed or not
  - Authenticate human to machine
  - Or, possibly, machine to machine
- Note: "access control" often used as synonym for authorization

Lampson's Access Control Matrix

- Authorization is a form of **access control**
- Classic view of authorization…
  - Access Control Lists (ACLs)
  - Capabilities (C-lists)
  - Subjects (users) index the rows
  - Objects (resources) index the columns

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	—	—
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Are You Allowed to Do That?

- **Access control matrix** has **all** relevant info
- Could be 100's of users, 10,000's of resources
  - Then matrix has 1,000,000's of entries
- How to manage such a large matrix?
- Note: We need to check this matrix before access to any resource by any user
- How to make this more efficient/practical?

Access Control Lists (ACLs)

- ACL: store access control matrix by **column**
- Example: ACL for **insurance data** is in **blue**

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	—	—
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Capabilities (or C-Lists)

- Store access control matrix by **row**
- Example: Capability for **Alice** is in **red**

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	—	—
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

ACLs vs Capabilities

- ACLs
  - Good when users manage their own files
  - Protection is data-oriented
  - Easy to change rights to a resource
- Capabilities
  - Easy to delegate — avoid the confused deputy
  - Easy to add/delete users
  - More difficult to implement
  - The "Zen of information security"
- Capabilities loved by academics
  - Capability Myths Demolished

Inference Control Example

- Suppose we query a database
  - Question: What is average salary of female CS professors at SJSU?
  - Answer: $95,000
  - Question: How many female CS professors at SJSU?
  - Answer: 1
- Specific information has leaked from responses to general questions!

Inference Control & Research

- For example, medical records are private but valuable for research
- How to make info available for research and protect privacy?
- How to allow access to such data without leaking specific information?

Naïve Inference Control

- Remove names from medical records?
- Still may be easy to get specific info from such "anonymous" data
- Removing names is not enough
  - As seen in previous example
- What more can be done?

Less-naïve Inference Control

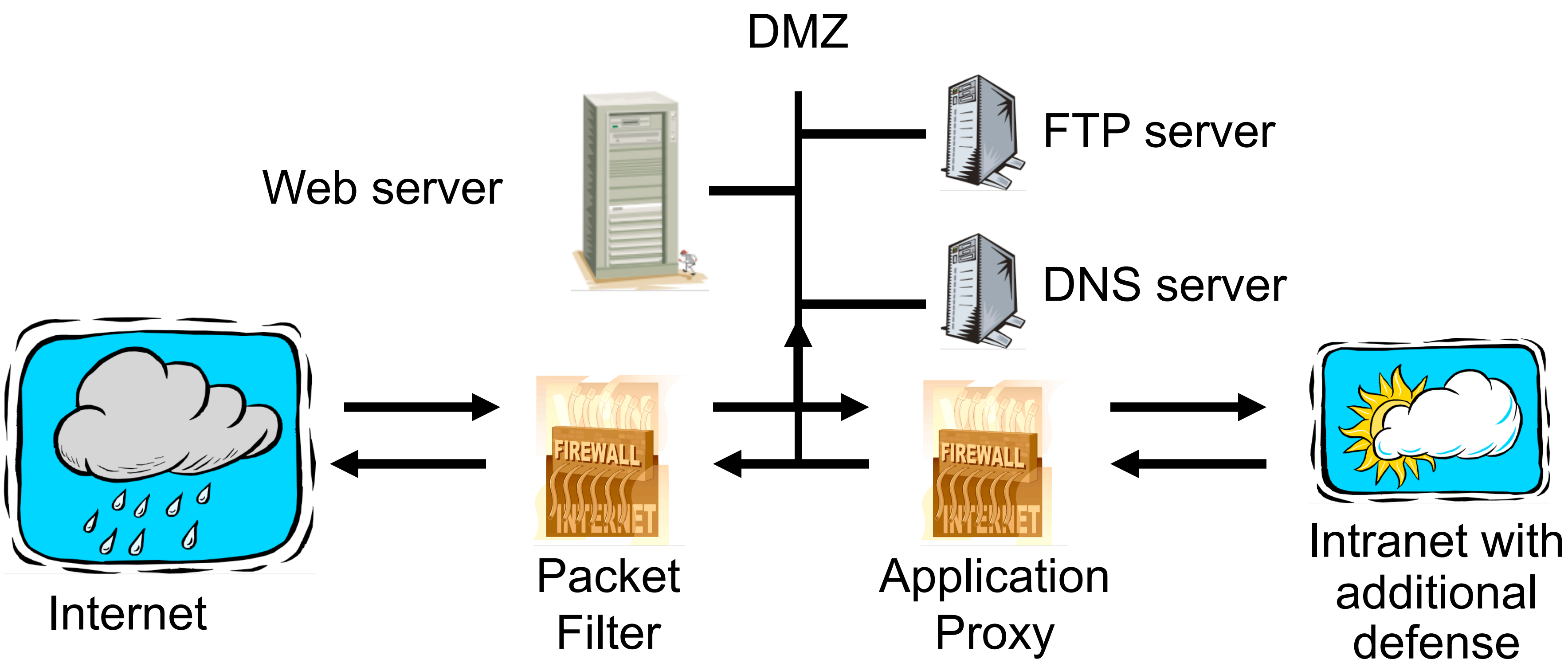
- Query set size control
  - Don't return an answer if set size is too small
- N-respondent, k% dominance rule
  - Do not release statistic if k% or more contributed by N or fewer
  - Example: Avg salary in Bill Gates' neighborhood
  - This approach used by US Census Bureau
- Randomization
  - Add small amount of random noise to data
- Many other methods — none satisfactory

