Clean Insights

Privacy-Preserving Measurement

THE GUARDIAN PROJECT

https://guardianproject.info

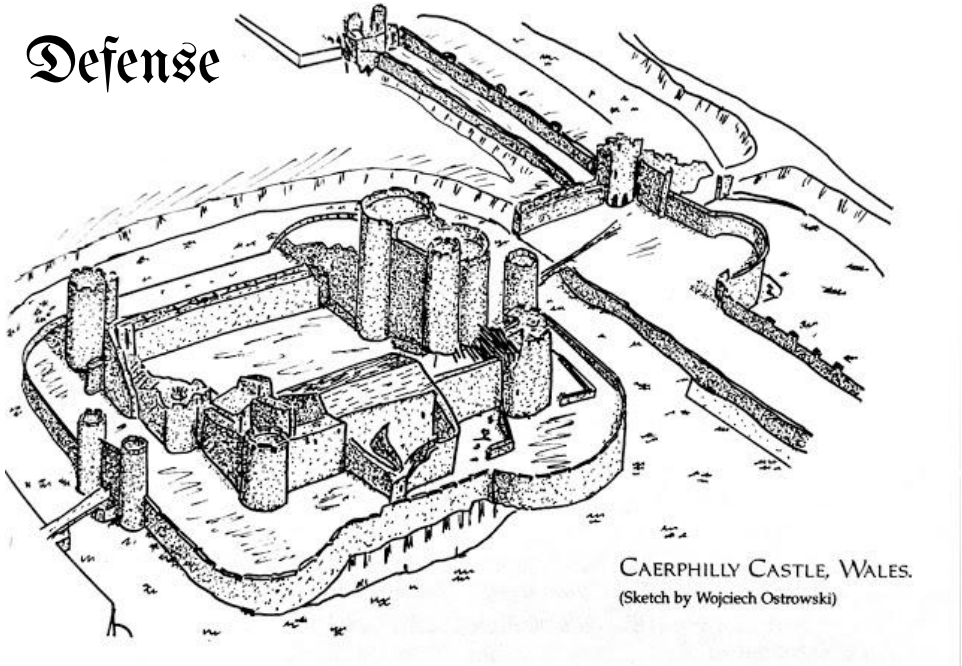
Threat Modeling
in a Medieval Mode

WARNING: This documents makes use of an intentionally humorous medieval warfare metaphor in order to make the topic of “mobile security threat modeling” more accessible and engaging!

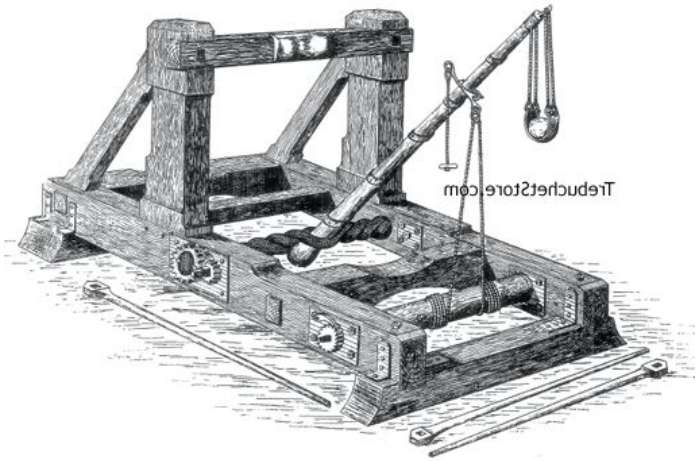
Do as thou wilt!

The Castle and the Trebuchet

Defense



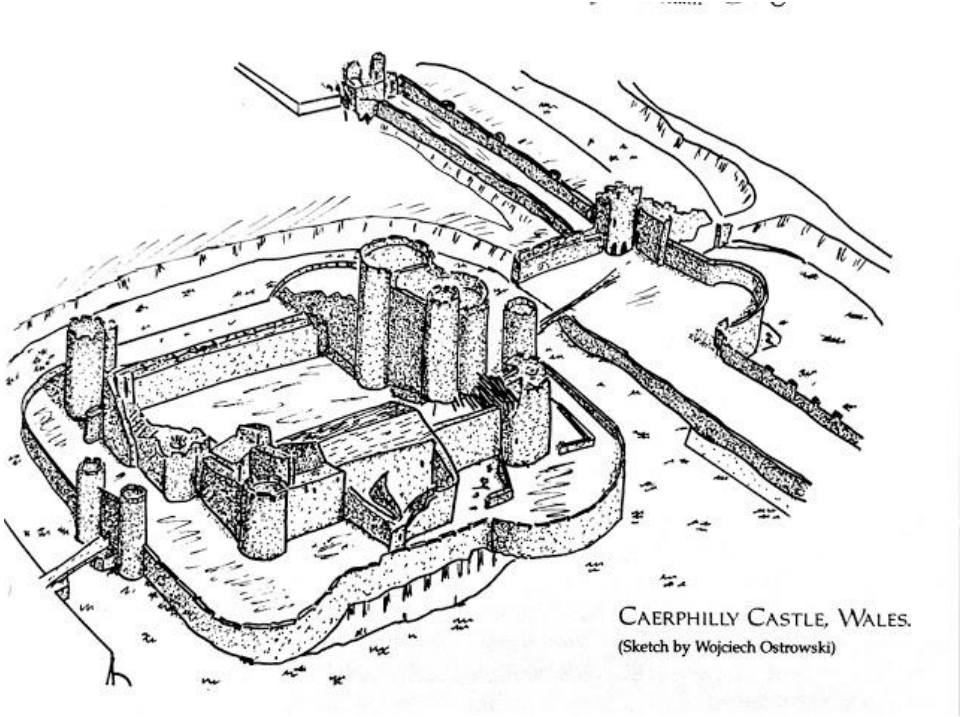
CAERPHILLY CASTLE, WALES.
(Sketch by Wojciech Ostrowski)



