CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

Objective

Use the simple CIA (confidentiality, integrity, availability) methodology to model threats to your online identity. Include diagrams and creative attack vectors. Think from the perspective of someone else who wants to take over/remove/destroy your online identity.

I've broken the model into the key components of value that make up my online identity, ranging from personal websites and eCommerce websites to social media accounts, emails, and accounts linked to payment methods. For many online services, enabling MFA is an obvious preventative measure.

Risk	**Attacker Behavior**	**Response/Preventative Measure**
Target: Personal Website(s); sketchyactivity, austinjhunt		
Availability	Attacker runs application-layer DDOS attack against website	Prevention: Place website behind Cloudflare and leverage built in DDoS protection service. Set up server-side alerting about resource usage spikes/request spikes. Reaction: Collect attacking IP addresses / CIDRs and block them explicitly. Use WAF (Web Application Firewall) to block traffic from likely attackers.
Confidentiality	Attacker triggers an exception on web app and scrapes sensitive values from exception trace on DEBUG page enabled as workaround for static file serving.	Prevention: Do not deploy Django web applications with DEBUG=True. If using an environment variable to set DEBUG, use default value of False in case environment variable is not present. Note: Django apps can often fail to function properly as a result of setting DEBUG=False, which is why many people leave it True even in prod to avoid having to deal with the static files issue. Reaction: If you notice a web app has been deployed with DEBUG=True, replace all sensitive values with new values. Redeploy with DEBUG=False.
Confidentiality	Attacker spoofs DNS with bettercap and spoofs website certificate with CarbonCopy, a tool that creates a spoofed certificate of any online website and signs an Executable for AV Evasion. Reference video. Alternatively attacker uses homograph attack to spoof site. Attacker collects credentials & credit card info of target victim. Reference: Can a fraudulent website still have a locked padlock icon?	Prevention: keep updated SSL certificates installed on all websites; doesn't fully prevent but makes attack more difficult as they have to register for spoofed SSL cert in addition to registering spoofed domain/spoofing DNS. Reaction: If the attacker is spoofing DNS and creating their own SSL certificate on their fake site, not a lot can be done on the server side to prevent this; user needs to take close look at URL, SSL cert, use bookmarks.

CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

Confidentiality, Integrity	Attacker leverages partial ("Flexible") Cloudflare encryption to view unencrypted traffic between Cloudflare server and web server, then signs in and tampers with accounts using discovered unencrypted credentials.	Prevention: More maintenance required but use Full encryption mode / end-to-end encryption, so traffic is encrypted between user and Cloudflare AND between Cloudflare and web server.
Target: Email (MFA)		
Confidentiality	Attacker accesses Gmail account or organization-related Outlook account.	Prevention: Enable authenticator app-based MFA for all Gmail accounts and Outlook accounts. Reaction: react to email alert about new login, force log out new device & change password.
Availability	Attacker accesses Gmail account and changes password, locks me out.	Prevention: Enable authenticator app-based MFA for all Gmail accounts and Outlook accounts. Reaction: Use linked phone number (assuming unchanged) to get an account recovery code. Keep a separate recovery email stored on account. *NOTE: Tested with test Gmail account, they sent a recovery code to my phone which I entered and then they also sent a separate required recovery code to the same email account of which I was claiming to be locked out. Realized it was due to not having a recovery email, so that is now fixed.*
Availability, Integrity, Confidentiality	Attacker signs into Gmail account, changes password, changes verification email addresses, changes MFA settings, locks me out. Attacker reads sensitive emails, sends malicious emails to others from my address, uses email access for password resets on other online accounts.	Prevention: Enable authenticator app-based MFA for all Gmail accounts and Outlook accounts. Don't store sensitive information in email inbox. Discourage others from sending such information via email. Reaction: If attacker has gotten this far, you're likely taking a loss. But Google does offer an account recovery page specifically for these kinds of situations.
Target: Social Media – Instagram (MFA), Twitter (MFA), LinkedIn (MFA), YouTube (MFA via Google authentication)		
Availability	Attacker locks LinkedIn / Twitter / Instagram account by intentionally attempting random string of likely wrong passwords rapidly. LinkedIn docs. Twitter docs. Instagram has no docs about this but tested with a test account which did lock.	Prevention: Do not store critical information on social media or have backups in alternative locations particularly for media files/milestone info. Cannot control social media server-side lock-out response but can control what you are being locked out of.
Confidentiality, Integrity	Attacker logs in to LinkedIn / Twitter / Instagram account.	Prevention: enable MFA on Instagram, LinkedIn, Twitter. Avoid further identity theft by not posting personal info.

CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

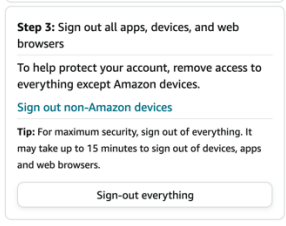
		Reaction: React to email alert from social media platform about login from new device. Immediately login, change password. Notify connections about potential compromise.
Availability	Attacker successfully logs in to LinkedIn / Twitter / Instagram account, changes security settings/recovery email addresses/MFA settings. Attacker creates malicious/reputation-damaging content on social media under my name or entirely deletes all content.	Prevention: enable MFA on Instagram, LinkedIn, Twitter. Reaction: for Twitter, contact support and report the hack. For Instagram, deny the email request to change email address or if already changed, request a security code/support. For LinkedIn, report hacked account immediately.
Target: Online Bank Account		
Availability	Attacker runs brute force login attempt and locks my account (MFA not an available feature).	Prevention: This is my bank's policy unfortunately; better solution would be for bank to allow MFA as a security feature and to block only offending IP addresses. Call the bank and request/suggest a policy change for online account access? Reaction: report problem to bank. Request support to regain access. Request that they block the attacker IP(s), though not likely to happen.
Confidentiality, Availability, Integrity.	Attacker first gets access to email address on file with bank. They call the bank and request a one-time PIN be sent to the email. They use the PIN to log in and reset the password to something of their choice. ALTERNATIVELY, they guess the password randomly with brute force (unlikely). They guess answer to security question based on social media. They change primary & secondary email addresses and phone numbers. They get access to account numbers. Fraudulent purchases are made, account is wiped.	Prevention: Root entry is ultimately email access, so enable MFA on all email accounts to make up for bank's lack of MFA. Monitor alerts about new logins to email accounts, since Google sends these. Use an obscure security question whose answer is not indicated anywhere online since bank allows for bypassing email verification with security question. Reaction: Report fraud to bank over phone. Verify identity, freeze account, have them revert account settings & change password & force log out current user. Also request creation of a different account with new information.
Target: Work & School-related O365 Account (MFA)		
Confidentiality, Integrity	Attacker correctly guesses or finds O365 password and leverages predictable work schedule to trigger MFA request around the time that I am also logging in for work/school such that I approve MFA request for attacker accidentally. Alternatively, they use a similar attack to the one drawn in the diagram on the last page to bypass MFA.	Prevention: use very long password, stored only in LastPass and nowhere else, for O365 accounts. Don't reuse. Change cyclically. Aside from that, accept risk as this is vector is very unlikely to be successful.

CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

O365 is locked down quite tightly; if you try to reset your password, you need to:

1. provide the linked email address, go through a CAPTCHA challenge,
2. go through a phone-based or authenticator-based verification step 1 which involves correctly entering a full phone number given the last two digits and then entering the code sent to that number
3. go through a phone or authenticator-based verification step 2 which involves approving an authentication request on a pre-configured authenticator mobile app.

So, I just keep my password hidden (I don't even know it) and long, and rely on MFA to protect my account.

Target: File Storage - OneDrive, Google Drive		
Availability	Attacker accesses OneDrive or Google Drive and completely deletes all stored files.	Prevention: MFA, keep a backup of files stored in the cloud.
Target: Password Storage – LastPass (MFA)		
Confidentiality	Attacker uses homograph attack to send a security alert email from an address like security-alert@lastpass.com (capital i instead of l) with a link to a spoofed site that has its own spoofed SSL cert, requesting master password entry. They use that master password entered to actually input it into LastPass, which triggers an MFA request to my phone, which I approve thinking it was triggered by my own input.	Prevention: protect LP account with complex master password and LastPass Authenticator based MFA. Check from addresses very carefully, especially for security/password-related emails. Check SSL certificates on links. Report email to LastPass admin in organization before taking any action.
Bank-linked Accounts: PayPal (MFA), Venmo (MFA), eBay (MFA), Internet & Utilities, Credit Card account (MFA for new devices), Spotify, Planet Fitness, Fidelity (MFA), Vanderbilt billing portal (MFA), Stripe for payment processing (MFA), Gemini (MFA), ezTaxReturn (MFA)		
Availability	Attacker attempts password incorrectly too many times & locks account.	Prevention: Not much I can do as a service user to prevent someone else from incorrectly guessing my password. Reaction: Report to support team for the affected service.
Target: Credit-linked accounts – Amazon (MFA)		
Confidentiality, Integrity	Attacker accesses Amazon account by tricking me into entering my password into a spoof homograph site amazom.com, which they enter into real amazon, which triggers MFA request to my phone; I provide code which they then take and enter into real amazon.	Prevention: Keep MFA enabled, pay clear attention to links clicked on, URLs in address bar, SSL certificates when entering secure information into web forms. Reaction: this would certainly trigger a "new device" alert to my email address on file; react to that email by signing in, changing password, and signing out all connected devices. Inconvenient but necessary. 

CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

Development: AWS (MFA), GitHub (MFA), Heroku (MFA), Docker Hub (MFA), Cloudflare (MFA), GoDaddy (MFA)		
Confidentiality, Integrity, Availability	Attacker accesses AWS credentials used for S3 storage & access of media files by public web application whose source is on GitHub. They either remove all media files or they replace them with their own media files with similar names. Note that `AWSCompromisedKeyQuarantine` Which Amazon auto applies when keys are compromised, only limits actions in IAM, EC2, Organizations, Lambda, and Lightsail. Not S3.	Prevention: Do not include AWS creds in GitHub repository; ignore with .gitignore. Keep a separate backup of media files in a different location either using a different set of AWS credentials for a separate S3 bucket, another cloud platform entirely, or simply a local hard drive. Use principle of least privilege for the AWS programmatic access; use granular RBAC to limit access only to specific S3 bucket. Reaction: Monitor and quickly react to alerts from Amazon about key exposures since they run scans automatically and apply that quarantine policy by default. disable AWS creds, create new ones. Restore corrupt/missing media files from backup in separate location.
Confidentiality, Integrity	Attacker accesses SSL private key file for web app deployed with Docker by pulling public Docker image that was built with the SSL cert and key inside for convenience. Attacker uses cert & private key on a separate illegitimate site for their own malicious purpose. Same vulnerability applies to anything that needs to be kept secret.	Prevention: If you include private keys in the build of a Docker image, that image needs to be stored privately, not publicly. Note that docker limits the number of private repositories you can have (1 for personal). Less convenient but more secure alternative is to not include those files in the Docker image, store the image either publicly or privately, keep the files directly (securely) on the web server / host file system and *mount* them into the running container(s). Reaction: Maybe run a job that automatically scans (within your own repo, for defense) Docker hub repositories for secret information; revoke or replace secret values when you get an alert that they have been exposed. If you get an alert triggered by a scan, it's quite possible that an attacker is already acting on that secret value
MFA via Authenticator Apps		
Availability	Non-backed-up mobile device breaks/stops functioning. All Authenticator application-based MFA is lost.	Prevention: enable MULTIPLE MFA methods on accounts that allow it (some only allow one method, e.g., only SMS or Authenticator). Also, more obviously, back up mobile device at least weekly so Authenticator app data can be restored. Reaction: use secondary verification/MFA to regain access & re-add Authenticator MFA for accounts that have multiple verification methods. For those that do not, request support to regain access and replace MFA method with new mobile device.
Physical Devices		

CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

Availability, Confidentiality, Integrity	Thief steals MacBook from table at a coffee shop. Uses built-in recovery mode to reset the password without needing any other creds. Scans filesystem for entry methods to other systems, tampers with files, uses browser caches to access unexpired web sessions, uses LastPass browser extension to view all passwords to other systems.	Prevention: Do not leave MacBook anywhere in public. Locking screen is not enough. Take it with you if you are walking away. Also, note that MacOS doesn't have a limit set on the number of incorrect password attempts you can make. To prevent thief's ability to bypass password on MacOS, enable FileVault for software-based disk encryption and enable a firmware password as hardware-based encryption, which requires a password when booting to recovery mode. Reaction: Use Find My service/app to locate & report or pursue the stolen device if enabled pre-theft. If not enabled, follow alternative steps here
Availability, Confidentiality, Integrity	Attacker sits somewhere behind me at a coffee shop where they can see over my shoulder. They watch me enter my iPhone passcode a few times until they've confirmed it. They steal it when I walk away temporarily, knowing the code and thus having full access.	Prevention: use low screen brightness or use a privacy screen protector; generally be aware of people around/behind you when entering sensitive values; do not leave your phone anywhere, always keep it on you in public. Reaction: Use Find My service/app to locate & report or pursue stolen device if enabled pre-theft. Follow instructions from Apple Support.

Realization during this assignment: A ton of apps are protected by MFA nowadays. What makes MFA work? Ultimately it seems to be timing; when you get an MFA code or an MFA approval request, it is the timing of that request that makes you believe it is real and triggered by you; *"I just entered my password so this approval request is expected"*. Hackers who leverage that timing to make you think their own MFA requests are yours can bypass MFA. Based on some research it looks like this would generally be done by spoofing the password entry forms that would trigger MFA, which means it is up to us as users to pay special attention to URLs, SSL certs, from addresses, etc., before entering credentials.

Security Self-Report

- Multifactor Authentication
  - Accounts that Support MFA
    - 28/29 have MFA enabled
    - Need to implement on 1 personal / custom web app with account management
  - Accounts that do not support MFA
    - 5 accounts, which each use long >=24char passwords
- Security/Recovery Questions
  - No accounts using security questions whose answers can be found on social media/online
- Passwords
  - No reuse; all passwords managed through LP password manager
  - I do not share passwords with anyone, and don't know them myself in most cases
- Antivirus
  - SentinelAgent for Windows machine, Norton 360 Standard for MacOS devices

CS 6387 – Topics in Software Engineering: Security
Week 2 Homework – Threat Modeling Your Online Identity
8 Sept 2022
Austin Hunt

- LastPass security score:
  - 85.5% - unfortunately it counts multiple sites behind Azure SSO as having a "reused password"
- Backups
  - Time Machine used for MacOS backups
  - iPhone hasn't been backed up in a few months at least

How could an attacker hack their way through MFA and gain access to all of my passwords?