Something Better Than Nothing?

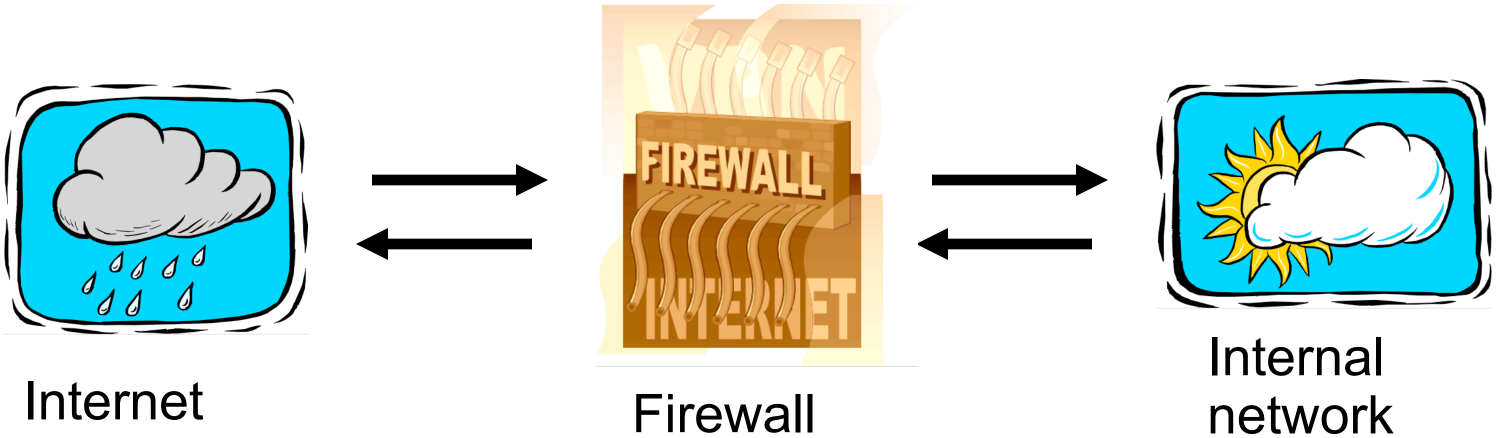
- Robust inference control may be impossible
- Is weak inference control better than nothing?
  - **Yes**: Reduces amount of information that leaks
- Is weak covert channel protection better than nothing?
  - **Yes**: Reduces amount of information that leaks
- Is weak crypto better than no crypto?
  - **Probably not:** Encryption indicates important data
  - May be easier to filter encrypted data

