Penetration Testing Application Programming Interface (API) Security

University of Chester

June - 2023

M.Sc. Research Dissertation

Department of Computer Science

Adam Thomas Wallwork
1912062 - J78878

Introduction

What’s an API?

- An API (Application Programming Interface) is a set of rules that allows software applications to communicate and interact with each other, enabling exchange of data and sharing of functionality (ibm, n.d).
- “API is a building block for developing complex and enterprise-level applications.” (OWASP, 2023)

What are they used for?

- Enables third party developers to build applications and services on top of existing platforms

Who uses them?

- Cars, operating systems, hardware, software, mobile apps, the cloud, web apps, network services, private and public APIs and financial companies such as cryptocurrency exchanges and banks

Why am i focussing on API security?

- There is a significant amount of research and tooling available for network and web application hacking, but I noticed a lack of focus on testing API security.

Three most commonly used APIs (Bombal, 2022)

- Representational state transfer (RESTful) API
- GraphQL API
- SOAP API

Why focus on API security (Web APIs)?

- APIs account for 83% of all web traffic, indicating their significant role in modern web applications (Mathur, 2020)

If so much of the web is powered by APIs exploiting them for attackers could return a high reward with selling user data.

- Two thirds of cloud breaches are due to misconfigured APIs (IBM Security X-Force Threat Intelligence, 2021)

Developers often hard code credentials and API tokens into their APIs and leak API tokens with high privileges providing attackers an easy way into your network.

API security not being prioritised

Developers not yet focussing on security testing their APIs

- A survey completed by stateofapis interviewed developers across multiple countries shows that all of the responses received (92.7%) shows that developers plan to test their APIs but only 4.0% plan to security test their APIs (stateofapis, 2022)

With only 4% of developers prioritising security tests this leaves a wide open door for attackers to test your APIs for you.

- According to Gartner's prediction, by 2022, API attacks will surpass any other exploitation methods and will become the most common way for cybercriminals to exploit vulnerabilities in enterprise web applications, leading to data breaches (Novikov, 2022)

As of 2023 we have seen an increase in API related exploits and data breaches as a result.

