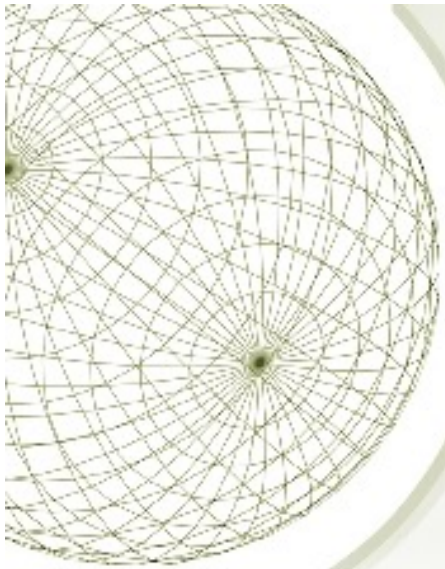
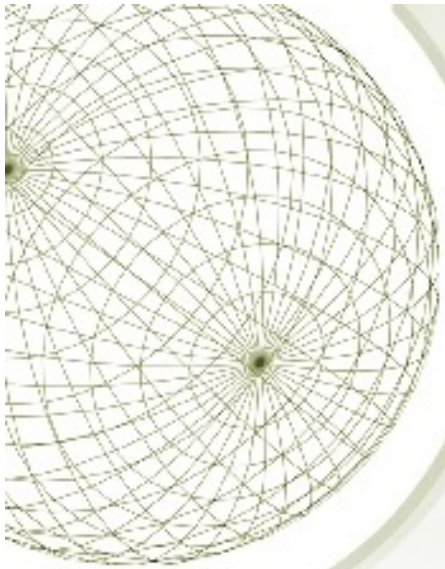


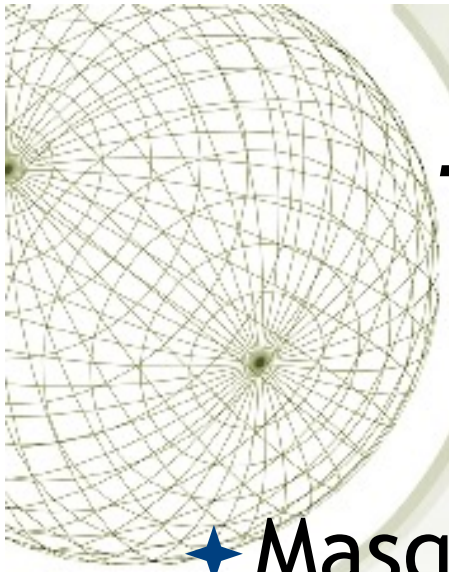
Intruders

Ola Flygt
Linnaeus University, Sweden
http://homepage.lnu.se/staff/oflmsi/
Ola.Flygt@lnu.se



Outline

- Intrusion Techniques
- Intrusion Detection
- Password Protection
- Password Selection Strategies



Three classes of intruders (hackers or crackers)

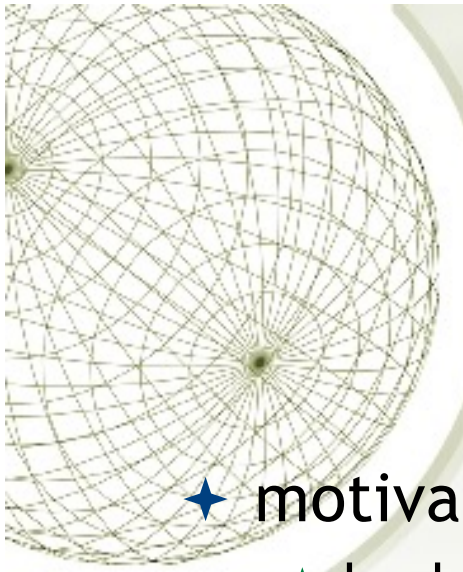
- Masquerader
  - An individual trying to exploit a legitimate user's account
- Misfeasor
  - A legitimate user misusing it's privileges
- Clandestine user
  - An individual seizing supervisory control to evade detection

Intruders

- clearly a growing publicized problem
  - from “Wily Hacker” in 1986/87
  - to clearly escalating CERT stats
- range
  - benign: explore, still costs resources
  - serious: access/modify data, disrupt system
- led to the development of CERTs (Computer Emergency Response Team)
- intruder techniques & behaviour patterns constantly shifting, have common features

Examples of Intrusion

- remote root compromise
- web server defacement
- guessing / cracking passwords
- copying viewing sensitive data / databases
- running a packet sniffer
- distributing pirated software from someone else's computer (anonymous FTP)
- Dial into an unsecured modem to access net
- impersonating a user to reset password
- using an unattended workstation



Hackers

- motivated by thrill of access and status
  - hacking community, a strong meritocracy
  - status is determined by level of competence
- benign intruders might be tolerable
  - But, do consume resources and may slow performance
  - can't know in advance whether benign or malign
- IDS / IPS / VPNs can help counter
- awareness led to establishment of CERTs
  - collect / disseminate vulnerability info / responses

Hacker Behavior Example

- select target using IP lookup tools
- map network for accessible services
- identify potentially vulnerable services
- brute force (guess) passwords
- install remote administration tool
- wait for admin to log on and capture password
- use password to access remainder of network

Criminal Enterprise

- organized groups of hackers now a threat
  - corporation / government / loosely affiliated gangs
  - typically young
  - Before focused on a few countries, now global with US as “leader”, see: https://threatmap.checkpoint.com/ThreatPortal/livemap.html
  - often target credit cards on e-commerce server or to install ransomware
- criminal hackers usually have specific targets
- once penetrated act quickly and get out
- IDS / IPS help but less effective
- sensitive data needs strong protection

Criminal Enterprise Behavior

- act quickly and precisely to make their activities harder to detect
- exploit perimeter via vulnerable ports
- use trojan horses (hidden software) to leave back doors for re-entry
- use sniffers to capture passwords
- do not stick around until noticed
- make few or no mistakes.

Actions against cybercriminals (Q2 2013)

- In April, the Russian Federal Security Service (FSB) and the Security Service of Ukraine (SBU) announced they arrested several individuals believed to be involved in the development of the Carberp banking Trojan. The leader of the group was a 28-year-old Russian citizen. The rest of the group — some 20 individuals between 25 and 30 years old—were arrested in Kiev, Zaporozhye, Lvov, Odessa, and Kherson. The ring was said to be responsible for stealing US$250 million (€193 million) in Ukraine and Russia alone.
- A 24-year-old Algerian who was arrested in Thailand in January, was extradited to the United States in April. Also known as "Bx1," he was listed in a North District of Georgia indictment as a coconspirator who helped develop SpyEye components. Known in the underground as "Gribodemon" and "Harderman," the real name of his partner, the presumed author of the SpyEye Trojan, was redacted in the indictment because he had not yet been arrested.

Source: McAfee Threats Report: Second Quarter 2013

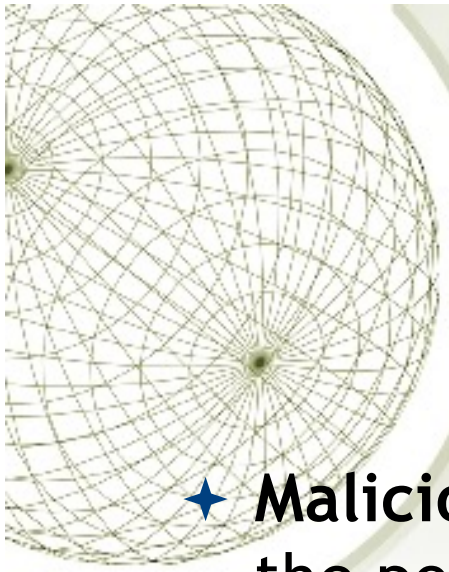
Actions against cybercriminals (Q2 2013) cont.

- On May 9, federal prosecutors unsealed charges against eight New York people linked with an international cybertheft ring accused of stealing US$45 million from banks around the globe. The alleged crooks used prepaid MasterCard debit. The defendants withdrew US$2.8 million from New York banks in two separate attacks this past December and February. While the eight were taking the money from the New York banks, additional coconspirators made more than US$42 million in withdrawals at other banks across the world.
- In May, the founder of digital currency system Liberty Reserve was indicted in the United States along with six other people for a US$6 billion money-laundering scheme. A Costa Rican citizen of Ukrainian origin and the founder of the currency system, was arrested in Spain, while others were arrested in Costa Rica and New York. Police in Costa Rica also raided three homes and five businesses linked to Liberty Reserve, according to the Associated Press.