Offense

Fortifying the Castle

Defensive Threat Modeling



CAERPHILLY CASTLE, WALES.
(Sketch by Wojciech Ostrowski)

Built from Wood and Stone…

“Materials that were used in the building of castles varied through history. Wood was used for most castles until 1066. They were cheap and were quick to construct. The reason wood fell into disuse as a material is that it is quite flammable. Soon stone became more popular.

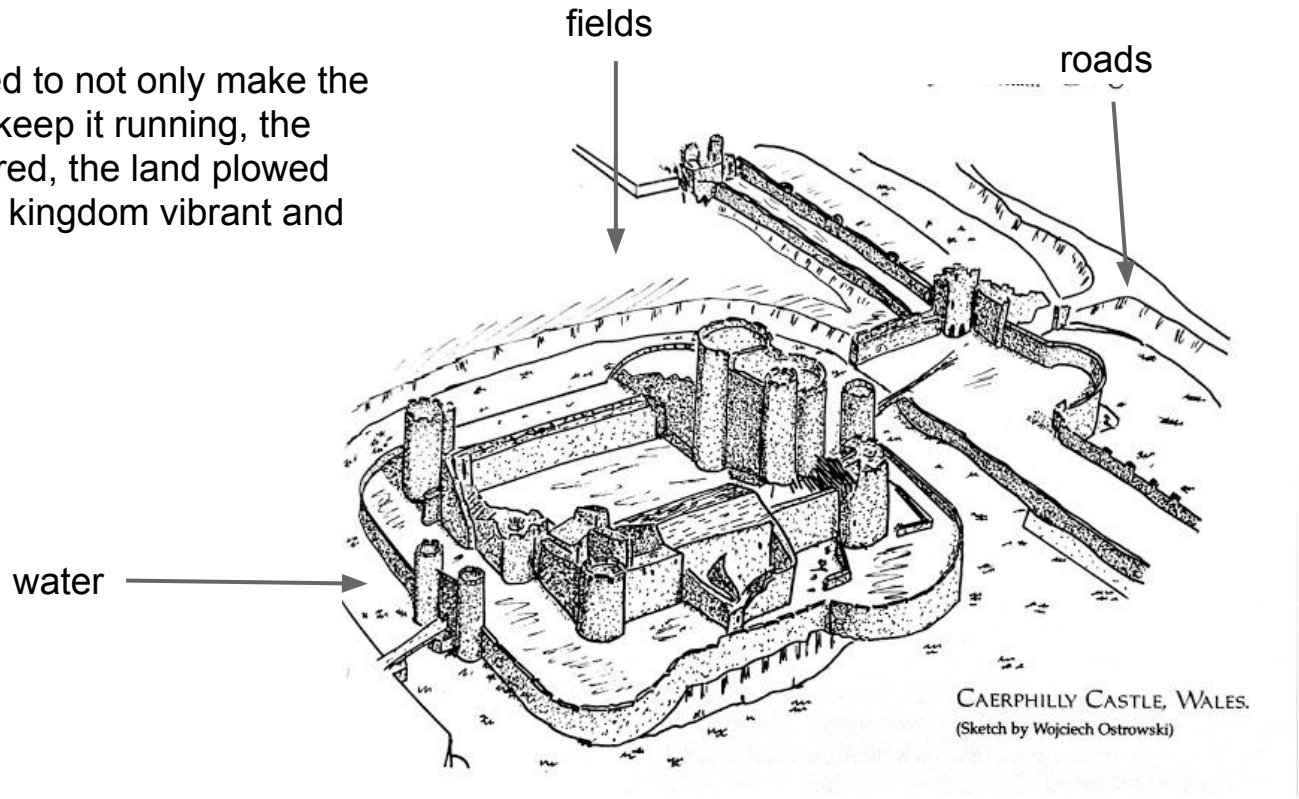
Stone castles took years to construct depending on the overall size of the castle. Stone was stronger and of course much more expensive than wood. Most stone had to be quarried miles away, and then brought to the building site. But with the invention of the cannon and gunpowder, castles soon lost their power.“

http://en.wikipedia.org/wiki/Medieval_fortification

External Dependencies

air!

All the things we need to not only make the castle strong, but to keep it running, the people fed and watered, the land plowed and planted, and the kingdom vibrant and bountiful!