Firewalls and Defense in Depth

- Typical network security architecture



Firewalls



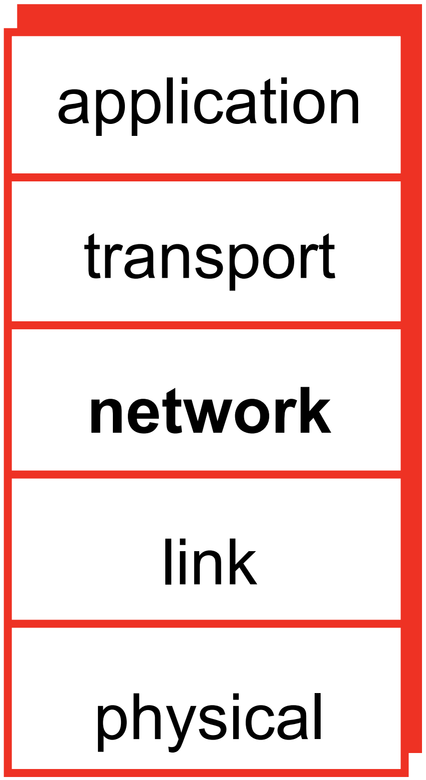
- Firewall decides what to let in to internal network and/or what to let out
- **Access control** for the network

Firewall Terminology

- No standard firewall terminology
- Types of firewalls
  - **Packet filter** — works at network layer
  - **Stateful packet filter** — transport layer
  - **Application proxy** — application layer
- Lots of other terms often used
  - E.g., “deep packet inspection”

Packet Filter

- Operates at network layer
- Can filters based on…
  - Source IP address
  - Destination IP address
  - Source Port
  - Destination Port
  - Flag bits (SYN, ACK, etc.)
  - Egress or ingress



Packet Filter

- Configured via Access Control Lists (ACLs)
  - Different meaning than at start of this lecture

Action	Source IP	Dest IP	Source Port	Dest Port	Protocol	Flag Bits
Allow	Inside	Outside	Any	80	HTTP	Any
Allow	Outside	Inside	80	> 1023	HTTP	ACK
Deny	All	All	All	All	All	All

- **Q**: Intention?
- **A**: Restrict traffic to Web browsing

Intrusion Prevention

- Want to keep bad guys out
- **Intrusion prevention** is a traditional focus of computer security
  - Authentication is to prevent intrusions
  - Firewalls a form of intrusion prevention
  - Virus defenses aimed at intrusion prevention
  - Like locking the door on your car

Intrusion Detection

- In spite of intrusion prevention, bad guys will sometime get in
- Intrusion detection systems (**IDS**)
  - Detect attacks in progress (or soon after)
  - Look for unusual or suspicious activity
- Who is likely intruder?
  - May be outsider who got thru firewall
  - May be evil insider
- What do intruders do?
  - Launch well-known attacks
  - Launch variations on well-known attacks
  - Launch new/little-known attacks
  - "Borrow" system resources

IDS

- Intrusion detection **approaches**
  - Signature-based IDS
  - Anomaly-based IDS
- Intrusion detection **architectures**
  - Host-based IDS
    - ✓ Monitor activities on hosts for Known attacks (i.e. signature), Suspicious behavior (i.e. anomaly)
  - Network-based IDS
    - ✓ Monitor activity on the network for Known attacks (i.e. signature), Suspicious network activity (i.e. anomaly)
- Any IDS can be classified as above
  - In spite of marketing claims to the contrary!

Signature Detection Example (Host Based)

- Failed login attempts may indicate password cracking attack
- IDS could use the rule "N failed login attempts in M seconds" as **signature**
- If N or more failed login attempts in M seconds, IDS warns of attack
- Note that such a warning is specific
  - Admin knows what attack is suspected
  - Easy to verify attack (or false alarm)

Signature Detection (Host Based)

- Suppose IDS warns whenever N or more failed logins in M seconds
  - Set N and M so false alarms not common
  - Can do this based on "normal" behavior
- But, if Trudy knows the signature, she can try (N – 1) logins every M seconds…
- Then signature detection slows down Trudy, but might not stop her

Signature Detection (Host Based)

- Many techniques used to make signature detection more robust
- Goal is to detect "almost" signatures
- For example, if "about" N login attempts in "about" M seconds
  - Warn of possible password cracking attempt
  - Can use statistical analysis, heuristics, etc.

