




usd HeroLab

ANNUAL REPORT 2020

We protect companies against hackers and criminals.

more security. **usd** 



CONTENTS

Introduction	4
Top 5 Most Prominent Vulnerabilities	6
Cross-Site-Scripting (XSS)	8
Broken Access Control	10
SQL Injection	12
Transport Layer Security 1.0 (TLS 1.0)	14
Server Message Block Protokol 1.0 (SMB 1.0) & SMB Signing	15
Top 3 Zero-Days	16
Top 3 – usd-2020-0060 (CVE-2020-15861) Net-SNMP	18
Top 2 – usd-2020-0002 (CVE-2020-6581) Nagios NRPE	19
Top 1 – usd-2020-0016 (CVE-2020-5836) Symantec Endpoint Protection	19
usd HeroLab Toolchain	20
Expert Voices	24
We Support You	26

IDENTIFY RISKS. ACT PURPOSEFULLY. RAISE YOUR IT SECURITY LEVEL.

“

As in previous years, we identified countless vulnerabilities in our customers' IT systems and applications through our pentests in 2020. It is therefore essential that you know your risks and are one step ahead of the attackers.”

Matthias Göhring
Co-Head of usd HeroLab

Threats from hacker attacks are constantly growing: IT infrastructures are becoming more complex and attackers are adopting increasingly sophisticated and methodical approaches.

Our mission „more security“ drives us to always stay on top of the current and future IT security landscape. This also means that the security analysts in our usd HeroLab constantly keep an eye on the risks our clients face and know what threats companies are exposed to. The crisis year 2020 in particular formed ideal conditions for cybercriminals: great uncertainties, accelerated digitalization, budget cuts, and higher online activity. For example, according to the latest situation report from the German Federal Office for Information Security (BSI) ¹, reports of stolen highly sensitive personal data rose sharply. This shows that IT security is indispensable now more than ever.

We are convinced that technical security analyses, as they are widely performed today, no longer meet the current threat situation and the demands on the market. For this reason, we are continuously investing in the development of the usd HeroLab Toolchain for more efficiency, transparency and higher quality. At the same time, a structured and efficient training of our specialists is necessary to be able to perform at a consistently high level. In 2020, we further developed our comprehensive internal training program, the „usd HeroLab Certified Professional“ (UCP). However, we also focused on our external commitment to education this year. As part of our university cooperation,

we taught IT security in a practical way to qualified young professionals in our „Hacker Contests“. Our numerous CST Academy seminars promoted an exchange and transfer of knowledge with the security community. Joining forces for more security. This is how we start into the new year as well.

¹ „Die Lage der IT-Sicherheit in Deutschland 2020“, Federal Office for Information Security
(Bundesamt für Sicherheit in der Informationstechnik)



TOP 5 MOST NOTABLE VULNERABILITIES

Our security analysts repeatedly uncover gateways into systems and applications that pose significant risks to corporate security. Some vulnerabilities are occurring more frequently in various IT systems than others. We have compiled the five most prominent vulnerabilities for you in this report - how does the hacker operate? What are the consequences for your company? How can you protect yourself better?

In the following, we give general recommendations for security measures.
We are happy to support you with individual solutions.

CROSS SITE SCRIPTING (XSS)

Cross site scripting refers to a category of vulnerabilities that allow an attacker to inject malicious JavaScript code into a web server's responses.

The web browser of other users then cannot distinguish the JavaScript code inserted by the attacker from the legitimate code of the application and executes malicious scripts accordingly. This usually leads to the attacker being able to completely take over the victim's current session.

The fact that cross site scripting appears within a statistic of the most common vulnerabilities is not really surprising. Nevertheless, it is interesting to see that despite increasing use of frameworks and rising awareness among software developers, more than two-thirds of the web applications we tested still exhibited such a vulnerability.

Cross-site scripting vulnerabilities are generally classified by usd as a critical security risk, as the confidentiality and integrity of user data is threatened acutely.

Recommended measures

User-controlled input should always be considered potentially dangerous and should never be embedded within server responses without sufficient filtering and encoding. Appropriate functions for filtering and encoding input are available in all common programming languages. The correct use of frameworks and regular training of developers are important measures to prevent cross site scripting vulnerabilities.

Example

We demonstrate an attack in which credentials stored in the web browser are read by the attacker via JavaScript.