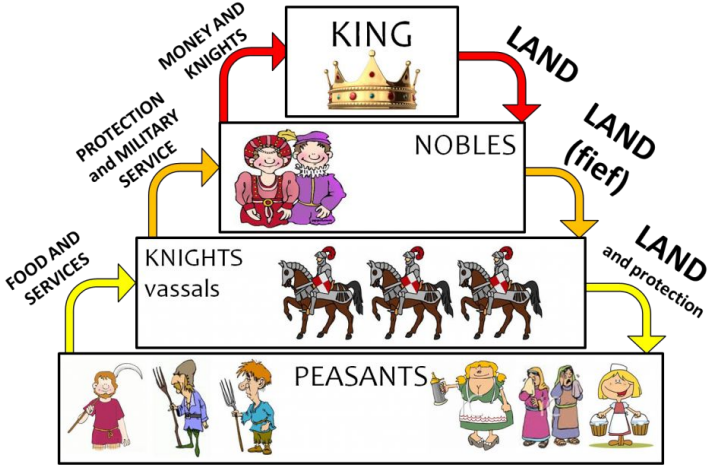


External Dependencies

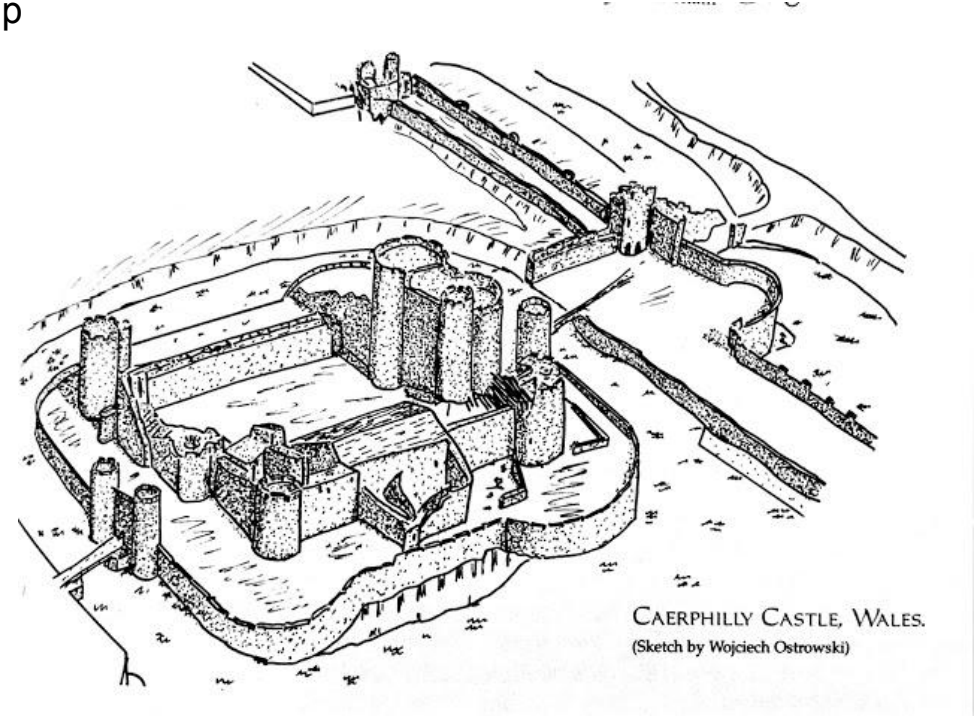
ID	Name	Description
1	**Mobile OS and Device**	Mobile app runs on the Android operating system, which has many dependencies of its own, and whose specific security settings (screen lock, disk encryption, random apps) are up to the user.
2	**Mobile Network and Internet Links**	Mobile app communicates over mobile networks, and the national backbones that connect those, and the Internet in order to download lessons data and upload & publish stories.
3	**Proxy and Circumvention Networks**	When possible, the communication will utilize a secure proxy network system such as Tor or Psiphon. The availability and accessibility of these networks are largely out of control of the Mobile team.
4	**Backend Servers and Platforms**	Backend systems hosted by Mobile and RZ teams utilize the Wordpress.org platform and Linux servers for publishing stories.
5	**App Distribution Points**	Mobile app is primarily distributed through the Google Play application distribution system.
6	**Lesson Bundles and Other Content**	Mobile relies upon bundles of lesson content that include text, photos, audio and video. These are stored on a remote server, and downloaded, unpacked and rendered by the user on-demand.

Ye Olde Trust Levels

Trust Levels are the duties, ranks, privileges assigned to all those with any relationship to, or within any distance of the castle.