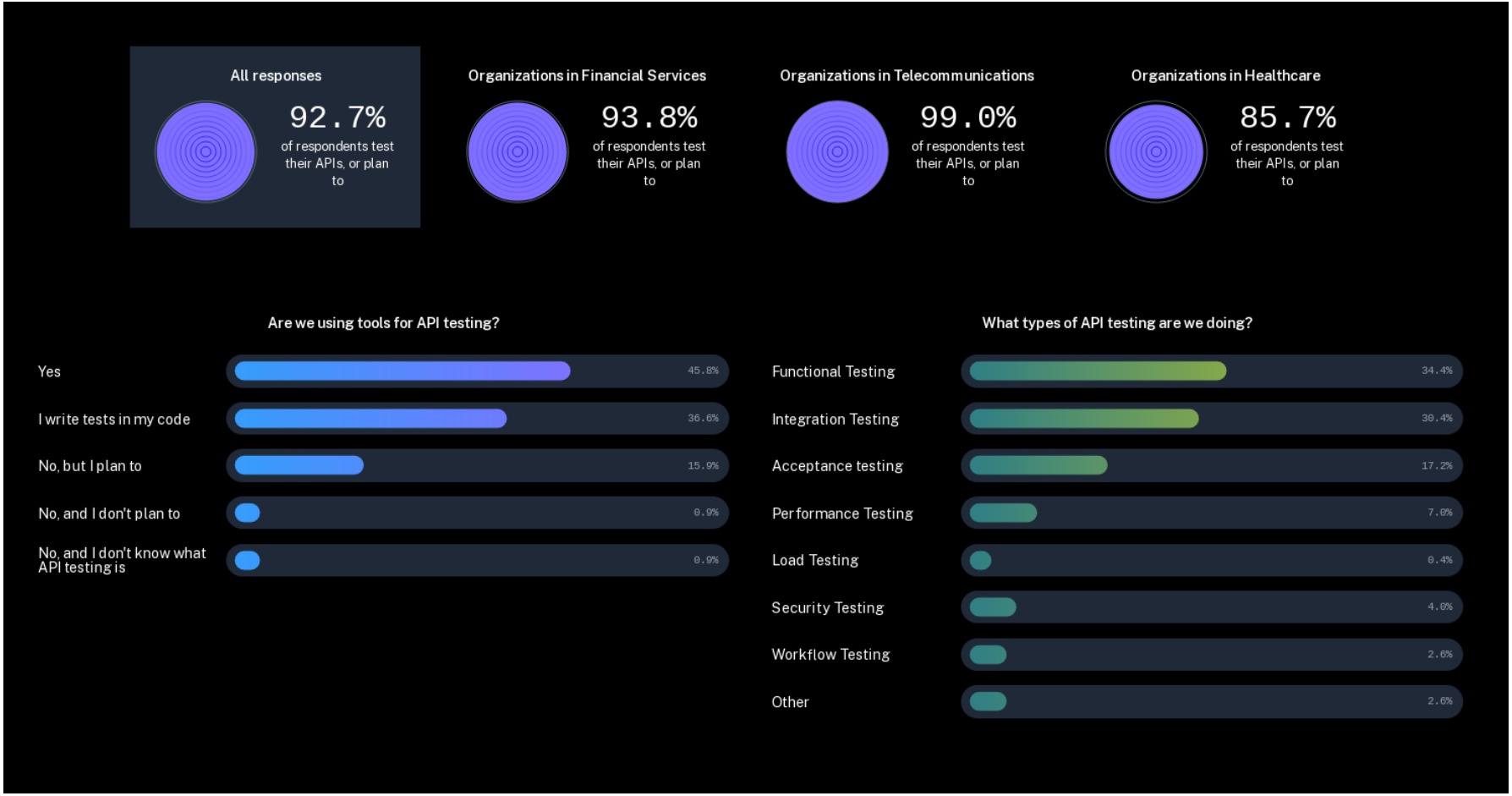


Figure 1: What types of API testing are we doing? (stateofapis, 2022)

Security professionals not yet focusing their efforts on API specific penetration testing as part of their penetration test

- As seen in the USPS penetration test report (Inspector General, 2018) penetration testers that focus on web primarily target web applications and not APIs specifically resulting in large data breaches as seen in the USPS 60 million user data breach in 2018 (Krebs, 2018)

During their assessment, USPS penetration testers examined both web applications and APIs for vulnerabilities. However, their focus on common web application bugs resulted in the oversight of a critical flaw in the API's functionality. This flaw, rather than a specific exploitable vulnerability, ultimately led to the breach.

APIs are different than web applications!

- Many researchers approach testing APIs the same way they would approach a web application, they would look for the same vulnerabilities, use the same tools and fuzzing word lists and use the web application hackers methodology.

- The problem with this is APIs are built and operate differently than web applications and if you test an API the same way you would a web application you’re going to miss critical vulnerabilities that exist in the logic and functionality in the API.

- APIs use different technologies, different paths and files so using web application fuzzing word lists for your content discovery won't be as effective as using a tailored API specific word list.

Data Breaches caused via the exploitation of APIs

- **Twitter:** Over 200 million Twitter users have had their private and public twitter profile information leaked online via exploited API which enabled attackers to scrape user data and later sell it (Abrams, 2023)

Threat actors were able to exploit a vulnerability in the twitter API which allowed them to scrape data for both public and private data associated with twitter profiles. The exploit relied on the threat actor already having access to pre-existing email addresses and phone numbers to then use them to search for users and then scrape their data.

- **T-mobile:** Customer information disclosure supported sim swap and dox attacks via exploited API leaking customer information via customer phone number (Gallagher, 2017)

T-mobile had an API endpoint which allowed users to input a victims phone number which returned all of their customer information. This API vulnerability was used mainly by sim swap hackers to steal social media account usernames and cryptocurrency wallet coins bypassing 2FA.

- **LinkedIn:** 700 million LinkedIn users had their information scraped via vulnerable API which was later put up for sale on underground forums (Taylor, 2021)

Cause of vulnerability was not explicitly made clear.

