Cobalt
Pen Testing as a Service

5 Things a Pen Tester Looks for When Evaluating an Application

By Robert Kugler

The best defense against attacks is to develop secure applications. It's important for developers to be aware of how application attacks work and use this knowledge to build secure software right into their apps.

When looking at an application, the average person sees the app the way that it is intended to be viewed. However, when a pen tester views an application they see it as a target or an opportunity to find flaws. Pen Testers have a broad set of skills to investigate the constantly-changing vulnerability landscape, and also have a strong grasp of the most prevalent vulnerabilities.

In this cheat sheet, I will delve into how I view an application and explore the 5 most common threats that I've come across. For each of these vulnerabilities, I will explore three questions:

1. What is it?
2. How to look for it?
3. What's the impact?

Let's get started!

1. Injections

What is it?

Injection attacks are those that allow an attacker to insert or pass along malicious code through an application to another system. When talking about injections the first thing that might come to mind is untrusted user input originating from web-based forms and API requests that are used to craft database queries or OS commands.

SQL Injection is one of the most common injection flaws, which refers to an injection attack where an attacker executes malicious SQL statements or a malicious payload that controls a web application's database server.

How to look for it?

You need to combine fuzzing the application using automated tools such as Burp's intruder with manual analysis; otherwise, you will only catch the obvious vulnerabilities while missing the dangerous hidden edge-cases. Always check common parameters such as $id, $product_id, $password or $email but don't forget to look at session cookies or HTTP headers.

What's the impact?

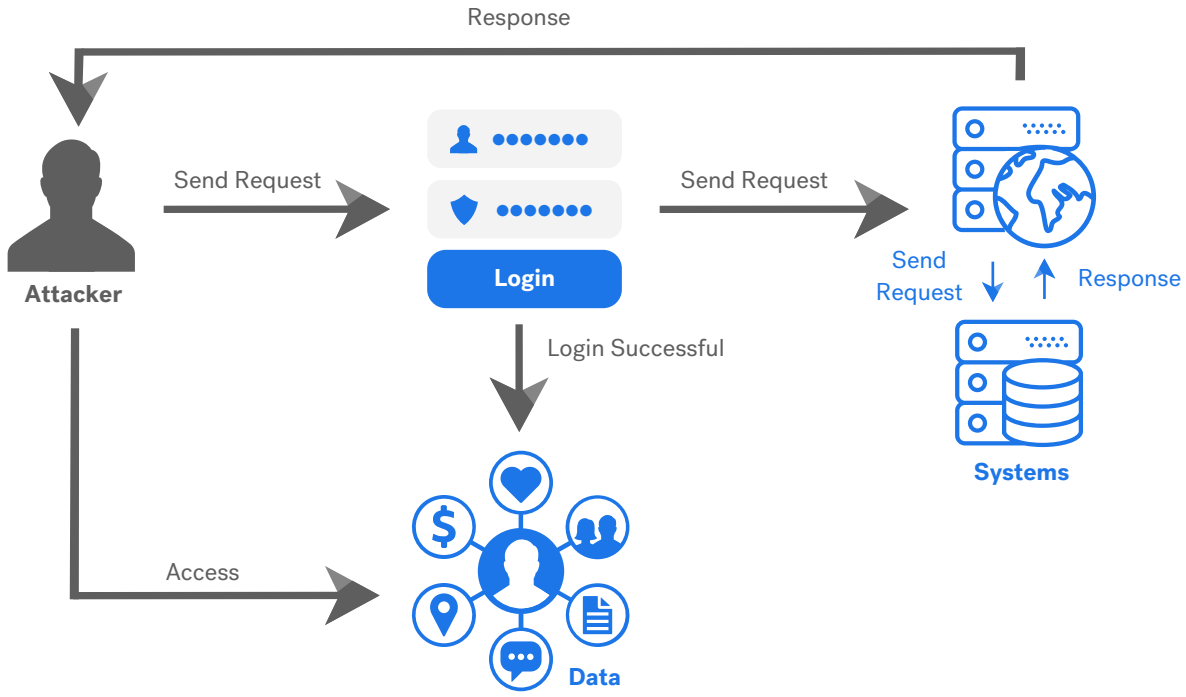
If an injection is done successfully, this can lead to discovery and loss of stored information, giving attackers the ability to change sensitive data or to side-step authentication measures. TalkTalk is an example of a company who was attacked using a SQL injection, attackers accessed the personal data of 156,959 customers including their names, addresses, dates of birth, phone numbers and email addresses. In some cases, even bank account details and sort codes were compromised. The company was punished with a £400,000 fine for failing to prevent the attack, now with GDPR in place, the punishment for this sort of data loss would be even worse. Baroness Dido Harding, the former CEO of Talk Talk, opened Infosecurity Europe 2018 by warning other business leaders of the importance of board level input when it comes to security risks.

