Threat modelling toolkit

How to run your workshop

/thoughtworks

How to run your workshop

Gather the team around a whiteboard. Invite product leaders and security stakeholders to get a rounded perspective

What are we defending?

- Draw a picture of what we are building
- Show relevant components, dataflows, users and collaborators
- Highlight the data or services we are protecting

What can go wrong?

- Show sources of threat - attackers and insiders
- Brainstorm many threats using these cards as cues
- Capture on stickies, e.g. “Spoofed Identity: Weak credentials allowed”

What are we going to do about it?

- Thinking about risk, dot vote top three threats
- Add actions to backlog that reduce the risk
- Take a photo to add to document or Wiki

How do we know that we did a good job?

- Perform a review of actions after 30 days
- Are the actions complete? If not why not?
- Time to threat model again!

/thoughtworks

Spoofed identity

How hard is it for an attacker to pretend to be someone with authority to use the system? Can someone spoof an identity and then abuse its authority? Spoofing identity allows attackers to do things they are not supposed to do.

An example of identity spoofing is an attacker illegally accessing and then using another user's authentication information, such as username and password.

Key concepts: Identity, Authentication

/thoughtworks

Examples of Weak Authentication

- Lack of any form of authentication
- Dependency on browser-based implementation
- Failure of user interface to obscure entry of credentials
- Failure to prevent users creating weak credentials
- Failure to prevent caching of credentials on a shared computer
- Authentication mechanism subject to brute force attack
- Can authenticate with credentials harvested from other breaches
- Failure to allow use of password manager and strong credentials
- Reliance on a single factor for authentication
- Weakness in process to reset credentials
- Guessable values such as IMEI number used in authentication

Examples of Weak Processes

- Lack of identity or entitlement checks in setting up a new account
- Use of shared accounts and credentials
- Weakness in offline process to reset credentials
- Failure to revoke access when someone leaves

Other Examples

- Authentication, authorisation or session management has been coded from scratch
- Failure of session to timeout after a reasonable duration of time
- Failure to configure authorisation, i.e. based roles and least privilege
- Failure to re authenticate session when taking destructive actions, such as delete account

And what else?

Tampering with input

How hard is it for an attacker to modify the data they submit to your system? Can they break a trust boundary and modify the code which runs as part of your system? Tampering with input can allow attackers to do things they are not supposed to do.

An example of tampering with input is when an attacker submits a SQL injection attack via a web application and uses that action to delete all the data in a database table.

Key concepts: Integrity, Injection, Validation, Whitelisting, Blacklisting

/thoughtworks

Lack of validation server-side

- Fails to prevent stored Cross Site Scripting (XSS)
- Fails to prevent reflected XSS
- Fails to prevent SQL, XML (XXE) or LDAP injection
- Fails to prevent shell injection
- Fails to prevent an open redirect
- Fails to prevent Cross-site request forgery (CSRF)
- Framework support for mass binding can be exploited
- Alternate character encodings can be used to circumvent protections
- It is possible for attacker to tamper with cookies
- File upload feature fails to block malware

Lack of validation in browser

- Fails to prevent DOM based XSS
- Relies on browser based business logic for validation
- Scripts to display advertising contain malicious code
- Code injection is possible via JSON responses received from server
- Transfers between DOM contexts are subject to code injection
- It is possible for attacker to tamper with cookies

And what else?

Repudiation of action

How hard is it for users to deny performing an action?
What evidence does the system collect to help you to prove otherwise?
Non-repudiation refers to the ability of a system to ensure people are accountable for their actions.

An example of repudiation of action is where a user has deleted some sensitive information and the system lacks the ability to trace the malicious operations.