Source: McAfee Threats Report: Second Quarter 2013

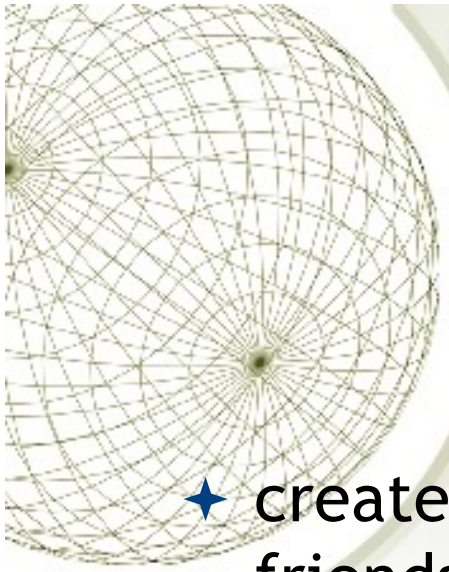
Insider Attacks

- among most difficult to detect and prevent
- employees have access & systems knowledge
- may be motivated by revenge / entitlement
  - when employment terminated
  - taking customer data when move to competitor
- IDS / IPS may help but also need:
  - least privilege, monitor logs, strong authentication, termination process to block access & mirror data



Insider Attacks

- **Malicious insiders** are the least frequent, but have the potential to cause significant damage due to their insider access. Administrators with privileged identities are especially risky.
- **Exploited insiders** may be “tricked” by external parties into providing data or passwords they shouldn’t.
- **Careless insiders** may simply press the wrong key and accidentally delete or modify critical information.

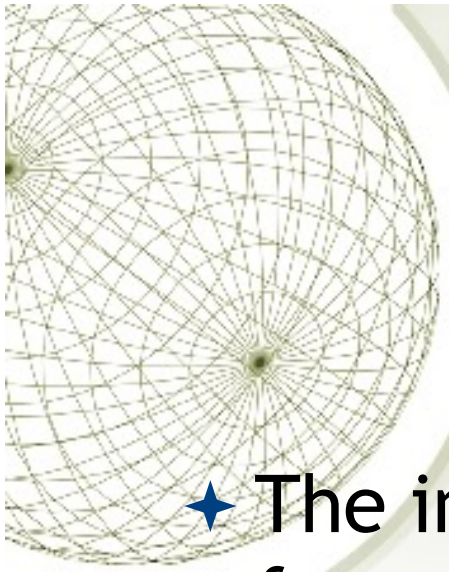


Insider Behavior Example

- create network accounts for themselves and their friends
- access accounts and applications they wouldn't normally use for their daily jobs
- e-mail former and prospective employers
- conduct furtive instant-messaging chats
- visit web sites that cater to disgruntled employees, such as http://fuckedcompany.com
- perform large downloads and file copying
- access the network during off hours

The Stages of a Network Intrusion

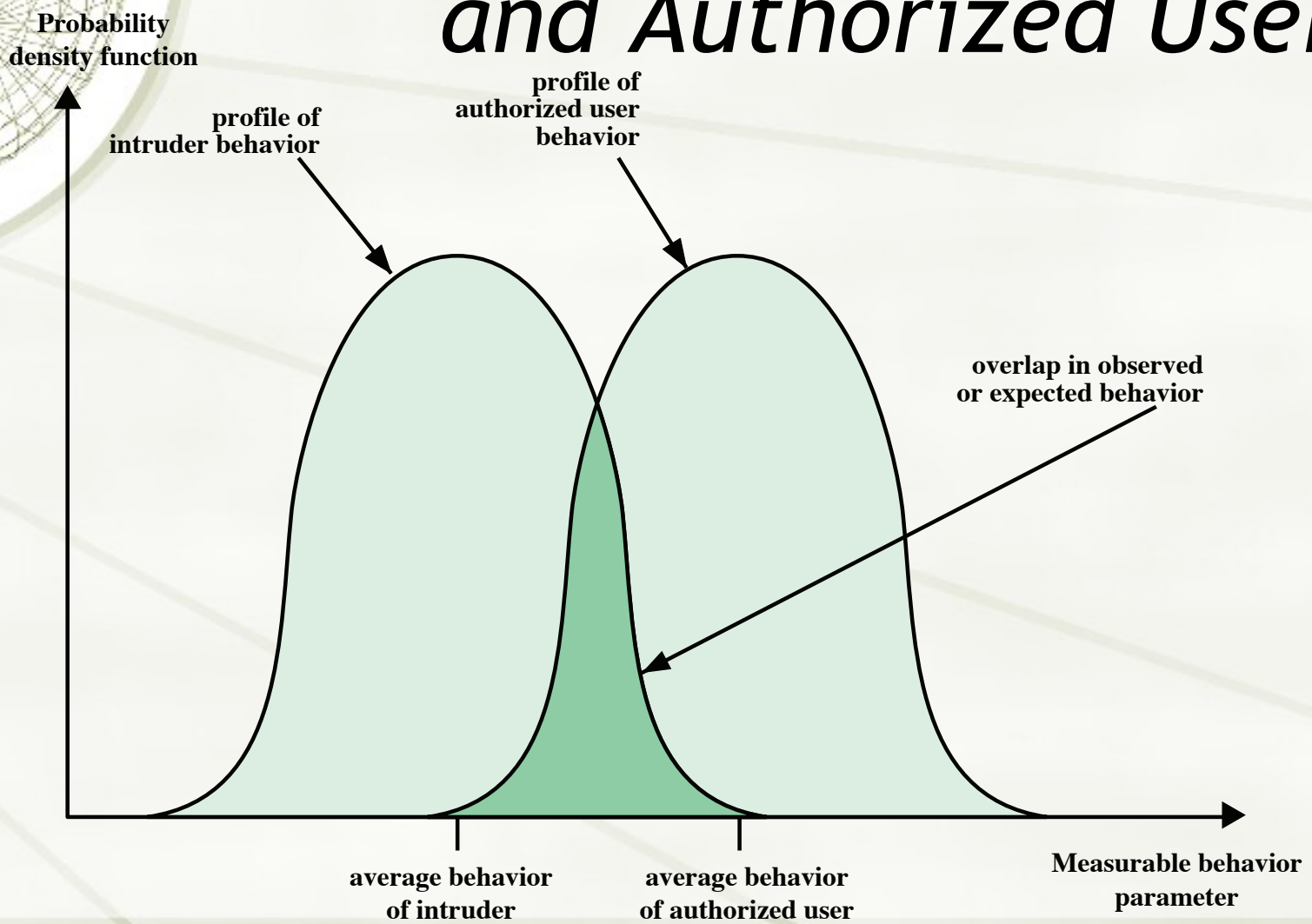
1. Scan the network to:
   - locate which IP addresses are in use
   - what operating systems are in use
   - what TCP or UDP ports are “open” (being listened to by Servers)
2. Run “Exploit” scripts against open ports
3. Get access to Shell program which is “suid” (has “root” privileges).
4. Download from Hacker Web site special versions of systems files that will let Cracker have free access in the future without his cpu time or disk storage space being noticed by auditing programs (rootkit).
5. Use IRC (Internet Relay Chat) to invite friends to the feast.

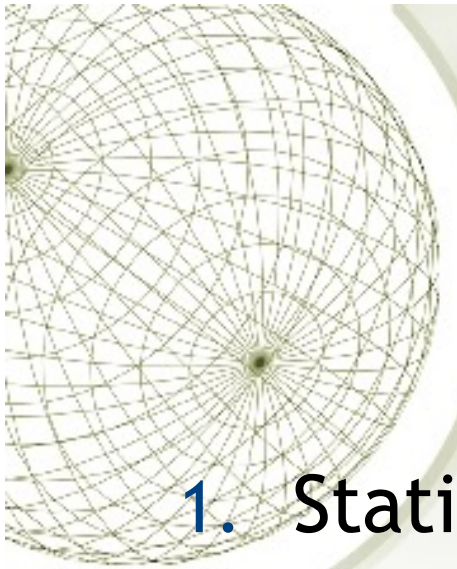


Intrusion Detection

- The intruder can be identified and ejected from the system.
- An effective intrusion detection can prevent intrusions. (Intrusion Prevention)
- Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Profiles of Behaviour of Intruders and Authorized Users





Intrusion Detection

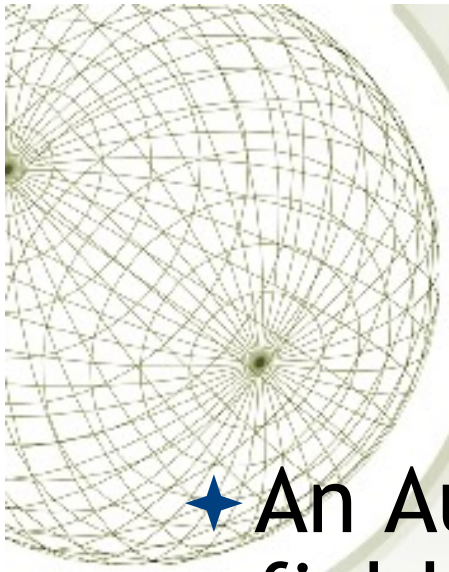
1. Statistical anomaly detection - what is not normal behaviour?
   a. Threshold detection (per system)
   b. Profile based (per user)
2. Rule based detection - what is not a proper behaviour?
   a. Anomaly detection - based on normal activities
   b. Penetration identification - expert system approach