- **USPS:** A security flaw in the U.S. Postal Service website API exposed account information of up to 60 million users (Krebs, 2018)

The USPS API had features that accepted wildcard search parameters, enabling users to retrieve all records without specifying specific terms. Access controls were not properly implemented allowing for any logged in user to query the system for account details of other users (Krebs, 2018)

Financial incidents via API vulnerabilities

- **Coinbase:** An API vulnerability in Coinbase's trading interface for retail advanced trading (missing logic validation check) allowed users to submit trades with mismatched source accounts, which could potentially result in unauthorized transactions on the Coinbase exchange (Coinbase, 2022)

- **Banks and Cryptocurrency exchanges:** "100% of the APIs tested were vulnerable to OWASP API1:2019 Broken Object Level Authorization (BOLA) vulnerabilities allowing Knight to change the PIN code of any bank customer's Visa ATM debit card number and transfer money in and out of accounts." (Knight, 2021)

- **Venmo:** Venmo's public API endpoint had a vulnerability which allowed an attacker to scrape customer data which resulted in exposing Venmo's users financial activity (Salmon, 2019)

Data breach risks?

- The risks of a data breach for users is that of potential identity theft, using someone's identity to perform sim swap attacks, doxxing of individuals and fraud.

Risks

- Fraud
- Identity theft
- Tailored social engineering
- Fines and penalties to companies
- Reputational harm for companies
- Selling off scraped data via API exploitation to advertising and marketing agencies

OWASP API TOP TEN	Vulnerability
1	Broken Object Level Authorization
2	Broken Authentication
3	Broken Object Property Level Authorization
4	Unrestricted Resource Consumption
5	Broken Function Level Authorization
6	Unrestricted Access to Sensitive Business Flows
7	Server Side Request Forgery
8	Security Misconfiguration
9	Improper Inventory Management
10	Unsafe Consumption of APIs

Table 1: OWASP Top Ten API Vulnerabilities (OWASP, 2023)

Broken Object Level Authorization (Bypass access controls)

Broken Object Level Authorization is basically Insecure direct object reference (IDOR). An example of this would be when user A can access user B's data when they are not suppose to and vice versa.

User A (123):

- Attacker makes a HTTP GET request to user B: GET /api/user?id=124 and finds their account information (date of birth, email address, phone number, full name and street address, etc)

User B (124):

- Attacker creates a second account, identifies their user ID to be 124 and makes a HTTP GET request for user ID 123 which is user A's account ID: GET /api/user?id=123 and finds their information and now has access to user A's data from user B when they are not supposed to be authorised to access this data.

An attacker could weaponise this to cycle through all possible combinations of user IDs and collect all user data from the site.

Research Question, Hypothesis and Rationale

Rationale

- The focus of this research is to develop a thorough API security specific penetration testing methodology to ensure the security of an API.

Hypothesis

- Implementing an effective API penetration testing methodology will significantly enhance the security of APIs and reduce the risk of data breaches.

Research question

- How can API penetration testing be conducted effectively to improve API security and prevent future data breaches?

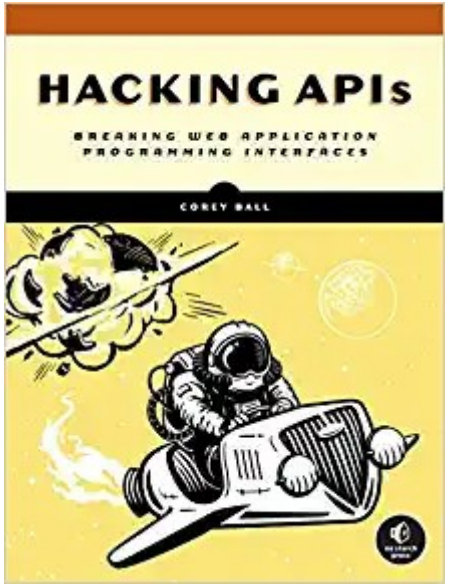


Figure 2: Hacking APIs (Ball, 2022)