Signature Detection (Host Based)

- Advantages of signature detection
  - Simple
  - Detect known attacks
  - Know which attack at time of detection
  - Efficient (if reasonable number of signatures)
- Disadvantages of signature detection
  - Signature files must be kept up to date
  - Number of signatures may become large
  - Can only detect known attacks

Anomaly Detection –learning by example #1 (Host Based)

- Suppose we monitor use of three commands:

  open, read, close
- Under normal use we observe Alice:

  open, read, close, open, open, read, close, …
- Of the six possible ordered pairs, we see four pairs are normal for Alice,

  (open,read), (read,close), (close,open), (open,read)
- Can we use this to identify unusual activity?

Anomaly Detection –learning by example #1 (Host Based)

- We monitor use of the three commands

  open, read, close

- If the ratio of abnormal to normal pairs is "too high", warn of possible attack
- Could improve this approach by
  - Also use expected frequency of each pair
  - Use more than two consecutive commands
  - Include more commands/behavior in the model
  - More sophisticated statistical discrimination

Anomaly Detection –learning by example #2 (Host Based)

- Over time, Alice has accessed file F_n at rate H_n
- Recently, "Alice" has accessed F_n at rate A_n

H_0	H_1	H_2	H_3
0.10	0.40	0.40	0.10

A_0	A_1	A_2	A_3
0.10	0.40	0.30	0.20

- Is this normal use for Alice?
- We compute $S = (H_0 - A_0)^2 + (H_1 - A_1)^2 + (H_2 - A_2)^2 + (H_3 - A_3)^2$

$$S = (0.1 - 0.1)^2 + (0.4 - 0.4)^2 + (0.4 - 0.3)^2 + (0.1 - 0.2)^2 = 0.02$$

- If we consider S < 0.1 to be normal, then this is normal
- How to account for use that varies over time?

Anomaly Detection –learning by example #2 (Host Based)

- To allow "normal" to adapt to new use, we update averages: $H_i = 0.2 \cdot A_i + 0.8 \cdot H_i \ \text{ for } i = 0, 1, 2, 3.$
- In this example, H_n are updated using the tables

H_0	H_1	H_2	H_3
0.10	0.40	0.40	0.10

A_0	A_1	A_2	A_3
0.10	0.40	0.30	0.20

- For example, we update H_2 and H_3

$$H_2 = 0.2 \cdot 0.3 + 0.8 \cdot 0.4 = 0.38 \quad \text{and} \quad H_3 = 0.2 \cdot 0.2 + 0.8 \cdot 0.1 = 0.12.$$

- And we now have

H_0	H_1	H_2	H_3
0.10	0.40	0.38	0.12

Anomaly Detection –learning by example #2 (Host Based)

- The updated long term average is

H_0	H_1	H_2	H_3
0.10	0.40	0.38	0.12

- Suppose new observed rates…

A_0	A_1	A_2	A_3
0.10	0.30	0.30	0.30

- Is this normal use?
- Compute

$$S = (0.1 - 0.1)^2 + (0.4 - 0.3)^2 + (0.38 - 0.3)^2 + (0.12 - 0.3)^2 = 0.0488$$

Since $S = .0488 < 0.1$ we consider this normal

- And we again update the long term averages:

$$H_i = 0.2 \cdot A_i + 0.8 \cdot H_i \quad \text{for } i = 0, 1, 2, 3$$

Anomaly Detection –learning by example #2 (Host Based)

- The starting averages were:

H_0	H_1	H_2	H_3
0.10	0.40	0.40	0.10

- After 2 iterations, averages are:

H_0	H_1	H_2	H_3
0.10	0.38	0.364	0.156

- Statistics slowly evolve to match behavior
- This reduces false alarms for SA
- But also opens an avenue for attack…
  - Suppose Trudy **always** wants to access F3
  - Can she convince IDS this is normal for Alice?