Audit Records

- A fundamental tool in an IDS
- Audit Records are used in two phases
  - To collect information about a system during normal use used to build a model of the system
  - To monitor the running system and detect intrusion attempts

Audit Records

- The information can be
  - Native - information already collected in the system, eg. log files for login
  - Detection specific - extra modules introduced in the system that generate information deemed to be interesting



Audit Records

- An Audit record may have different fields of information
  - Subject
  - Action
  - Object
  - Exception-Condition
  - Resource-Usage
  - Time-Stamp

Measures used for Intrusion Detection

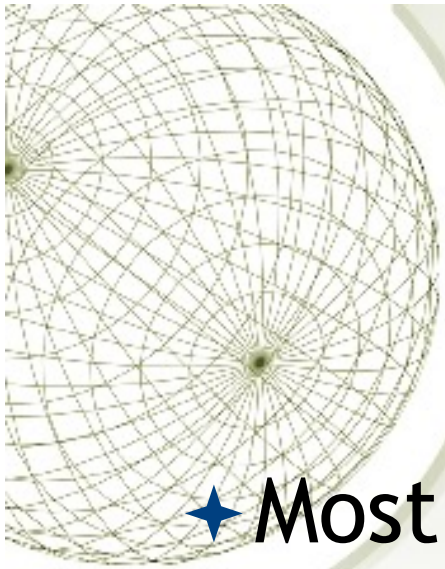
Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off-hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a "dead" account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified terminals	Operational	Attempted break-in.

Measures used for Intrusion Detection

Measure	Model	Type of Intrusion Detected
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.

Measures used for Intrusion Detection

Measure	Model	Type of Intrusion Detected
File access activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access unauthorized files.



Different IDS

- Most systems use the Audit Records to either create a statistical model for the system or as a input to a rule generator
- Rule-based penetration identification however are based on rules created by security experts

Example of heuristics in a Rule-based penetration identification

1. Users should not read files in other users' personal directories
2. Users must not write in other users' files
3. Users who log in after ours often access the same files they used earlier
4. Users do not copy system files

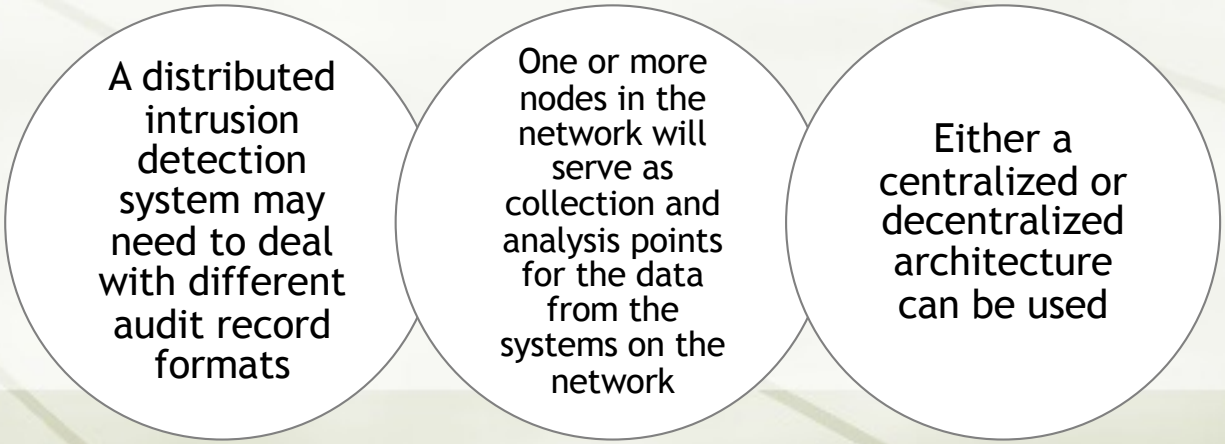
USTAT – state transition model

- USTAT is a way to simplify the creation of rules by limiting the number of actions used
- May also be used to standardize the IDS system

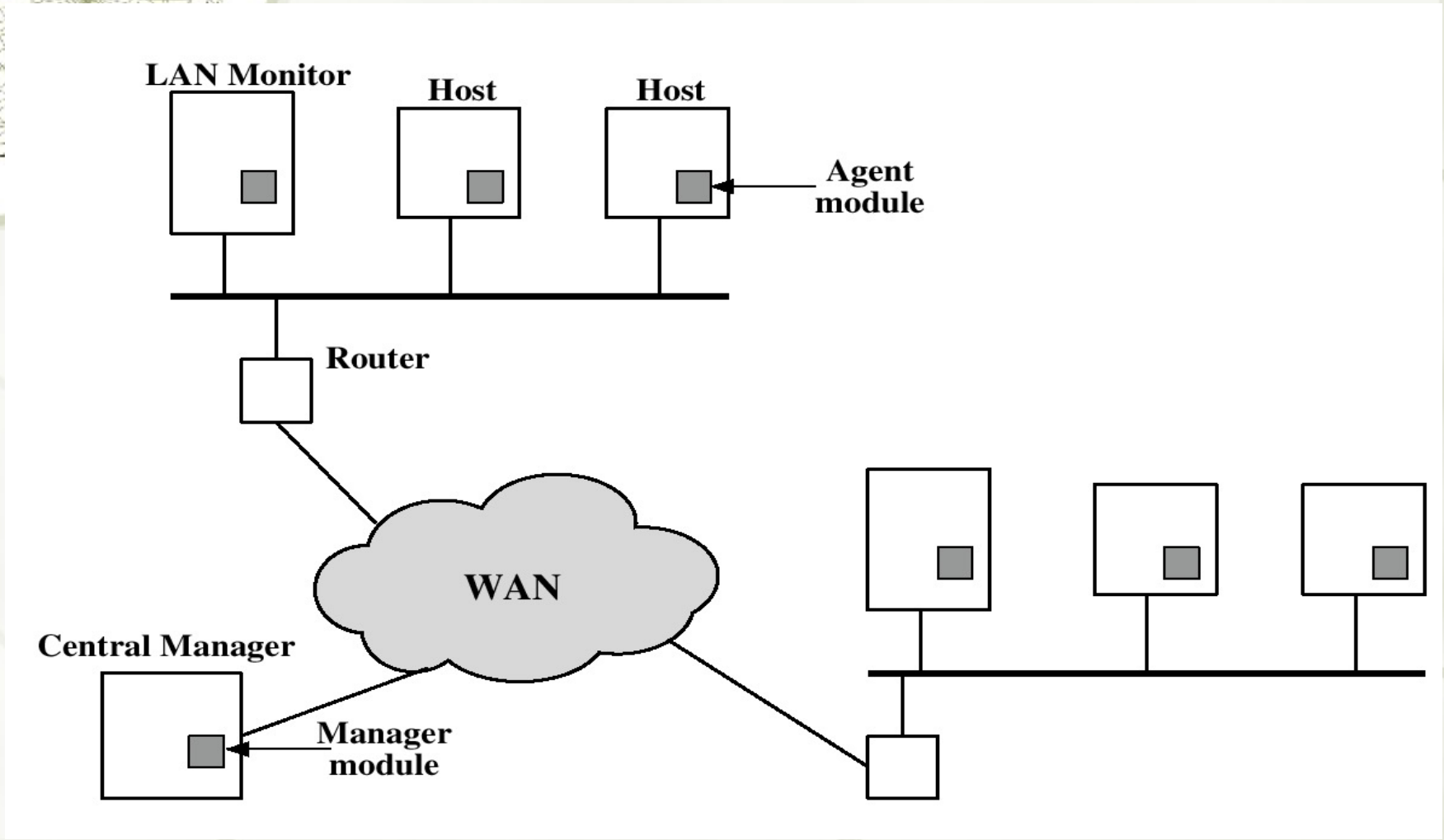
USTAT Action	SunOS Event Type
Read	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Rename	rename
Hardlink	link

