OWASP TOP 10 2017

Flashcard Guide on The 10 Most Critical Web Security Risks of 2017
One of the most effective security awareness training tools for your company.

WHAT IS IT?
Websites and apps occasionally need to run commands on the underlying database or operating system to add or delete data, execute a script, or start other apps. If unverified inputs are added to a command string or a database command, attackers can launch commands at will to take control of a server, device, or data.

HOW DOES IT WORK?
If a website, app, or device incorporates user input within a command, an attacker can insert a “payload” command directly into said input. If that input is not verified, an attacker then “injects” and runs their own commands.

WHY IS IT BAD?
Once attackers can make commands, they can control your website, apps, and data.

FUN FACTS
SQL injection was leveraged in the infamous Sony Pictures hack of 2014, when suspected North Korean operatives gained access to confidential data. According to US-CERT, the attackers used a Server Message Block Worm Tool to install several malicious components, including a backdoor and other destructive tools.
With just 5 easy steps, you’re ready to begin:

- Step 1: Download
- Step 2: File > Print
- Step 3: Print 2-Sided
- Step 4: Trim cards around edge
- Step 5: Get learning!
OWASP TOP 10 2017
A Flash Card Reference Guide to the
10 Most Critical Web Security Risks of 2017

1
INJECTION
Allowing untrusted data to be sent as part of a command or query

2
BROKEN AUTHENTICATION
Incorrectly implemented authentication and session management functions

3
SENSITIVE DATA EXPOSURE
Many web technologies weren’t designed to handle financial or personal data transfers
What is it?
Authentication is the process for making sure it’s really you accessing your accounts and data. Generally, it’s facilitated by a username and password combination, but complexity is added when people forget or change their passwords or want to update their email addresses. It gets even more complex as a site, app, or device itself becomes bigger, broader, and more connected with other sites, apps, or devices.

How does it work?
In the simplest attacks, passwords can be guessed or stolen if left unprotected. As complexities are added, attackers can find other areas where user credentials or sessions have inadequate protections and then hijack a user’s access, and eventually their data.

Why is it bad?
If attackers can hijack a user’s or administrator’s session, they have access to everything available within that account, from data to account control.

Fun facts
The simplest examples of this vulnerability are either storing user credentials without encryption or allowing them to be easily guessed. Other examples include using session IDs in the URL and enabling unreasonably long session timeouts.

Sensitive data exposure
Sensitive data, such as credit card numbers, health data, or passwords, should have extra protection given the potential of damage if it falls into the wrong hands. There are even regulations and standards designed to protect sensitive data. But, if sensitive data is stored, transmitted, or protected by inadequate methods, it can be exposed to attackers.

How does it work?
If data is stored or transferred as plain text, if older/weaker encryption is used, or if data is decrypted carelessly, attackers can gain access and exploit the data.

Why is it bad?
Once an attacker has passwords and credit card numbers, they can do real damage.

Fun facts
Wireless routers offer notoriously weak data protections. Researchers recently found that the cryptography protecting WPA2, the industry standard, exposes data and allows it to be read or manipulated as it’s wirelessly transferred.

Injection
Websites and apps occasionally need to run commands on the underlying database or operating system to add or delete data, execute a script, or start other apps. If unverified inputs are added to a command string or a database command, attackers can launch commands at will to take control of a server, device, or data.

How does it work?
If a website, app, or device incorporates user input within a command, an attacker can insert a “payload” command directly into said input. If that input is not verified, an attacker then injects and runs their own commands.

Why is it bad?
The data in or out of the database can be used to exploit the software, hijack data, or act as an attack vector.

Fun facts
SQL injection was leveraged in the infamous Sony Pictures hack of 2014, when suspected North Korean operatives gained access to confidential data. According to US-CERT, the attackers used a Server Message Block Worm Tool to install several malicious components, including a backdoor and other destructive tools.

About HackerOne
More than 1,000 organizations, including The U.S. Department of Defense, General Motors, Lufthansa and Starbucks trust HackerOne to find critical software vulnerabilities before criminals can exploit them. HackerOne customers have resolved more than 57,000 vulnerabilities and awarded more than $22M in bug bounties. To learn more, visit www.hackerone.com.

Let’s make the internet safer, together.
SECURITY MISCONFIGURATION
Manual, ad hoc, insecure, or lack of security configurations that enable unauthorized access

CROSS-SITE SCRIPTING (XSS)
A web application includes untrusted data in a new web page without proper validation

XML ETERNAL ENTITIES
XML “entities” can be used to request local data or files

BROKEN ACCESS CONTROL
Improper enforcement of what authenticated users are allowed to do
**Security Misconfiguration**

*What Is It?*

Exactly what its name implies, security misconfiguration is when you’ve overlooked some vulnerabilities. This includes using default credentials, leaving files unprotected on public servers, having known-but-unpatched flaws, and more, and at any layer of the software stack. In other words, it’s your fault.

*How Does It Work?*

People get busy, things get missed, prioritization decisions are made... and vulnerabilities are left unchecked.

*Why Is It Bad?*

It makes it easy for even novice attackers to find and access your valuable systems and data. Luckily most of these types of vulnerabilities are also easy for you to find and fix.

**Fun Facts**

- The infamous Mirai botnet of 2016 relied on unchanged default credentials (such as a login of “admin” and a password of “1234”) of just over 60 specific IoT devices. When exploited, it eventually infected nearly 400,000 units of just those 60 unprotected devices.