Feudal Pyramid of Power

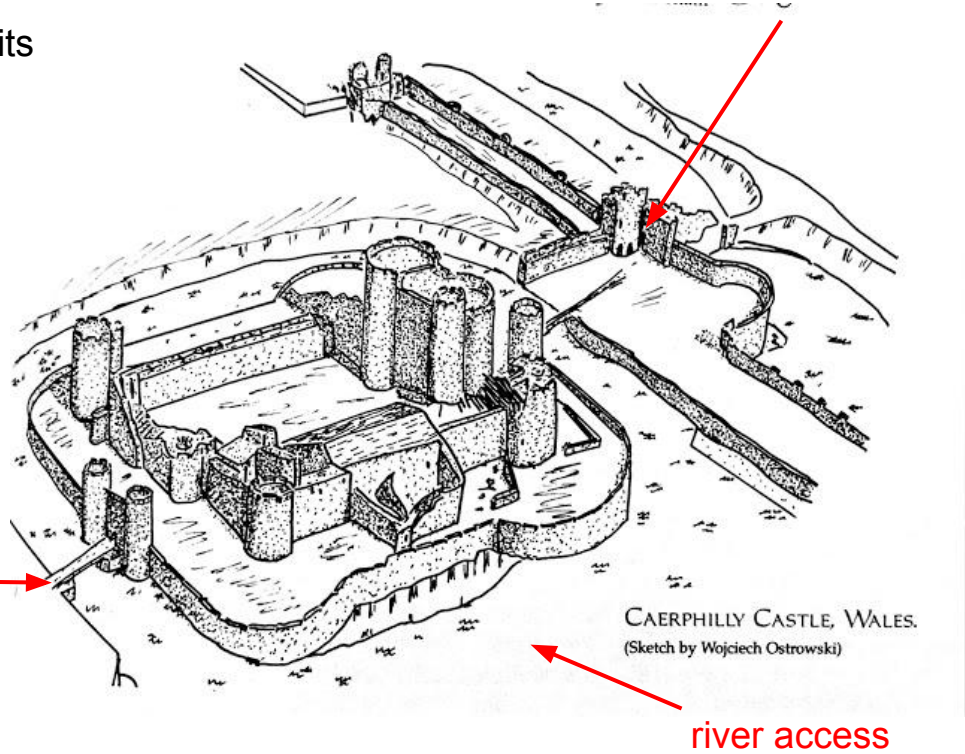


Trust Levels

ID	Name	Description
1	Anonymous App User	A user who has downloaded the app but who has not registered with any site
2	App Users with Login	A user with the app who has registered, and has a valid account
3	App User with Invalid Login	A user with the app who is trying to access an account using invalid credentials
4	Content Manager	The person managing content being published to the site and interacting with users
5	Site Administrator	The administrator of the site, managing status, updates, security, etc
6	App Developer	A software developer who has the ability to commit code to the open-source project
7	App Release Manager	A technical staff member who compiles, signed and distributes the binary mobile application.

Entry Points Ahoy!

Entry points are the ways into the castle, official or not, known or secret, or any of its surrounding areas.

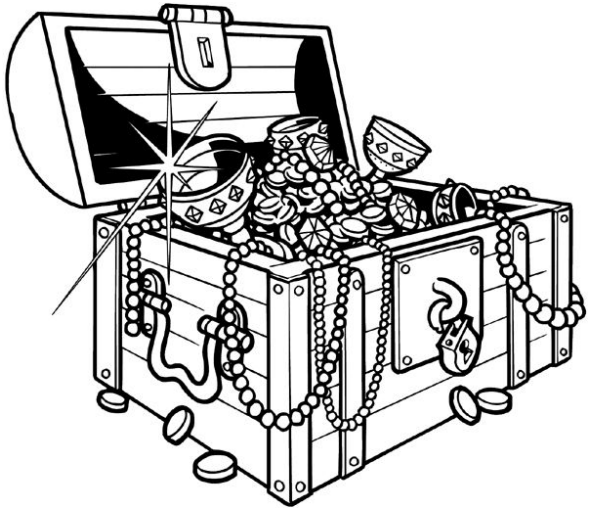


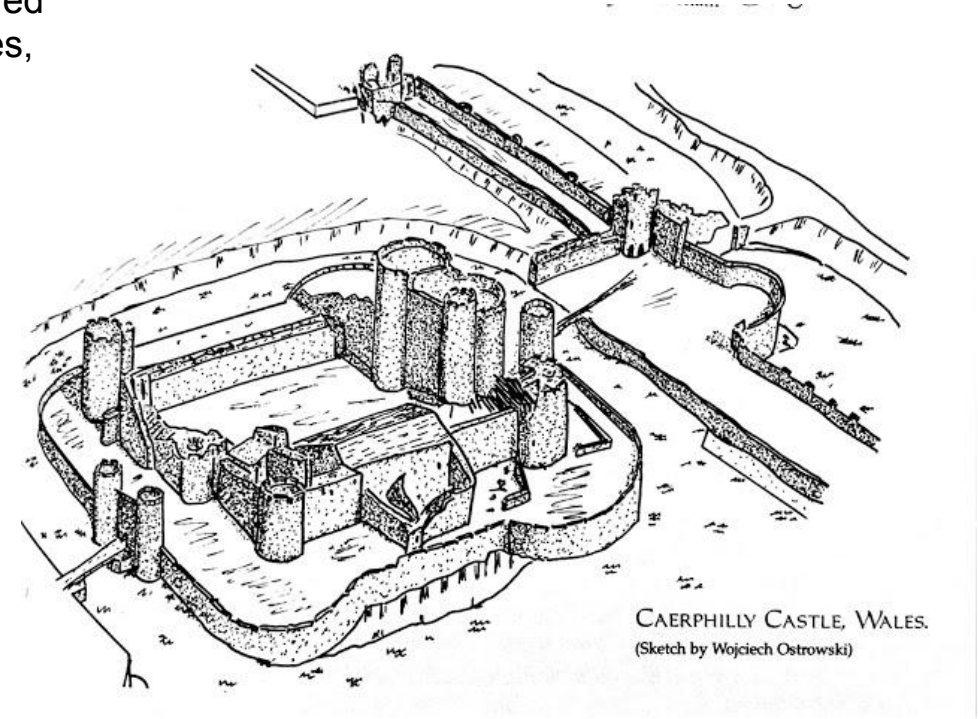
Entry Points

ID	Name	Description
1	HTTPS Port	Servers that the app connects to will be available via HTTPS
1.1	Lesson Bundles	Lesson content is stored as ZIP files that contain text and media components.
1.2	Account Setup	The app allows for users to configure accounts on various remote services and sites.
1.3	Mobile Media	The app can import media files from the phones shared gallery and storage.
1.4	App Source Code	Application is powered by open-source code hosted on Github.com
1.5	Wordpress	Server hosts a Wordpress server that has open registration
1.6	Media Upload Service	System may offer a separate method to upload media stories

Assets (Gold and Doubloons)

Assets are the valuable treasure contained in the castle - gold, jewels, scrolls, horses, people, magical swords, and so forth...





