# ЗАШТИТА ПОДАТАКА

## ЗАШТИТА СИСТЕМА

Детекција упада у систем

# Преглед

- Биће објашњено:
  - Детекција упада у систем
  - Приступи детекцији упада у систем
    - детекција статистичких аномалија
      - детекција помоћу граничних вредности
      - детекција заснована на профилима
    - детекција заснована на правилима
      - детекција аномалија
      - идентификација пенетрације
  - Дистрибуирана детекција упада
  - Ћупови са медом (Honeypots)

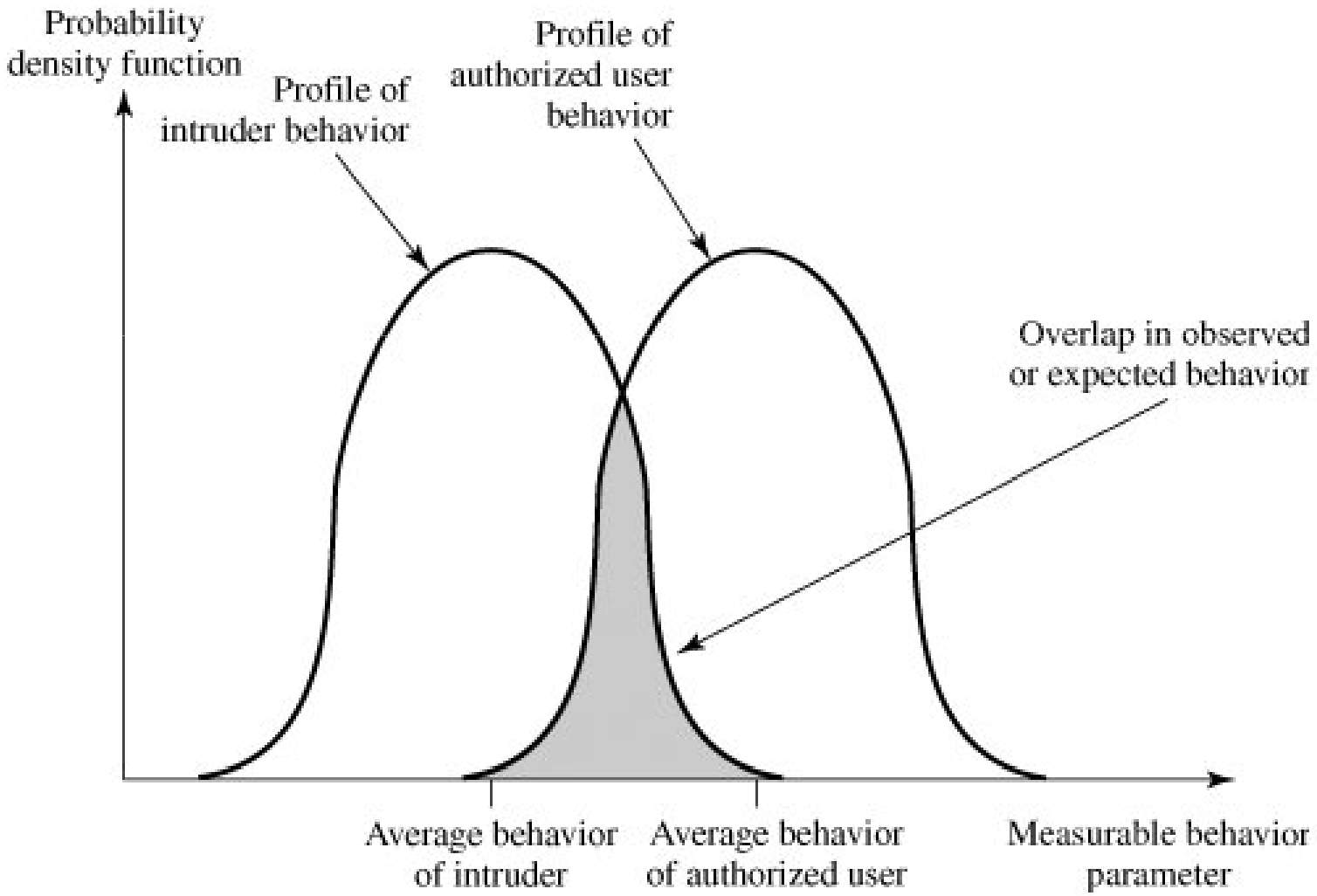
# Детекција упада у систем

- Неизбежно, и најбољи системи за превенцију упада у систем ће заказати (имаће пропусте).
- Друга линија одбране система је детекција упада у систем, која омогућава:
  - да ако се упад детектује довољно брзо, може се идентификовати уљез и избацити из система (пре него што направи штету или уз минималну штету),
  - ефикасна детекција уљеза може да застраши потенцијалне уљезе и на тај начин их одврати од покушаја упада у систем
  - да се сакупљају подаци о техникама упада, који се могу користити да се ојача систем за превенцију упада.

# Детекција упада у систем(2)

- Детекција упада се заснива на претпоставци да се понашање уљеза разликује од понашања легитимног корисника, на начин који је мерљив.
- Наравно, не може се очекивати да ће постојати егзактна дистинкција, већ се мора претпоставити да ће постојати преклапање у понашању легитимног корисника и уљеза.

# Детекција упада у систем(3)



# Детекција упада у систем(4)

- Због преклапања у понашању легитимног корисника и уљеза, пројектовање система за детекцију упада је изазован задатак.
- Лабава дефиниција понашања уљеза, која ће детектовати више уљеза, довешће и до већег броја “лажно позитивних” детекција, тј. легитимних корисника који су детектовани као уљези.
- Чврста дефиниција понашања уљеза, која ће имати мањи број “лажно позитивних” детекција, довешће до већег броја “лажно негативних” детекција, тј. уљеза који нису детектовани као уљези.
- Мора постојати компромис приликом дизајнирања система за детекцију упада.

# Детекција упада у систем(5)

- Једна студија, показала је следеће:
  - разлика између улъеза типа *masquerader* и легитимног корисника, може се, јако поуздано, уочити праћењем понашања легитимног корисника, односно одступањем овог улъеза од таквог понашања,