Request

RawParamsHeadersHex

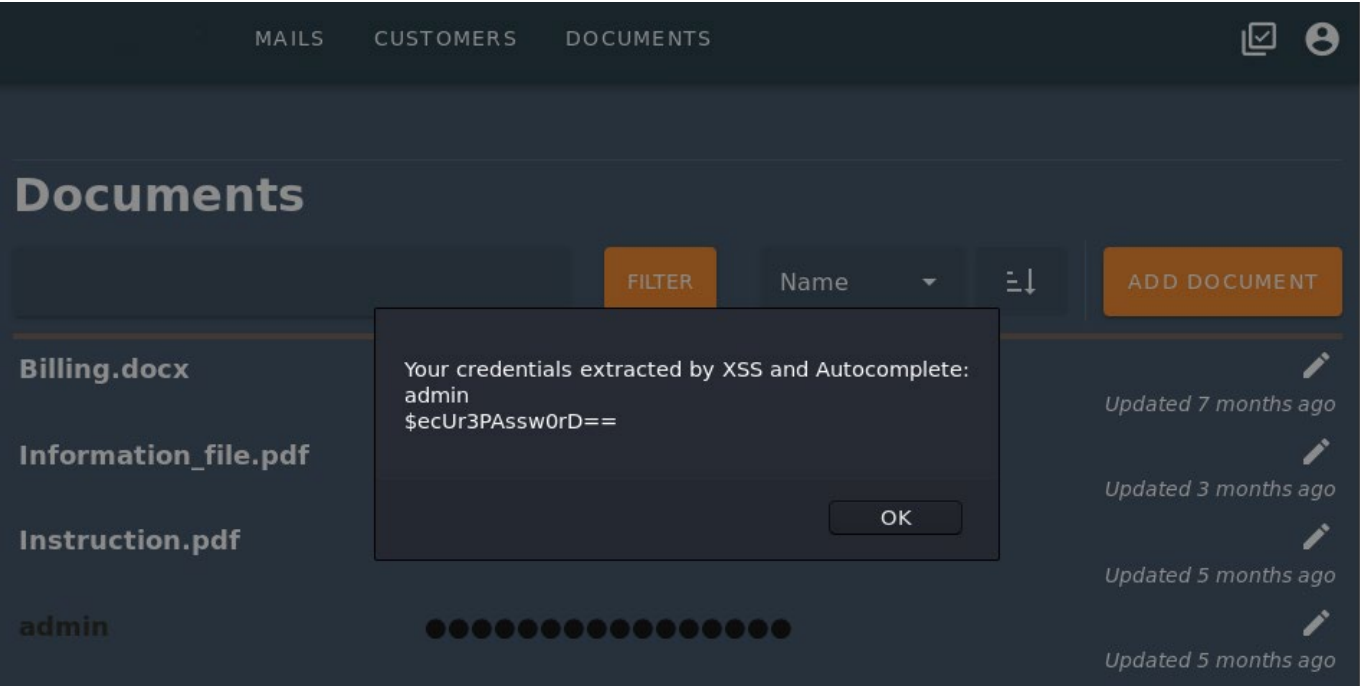
```
1 POST /documents/create/ HTTP/1.1
2 Host: example.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://example.com/documents/list/
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Content-Length: 1110
10 Connection: close
11 Cookie: csrfToken=XvWGRsfmyc2pmvGLc0oT6BM2bpJINyIQsiZt5cmZFqGrblr9bzoIbeGOizdUEM0; PHPSESSID=u2xx25t882woe6jz8i6353ui4mgu6qzz
12
13 csrfmiddlewaretoken=rTKNgg6ROHJQ0nR0FzzCWZEGJuhh4gmdWGNau0duVN7753wGCKK7yz6knn7Mbmgntitle=<script>function+d()+{var+u+%3d+document.ge
tElementById("username").value%3bvar+p+%3d+document.getElementById(
"password").value%3balert("Your+credentials+extracted+by+XSS+and+Au
tocomplete%3a\n".concat(u,+"\n",p))%3b}</script><input+type%3d"text
"+id%3d"username"+name%3d"username"><input+type%3d"password"+id%3d"
password"+name%3d"password'+onchange%3d'd()'>
```

