Threat Modeling; The misunderstood, misapplied, and most misused tool in the development toolbox.

Greg Sternberg
March 2, 2023

Things You Won’t Find In My Bio

- Can’t make my mind up about a career – Geo-Physicist, Chemical Engineer, Hacker, Red/Blue/Purple/Mauve teamer, Programmer, Lead, Manager, Architect, CISO, Teacher, Privacy Advocate, Grumpy Old Guy, …
- Like to say “Just one more thing” in architecture/security reviews.
- Like to open doors that say “Do Not Open” and push buttons that say “Do Not Push”
- Have gotten ‘the look’ from significant others when asked, “What are you thinking about?” and I reply, “Wondering how one might rob this movie theater.”

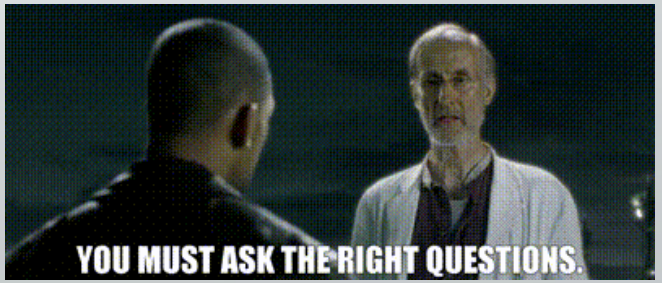
What is Threat Modeling*?

* Disclaimer: Portions of this presentation shamelessly ~~stolen~~, er, borrowed, from a presentation given by Adam Shostack.

Bast: “We’ve analyzed their attack, sir, and there is a danger. Should I have your ship standing by?”

Tarkin: “Evacuate? In our moment of triumph? I think you overestimate their chances!”

- ‘Official’ definitions:
  - Threat modeling is a process by which potential threats, such as structural vulnerabilities or the absence of appropriate safeguards, can be identified and enumerated, and countermeasures prioritized.
  - The purpose of Threat Modeling is to provide defenders with a systematic analysis of the probable attacker’s profile, the most likely attack vectors, and the assets most desired by an attacker.
- Mine
  - “What if …?”
- Speaking of Adam Shostack – here are four questions he suggests we ask:
  - What are we building?
  - What can go wrong?
  - What are we going to do about that?
  - Did we do a good job?



And Threat Modeling Is Important, Because?

- Not all threats are created equal
- Systems are so complex it’s impossible to understand all the permutations
- Threats are always evolving
- Ways of exploiting them change
- Ties attacks with defenses

S.T.R.I.D.E vs A.I.N.C.A.A.

- S.T.R.I.D.E.
  - Spoofing - Pretending to be someone you aren't
  - Tampering - Changing data or code you're not authorized to change
  - Repudiation – Did you, or did you not, perform an action
  - Information disclosure – Exposing information to someone not authorized
  - Denial of service – Deny/degrade/interrupt service to legitimate users
  - Elevation of privileges – Gain capabilities without proper authorization
- A.I.N.C.A.A.
  - Authentication - Who are you, really?
  - Integrity – Information hasn't been incorrectly or inappropriately modified.
  - Non-repudiation – Verification of what you did.
  - Confidentiality – Sensitive information is only accessed by authorized people
  - Availability – Information/Resources are available to those who need them
  - Authorization – Are you allowed to do what you're trying to do?n!