**Cross-Site Scripting (XSS)**

*What Is It?*

Access control, or authorization, is how web apps let different users access different data, features, or functionality. It’s kind of how Netflix limits people on their “standard” plan to HD content, while “premium” users can watch 4K. When it’s broken, you have access to (and can modify) data you shouldn’t.

*How Does It Work?*

Sometimes, gaining unauthorized access is as simple as manually entering an unlinked URL in a browser, such as `http://example.com/admin`.

*Why Is It Bad?*

As with other vulnerabilities, attackers can gain access to, and modify, data they shouldn’t.

**Fun Facts**

- A web meeting platform, Fuze, enabled meeting access via a simple URL ending with an incrementing seven-digit number. Using any number provided access to replays of the corresponding meeting. Since the URLs were unprotected, the content was then indexed by — and searchable through — popular search engines.

**XML External Entities**

*What Is It?*

XML is a data format used to describe different data elements. XML also uses “entities” to help define related data, but entities can access remote or local content, as harmless as pulling a current stock price from a third party website. Entities can, however, be used to request local data or files, which could then be returned — even if that data was never intended for outside access.

*How Does It Work?*

An attacker sends malicious data lookup values asking the site, device, or app to request and display data from a local file. If a developer uses a common or default filename in a common location, an attacker’s job is easy.

*Why Is It Bad?*

Attackers can gain access to, and modify, data on your site. Even if the data is harmless, you may have other sensitive data nearby — and attackers can exploit that, too.

**Fun Facts**

- The classic “billion laughs” attack exploits XXE by defining 10 elements that refer to each other, quickly exceeding any available memory and disrupting entire services.

**Broken Access Control**

*What Is It?*

Access control or authorization is how web apps let different users access different data, features, or functionality.

*How Does It Work?*

Some web apps have a “feature” that lets users define new rules that can access data and modify data.

*Why Is It Bad?*

Without tight restrictions, attackers can gain access to (and modify) data.

**Fun Facts**

- When it’s broken, you can access more data than you should be able to when it’s working. How? Stripped-down, broken access controls can allow an attacker to define new rules that grant access to, and modify, data.
INSECURE DESERIALIZATION
Receipt of hostile serialized objects resulting in remote code execution

USING COMPONENTS WITH KNOWN VULNERABILITIES
Finding and exploiting already-known vulnerabilities before they are fixed

INSUFFICIENT LOGGING & MONITORING
Insufficient monitoring allows attackers to work unnoticed

START UNCOVERING CRITICAL VULNERABILITIES TODAY
More security teams use HackerOne to manage vulnerability disclosure and bug bounty programs than any other platform.
WHAT IS IT?
If you’re not looking for attackers or suspicious activities, you’re not going to find them.

HOW DOES IT WORK?
Software and systems have monitoring abilities so organizations can see logins, transactions, traffic, and more. By monitoring for suspicious activity, such as failed logins, organizations can potentially see and stop suspicious activity.

WHY IS IT BAD?
Attackers rely on the lack of monitoring to exploit vulnerabilities before they’re detected. Without monitoring and the logging to look back and see what happened, attackers can cause damage now and in the future.

Insufficient Logging & Monitoring

WHAT IS IT?
When vulnerabilities become known, vendors generally fix them with a patch or update. The process of updating the software eliminates or mitigates said vulnerability.

HOW DOES IT WORK?
Organizations sometimes fail to keep software up-to-date, especially if their stacks are large or complex, or if it would require a significant undertaking to validate their systems or products after an update. When an exploit is made public or a patch is released, attackers know some organizations will not act immediately. Attackers now have a window, from days to years, to search for systems or applications where the known vulnerability is still in place.

WHY IS IT BAD?
Because it’s public information, attackers have a recommended path to exploit, and organizations may face legal or regulatory penalties for failing to act.

FUN FACTS
The former CEO of Equifax, while testifying to Congress regarding their infamous 2017 breach, blamed it on someone in IT, stating “The human error was that the individual who’s responsible for communicating in the organization to apply the patch, did not.”

Insecure Deserialization

WHAT IS IT?
Before data is stored or transmitted, the bits are often serialized so that they can be later restored to the data’s original structure. Reassembling a series of bits back into a file or object is called deserialization.

HOW DOES IT WORK?
Deserialized data can be modified to include malicious code, which is likely to cause issues if the application does not verify the data’s source or contents before deserialization.

WHY IS IT BAD?
Attackers can build illegitimate objects that execute commands within an infected application because of untrusted inputs to deserialization functions.

FUN FACTS
During 2015 and 2016, insecure deserialization was found in so many Java applications, including one at PayPal, the wave of vulnerabilities was dubbed the “Java Deserialization Apocalypse”.

Known Vulnerabilities

WHAT IS IT?
Using components with known vulnerabilities can cause damage now and in the future. What happened? Hackers can cause damage now and in the future.

HOW DOES IT WORK?
Attackers typically exploit known vulnerabilities. They or their tools can poll for networks and start pinging addresses in hopes of finding victims so organizations can look for inputs to any functions.

WHY IS IT BAD?
If you’re not looking for attackers or suspicious activities, you’re not going to find them.

FUN FACTS
The former CEO of Equifax, while testifying to Congress regarding their infamous 2017 breach, blamed it on someone in IT, stating “The human error was that the individual who’s responsible for communicating in the organization to apply the patch, did not.”