  - детекција улъеза типа *misfeasor* је много тежа, јер разлика између регуларног и нерегуларног понашања може бити јако мала; ипак, овај улъез се може детектовати постављањем услова који означавају неауторизовани приступ,
  - детекција улъеза типа *clandestine user* је изван опсега чисто аутоматизованих техника.
- Ове обзервације које су начињене 1980. и данас су истините.

# Приступи детекцији упада

- Детекција статистичких аномалија (**statistical anomaly detection**) - укључује колекције података повезаних са понашањем корисника у одређеном временском периоду. Над оваквим подацима се примењују статистички тестови, којима се проверава да ли је понашање корисника легитимно.
  - детекција помоћу граничних вредности (**threshold detection**) - дефинисање граничних вредности фреквенција догађања појединих догађаја (независно од корисника),
  - детекција заснована на профилима (**profile based**) - развија се профил активности сваког корисника и затим се користи да се детектују промене у понашању.
- Детекција заснована на правилима (**rule-based detection**) - покушај да се дефинише сет правила, која се могу користити да се одлучи да ли је неко понашање нерегуларно.
  - детекција аномалија (**anomaly detection**) - развијају се правила, која детектују девијације од претходних шаблона коришћења,
  - идентификација пенетрације (**penetration identification**) - експертски систем који трага за сумљивим понашањем.

# Приступи детекцији упада (2)

- У суштини, статистички приступи покушавају да дефинишу очекивано (нормално) понашање, док приступи базирани на правилима покушавају да дефинишу правилно (исправно) понашање.
- Ако говоримо о типовима нападача, које смо поменули раније, онда можемо да констатујемо следеће:
  - детекција статистичких аномалија је ефикасна против уљеза типа *masquerader*, који тешко да ће успети да имитира шаблоне понашања налога које присвоје.
  - детекција заснована на правилима је ефикасна против уљеза типа *misfeasor,* јер може да препозна догађаје и секвенце, који могу открити пробој у систем.
- У пракси, систем може користити комбинацију оба приступа, како би био ефикасан против ширег спектра напада.

