x 40-ish slides on analyzing threats x

owasp helsinki meeting, 12.12.2006

olli@juurihoito.org

introductions

me:

- **work:** incident response, testing and consulting internally
  - unix back and foreground
- **life:** movies, music, parrots and computers for soul
  - my better half calls me 'napu-napu', so i ought to select more carefully stuff from life section i suppose :-(

- native from turku, i bet that explains a lot

what about you?

toc

- why threat analysis / modeling / whatcha call it
- what's it about
- some experiences from real life
- suggested reading material
- chit chat (optional, we’re finns, right?)

opinions, questions -> interrupt me

well, why?

- a lot of presentations out there on
  - testing web application security,
  - coding practices,
  - design patterns,
  - selling you stuff (or a consultant)
- they're all great, but ...

well, why?

:-(

..you always end up with hidden use cases, features and design compromises (yes, bugs)

more on this 'why' thingy

- threat analysis can help along this way to..
  - understand the operating environment your gizmo is heading
  - identify, analyze and document (and thus hopefully mitigate) threats
  - be your source of “told you so's” after compromises
    - this might not bring you friends though :)

more on this 'why' thingy

- even still, according to 'surprise pattern'..

you still end up with bugs, design compromises and stuff :-(

hopefully almost last slide on this 'why' phenomenon

good news everyone!,

threat analysis when

- documented properly
- communicated to relevant parties
  - from business to developers through architects and project managers..

last slide on this 'why' phenomenon

..can assist you to focus on most critical security issues as they're known and weighted.

identifying threats - overview

- assets
- input/output (and of course “temput”)
- exposure (internal, external, distributed, centralized..)
- threat types (patterns)
- impact (who, how, why)

identifying threats - assets

assets

- have typically some value
- from photograph to leatherman through order history and credit card numbers depending on context
- losing control/sight/availability/whatever of such, can cause you some sort of grief *) indirectly
  - like women, which are often thus mistakenly thought as assets but in fact are threats, wallet being the asset endangered :)

*) we often feel bad at least when losing money, see threats toward a wallet **)

identifying threats -

**) no, i wouldn’t say all this out loud to my girlfriend, please don’t tell

identifying threats – inputs, outputs

inputs, outputs

- input: something an entity (human, software) can utilize to access (RWX) an asset
- ouput: something resulting from above mentioned
  - like XSS through, say, an error page which accept user supplied data and stick it in resulting pages as is without sanitation

identifying threats - exposure

exposure

- anonymous access to any inputs can be bad and should be carefully studied
- depending on system, authentication alone can help to mitigate likelihood of a threat considerably
  - root cause may need to be removed of course if a valid vector exists (business decision)
  - in case threat exposure is limited, this may be sufficient mitigation alone (risk management).

identifying threats – system modeling

your way can be..

- data flow & context diagrams
  - depth of modeling based on need, available resources including time
- sequence diagrams
- anything preferably already existing
  - people don't often have time to model things just because of security
- state diagram thingies
- <insert your favorite here>

identifying threats – context diagram

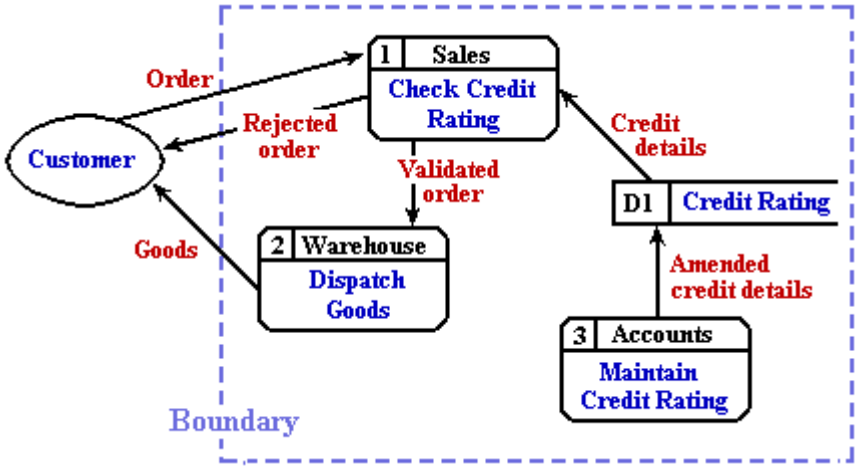
context diagrams

- very high level abstraction of a system
- eases in understanding the big picture
- dfd models by this Tony Drewry person, not me
  - borrowing with pride:)