"Aren't you a little short for a Stormtrooper?" - Princess Leia

Spoofing

Pretending to be someone you aren’t

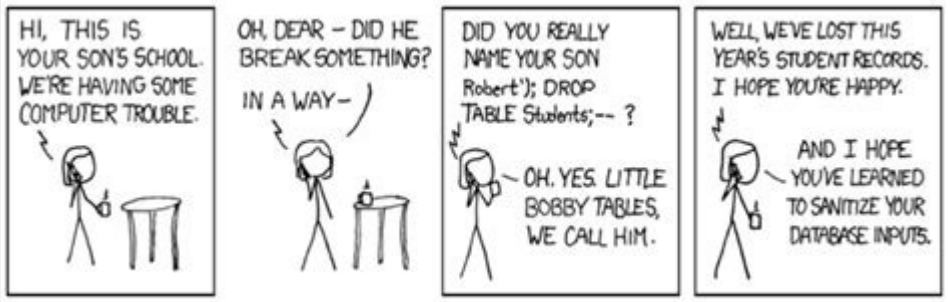
#1, 7, 10

- Examples:
  - #1
    - http://www.company.com/recoveryapp/userID=20482,phone=3033033003
    - http://www.company.com/recoveryapp/userID=20483,phone=3033033003
  - #2
    - http://company.com/app/getappInfo
    - http://company.com/app/admin_getappInfo
  - #3
    - admin/admin
- Questions:
  - Could you pretend to be someone else?
  - root/admin is always bad, even when you have to use root/admin
- Alternatives:
  - Used tested & reputable authentication w/ MFA
  - Time-limited tokens
  - Wrapper every ACID/CRUD in authorized check (specially in APIs)
  - Don’t share accounts

Tampering

Changing data or code you're not authorized to change

#3, 8, 9



- Examples:
  - "Little Bobby Tables"
  - Reasonability filter
- Questions:
  - What can access the data?
  - Can it be changed?
  - How can it be changed?
  - Are changes logged/monitored?
- Alternatives:
  - Data hashing and signing
  - Digital signatures
  - Least privilege (if there's no specific requirement then the answer is 'no access')

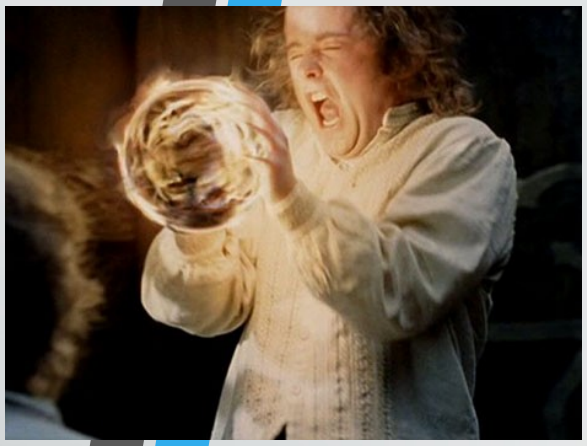
```
POST /echo/post/json HTTP/1.1
Authorization: Bearer mt0dgHmLJMVQhvjpNXDyA83vA_Pxh33Y
Accept: application/json
Content-Type: application/json
Content-Length: 85
Host: reqbin.com

{
   "Id": 12345,
   "Customer": "John Smith",
   "Quantity": -1,
   "Price": 10.00
}
```

Repudiation

Did you, or did you not, perform an action

#3, 7

- Example:
  - “But it wasn’t me who ordered 10,000,000 widgets!?”
  - If you broke your mom’s favorite vase, what’s the first thing you did (other then blame your sibling)?
- Questions:
  - Can I prove it in a court of law? Or to my CEO?
  - Who has CRUD to it and is that access logged and monitored?
- Alternatives:
  - Audit trails (and secure them!)
  - Digital signatures



Information Disclosure

Exposing information to someone not authorized

#1, 2, 3, 7, 10

- Example:
  - GetEmployeeAge() returned a json record with:
    - Organization, Title, GivenName, MiddleName, FamilyName, DisplayName, PrintOnCheckName, Active, PrimaryPhone, PrimaryEmailAddr, Address, EmployeeType, status, Id, SyncToken, CreateTime, LastUpdatedTime, PrimaryAddr, etc…
  - Using poor encryption (i.e. XOR or ROT13)
  - $prod_id = $_GET["prod_id"]; $sql = "SELECT * FROM Products WHERE product_id = " . $prod_id;
- Question:
  - Do I *need* to return/store that data?
  - Is the data stored/sent/used appropriately?
  - Where and how does the data flow through the system?
- Alternatives:
  - Strong encryption (please don’t write your own)
  - Only allow what you need and reject everything else