# Записи ослушкивања (Audit Records)

- основни алат за детекцију упада.
- Некакав запис активности корисника мора се одржавати као улаз система за детекцију упада. Постоје два начина:
- изворни записи ослушкивања (**native audit records)**
  - део свих вишекорисничких оперативних система
  - предност: већ постоји и може се користити
  - мана: можда не постоје сви потребни подаци, или неки подаци нису у одговарајућој форми
- записи ослушкивања специфични за детекцију (**detection-specific audit records)**
  - креиран посебно за прикупљање података потребних систему за детекцију
  - предност: независан у односу на платформу
  - мана: додатно оптерећење система, при покретању додатног софтвера за сакупљање података о понашању корисника

# Записи ослушкивања специфични за детекцију пример

- Сваки запис ослушкивања (*audit record)* садржи следећа поља:
  - субјекат (**Subject)**: иницијатор акције (корисник или процес),
  - акција (**Action)**: операција коју изводи *субјекат* над *објектом* (логовање, читање, упис, улаз/излаз, извршавање),
  - објекат (**Object)**: на кога се односи акција (фајл, програм, ...),
  - услов за изузетак (**Exception-Condition)**: који изузетак, ако постоји неки, се баца при повратку,
  - искоришћење ресурса (**Resource-Usage)**: листа ресурса и количине за сваки употребљен ресурс,
  - временски жиг (**Time-Stamp)**: време и датум када се догодила акција.

# Записи ослушкивања специфични за детекцију пример(2)

- Већина операција корисника састоји се од елементарних акција.
- Нпр, копирање фајла подразумева извршавање команде, која обухвата валидацију приступа и припрему копирања, читање из једног фајла и упис у други фајл.
- Размотримо команду:

  COPY GAME.EXE TO <Library>GAME.EXE

  коју издаје корисник *Smith* да би копирао извршни фајл *GAME* из текућег директоријума у <*Library*> директоријум. Генерисани су следећи записи ослушкивања (*audit records)*:

| | | | | | |
|---|---|---|---|---|---|
| Smith | execute | <Library>COPY.EXE | 0 | CPU = 00002 | 11058721678 |

| | | | | | |
|---|---|---|---|---|---|
| Smith | read | <Smith>GAME.EXE | 0 | RECORDS = 0 | 11058721679 |

| | | | | | |
|---|---|---|---|---|---|
| Smith | write | <Library>GAME.EXE | write-viol | RECORDS = 0 | 11058721680 |

# Записи ослушкивања специфични за детекцију пример(3)

- Декомпозиција операције корисника на елементарне акције има три предности:
  1. Због тога што су објекти заштићени ентитети у систему, коришћење елементарних акција омогућава праћење свих понашања која утичу на објекте. Тако да систем може да детектује покушаје нарушавања контроле приступа, пратећи абнормалности у броју изузетака који су бачени приликом повратка, и може да детектује успешне покушаје нарушавања контроле приступа, пратећи абнормалности у скупу објеката којима субјекат има приступ.
  2. Један објекат, једна акција записи ослушкивања упрошћавају модел и имплементацију.
  3. Због једноставне, униформне структуре записа ослушкивања специфичних за детекцију, може бити релативно лако доћи до потребних информација или барем до дела информација једноставним мапирањем из постојећих изворних записа ослушкивања (*native audit records)* у записе ослушкивања специфичне за детекцију (*detection-specific audit records)*.

# Детекција статистичких аномалија

- Детекција помоћу граничних вредности
  - подразумева бројање догађаја одређеног типа у одређеном временском интервалу и упоређивање тог броја са унапред задатом граничном вредношћу за тај догађај,
  - ова метода сама за себе је груба и неефикасна, јер морају да се одреде и гранична вредност и временски интервал,
  - али уз неку софистицираниј у методу, може да буде корисна.
- Детекција заснована на профилима
  - подразумева дефинисање претходног понашања корисника (или група корисника) и затим проналажење одступања од таквог понашања,
  - профил може да се састоји од мноштва параметара, па онда не мора да се сигнализира узбуна за промену само једног од њих, већ могу да се дефинишу услови за узбуну,