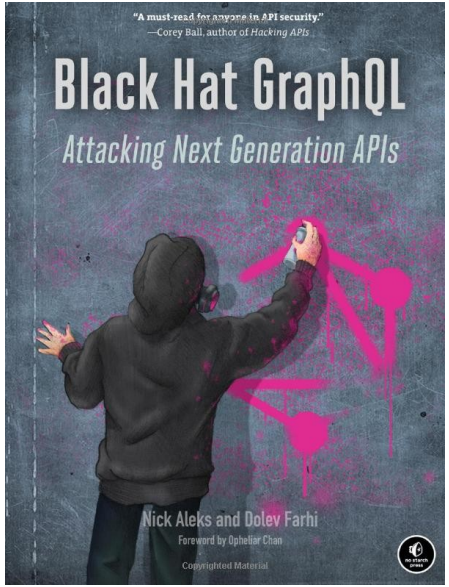


Figure 3: Black Hat GraphQL (Aleks, et al. 2023)

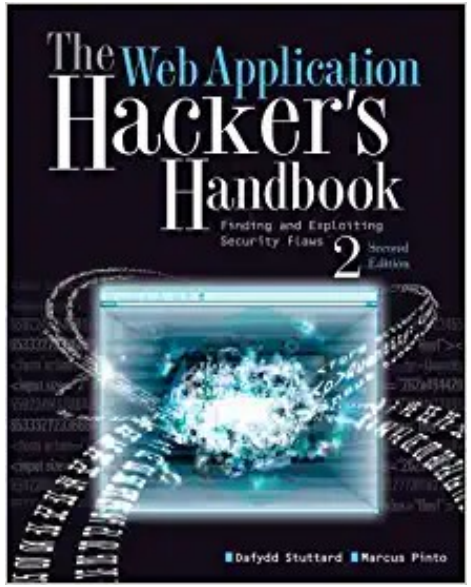


Figure 4: Hackers Handbook (Stuttard. et al. 2011)

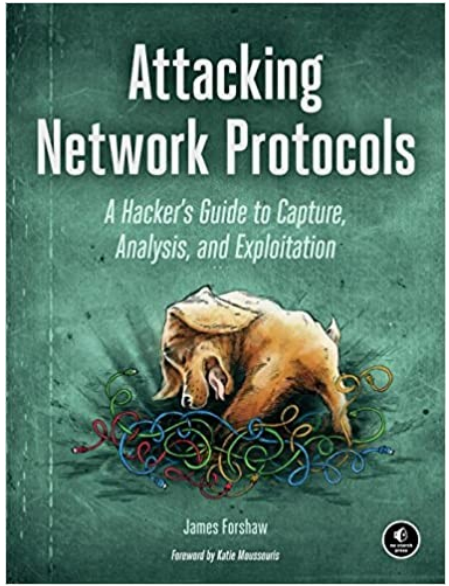


Figure 5: Attacking Network Protocols (Forshaw, 2017)

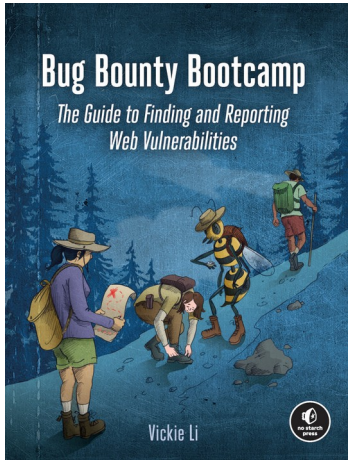


Figure 6: Bug Bounty Bootcamp (Li, 2021)

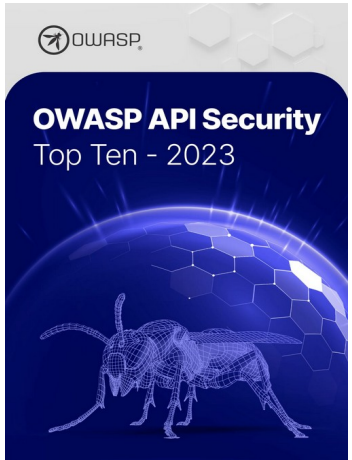


Figure 7: OWASP API Security (OWASP, 2023)

Literature Review

- Hacking APIs: Breaking Web Application Programming Interfaces (Ball, 2022)

This book is dedicated solely to API hacking and API specific security testing using the OWASP API Top Ten vulnerabilities and tools specific to API testing.