Key concepts: Non-repudiation, Logging, Audit, Signing

/thoughtworks

Examples of insufficient logging

- Lack of logging showing user access to sensitive data
- Lack of logging sensitive user actions, such as delete account
- Lack of logging of administrative activities
- Lack of logging of session management or authentication failures
- Lack of logging of validation errors
- Lack of centralisation of logs to a central store
- Lack of audit log showing user actions within delivery infrastructure

Examples of logs being vulnerable to tampering

- Lack access control on log files allowing attackers to cover tracks
- Lack of integrity signatures on logs for sensitive actions
- Lack of integrity signatures on artefacts passing through delivery pipeline

Examples of weakness in process

- Lack awareness of audit and access policy
- Lack of awareness material to communicate audit and abuse policy to user
- Lack of operational process or team to respond to suspicious events

And what else?

Information disclosure

Can someone view information they are not supposed to have access to? Information disclosure threats involve the exposure or interception of information to unauthorised individuals.

An example of information disclosure is when a user can read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers.

Key concepts: Confidentiality, Encryption, leakage, Man-in-the-middle

/thoughtworks

Poor handling of secrets

- Lack of tooling to prevent pushing configuration secrets to source control
- Secrets are stored in plain text in source control
- Possible for malicious process to read plaintext credentials

Encryption of data in transit

- Cleartext transport of credentials or data over WiFi and/or Internet
- Cleartext transport of credentials or data between components within the system
- TLS Cypher configuration is weak
- Configuration of TLS is vulnerable to a 'downgrade' attack
- Lack of measures to prevent domain spoofing, such as Strict Transport Security

Information leakage

- Sensitive information is present in log files
- Leakage of unnecessary system information which can assist an attacker
- Triggering an exception leaks unnecessary information that can assist attacker
- Lack of access control on resources not intended to be discoverable to user
- Possible for another tenant to read deallocated cloud storage

Other examples

- Sensitive data stored in unencrypted storage
- Possible for malicious process to read sensitive information from logs
- Sensitive data is stored in predictable locations in memory
- Lack of anti-caching headers to prevent caching of sensitive HTTP requests or responses
- Lack of rate limiting allows 'scraping' or 'spidering' of valuable data

And what else?

Denial of service

Can someone break a system so valid users are unable to use it? Denial of service attacks work by flooding, wiping or otherwise breaking a particular service or system.

An example of denial of service is where a Web server has been made temporarily unavailable or unusable with a flood of traffic generated by a botnet.

Key concepts: Availability, Botnets, DDoS, Content delivery network

Flooding of network traffic

- Failure to apply network isolation to a service which does not need to be on the internet
- Exposure of unnecessary services to the Internet
- Fails to filter network flooding attacks at OSI network layers 2 or 3
- Failure to use a CDN (for example Fastly, Cloudflare or AWS CloudFront)
- System was not designed to meet current traffic demands

Scripted application attacks

- Lack of rate limiting in Internet facing user interfaces

Operational concerns

- Lack of logging to determine source of flooding
- Lack response plans to block traffic from a particular source
- Lack of response plan to report issue to upstream infrastructure suppliers

And what else?

Elevation of privilege

Can an unprivileged user gain more access to the system than they should have? Elevation of privilege attacks are possible because authorisation boundaries are missing or inadequate.

An example of elevation of privilege is where a user can manipulate the URL string to gain access to sensitive records they should not be able to see.

Key concepts: Authorisation, Isolation, Blast Radius, Remote Code Execution

Known vulnerabilities

- A known vulnerability in infrastructure component is exploited due to failure to apply patches in production
- A known vulnerability in application component is exploited due failure to apply patching in production

Lack of isolation in architecture

- Service which do not need to be exposed to Internet are
- Possible to escalate privilege from another system
- Possible to mount attack on other system components via network

Lack of hardening of infrastructure

- Developer mode tools or default admin credentials are enabled
- Unnecessary services exposed by underlying infrastructure
- Able to escalate privilege via cloud vendor side channel attack

Absence of authorisation in web UI

- Failure to check authorisation to more sensitive resources
- Fails to prevent clickjacking
- Lack of Client Security Policy (CSP) configuration allows loading of untrusted resources

Implementation weakness

- A security enforcing function, such as authentication, authorisation or session management has been coded from scratch

And what else?

/thoughtworks