identifying threats – data flow diagrams

dfd, level0

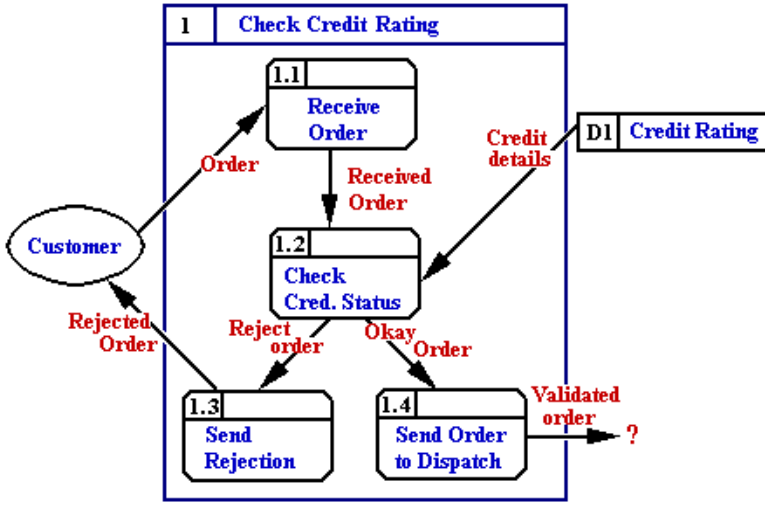
- contains the major processes, system boundaries
- .. interactions with external entities,



identifying threats – DFD's

dfd, level1

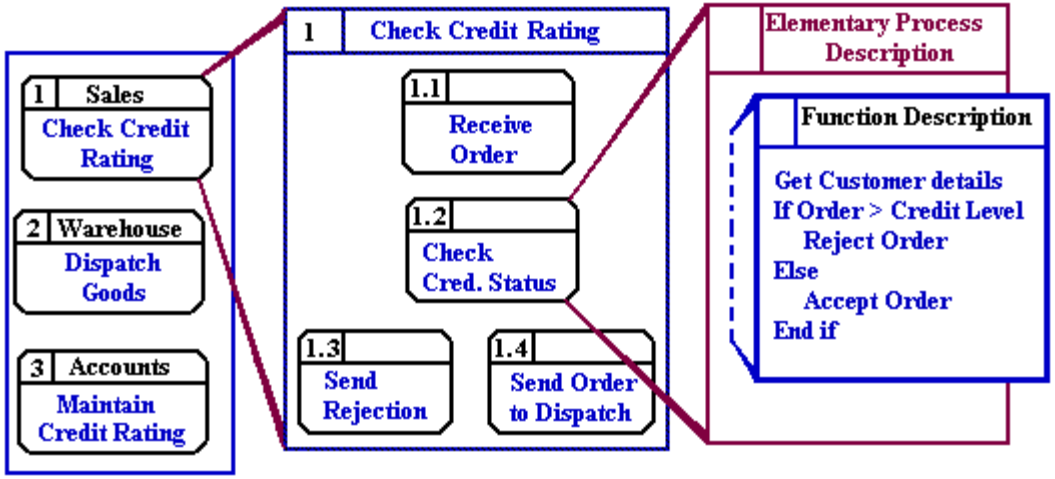
- further decomposition of high level processes
- .. and so on, untill