- Black Hat GraphQL: Attacking Next Generation APIs (Aleks. et al. 2023)

There are many different types of APIs from Restful APIs, GraphQL APIs and OpenAPI APIs but GraphQL API hacking focusses on GraphQL and hardening GraphQL security controls as GraphQL increases in adoption with developers.

- The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws 2nd Edition. (Stuttard. et al. 2011)

This book teaches you everything you need to know and all the different vulnerability classes associated with web applications such as SQLi, XSS, LFI, RFI, etc. Teaches you how to look for those vulnerability types in different web applications and how to use tools to identify and exploit these vulnerabilities.

- Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation 1st Edition (Forshaw. et al. 2017)

This book focuses on different network protocols and how you can look for and exploit misconfigurations and vulnerabilities and is an important read when dealing with web applications and APIs as they use protocols such as HTTP and HTTP methods such as GET, PUT, DELETE and POST.

- Bug Bounty Bootcamp: The Guide to Finding and Reporting Web Vulnerabilities (Li, 2021)

Focusses on web application hacking from recon to exploitation and focuses more on bug bounties and teaching you everything you need to know to become a bug bounty hunter, however it still provides a good insight into the hacker mindset, how to hunt for vulnerabilities and how to use tools such as Burpsuite to find and exploit bugs.

- OWASP API Security Project (OWASP, 2023)

The OWASP project is a community driven project that focuses on comprising the ten most critical types of vulnerabilities in web applications and as of 2019/2023 APIs and tools that can look for, exploit and abuse these vulnerabilities. This resource is primarily designed for developers to ensure that they are aware of the vulnerability types in an effort to squash the bugs before you roll out your software into production.

Comparable studies

- The Tangled Web: A Guide to Securing Modern Web Applications 1st Edition (Zalewski, 2011)

- Web Application Defender's Cookbook: Battling Hackers and Protecting Users 1st Edition (Barnett, et al. 2013)

- API Security in Action (Madden, 2020)

- Web Application Security: Exploitation and Countermeasures for Modern Web Applications (Hoffman, 2020)

Gaps in the literature

- Lack of comprehensive and standardised methodologies
- Limited focus on real world scenarios
- Emerging threats and vulnerabilities
- Integration of secure coding practices during the development life cycle

- Standardising and benchmarking API security testing tools
- Impact of exploited API vulnerabilities and it’s long term repercussions due to data breaches
- Sufficient arsonal of API security testing tools focused on APIs and not web applications
- Lack of research focus on mobile applications and their APIs
- Compliance and regulatory considerations specific to APIs

Work Completed So Far

- Finding relevant and up to date literature on API security

See literature review, comparable studies and data breach via API exploitation examples

- Finding purposefully vulnerable API security sandbox for security testing

Checkmarx purposefully vulnerable API sandbox is what i’ve chosen thus far but setting it up and configuring it is seeming difficult and their github repository has not been updated since it’s release

- Learning the OWASP API Top Ten vulnerabilities

See slide 13

- Finding relevant penetration testing tools specific to API security testing

See slide 22

- Research of API exploitation in real world cases that resulted in data breaches

See slides 8, 9 and 10

- Becoming familiar with the tools and their configurations for security testing

Finding the established API security researchers to follow and learn from their research

- Alissa Knight (Knight, n.d)
- Cory Ball (SANS, n.d)
- Katie Paxton-Fear (Paxton, n.d)
- OWASP (OWASP, 2023)
- David Sopas (Sopas, n.d)

API security specific testing tools

- Postman (Postman, n.d)
- Burpsuite (Portswigger, n.d)
- MindAPI (Sopas, n.d)
- Kiterunner (Assetnote, n.d)
- Wordlists (Assetnote, n.d)

- Finding the common causes of API vulnerabilities and the most common way attackers exploit APIs to steal data (scraping)