Assets

ID	Name	Description
1	**Users and Content**	**Assets relating to app users and the content**
1.1	User Login Details	The login credentials that an app user has.
1.2	Content Manager Login Details``	The login credentials that a content manager has.
1.3	Admin Login Details	The login credentials that a admin has.
1.3	Published Stories (Text and Media)	Content at rest on the server once a story is published
1.4	Lesson (Curriculum) Bundles	Archives stored in public place on server for download by app
2	**System**	**Assets relating to the underlying system.**
2.1	Availability of Website	The website should be available 24 hours a day and can be accessed by all users.
2.2	Ability to Execute Code/SQL as a Web Server User	This is the ability to execute source code on the web server as a web server user.
2.3	Backups of site content	Regular backups of content to be able to restore all published stories and media after attack, or to replicate/mirror in case of DDoS
2.4		
3	**Website**	**Assets relating to the site content**
3.1	Login Session	This is the login session of a user to th site. This user could be an app user of the content manager.
3.3	Ability to Create Users	The ability to create users would allow an individual to create new users on the system. These could be ...
3.4	Access to Audit Data	The audit data shows all audit-able events that occurred within the application.

Application Security Framework

(8 aspects of defensive threat modeling)

- Authentication
- Authorization
- Configuration Management
- Data Protection
- Data Validation
- Error Handling
- User / Session Management
- Auditing and Logging

Authentication

General Countermeasure

- Credentials and authentication tokens are protected with encryption in storage and transit
- Protocols are resistant to brute force, dictionary, and replay attacks
- Strong password policies are enforced
- Trusted server authentication is used instead of SQL authentication
- Passwords are stored with salted hashes
- Password resets do not reveal password hints and valid usernames
- Account lockouts do not result in a denial of service attack

Mobile Countermeasure

- All remote authentication tokens and local credentials must be stored in a secure manner using SQLCipher and/or CacheWord
- Review and implement strong but usable password policy in the Mobile lockscreen
- Wordpress.org server instance must be audited to ensure password management is strong, and procedure for detecting attack via password reset or auth mechanism is handled
- Mobile app should implement lockout/failure/reset mechanism (extend CacheWord to support this possibly)

Authorization

General Countermeasure

- Strong ACLs are used for enforcing authorized access to resources
- Role-based access controls are used to restrict access to specific operations
- The system follows the principle of least privilege for user and service accounts
- Privilege separation is correctly configured within the presentation, business and data access layers

Mobile Countermeasure

- Ensure account creation from Mobile app grants the proper least privilege user status to new users
- Review administrative security controls for servers (password, key storage, logs, detection, firewalls, etc)

Configuration Management

General Countermeasure

- Least privileged processes are used and service accounts with no administration capability
- Auditing and logging of all administration activities is enabled
- Access to configuration files and administrator interfaces is restricted to administrators

Mobile Countermeasure

- Privileges for updating Mobile content bundles and binary apps must be restricted
- Monitoring systems for unauthorized changes on sites, distributions, code must be in place

Data Protection: Storage & Transit

General Countermeasure

- Standard encryption algorithms and correct key sizes are being used
- Hashed message authentication codes (HMACs) are used to protect data integrity
- Secrets (e.g. keys, confidential data) are cryptographically protected both in transport and in storage
- Built-in secure storage is used for protecting keys
- No credentials and sensitive data are sent in clear text over the wire

Mobile Countermeasure

- SQLCipher and IOCipher implemented properly for all sensitive data
- CacheWord utilized for protect keys
- NetCipher and/or proper SSL implementation for over the wire security
- All possible SSL/TLS certificates should be pinned to defend against MiTM

Data Validation

General Countermeasure

- Data type, format, length, and range checks are enforced
- All data sent from the client is validated
- No security decision is based upon parameters (e.g. URL parameters) that can be manipulated
- Input filtering via white list validation is used
- Output encoding is used

Mobile Countermeasure

- Server-based endpoints for auth, upload, publishing must be validated properly

Error Handling

General Countermeasure

- All exceptions are handled in a structured manner
- Privileges are restored to the appropriate level in case of errors and exceptions
- Error messages are scrubbed so that no sensitive information is revealed to the attacker

Mobile Countermeasure

- Android app logging should be properly scrubbed of sensitive data
- Error handling must be properly configured and implemented on the server

User & Session Management

General Countermeasure

- No sensitive information is stored in clear text in the cookie
- The contents of the authentication cookies is encrypted
- Cookies are configured to expire
- Sessions are resistant to replay attacks
- Secure communication channels are used to protect authentication cookies
- User is forced to re-authenticate when performing critical functions
- Sessions are expired at logout

Mobile Countermeasure

- Cookies should not be used in app-to-server interaction
- Server-side management of sessions based on API calls from mobile client must be reviewed and hardened
- Review and implement policy for re-authentication and session expiry

Auditing and Logging

General Countermeasure

- Sensitive information (e.g. passwords, PII) is not logged
- Access controls (e.g. ACLs) are enforced on log files to prevent un-authorized access
- Integrity controls (e.g. signatures) are enforced on log files to provide non-repudiation
- Log files provide for audit trail for sensitive operations and logging of key events
- Auditing and logging is enabled across the tiers on multiple servers

Mobile Countermeasure

- Audit and review server log management
- Monitor all known app distribution points and user community for malware
-