identifying threats – DFD's

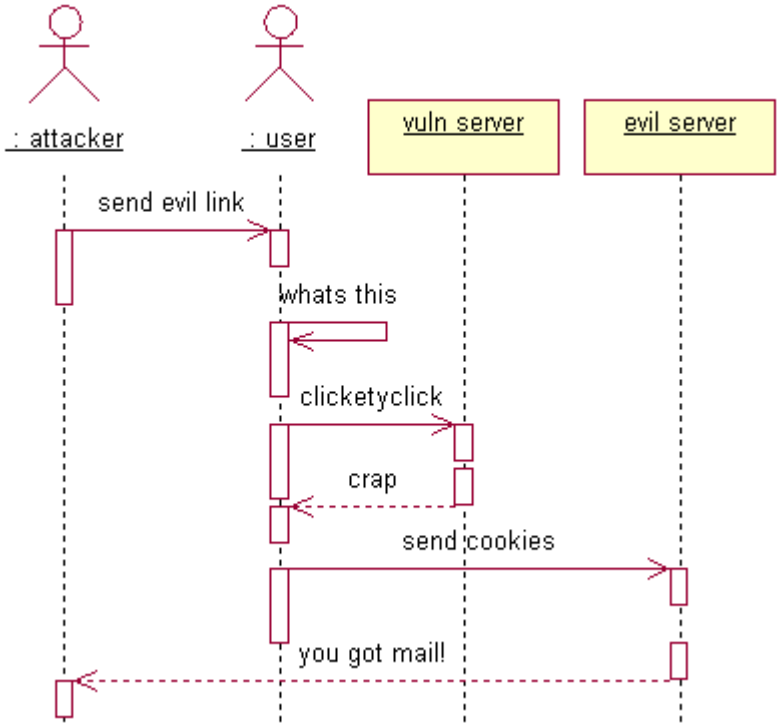
..things can be described in, like in the example, pseudocode.

more information from the author's site,
http://www.cems.uwe.ac.uk/~tdrewry/dfds.htm



identifying threats – sequence diagrams

.. another way of modeling things



identifying threats – quantifying and categorizing

- various acronyms used, two most known are
  - microsoft dread, stride
    - **damage potential, reproducibility, exploitability, affected users, discoverability**
    - spoofing, tampering, repudiation, information disclosure, denial of service, **e**scalation of privileges
  - more describing than ones based on impact and likelihood scale
  - stride's good too, if you do a lot of them you can see what are root causes for n most severe types of vulns or something.

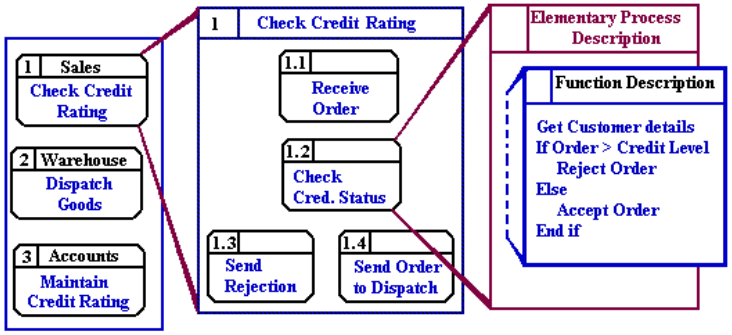
identifying threats – in general,

- most important thing is that used models assist in understanding and communicating
  - the system functionality
    - uniform models are good if you'll do comparison as well
  - the threats also to non-techies such as managers

- It really does not matter that much how you choose to model, document.
  - pick something you're cool with.

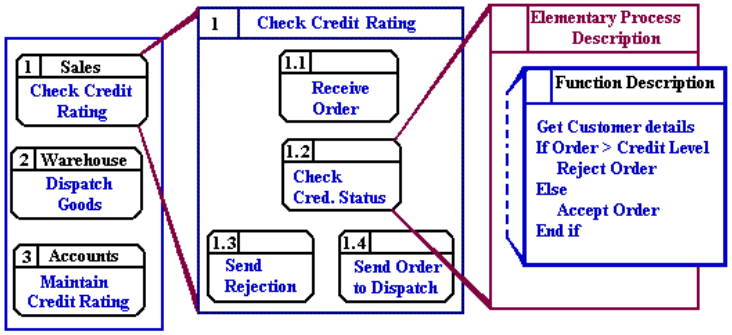
identifying threats – example

let's leave this one as an example



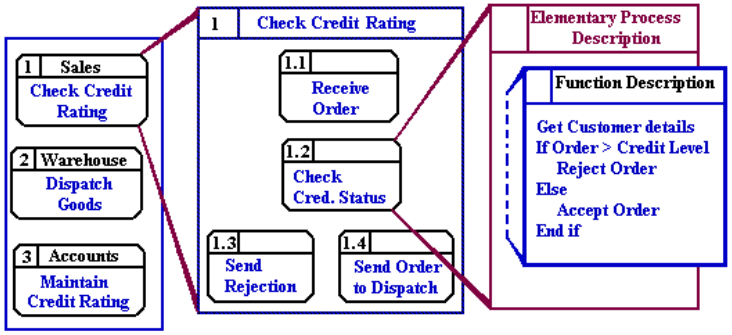
- assets (some):
  - uid,passwd
    - how transmitted
    - how stored, where
      - password auto complete?
  - personal information
    - useful in social engineering
  - email addresses
    - useful in mass mailing purposes
  - credit card numbers
    - Are they stored or just proxied ..?