# Детекција заснована на профилима

- У основи ове методе је анализа записа ослушкивања.
- Записи ослушкивања обезбеђују улаз функцији за детекцију упада на два начина:
  - анализа записа ослушкивања кроз одређени временски интервал, омогућава да се направи профил активности просечног корисника,
  - и друго, текући записи ослушкивања су улаз за детекцију упада.
- За дефинисање профила корисника, потребно је дефинисати метрику, којом ће се мерити одређени догађаји. Код детекције засноване на профилима примери параметара метрике су:
  - бројач - **counter** (нпр. број логовања корисника у току једног сата, број извршавања одређеног програма у току једне сесије,...)
  - штоперица за интервал - **interval timer** (нпр. време између два узастопна логовања, ...)
  - употреба ресурса - **resource utilization** (нпр. број одштампаних страница у оквиру сесије, укупно трајање извршавања програма, ...)

# Детекција заснована на профилима (2)

- На основу дефинисане метрике, могу се применити различити тестови, који помажу да се открије да ли понашање одступа од профила. Неке од метода су:
  - **Средња вредност и стандардна девијација** - мери се стандардна девијација неког параметра у временском периоду,
  - **Вишеваријацијски модел** - заснива се на корелацијама два или више параметара (нпр. процесорско време и употреба ресурса),
  - **Марковљев модел** - користи се да се установе вероватноће прелаза између различитих стања,
  - **Временске серије** - заснива се на временским интервалима и на тражењу секвенци догађаја који се одигравају превише брзо или превише споро,
  - **Операциони модел** - се базира на процени тога шта је абнормално, уместо на аутоматској анализи *audit records* (нпр. велики број покушаја логовања у кратком временском интервалу се може сматрати нападом).

# Детекција заснована на профилима (3)

| 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. |

| 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. |
| **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. |

# Детекција заснована на профилима (3)

- Кључна предност коришћења статистичких профила је да није потребно претходно знање о сигурносним недостацима.
- Програм за детекцију научи шта је нормално и онда тражи одударања од тога.
- Ово га чини портабилним.

# Детекција заснована на правилима

- Функционишу тако што посматрају догађаје у систему и на основу скупа правила доносе одлуку да ли је неки шаблон понашања сумњив или није.
- Детекција аномалија
  - слична је по приступу и предностима као детекција статистичких аномалија,
  - у овом случају, историјски записи ослушкивања (*audit records*) се користе да идентификују шаблоне коришћења и да аутоматски генеришу правила која описују те шаблоне,
  - затим се посматра текуће понашање и свака трансакција се пропушта кроз скуп правила да би се установило да ли се поклапа са неким историјским шаблоном понашања,
  - као и код детекције статистичких аномалија, не захтева се претходно познавање безбедоносних пропуста система,
  - да би овај метод био успешан потребна је велика база података са правилима.

# Детекција заснована на правилима (2)

- Детекција пенетрације
  - користи технологију експертских система за детекцију упада у систем,
  - главна карактеристика таквих система је коришћење правила за идентификовање познатих пенетрација или пенетрација које искоришћавају познате слабости,
  - могу се, такође, дефинисати и правила која идентификују сумњиво понашање, чак и када је понашање у оквиру установљених шаблона коришћења,
  - правила одређују стручњаци, а не добијају се аутоматски анализом записа ослушкивања (*audit records*),
  - процедура се састоји у томе да се интервјуишу администратори система и особље задужено за безбедност и да се установе познати сценарији пробоја у систем, као и могуће претње,
  - јачина овог приступа зависи од умећа онога ко саставља правила.

# Детекција заснована на правилима (3)

- Пример једне хеуристике која се може користити у систему за детекцију пенетрације:
  1. Корисници не би требали да читају фајлове из фолдера других корисника.
  2. Корисници не би смели да уписују у фајлове других корисника.
  3. Корисници који се пријаве на систем после радног времена, обично приступају истим фајловима као раније.
  4. Корисници обично не приступају дисковима директно, већ се ослањају на сервисе оперативног система.
  5. Корисници не би требало да буду улоговани више од једанпут на истом систему.
  6. Корисници не праве копије системских програма.