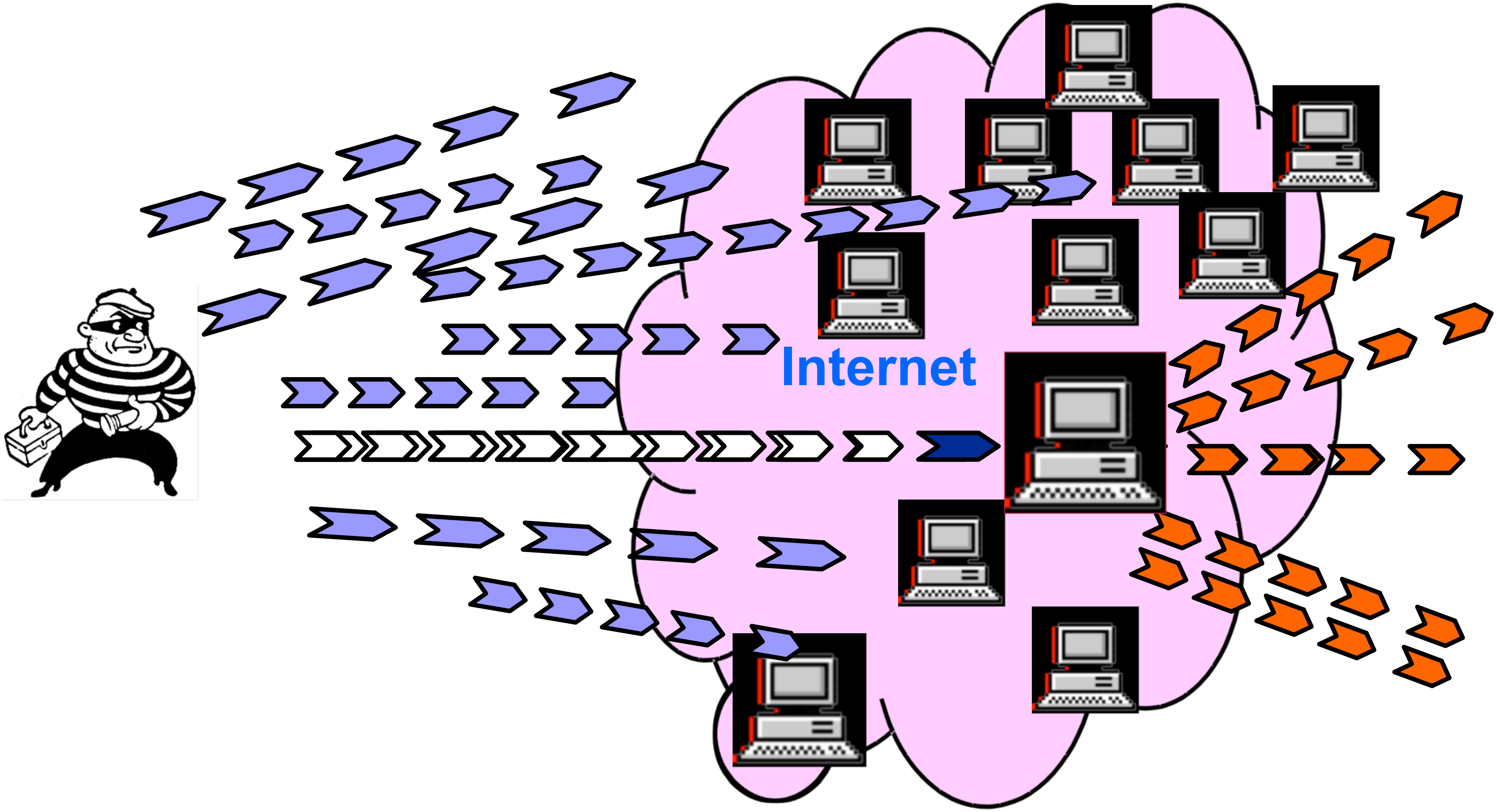
Anomaly Detection –learning by example #2 (Host Based)

- To make this approach more robust, must incorporate the variance
- Can also combine N stats Si as, say,