SQUIRREL!!!
imgflip.com

Privacy

- “Ask not what you can do with the data; ask what the data can do for (to?) you”
- If you don’t collect it you don’t have to care
- Value vs Yes/No
- Privacy by Design (https://en.wikipedia.org/wiki/Privacy_by_design)
- Privacy Design Patterns (https://privacypatterns.org/)
- Dark Patterns (https://www.deceptive.design/types)

Denial of Service

Deny/degrade/interrupt service to legitimate users

#4, 5, 8, 9

- Example:
  - (Unintentional) Self-modifying program that can remove itself
  - Request http://127.0.0.1/delete.php?filename=bob.txt;id

```php
<?php
print("Please specify the name of the file to delete");
print("<p>");
$file=$_GET['filename'];
system("rm $file");
?>
```

- Question:
  - Code will do something unexpected or be used in an unexpected fashion. Or both. Period.
  - For *every* parameter ask the questions:
    - What happens if it's 0/empty/null or too big/small or garbage?
- Alternatives:
  - Validate input
  - Throttling (often, frequently, and repeatedly)
  - Exception handling
  - Did I say validate input?

```
bool keep_looping = true;
for (int I = 0; keep_looping; i++) {
   Page p = getPage (http://company.com/script?p= + i);
   keep_looking = testPage (p);
}
```

Elevation of Privileges

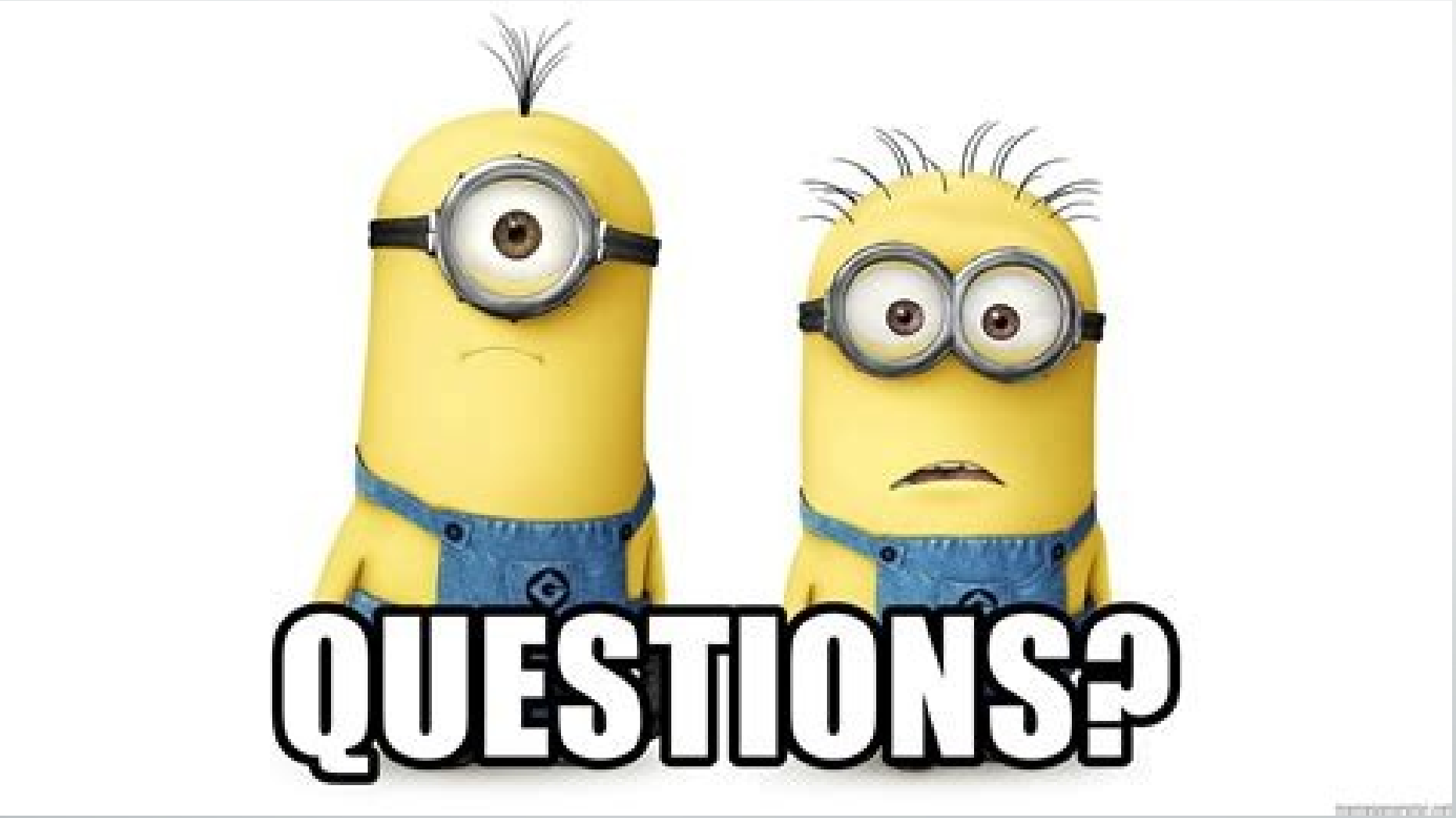
Gain capabilities without proper authorization