# Циљ система за детекцију упада

- Да би био од користи, систем за детекцију упада треба да детектује значајан проценат упада у систем, а да при томе има што мањи број лажних аларма.
  - мали проценат детектованих упада = лажни осећај сигурности
  - много лажних аларма = трошење времена на анализу лажних аларма
- Системи за детекцију упада нису још увек успели да превазиђу овај проблем.

# Дистрибуирана детекција упада

- До недавно, развој система за детекцију упада базиран је на stand-alone системима.
- Иако су рачунари који су требали да буду покривени детекцијом упада, били повезани у мрежу (LAN).
- Иако је могуће користити stand-alone системе за детекцију упада на сваком рачунару понаособ, ефикаснија одбрана се постиже координацијом и кооперацијом система за детекцију упада широм мреже.

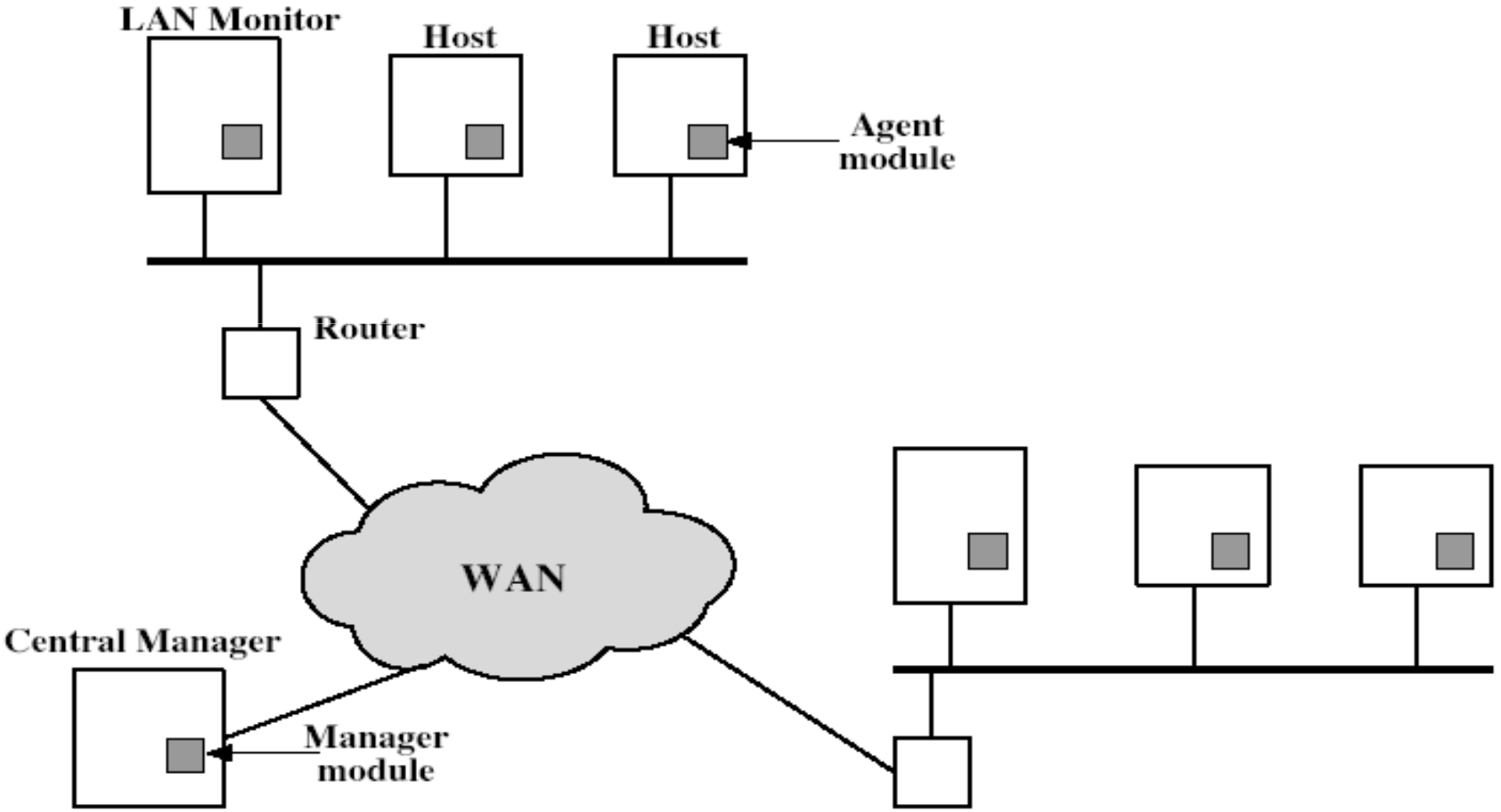
# Дистрибуирана детекција упада(2)