Distributed Intrusion Detection

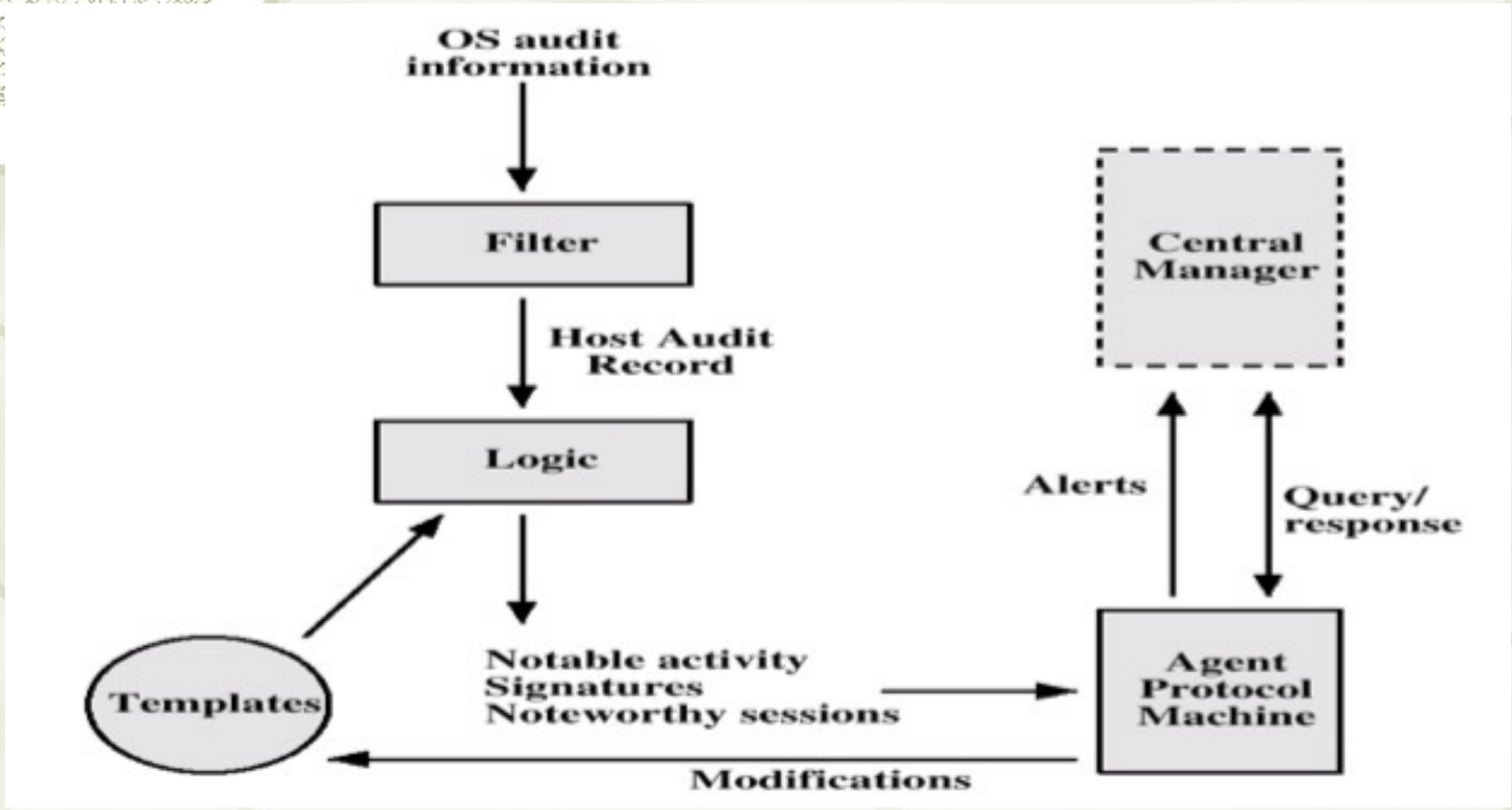
- Traditional systems focused on single-system stand-alone facilities
  - The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork
  - A more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network
- Major design issues:

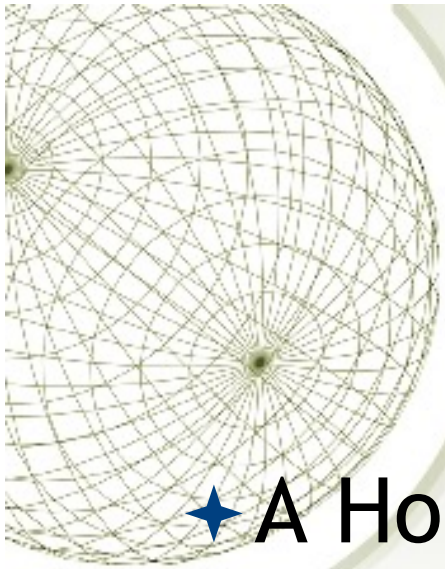


Distributed Intrusion Detection



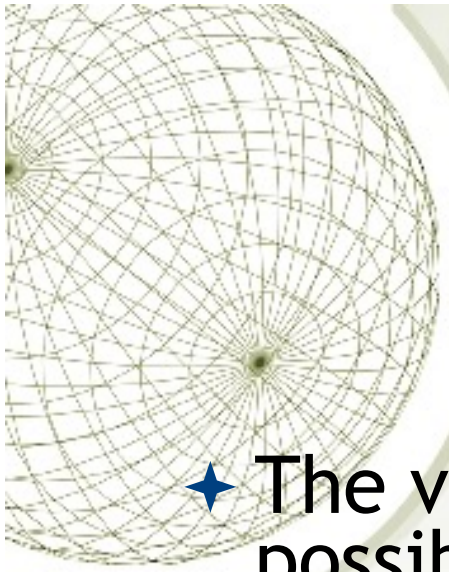
Distributed Intrusion Detection Agent Architecture





Honey pots

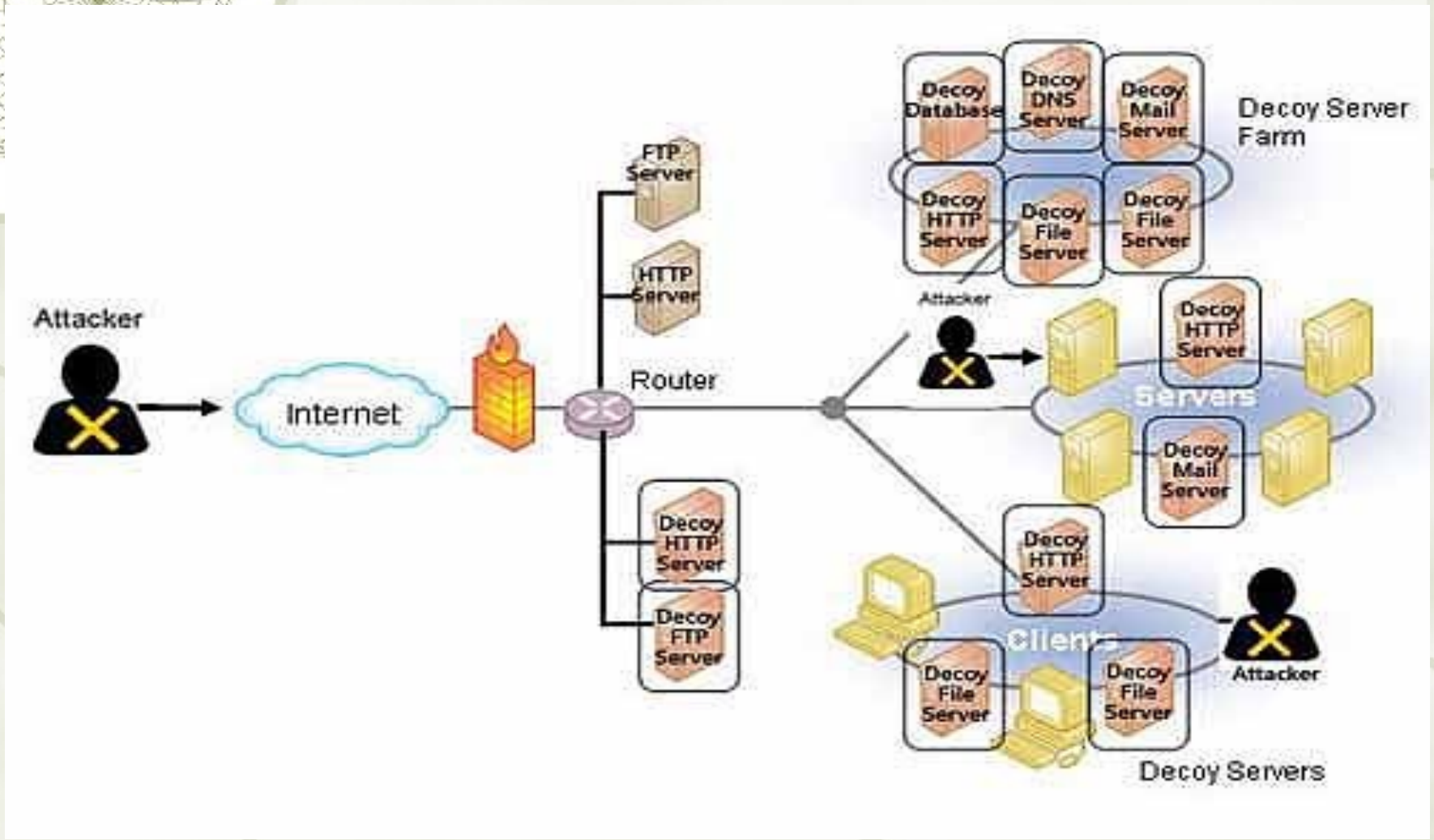
- A Honey Pot is an intrusion detection technique used to study hacker movements and probing to help better system defenses against later attacks
- Usually made up of a virtual machine that sits on a network or single client.