#1, 3, 4, 7, 9

- Example:
  - root
  - admin/Password1!
  - Hardcoding secrets
- Question:
  - Is this the minimum set of authorizations needed to perform the function?
  - If root/admin is being used how can it not be used?
  - Should <general user> be allowed to perform admin functions or should there be two accounts?
- Alternatives
  - Strong authentication and authorization
  - Separation of duties
  - Least privilege (yes, even for root/admin)

Threat Modeling isn’t a 

- A01 Broken Access Control
- A02 Cryptographic Failures
- A03 Injection
- A04 Insecure Design
- A05 Security Misconfiguration
- A06 Vulnerable and Outdated Components
- A07 Identification and Authentication Failures
- A08 Software and Data Integrity Failures
- A09 Security Logging and Monitoring Failures
- A10 Server-Side Request Forgery


QUESTIONS?

Supporting Slides

References

- https://safecode.org/wp-content/uploads/2017/05/SAFECode_TM_Whitepaper.pdf
- https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html
- https://owasp.org/www-project-application-security-verification-standard/
- https://explore.skillbuilder.aws/learn/course/external/view/elearning/13274/threat-modeling-the-right-way-for-builders-workshop
- https://aws.amazon.com/blogs/security/how-to-approach-threat-modeling/
- https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats
- https://owasp.org/www-community/Threat_Modeling_Process
- https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/
- https://www.synopsys.com/glossary/what-is-threat-modeling.html
- https://www.exabeam.com/information-security/threat-modeling/
- https://snyk.io/learn/threat-modeling/
- https://www.upguard.com/blog/what-is-threat-modelling

Threat Modeling Frameworks

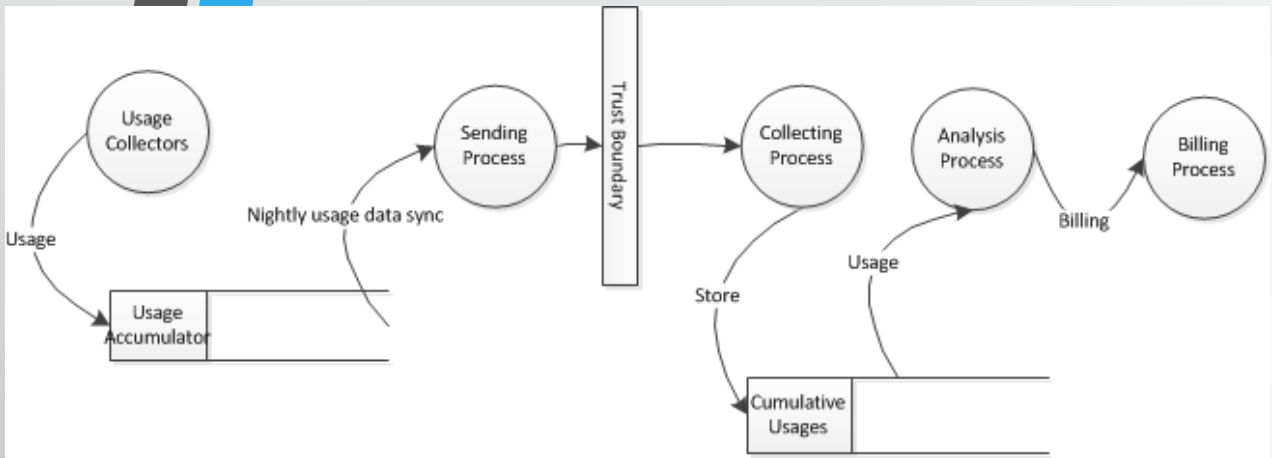
- S.T.R.I.D.E.
  - Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privileges
- D.R.E.A.D.
  - Damage potential, Reproducibility, Exploitability, Affected users, Discoverability
- PASTA
  - Process for Attack Simulation and Threat Analysis
- LINDDUN
  - Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Noncompliance
- Quantitative TTM
  - Quantitative Threat Modeling Method
- VAST
  - Visual, Agile, and Simple Threat Modeling
- OCTAVE
  - Operationally Critical Threat, Asset, and Vulnerability Evaluation