Deploy the Trebuchet

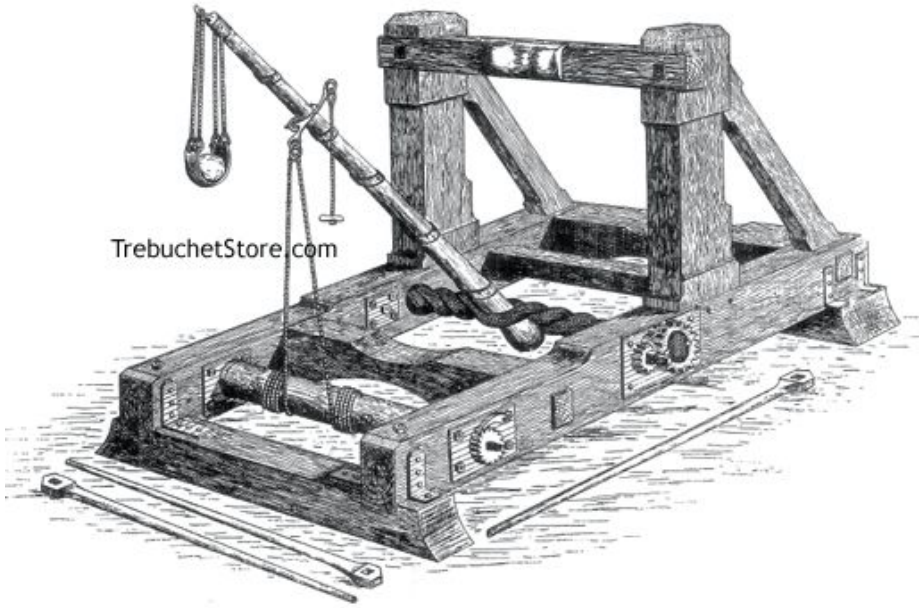


Fig. 1.—Sketch plan of a Catapult for slinging Stones, its Arm being partly wound down.
Approximate scale : ½ in. = 1 ft.

Offensive Threat Modeling

Disease Defeats the Strongest Walls…

“Biological warfare isn't only a product of the 20th and 21st centuries. It dates right back to medieval days, when huge catapults hurled dead horses and other animals into castles under siege, to spread disease. Facing starvation, the defenders ate the putrid flesh, and promptly succumbed to the dreaded plague. “

http://www.senioryears.com/catapult.html

STRIDE Threat List

Type	Security Control	Mobile examples
Spoofing	Authentication	Threat action uses Mobile app to attempt to login to someone elses account using their public username and guessing their password.
Tampering	Integrity	Interception or modification of lesson bundle data aimed to disable or infect app or device
Repudiation	Non-Repudiation	Threat action attempts to call administrative server endpoints using the user generated credentials
Information disclosure	Confidentiality	Threat from man-in-the-middle attack ("SSLStrip") against network data, providing access to users account. Also threat from local malicious app installed on device that can access data stored, captured in the clear.
Denial of service	Availability	Server media publishing endpoints can be attacked, filtered or otherwise made inaccessible to mobile app users.
Elevation of privilege	Authorization	Threat action can be made against mobile app by sending false data from network, allowing execution of code on mobile device. Alternately, mobile app can attempt to elevate access on server through media publishing API endpoints.

STRIDE Mitigation

Threat Type	Mitigation Techniques	Mobile Countermeasure
Spoofing Identity	1.Appropriate authentication 2. Protect secret data 3. Don't store secrets	• Ensure server auth tokens are encrypted on the device • Only store necessary secrets (i.e. don't store passwords, just store tokens)
Tampering with data	1. Appropriate authorization 2. Hashes 3. MACs 4. Digital signatures 5.Tamper resistant protocols	• Implement file, data and network security appropriately, using NetCipher SDK when possible • Alert user if data tampering or corruption is detected
Repudiation	Digital signatures Timestamps Audit trails	• Consider use of public key cryptography to sign published content to verify authorship/control beyond just user/password • Implement server-side auditing of user access, uploads to detect abuse, attack, infiltration
Information Disclosure	Authorization Privacy-enhanced protocols Encryption Protect / Don't store secrets	• Use NetCipher SDK for device side protection • Consider what data is stored in the clear, available via Gallery / outside of encryption • Support use of ObscuraCam app for face blurring in video before editing in Mobile
Denial of Service	Appropriate authentication Appropriate authorization Filtering /Throttling Quality of service	• Servers should implent DDoS mitigation using Deflect, Cloudflare, or some other system • Media upload should be filtered to only support media content types and not executable files • Ensure auth mechanisms can't be overwhelmed by repitive attacks
Elevation of privilege	Run with least privilege	• Servers should run without root/admin privs as much as possible • Mobile app should not require root/superuser on device

Measurement Magic Gone Wrong!

Magical Runes of #FAIL

TRUST US

Photo editing app Meitu says it needs permissions for analytics, denies selling user data

BY HARISH JONNALAGADDA Friday, Jan 20, 2017 at 7:25 am EST

4 Comments

Meitu details why it needs all those permissions.