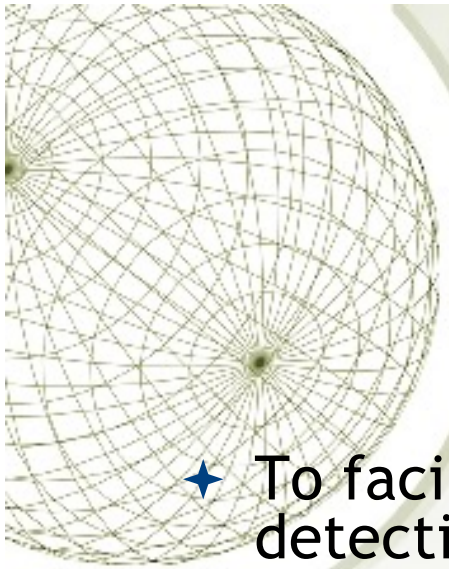


Three goals of a Honey Pot System

- The virtual system should look as real as possible, it should attract unwanted intruders to connect to the virtual machine for study.
- The virtual system should be watched to see that it isn't used for a massive attack on other systems, i.e. smurfing.
- The virtual system should look and feel just like a regular system, meaning it must include files, directories, and information that will catch the eye of the hacker.

Example Honey Pot System

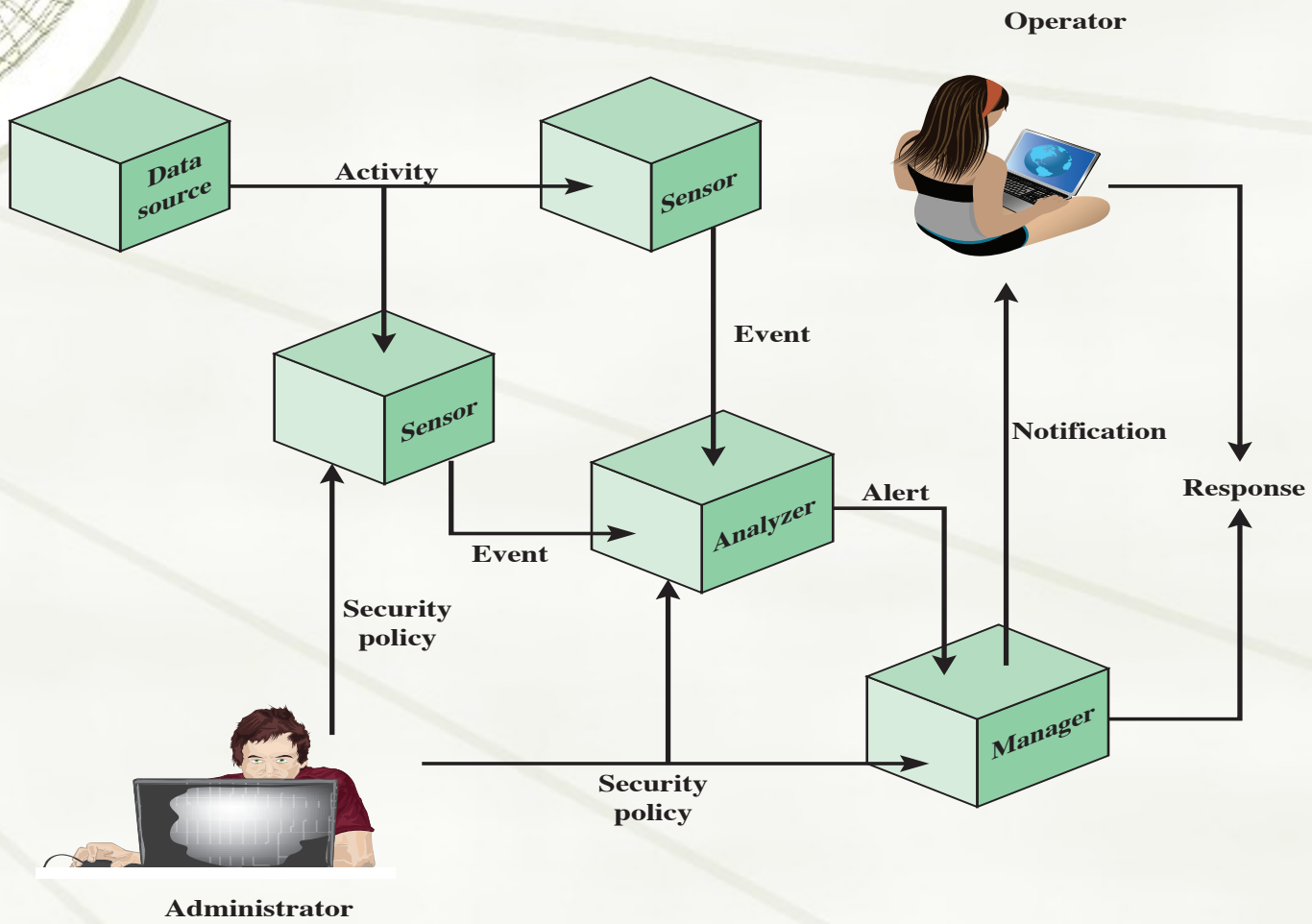




Intrusion detection exchange format

- To facilitate the development of distributed intrusion detection systems that can function across a wide range of platforms and environments, standards are needed to support interoperability
- IETF Intrusion Detection Working Group
  - Purpose of the group is to define data formats and exchange procedures for sharing information of interest to intrusion detection with response systems and to management systems that may need to interact with them
  - Have issued the following RFCs:
    - Intrusion Detection Message Exchange Requirements (RFC 4766)
    - The Intrusion Detection Message Exchange Format (RFC 4765)
    - The Intrusion Detection Exchange Protocol (RFC 4767)

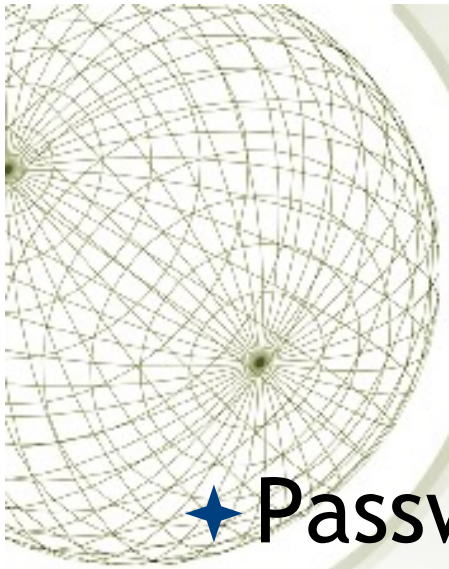
Model for Intrusion Detection Message Exchange



Examples of Intrusion Detection Systems

Bro	Unix, Linux, Mac-OS	NIDS
OSSEC	Unix, Linux, Windows, Mac-OS	HIDS
Snort	Unix, Linux, Windows	NIDS
Suricata	Unix, Linux, Windows, Mac-OS	NIDS
Security Onion	Linux, Mac-OS	HIDS, NIDS

Source: https://www.softwaretestinghelp.com/intrusion-detection-systems/



Password Management

- Passwords are interesting for a hacker since
  - they may be used to gain access to different types of assets
  - they are used in almost all systems
  - they often are of bad quality or easily compromised

Attack strategies and countermeasures

Workstation hijacking

- The attacker waits until a logged-in workstation is unattended
- The standard countermeasure is automatically logging the workstation out after a period of inactivity

Exploiting user mistakes

- Attackers are frequently successful in obtaining passwords by using social engineering tactics that trick the user or an account manager into revealing a password; a user may intentionally share a password to enable a colleague to share files; users tend to write passwords down because it is difficult to remember them
- Countermeasures include user training, intrusion detection, and simpler passwords combined with another authentication mechanism

Offline dictionary attack

- Determined hackers can frequently bypass access controls and gain access to the system's password file
- Countermeasures include controls to prevent unauthorized access to the password file, intrusion detection measures to identify a compromise, and rapid reissuance of passwords should the password file be compromised

Specific account attack

- The attacker targets a specific account and submits password guesses until the correct password is discovered
- The standard countermeasure is an account lockout mechanism, which locks out access to the account after a number of failed login attempts

Attack strategies and countermeasures

Electronic monitoring

- If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping
- Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary

Password guessing against single user

- The attacker attempts to gain knowledge about the account holder and system password policies and uses that knowledge to guess the password
- Countermeasures include training in and enforcement of password policies that make passwords difficult to guess