This information has been gathered by finding data breaches caused by exploiting APIs and looking for the root cause of the breaches

Planned Work

Future work planned

- Finish reading the sourced body of literature
- Find and use a purposefully vulnerable API security sandbox for testing
- Set up and configure an API security sandbox for security testing OWASP Top Ten API vulnerabilities
- Create the hackers API penetration testing methodology
- Conduct API penetration testing against all of the OWASP API Top Ten vulnerabilities
- Evaluate test results
- Write research report
- Finalize and submit dissertation

Challenges faced

Challenges faced so far

- Finding sources. API security is still up and coming and is not as heavily documented as Active Directory, phishing and web application hacking. Within the past couple of years there has been some major improvements in the educational space with books being released and more security researchers spending the time to learn and document API hacking.

Potential future challenges

- Finding and setting up a good API sandbox for security testing as there are many to choose from (arainho, n.d) and not a lot that are actively maintained.

Conclusion

The Hackers API Penetration Testing Methodology

- To create a comprehensive framework designed specifically for developers and security researchers to effectively test and assess the security of APIs. This methodology will provide a guide, enabling individuals to identify vulnerabilities and evaluate the overall security posture of their APIs.

- It provides a structured framework to identify vulnerabilities, evaluate risks, and recommend appropriate remediation measures, ultimately leading to stronger and more secure APIs.