- Главна разматрања при дизајнирању система за дистрибуирану детекцију упада:
  - систем за дистрибуирану детекцију упада ће се можда суочити са различитим форматом записа ослушкивања,
  - један или више чворова у мрежи ће бити задужени за сакупљање и обраду података система у мрежи, па због тога мора да се размотри питање сигурног слања података кроз мрежу (интегритет и поверљивост),
  - могу се користити централизована или децентрализована архитектура. Код централизоване, олакшано је повезивање примљених извештаја, али се ствара потенцијално уско грло система и јединствена тачка отказа система. Код децентрализоване, постоји више центара за анализу, па се мора успоставити комуникација и размена информација између њих.

# Дистрибуирана детекција упада - пример

- Систем се састоји од три главне компоненте:
  - **Host agent module**: модул за сакупљање audit записа који функционише у позадини посматраног система. Сврха овог модула је да сакупља податке о догађајима који имају везе са безбедношћу и да их шаље централном менаџеру.
  - **LAN monitor agent module**: функционише на исти начин као *host agent module*, осим што анализира LAN саобраћај и извештава о резултатима централног менаџера.
  - **Central manager module**: Прима извештаје од LAN monitor и host агената и процесира и повезује ове извештаје како би детектовао упад.

# Дистрибуирана детекција упада - пример (2)



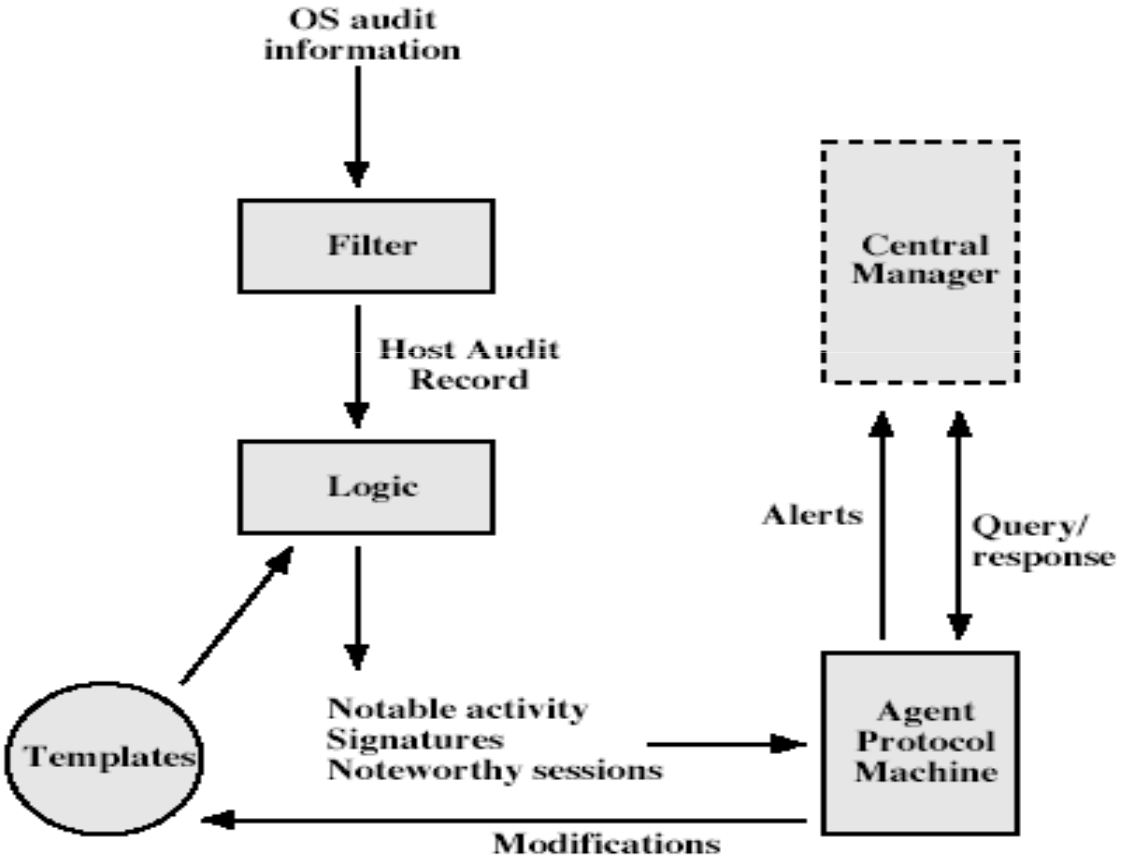
# Дистрибуирана детекција упада - пример (3)