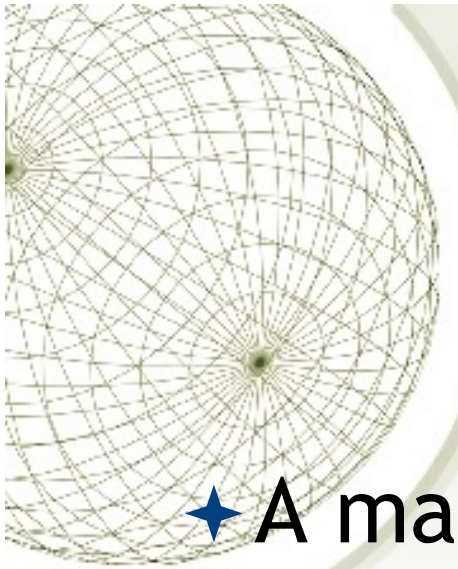
Exploiting multiple password use

- Attacks can become much more effective or damaging if different network devices share the same or a similar password for a given user
- Countermeasures include a policy that forbids the same or similar password on particular network devices

Popular password attack

- Attack is to use a popular password and try it against a wide range of user IDs
- Countermeasures include policies to inhibit the selection by users of common passwords and scanning the IP addresses of authentication requests and client cookies for submission patterns

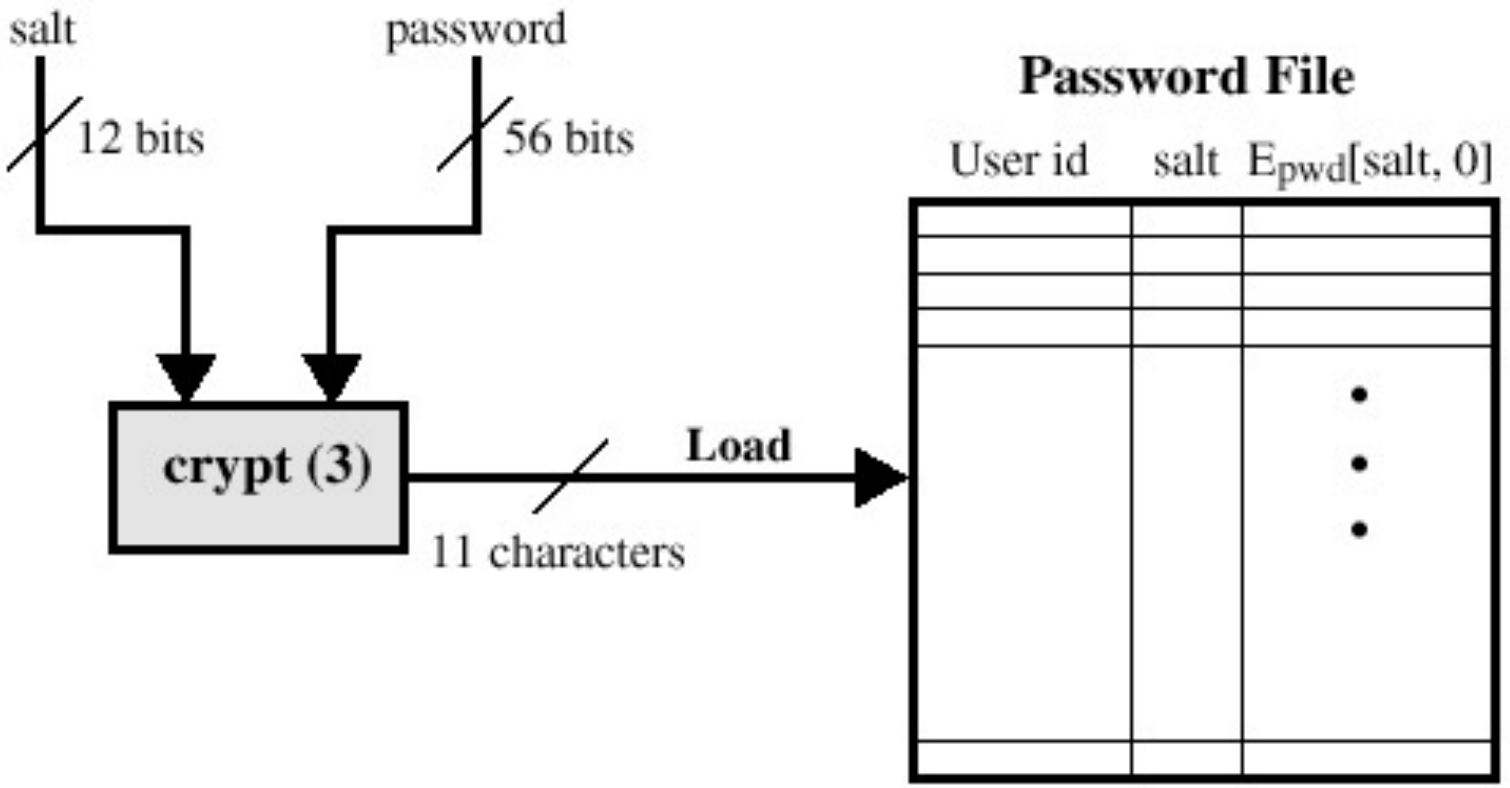
Have your account been compromised? Check it at https://haveibeenpwned.com



Intrusion Techniques

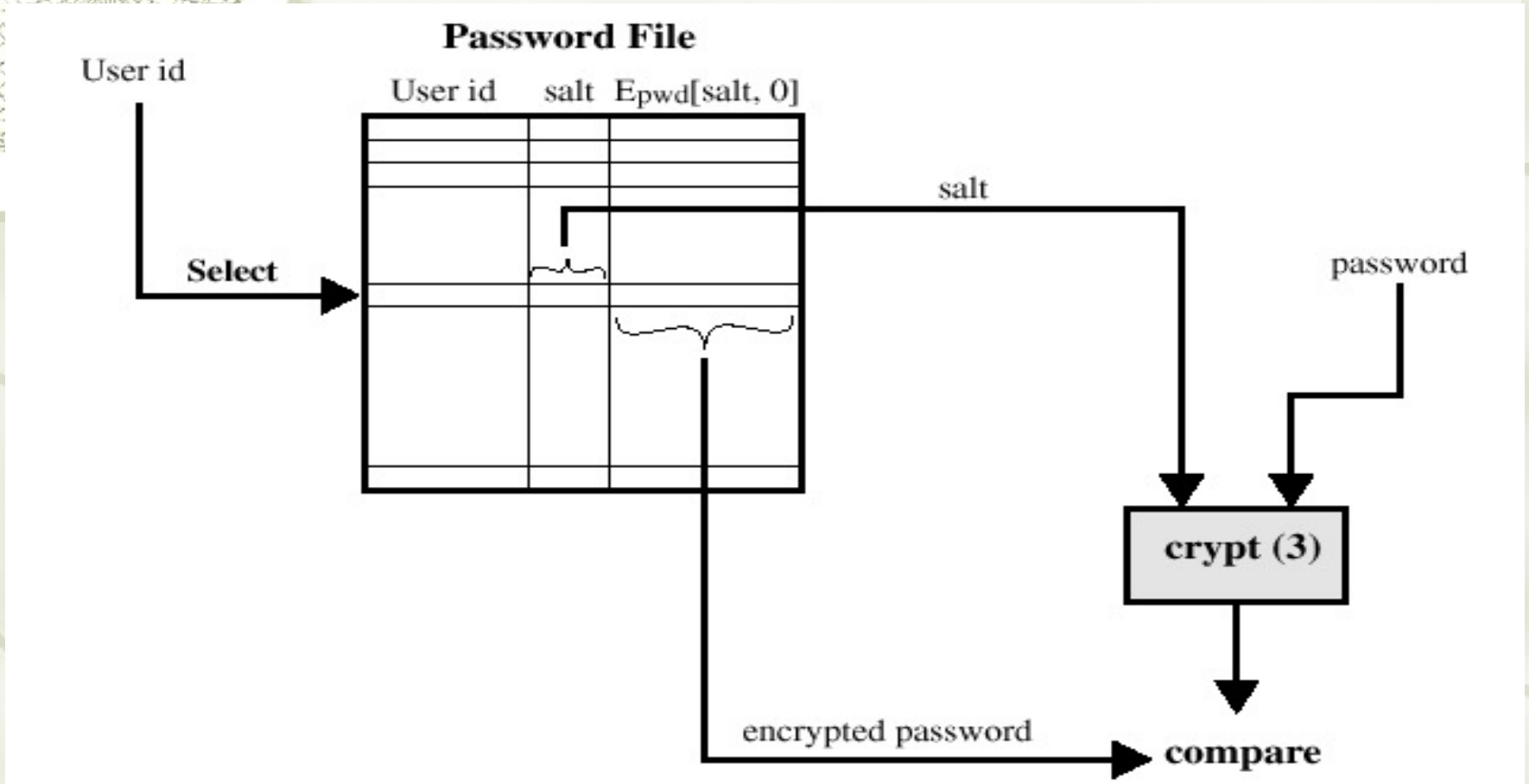
- A main goal of an attacker is to gain access to computing resources
- Systems maintain a file that associates a password with each authorized user.
- Password file can be protected with:
  - One-way encryption
  - Access Control