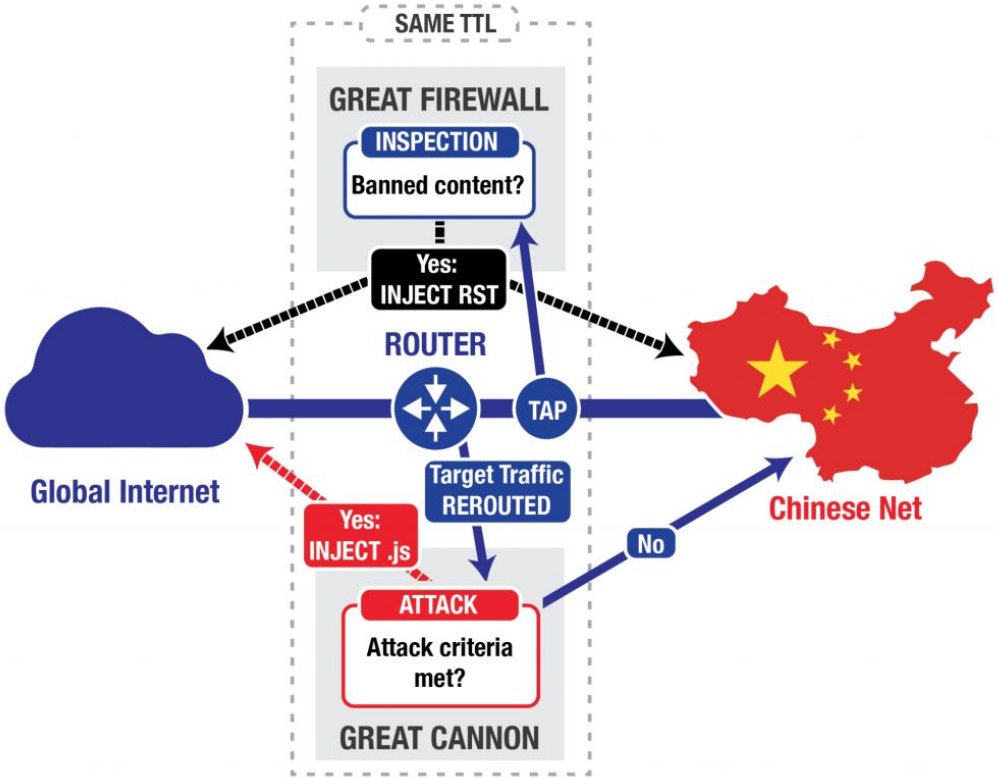
Chinese photo editing app Meitu made landfall in the U.S. recently, with the free app shooting up the Play Store rankings over the course of the week. The app adds anime-style filters to photos, an the final results end up being equal parts wonderful and weird.

Meitu also went into detail over the permissions it requires:

- **MAC address/IMEI number:** In some cases, Meitu cannot get both info at the same time and in some cases different devices even have the same IMEI number, so we combine these two details into one unique ID to track user devices.
- **LAN IP address** is used to prevent business fraud.
- **SIM card country code** is used for a rough location detection.
- **GPS and network location** are used for detecting countries and regions for Geo-based operation and advertisement placement.
- **Phone carrier info** is used as a standard tracking channel for analytics, just like the other third-party analytics tools(e.g., Flurry).
- **RUN_AT_START:** because the Google service (including GCM) is not available in mainland China, Meitu uses a third-party push notification service called Getui (www.getui.com).

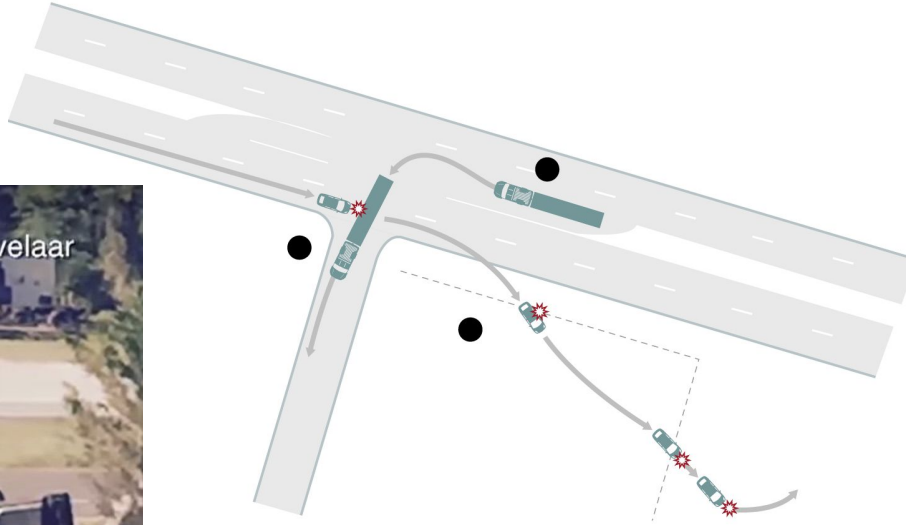
That's certainly a lot to put up with for a photo filter app. If you're satisfied with Meitu's explanation, the app is available for free from the Play Store.

All your permissions and data belong to us!



In the attack on GitHub and GreatFire.org, the GC intercepted traffic sent to Baidu infrastructure servers that host commonly used analytics, social, or advertising scripts.