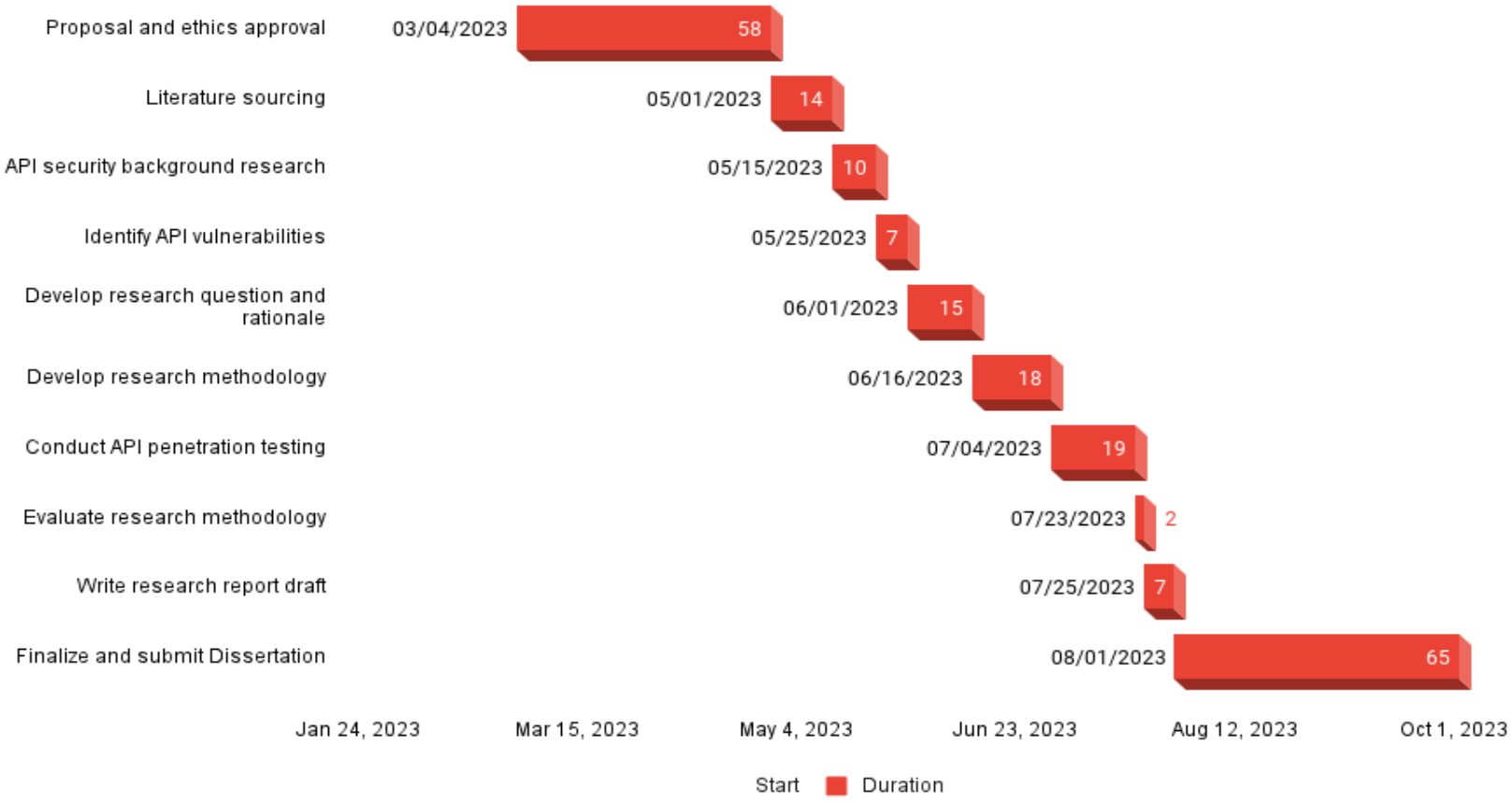
API Hacking Methodology

Jason Haddix (jhaddix, n.d) is a bug bounty hunter and penetration tester who authored the “The Bug Hunter's Methodology (TBHM)” (HackerOne, 2022) for bug bounty hunting focussing on web application hacking. I’m striving for API specific methodology.

- Philosophy (where to look for vulns)
- Recon and Discovery
- Mapping of the target
- Authorisation and session management (Does the API support authentication?)
- Looking for bugs within the API functionality and code base
- Finding the APIs documentation
- Exploring the API functionality

The focus here is not to teach someone how to hack APIs in a standardised way but how to think like a hacker whilst security testing an API to find vulnerabilities in an application and to show tips and tricks to help security professionals test their APIs.

Time Management



References

Mathur, A. (2020). API Discovery and Profiling -- Visibility to Protection. https://www.akamai.com/blog/security/api-discovery-and-profiling-visibility-to-protection

IBM Security X-Force Threat Intelligence, (2021). 2021 IBM Security X-Force Cloud Threat Landscape Report. https://www.ibm.com/downloads/cas/WMDZOWK6

Novikov, I. (2022). How To Address Growing API Security Vulnerabilities In 2022. https://www.forbes.com/sites/forbestechcouncil/2022/07/25/how-to-address-growing-api-security-vulnerabilities-in-2022

Abrams, L. (2023). 200 million Twitter users' email addresses allegedly leaked online. https://www.bleepingcomputer.com/news/security/200-million-twitter-users-email-addresses-allegedly-leaked-online

Gallagher, S. (2017). T-Mobile customer data plundered thanks to bad API. https://arstechnica.com/information-technology/2017/10/t-mobile-website-bug-apparently-exploited-to-mine-sensitive-account-data

Taylor, S. (2021). New LinkedIn Data Leak Leaves 700 Million Users Exposed. https://restoreprivacy.com/linkedin-data-leak-700-million-users

Bombal, D. (2022). Hacking APIs and Cars: You need to learn this in 2023! https://youtu.be/4VaHN4CG34w

Ball, C. (2022). Hacking APIs: Breaking Web Application Programming Interfaces. [Book]

Aleks, N. et al. (2023). Black Hat GraphQL: Attacking Next Generation APIs. [Book]

Stuttard, D. et al. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws 2nd Edition. [Book]

Forshaw, J. (2017). Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation 1st Edition. [Book]

Erickson, J. (2008). Hacking: The Art of Exploitation, 2nd Edition 2nd Edition. [Book]

Li, V. (2021). Bug Bounty Bootcamp: The Guide to Finding and Reporting Web Vulnerabilities. [Book]

OWASP, (2023). OWASP API Security Project. https://owasp.org/www-project-api-security

Coinbase, (2022). Retrospective: Recent Coinbase Bug Bounty Award. https://www.coinbase.com/blog/retrospective-recent-coinbase-bug-bounty-award