Response

RawHeadersHex

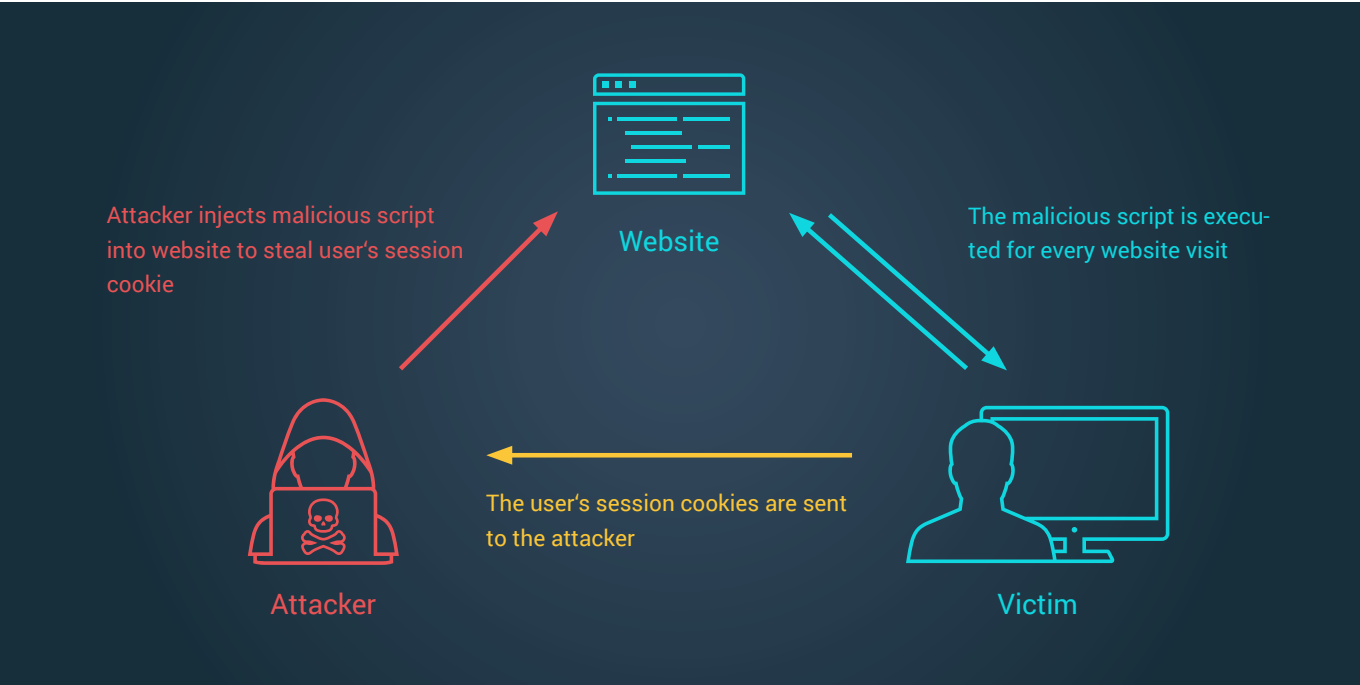
```
1 HTTP/1.1 200 OK
2 Date: Mon, 30 Nov 2020 18:39:28 GMT
3 Server: Apache/2.4.38 (Debian) OpenSSL/1.1.1d mod_wsgi
4 Cache-control: no-store
5 Content-Length: 51
6 Vary: Cookie
7 Connection: close
8 Content-Type: application/json
9
10 {
  "success_url":"/documents/list/",
  "error":false
}
```

Attacker places malicious JavaScript code inside a vulnerable application



A user visits the vulnerable page - their credentials are extracted

While the victim's credentials were displayed here for better visibility, a real attack would take place without any traces visible to the victim. Instead of being displayed on screen, the access data would have been sent over the network to a server controlled by the attacker.



BROKEN ACCESS CONTROL

Broken access control refers to vulnerabilities in which endpoints or functionalities in an application are not sufficiently protected by authentication and authorization mechanisms. Attackers can access these endpoints or use corresponding functionalities without having sufficient authorization to do so.