  $T = (S_1 + S_2 + S_3 + \ldots + S_N) / N$

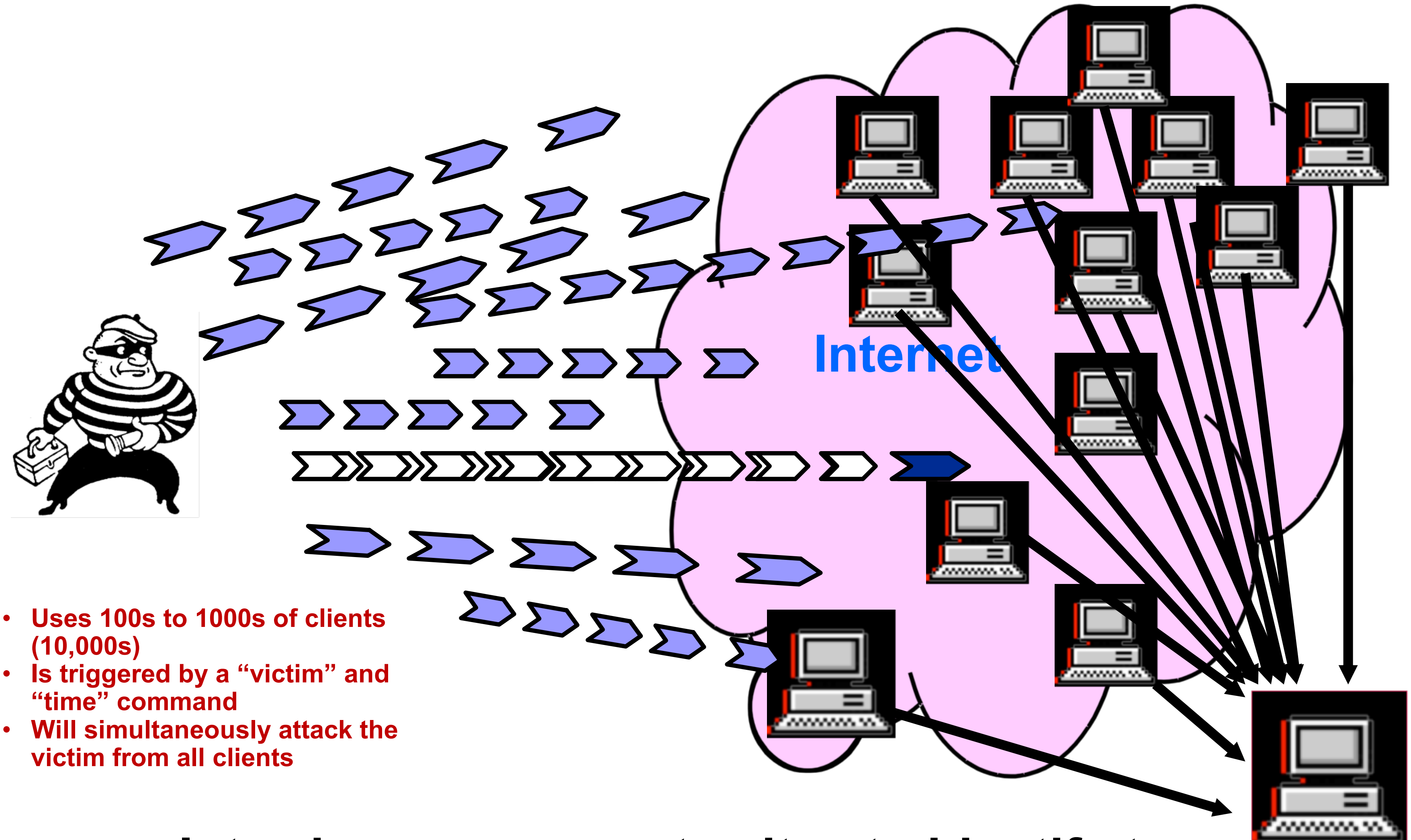
  to obtain a more complete view of "normal"
- Similar (but more sophisticated) approach is used in an IDS known as **NIDES**
- NIDES combines anomaly & signature IDS

Network Based Intrusion: Typical Intruder Attack



Intruder scans remote sites to identify targets, then attacks vulnerable or misconfigured hosts

Network Based Intrusion: Distributed Coordinated Attack



- Uses 100s to 1000s of clients (10,000s)
- Is triggered by a “victim” and “time” command
- Will simultaneously attack the victim from all clients

Intruder scans remote sites to identify targets, then attacks vulnerable or misconfigured hosts