UNIX Password Scheme



Loading a new password

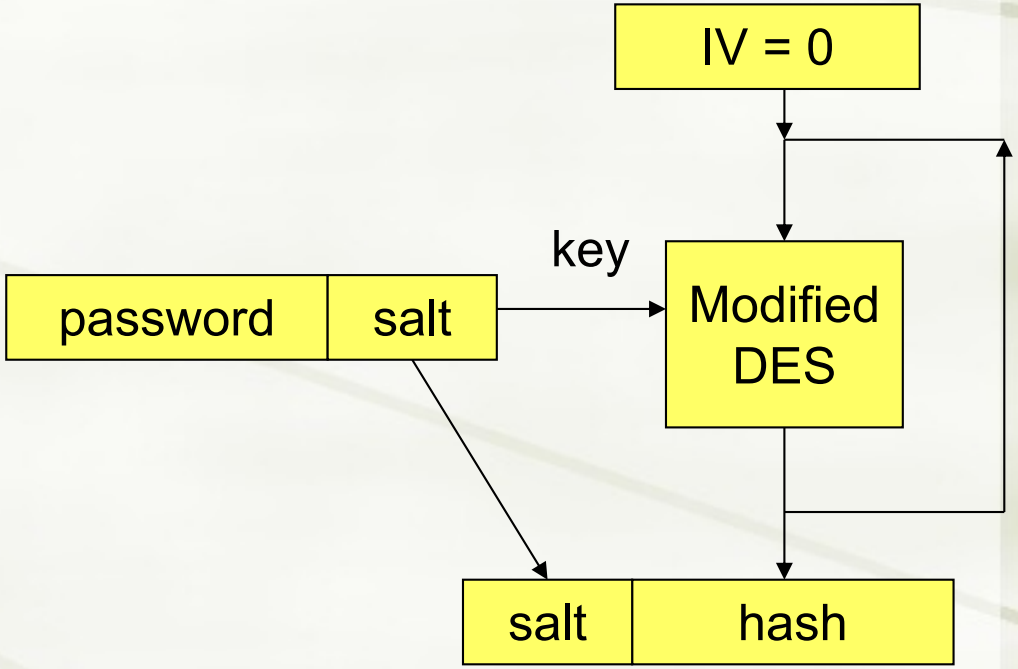
UNIX Password Scheme

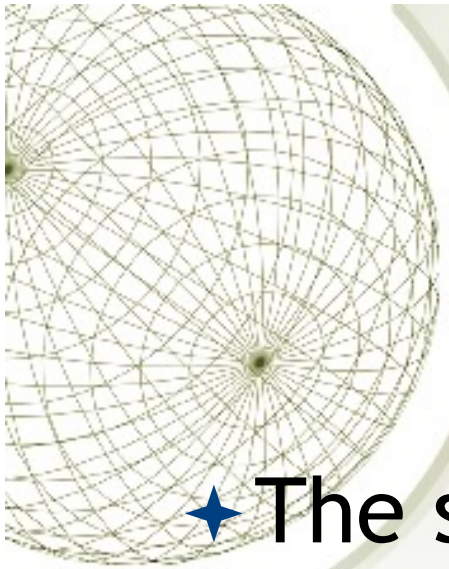


Verifying a password file

UNIX Password Scheme

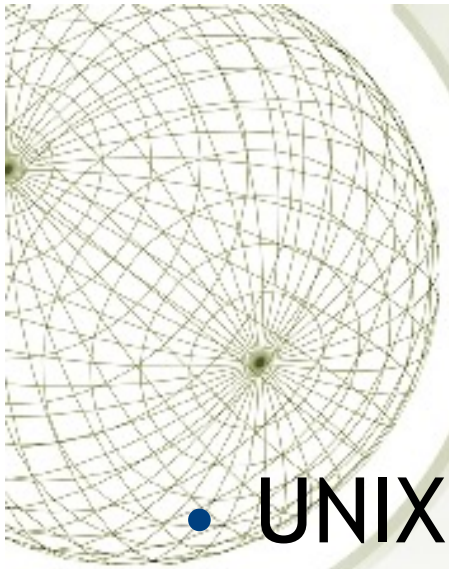
- Hash is stored in /etc/passwd (public) or /etc/shadow (readable by root)
- 8 byte ASCII password is used as 56-bit key to a modified DES
- Iterated thousands of times to slow down brute force guessing
- 12 bit salt used to thwart table lookup and detection of reused passwords
- DES modified to thwart hardware acceleration





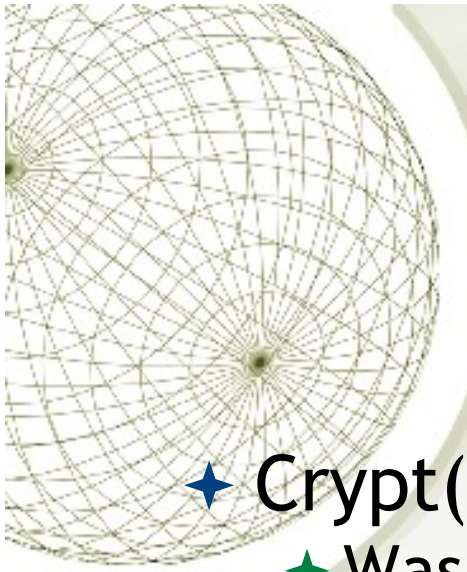
"Salt"

- The salt serves three purposes:
  - Prevents duplicate passwords.
  - Effectively increases the length of the password.
  - Prevents the use of hardware implementations of DES



Storing UNIX Passwords

- UNIX passwords were historically kept in in a publicly readable file, etc/passwords
- Now they are kept in a “shadow” directory and only visible by “root”



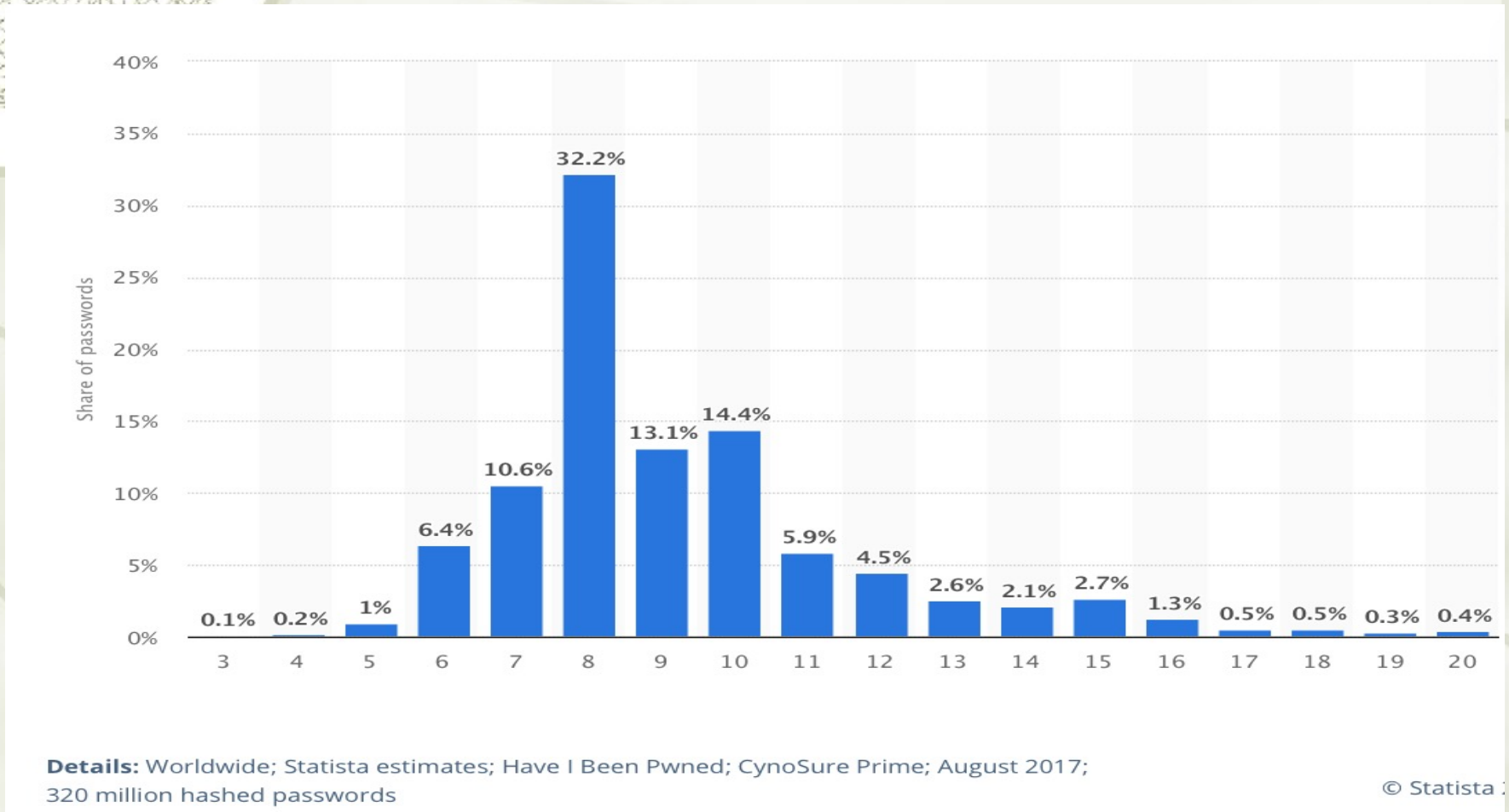
Problems with Crypt(3)