SQL injection can also lead to remote code execution, Yahoo is an example of a company who has been affected by this kind of attack. Since then, Yahoo has decided to grow its advanced persistent threat team and fund a red team to test its own products to help strengthen its security. The issue was resolved, likely by improving input validation and the QA process.

2. Broken Authentication

What is it?

For this vulnerability, attackers use weak authentication and session management flaws in an application to impersonate a user. For instance, are attackers able to bypass your two-factor authentication? Do you use weak default passwords? Are your password reset tokens properly randomized?



How to look for it?

When a security researcher attacks an application it is likely that they will start off by going after something like default credentials, and then moving on to check the two-factor authentication (2FA) configuration. Is rate-limiting implemented? Are 2FA codes reused? Can you somehow skip it or reset it? After that, you should test for session-specific vulnerabilities like session fixation or missing session timeouts. Are JSON Web Tokens (JWT) used, and if so, are they properly secured? Another thing that's fun to test is the password reset or account confirmation process. Check for predictable tokens and always test the lifetime. Are they leaked to a third party? Can you control where those tokens are leaked to?

What's the impact?

Such vulnerabilities may allow some or -- worst case scenario -- all accounts, to be

attacked. If successful, the attacker has the ability to do anything the victim can do. Broken authentication often leads to data loss or privilege escalation. In the past, Uber was affected by a vulnerability that could be used to brute-force promo codes and PayPal was affected by a 2FA bypass that could be used to take over accounts after a successful phishing campaign. Both companies quickly patched the vulnerabilities and awarded the researchers a bounty

3. Sensitive Data Exposure

What is it?

Sensitive data exposure is all about leaking data through man-in-the-middle attacks, detailed error logs, or unencrypted backups inside the web server's root directory.

How to look for it?

The questions you need to ask yourself: Is sensitive data encrypted in transit? Are passwords properly hashed using a secure hashing function like bcrypt or scrypt? Does the server use old TLS ciphers? Is the server vulnerable to TLS attacks like Heartbleed or Poodle? Use Dirbuster to brute-force common files & directories to check for backup files or Git repositories.

What's the impact?

Sensitive data exposure always leads to data loss, depending on the attack vector that could lead to the disclosure of sensitive messages, credentials, or files. For example, insecure AWS buckets are known for leaking sensitive data, but a web application penetration test or an AWS configuration review can help to reduce the risk.

4. XML External Entities

What is it?

XML External Entity (XXE) attacks are similar to server-side request forgery attacks, they allow an attacker to exploit a vulnerable XML parser to access local or remote files and services.

How to look for it?

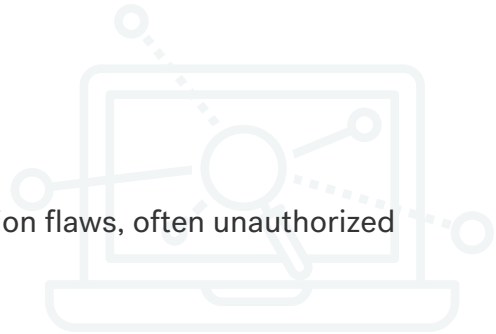
Use Data Type Definitions (DTD) to inject external entities in requests carrying an XML payload. Sometimes there are XML parsers installed with JSON endpoint,

always try to switch the content-type to catch those deprecated endpoints. If you don't have access to the output, use time-specific payloads and try to measure the response time to see if an exploitation attempt was successful.

What's the impact?

Successful XXE attacks will lead to read access on your server and may even lead to remote code execution. Numerous well-known online services were affected by XXE vulnerabilities like Google, Facebook, and Uber. (These vulnerablites were responsibly disclosed, updates were made to their software, and vulnerabilities were patched)

5. Broken Access Control



What is it?

This vulnerability is closely related to broken authentication flaws, often unauthorized users are able to access sensitive data or functions.

How to look for it?

Test all available roles inside the application for privilege escalation attacks to check if strong access control mechanisms are used. But don't forget to look for cross-origin resource sharing (CORS) misconfigurations. Furthermore, nobody should miss testing for insecure direct object references (IDOR) although they might look not that dangerous they are definitely.