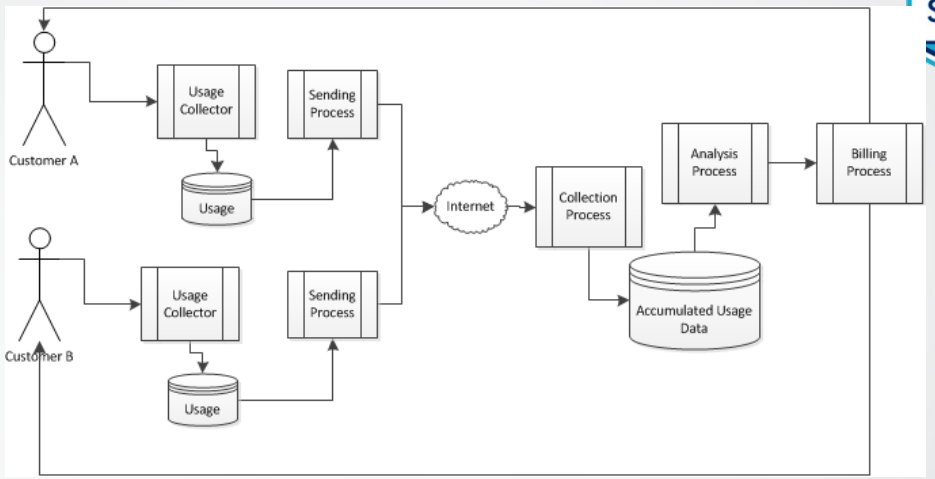
Etc…

Our Brains Ain’t Our Allies

- Our brains put up barriers to protect us from change!
  - It won't happen to me (a.k.a. I wouldn't let it happen that way)
  - The more you know the less you think you know
    - Converse: the less you know the more you think you know
  - Better at evaluating immediate risks (rock flying at us) then evaluating delayed risks (health problems later in life)
  - Trust
  - Small change blindness
  - Familiarity blindness
  - Heuristics

Diagramming





- Client
  - Usage data could be modified
  - Information disclosure
  - Usage collectors could be (accidentally) removed
- Transit
  - Disclosure
  - Modification
- Server
  - Data could be read or modified
  - Collector could be subject to DoS
  - Information disclosure

- Client
  - Encryption / signing / hashing
  - Encryption
  - Business process – how to handle missing or invalid usage
- Transit
  - Encryption / secure communication
  - Encryption / signing / hashing
- Server
  - Encryption / Least privilege
  - Throttling
  - Encryption / Least privilege

Threat Matrix

Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data Flows		X		X	X	
Data Stores		X		X	X	
Process	X	X	X	X	X	X
Interactions	X		X			

It’s 3 am. Do You Know Where Your Assets Are?

- You can’t protect against what you don’t know about
  - Libraries, third-party components
  - Unused code
  - Data (including caches)
  - Open ports
  - Installed software (specially unused software)
  - User accounts
  - APIs (especially old versions)
  - VM running on a dev’s computer that was used for a PoC but never shutdown