- Агент дохвата сваки запис ослушкивања који произведе изворни систем за сакупљање записа ослушкивања (*nativ audit records*).
- Примени се филтер који задржава само оне записе, који су од безбедносног интереса.
- Ови записи се затим реформатирају у стандардизовани формат, који се назива **host audit record** (HAR).
- Затим модул заснован на шаблонској логици анализира записе ради утврђивања сумљивих активности. На најнижем нивоу, агент тражи карактеристичне догађаја, невезано за прошле догађаје (нпр. неуспешан приступ фајлу, приступ системским фајловима, ...). На вишем нивоу, агент тражи секвенце догађаја, као што су познати шаблони напада. На крају, агент тражи понашања која одударају од профила корисника.

# Дистрибуирана детекција упада - пример (4)

- Када се детектује сумњива активност, централном менаџеру се шаље сигнал за узбуну.
- Централни менаџер обухвата експертски систем, који може да изводи закључке из добијених података. Менаџер може и да испитује (query) поједине системе и да тражи од њих копије HAR записа како би их довео у везу са записима добијеним од осталих агената.
- LAN monitor agent такође шаље податке централном менаџеру. LAN monitor agent прати конекције између host-ова, сервисе који се користе и проток саобраћаја. Тражи значајне догађаје, као што су изненадне промене у протоку саобраћаја, коришћење сервиса везаних за безбедност, и сл.

# Дистрибуирана детекција упада - пример (5)



# Ћупови са медом (Honeypots)

- Иновација код система за детекцију упада.
- Представљају мамац системе, који су дизајнирани да одвуку потенцијалне нападаче од критичних тачака система.
- Ћупови са медом су дизајнирани да:
  - одвуку нападача од приступања критичним системима
  - сакупљају податке о активностима нападача
  - охрабре нападача да остане у систему довољно дуго да администратори могу да одреагују

# Ћупови са медом (Honeypots) (2)

- Ови системи су напуњени са измишљеним подацима, смишљеним тако да изгледају значајно, а којима легалан корисник не би могао да приступи. Тако да је сваки приступ овим подацима сумњив.
- Систем је опремљен осетљивим агентима за праћење и прикупљање информација о акцијама корисника, тако да сакупља све информације о активностима нападача.
- Пошто је сваки напад на ћупове са медом приказан нападачу као успешан, администратори имају довољно времена да прате нападача, а да он то и не примети.
- Првобитни системи су подразумевали један рачунар који је представљао ћуп са медом и који је имао IP адресу која је дизајнирана да привуче нападаче. Скорашњи системи подразумевају читаву мрежу ћупова са медом, који симулирају компанију. Када се хакери нађу у мрежи, администратори могу да посматрају и анализирају њихово понашање у циљу смишљања адекватних одбрана.