Weaponized users though insecure analytics



REPORT: TESLA'S FATAL CRASH CAN'T BE BLAMED ON SOFTWARE ERRORS

“Blackbox” exonerates corporation

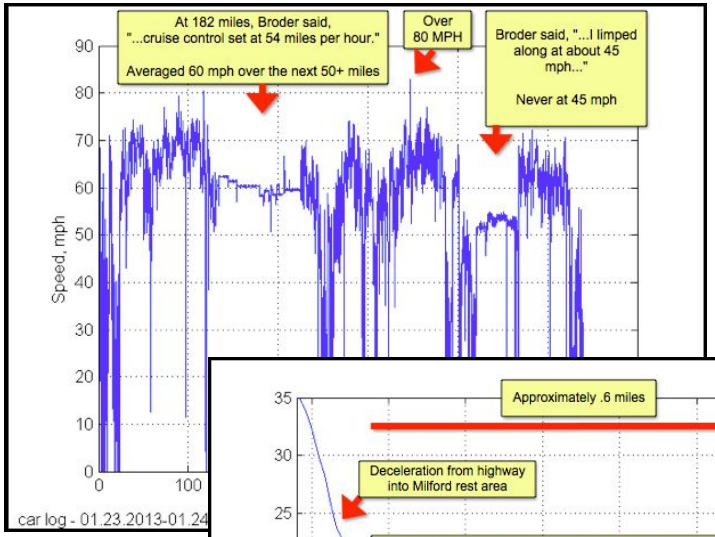
Tesla publishes Model S driving logs that show The New York Times' blatant lies

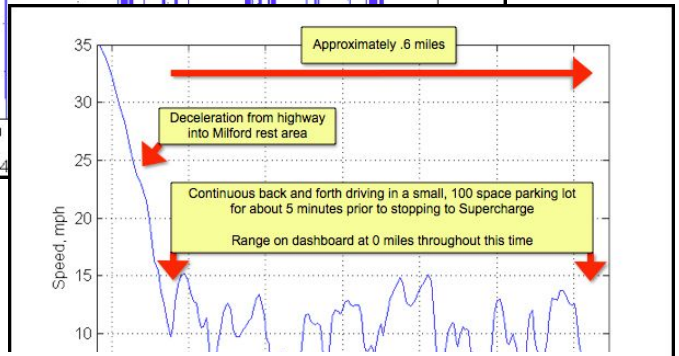
By Sebastian Anthony on February 14, 2013 at 8:17 am | 143 Comments

0 shares

Following Elon Musk's initial denouncement of The New York Times for publishing a fake review of the Tesla Model S electric car, he has now published the actual logs recorded by the car — and boy are they damning. In short, the NYT's John Broder lied through his teeth to smear electric vehicles in general, and the Model S in specific.

The basic premise of John Broder's story for The Times was that the car lied about its self-reported estimated remaining range; when it said there was 79 miles left in the batteries, there was in actual fact only 60. Eventually, after a few such cases of the on





A Most Peculiar Test Drive

Elon Musk, Chairman, Product Architect & CEO February 13, 2013

"Blackbox" incriminates the user

NETWORKWORLD
FROM IDG

PRIVACY AND SECURITY FANATIC

By Ms. Smith | Follow

About
Ms. Smith (not her real name) is a freelance writer and programmer with a special and somewhat personal interest in IT privacy and security issues.

Cops use pacemaker data to charge homeowner with arson, insurance fraud

Police called pacemaker data an 'excellent investigative tool' that provided 'key pieces of evidence' to charge a man with arson and insurance fraud

Network World | JAN 30, 2017 7:08 AM PT

RELATED


Cool Yu
disrupti

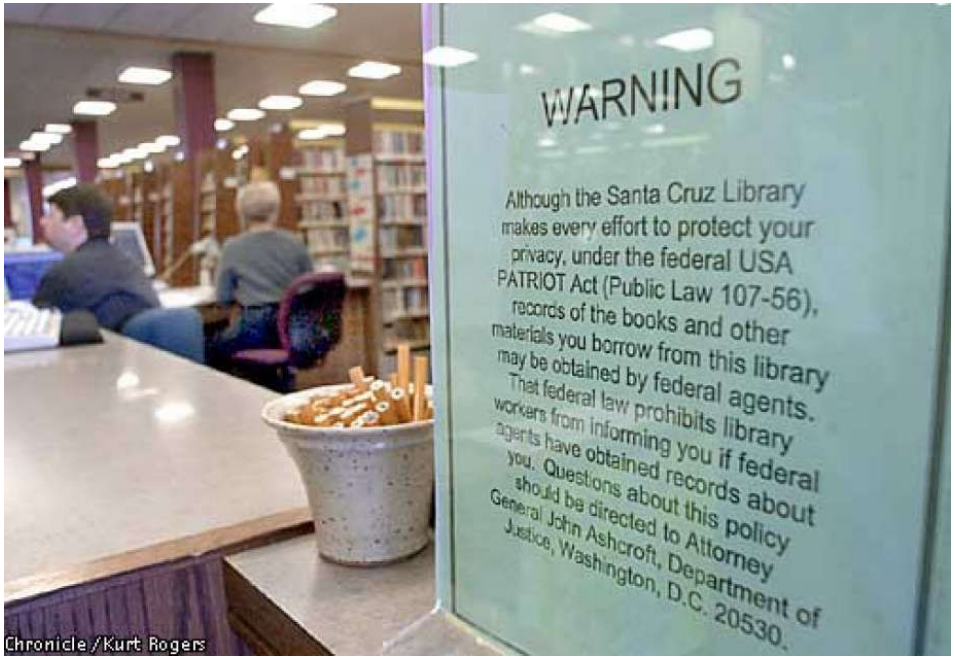

3 replac
phones
week


Pastor:
wife's n
sent pic


VIDEO

Middletown Police said this was the first time it had used data from a heart device to make an arrest, but the pacemaker data proved to be an “excellent investigative tool;” the data from the pacemaker didn’t correspond with Compton’s version of what happened. The retrieved data helped to indict Compton.

Your body will be used against you



Be careful what you read

Have fun storming the castle!