Knight, A. (2021). SCORCHED EARTH: HACKING BANKS AND CRYPTOCURRENCY EXCHANGES THROUGH THEIR APIS. https://nonamesecurity.com/hubfs/Scorched-Earth-Alissa-Knight/Scorched-Earth-Whitepaper.pdf

Salmon, D. (2019). I Scraped Millions of Venmo Payments. Your Data Is at Risk. https://www.wired.com/story/i-scraped-millions-of-venmo-payments-your-data-is-at-risk

Freeman, E. (2020). API Security for Dummies. https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWJ9kN

Krebs, (2018). USPS Site Exposed Data on 60 Million Users. https://krebsonsecurity.com/2018/11/usps-site-exposed-data-on-60-million-users

Inspector General, (2018). Office of Inspector General | United States Postal Service Audit Report Informed Visibility Vulnerability Assessment. https://web.archive.org/web/20190412233722/https://uspsoig.gov/sites/default/files/document-library-files/2018/IT-AR-19-001.pdf

Stateofapis. (2022). How important is API testing? https://stateofapis.com/#testing

TryHackMe, (2019). OWASP API Security Top 10 - 1. https://tryhackme.com/room/owaspapisecuritytop105w

TryHackMe, (2019). OWASP API Security Top 10 - 2. https://tryhackme.com/room/owaspapisecuritytop10d0

Arainho, (n.d). Deliberately vulnerable APIs. https://github.com/arainho/awesome-api-security#deliberately-vulnerable-apis

Zalewski, M. (2011). The Tangled Web: A Guide to Securing Modern Web Applications 1st Edition. [Book]

Barnett, R. et al. (2013). Web Application Defender's Cookbook: Battling Hackers and Protecting Users 1st Edition. [Book]

Madden, N. (2020). API Security in Action. [Book]

Hoffman, A. (2020). Web Application Security: Exploitation and Countermeasures for Modern Web Applications 1st Edition. [Book]

Zhirinovskiy, (2022). Discoverability by phone number/email restriction bypass. https://hackerone.com/reports/1439026

Ibm, (n.d). What is an API? https://www.ibm.com/topics/api

Jhaddix, (n.d). The Bug Hunters Methodology. https://github.com/jhaddix/tbhm

HackerOne, (2022). The Bug Hunter's Methodology - Application Analysis | Jason Haddix. https://youtu.be/FqnSAa2KmBI

Knight, A. (n.d). A.V.Knight. https://www.alissaknight.com

Sopas, D. (n.d). Organize your API security assessment by using MindAPI. It's free and open for community collaboration. https://github.com/dsopas/MindAPI

Paxton, K. (n.d). Katie Paxton-Fear. https://insiderphd.dev

Ball, C. (n.d). SANS. https://www.sans.org/profiles/corey-j-ball/

Assetnote, (n.d). Contextual Content Discovery Tool. https://github.com/assetnote/kiterunner

Assetnote, (n.d). Assetnote Wordlists. https://wordlists.assetnote.io

Postman, (n.d). Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs - faster. https://www.postman.com

Portswigger, (n.d). Burpsuite. https://portswigger.net/burp

Figures

Figure 1: Stateofapis. (2022). How important is API testing? https://stateofapis.com/#testing

Figure 2: Ball, C. (2022). Hacking APIs: Breaking Web Application Programming Interfaces. https://a.co/d/0NunOv2

Figure 3: Aleks, N. et al. (2023). Black Hat GraphQL: Attacking Next Generation APIs. https://a.co/d/epbIAuJ

Figure 4: Stuttard, D. et al. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. https://a.co/d/6M3716S

Figure 5: Forshaw, J. (2017). Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation. https://a.co/d/5M0jr01

Figure 6: Li, V. (2021). Bug Bounty Bootcamp: The Guide to Finding and Reporting Web Vulnerabilities. https://a.co/d/dSjLakb

Figure 7: OWASP, (2023). OWASP API Security Top Ten - 2023. https://owasp.org/API-Security/editions/2023/en/0x00-header/