identifying threats – example



- input,output (some):
  - interfaces modeled
    - *M* transport layer security, authentication?
  - any other services on servers not filtered
    - *M* hardening the system
  - human beings (social engineering)
    - *M* training, awareness

identifying threats – example



- threats (some):
  - interface trust model
    - does order sending service authorize and authenticate requests properly – or does it receive account information from some other layer?
      - how much this other layer should be trusted?
  - fraudulent orders,
    - certain patterns exist such as large purchase amounts

identifying threats – example documentation

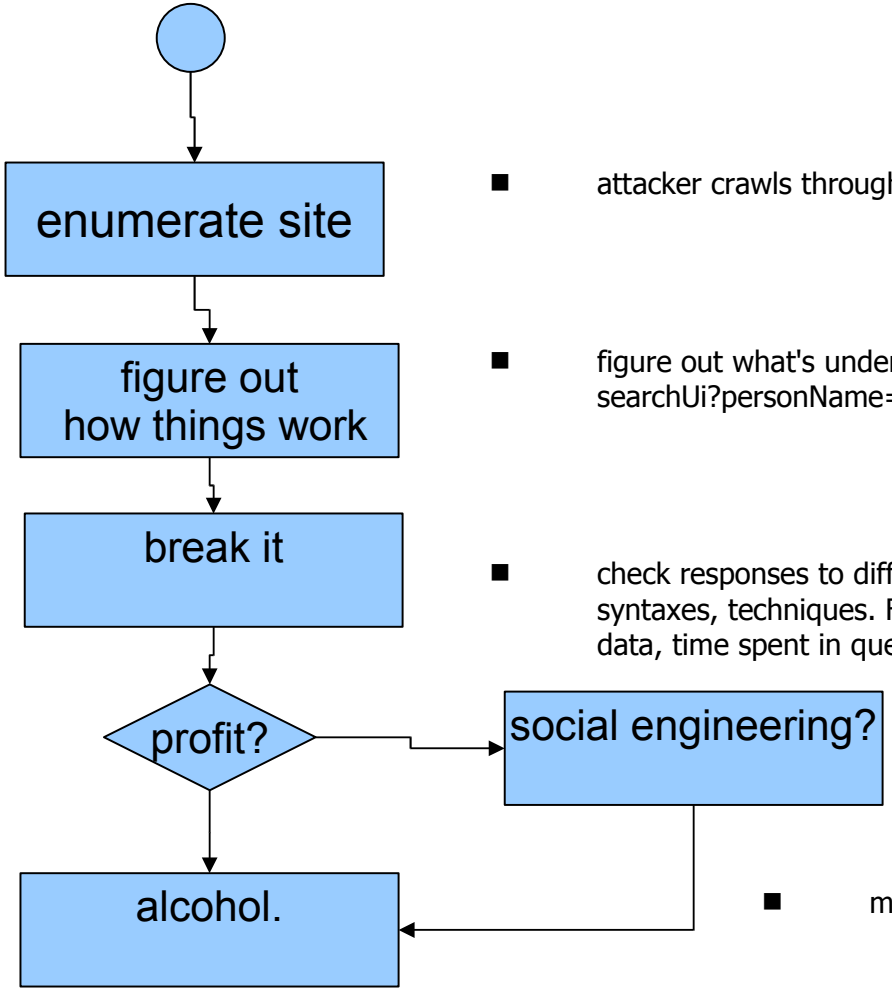
- threat1: interface trust model flawed
  - uid passed from presentation layer
    - **known fix:** Service layer should perform validation also, not just trust (more) insecure layer.
    - **current Mitigation: attack feasible only 1. by gaining access to first layer through another vulnerability 2. when performed by insiders, who on the other hand have better vectors.**
    - **verdict: Damage potential: 6, Affected users: 10, Discoverability 4, Detectability: 5 -> score: 6.25**
    - **responsible person: john doe**

identifying threats – quantifying and categorizing

- depending on vuln, creating sort of an attack tree can help in communicating
  - **when your communicating goes through, someone can actually make an educated decision**
    - **In case right people are at the desk**