What's the impact?

In 2016 an insecure direct object reference allowed an attacker to change your Uber password and one year later an IDOR vulnerability could be used to leak everyone's Airbnb messages. These companies are often targets to hackers as they store a lot of valuable personally identifiable information that could be abused for identity theft, fraud or blackmailing.

Both companies take security very seriously and they all have experienced security teams but still incidents will happen. It's important to remember that how you handle security incidents and how you put measures in place to prevent them in the future are key. Both companies have put programs in place to help strengthen their security from such attacks.

Bonus: Server-Side Request Forgery

What is it?

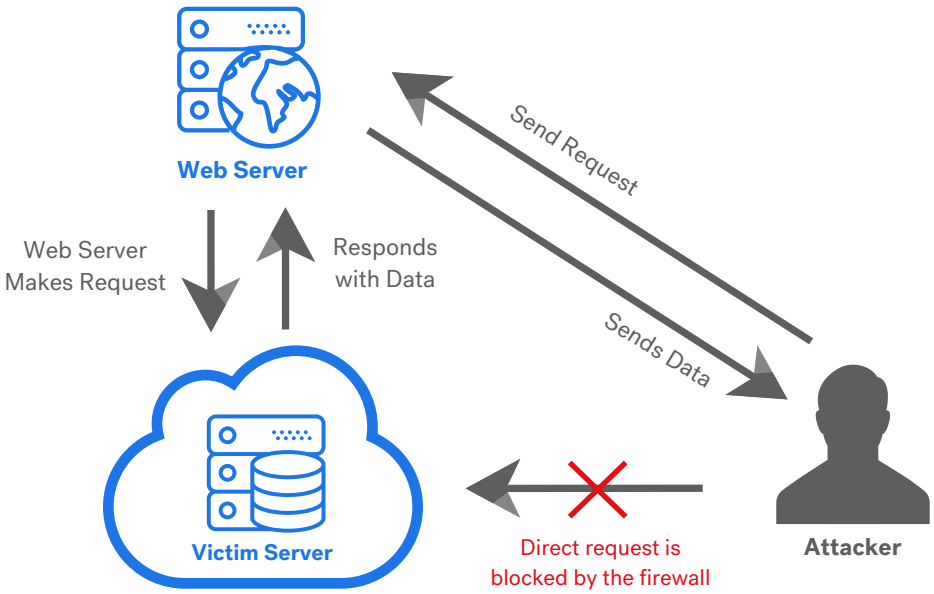
If an attacker is able to send a crafted request from your application to an arbitrary target you are likely vulnerable to server-side request forgery (SSRF). It includes any attack where the attacker can abuse functionality on the server to read or update internal resources.

How to look for it?

Check common vulnerable endpoints like webhooks or features that allow media embedding through online resources and inject your testing server's IP address to see if something connects back to you. If so, try to access local services and record the server's response times to guess if a port is closed or open. Try to enumerate all supported protocols. If you're lucky you might be able to get some hashes from a Windows system or juicy information from the Instance Metadata Service of an AWS server.

What's the impact?

An SSRF vulnerability often allows access to the internal network of a company and may expose endpoints that aren't reachable from the Internet. This can lead to remote code execution if an attacker is able to access local or vulnerable internal web services.



Concluding Thoughts

It's important to note that majority of application-based companies out there can be susceptible to attacks like these and that we should not have a point-and-laugh mentality toward these unfortunate cases, but use it as an opportunity to learn and not repeat them.

Tips to prevent vulnerabilities like this from happening:

- Monitor your vulnerability trends, keep an eye on things that keep popping up.
- Leverage resources that are available to keep up-to-date on overall security trends (blogs, conferences, training).
- Pen test on a continuous basis (Monthly, Quarterly, Semi-Annually, Annual whichver is right for your business) Know where your weakness are.

About the Author

Robert Kugler is an information security researcher and pen tester who has made his passion for breaking things his job. His background stems from over 8 years of data protection, security management and consulting as well as penetration testing.

Robert has helped strengthen the security of companies such as Mozilla, Axel Springer, PayPal, Spotify, Sophos, Sony, Fitbit, and Deutsche Telekom. In the past, he has given several presentations on IoT security, digital self-defense, the security risks of anti-virus software, and application vulnerabilities. Currently, Robert is working with Cobalt.io on managing penetration testing quality.

Visit Cobalt.io to learn more about how pen testing can help strengthen your application.