- Crypt(3)
  - Was designed to discourage guessing attacks
  - This particular implementation is now considered inadequate
  - Despite its known weaknesses, this UNIX scheme is still often required for compatibility with existing account management software or in multivendor environments
  - To gain greater cryptographic security and resistance to brute-force attacks, modern versions of Unix now have a variety of new password hash schemes implemented using the crypt() interface

Other Unix implementations

- BSDi
  - Modified the original DES-based scheme, extending the salt to 24 bits and making the number of rounds variable (up to 2^{24}-1). The chosen number of rounds is encoded in the stored password hash
  - The BSDi algorithm also supports longer passwords, using DES to fold the initial long password down to the eight bytes supported by the original algorithm.
- MD5 secure hash algorithm
  - The recommended hash function for many UNIX systems, including Linux, Solaris, and FreeBSD
  - Far slower than crypt(3)
- Bcrypt
  - Developed for OpenBSD
  - Probably the most secure version of the UNIX hash/salt scheme
  - Uses a hash function based on the Blowfish symmetric block cipher
  - Slow to execute
  - Includes a cost variable

Average number of characters of leaked user passwords worldwide (2017)



Source: https://www.statista.com/statistics/744216/worldwide-distribution-of-password-length/

Cracked passwords

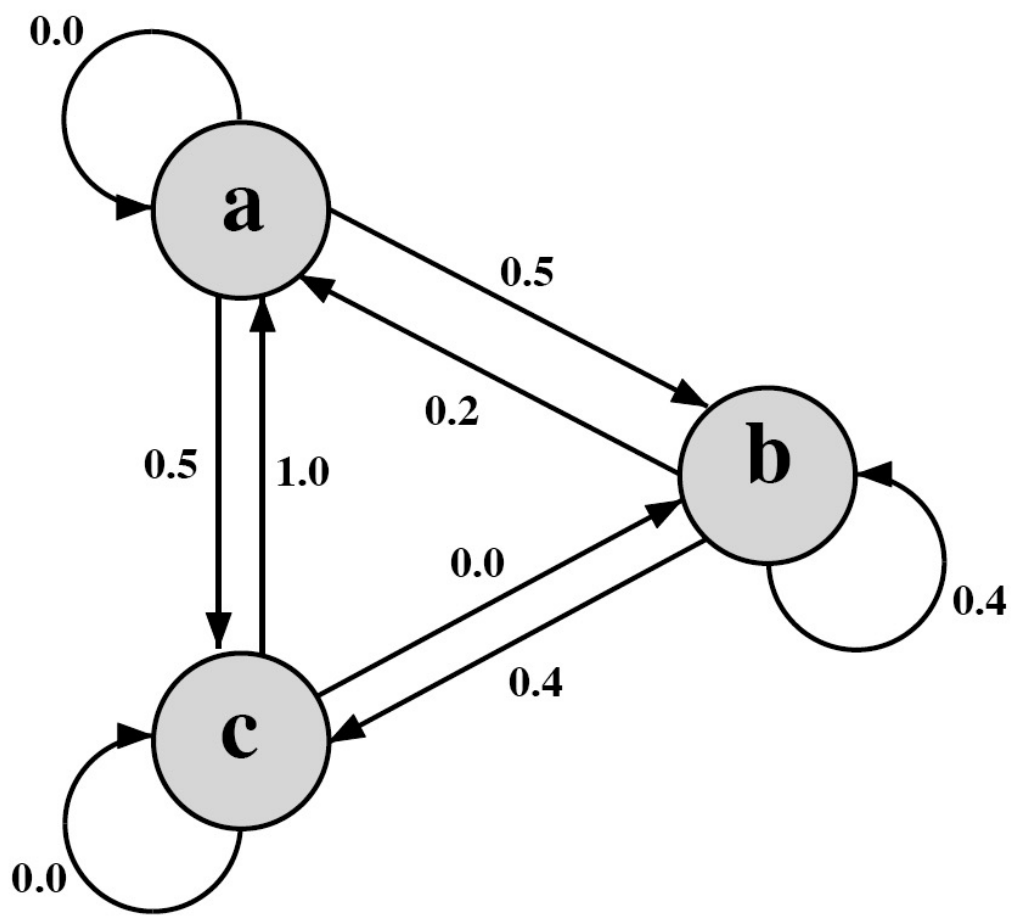
Type of Password	Search Size	Number of Matches	Percentage of Passwords Matched	Cost/Benefit Ratio[a]
User/account name	130	368	2.7%	2.830
Character sequences	866	22	0.2%	0.025
Numbers	427	9	0.1%	0.021
Chinese	392	56	0.4%	0.143
Place names	628	82	0.6%	0.131
Common names	2239	548	4.0%	0.245
Female names	4280	161	1.2%	0.038
Male names	2866	140	1.0%	0.049
Uncommon names	4955	130	0.9%	0.026
Myths & legends	1246	66	0.5%	0.053
Shakespearean	473	11	0.1%	0.023
Sports terms	238	32	0.2%	0.134
Science fiction	691	59	0.4%	0.085
Movies and actors	99	12	0.1%	0.121
Cartoons	92	9	0.1%	0.098

Cost/Benefit Ration is Computed as the number of matches divided by the search size.
The more words that needed to be tested for a match, the lower the cost/benefit ratio.

Password Selecting Strategies

- User education
- Computer-generated passwords
- Reactive password checking
- Proactive password checking

Markov Model – proactive password checker

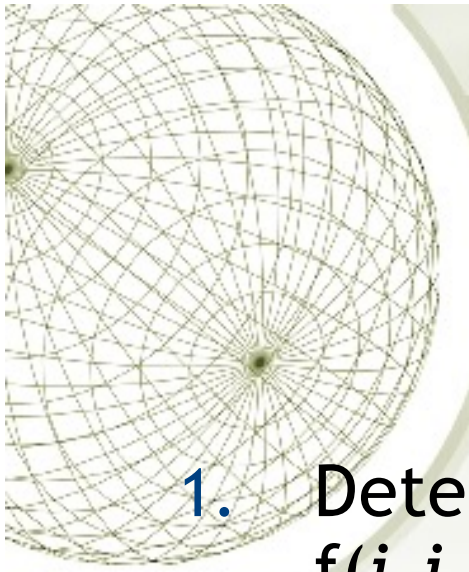


M = {3, {a, b, c} , T, 1 } where

$$T = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.4 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

e.g., string probably from this language: abbcacaba

e.g., string probably not from this language: aacccbaaa



Transition Matrix (second order model)

1. Determine the frequency matrix f, where f(*i*,*j*,*k*) is the number of occurrences of the trigram consisting of the *i*th, *j*th and *k*th character.
2. For each bigram *ij*, calculate f(*i*,*j*,∞) as the total number of trigrams beginning with *ij*.
3. Compute the entries of T as follows:

$$T(i,j,k) = \frac{f(i,j,k)}{f(i,j,\infty)}$$

Spafford (Bloom Filter)

$$H_i(X_i) = y \quad 1 \le i \le k; \quad 1 \le j \le D; \quad 0 \le y \le N-1$$

where

$$X_i = jth\ word\ in\ password\ dictionary$$

$$D = number\ of\ word\ in\ password\ dictionary$$

The following procedure is then applied to the dictionary:

1. A hash table of N bits is defined, with all bits initially set to 0.
2. For each password, its *k* hash values are calculated, and the responding bits in the hash table are set to 1

Spafford (Bloom Filter)

- Design the hash scheme to minimize false positive.
- Probability of false positive:

$$P \approx (1 - e^{-kD/N})^k = (1 - e^{-k/R})^k$$

$$or, equivalently,$$

$$R \approx \frac{-k}{\ln(1 - P^{1/k})}$$

where

$k = number\ of\ hash\ function$

$N = number\ of\ bits\ in\ hash\ table$

$D = number\ of\ words\ in\ dictionary$

$R = N / D, ratio\ of\ hash\ table\ size\ (bits)\ to\ dictionary\ size\ (words)$

Performance of Bloom Filter