identifying threats – quantifying and categorizing

- could be something like,



- attacker crawls through site and maps different parameters, etc
- figure out what's under the hood, like searchUi?personName=jack likely results a sql/ldap/? query
- check responses to different data, attempt multiple vendor syntaxes, techniques. Figure out from results (error messages, data, time spent in query) whether plausible
- some other vector?
- make the voices go away

identifying threats – quantifying and categorizing

- this would do as well

- 1 *__t__* exploiting searchUi input validation vuln
  - 1.1 enumerating site
  - 1.2 figure out how things work
  - 1.3 break it
  - *__m__* no detailed error messages are displayed, *client* side validation in place
  - score: medium

- 2 *__t__* performing social engineering attack
  - 2.1 figuring out stuff from developers comments, whois information, search engines (life beyond google)
  - *__m__* development process includes stripping out comments, people running system do not have time to spend online
  - 2.2 figure out what you just figured out
  - 2.3 be very convincing
  - *__m__* clear processes, roles and responsibilities. training and awareness
  - score: low

identifying threats – quantifying and categorizing

- once again, order of importance

  I. data gathered in analysis

  II. how well the data is communicated and understood

  III. what font you're using

  IV. Where do babies come from

  V. how you model the before mentioned

identifying threats – threat types

- depending on what you do, following could be considered
  - there are a whole lot more, variations, study patterns affecting you
  - malformed data
    - from overflows through sql injection via data corruption due to crappy code
  - session handling
    - random? lengthy enough? protected enough?

identifying threats – threat types

- continued
  - authentication, authorization
    - account locking? eternal, or say 15 minutes?
    - uid,pw vs otp vs pki?
  - networking in general
    - whatever has an ip can attack you, or be attacked
    - good to remember OSI layer 7 is just one layer, though significant
  - bad crypto use
    - the "'i just figured out this new cool algorithm', 'dang you'" conversation as opposed to standard offerings, PKCS etc

identifying threats – threat types

- continued
  - data storage issues
    - filesystem key store sufficient, or something drastic like hsm required.
      - physical protection? processes? degaussing/wiping?
    - data related issues, safe harbour, privacy (loss of it)
    - is it necessary to store data in plaintext? Would one-way hashes work in, say, passwords?
      - would that help? Against what threats?
        - new ones created?

identifying threats – threat dissection

- be specific, dissect threats
- high level ones can be documented, and thought of but...
  - threat 'hacking' difficult to grasp

identifying threats – threat dissection

- threat 'hacking' difficult to grasp
  - threat DoS of productQuery SOA interface is better, one could think different scenarios
    - continuous (recursive) queries from (multiple) clients, would authentication prior queries help? how and why?
    - odd unspecified data? how could that happen..
    - ips like mitigation resulting in DoS threat itself?
    - what about os etc protocol stack, patch management needed? hardening? filtering?

experiences from rl [1/3]

- perceived as worthwhile
  - developers like a whole lot more than 'avoid this' type of issues
  - helps when figuring out resource estimates
    - oh, so you don't create a pki deployment with this as well?
  - assist in grounding & debunking of requirements
    - some you might want to kill, others to base additional budget on
  - is great fun and increases awareness
    - people start to think more security oriented
      - "we have to scrap/alter this feature because of threat xyz"-pattern

experiences from rl [2/3]

- downfalls are plenty of course [1/2]
  - commitment necessary, including management (dang, one needs to do something)
  - despite models, kitchen analogy, cartoons and ppt's...
    - suits have 'omg its technical, cant possibly understand this concept of ones and zeroes' handicap which endangers discussions.
    - split reality horizon

experiences from rl [3/3]

- downfalls are plenty of course [2/2]
  - needed to be done by the persons who know the system
    - outsider good in identifying vectors, internal knows the system and should present it
  - as good results as experience on field
    - if you don't know better, you might think digital watches are still pretty cool

little evening reading

- http://www.octotrike.org , tool to do CRUD models and seems interesting
- threat modelling book
  - ms professional press / swiderski, snyder
  - isbn 0735619913
- tool could be interesting, also ms app threat modeling blog
  - http://blogs.msdn.com/threatmodeling/
  - contains link to current tool download

the near end experience(tm),

- i'm pretty much done here
- questions, suggestions, verbal abuse welcome:)
  - i'll lie the best i can