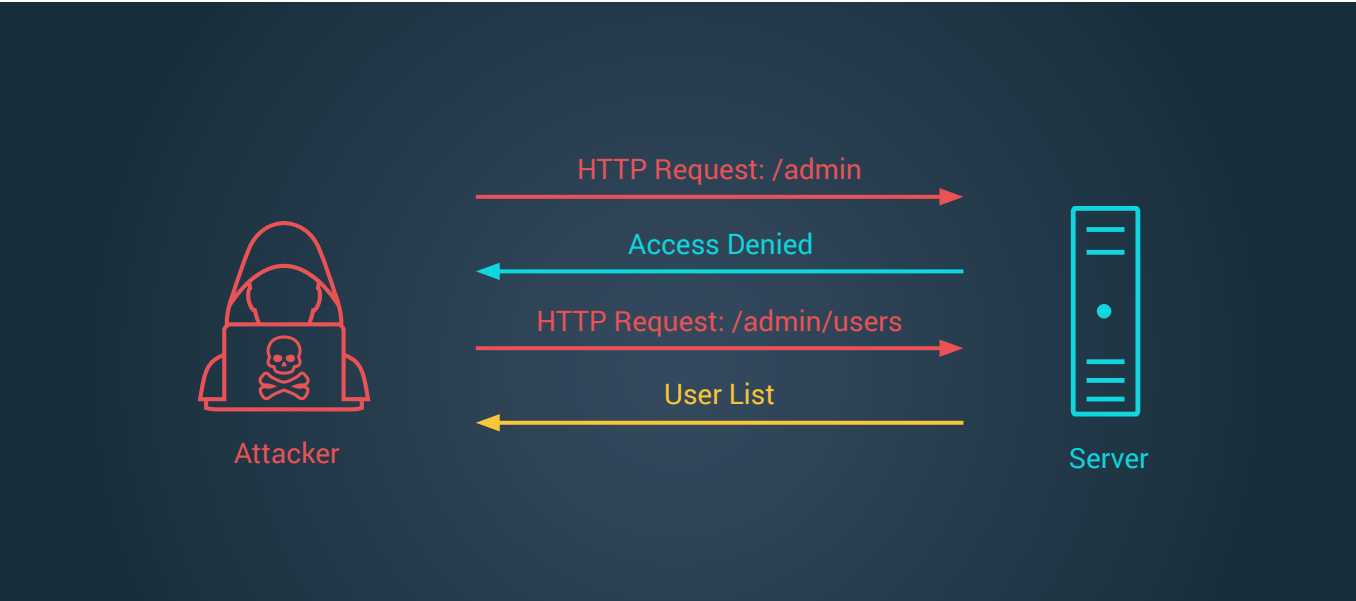
With a frequency of 52%, this vulnerability occurred in more than every second pentest in 2020. One of the most common reasons is that only client-side validation of requests is used, while no further checking is performed on the server side. The example below demonstrates this with an application that does not validate user requests on the server side.

Within the administrative section of the application, administrators have the possibility to set passwords of users. When the corresponding action is performed,

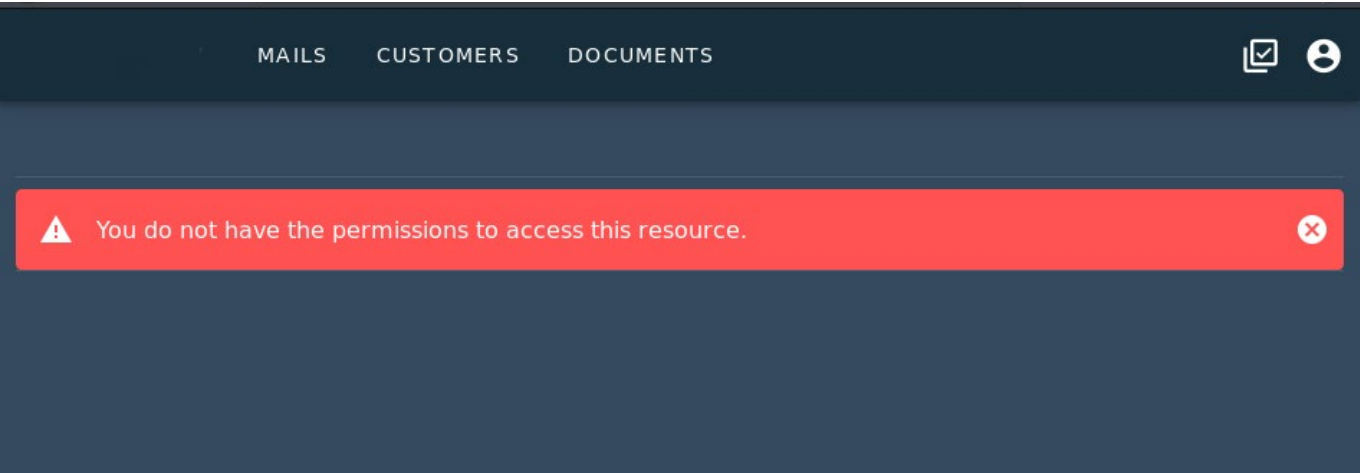
an HTTP POST request is sent to the application, which triggers the corresponding process.

Recommended measures

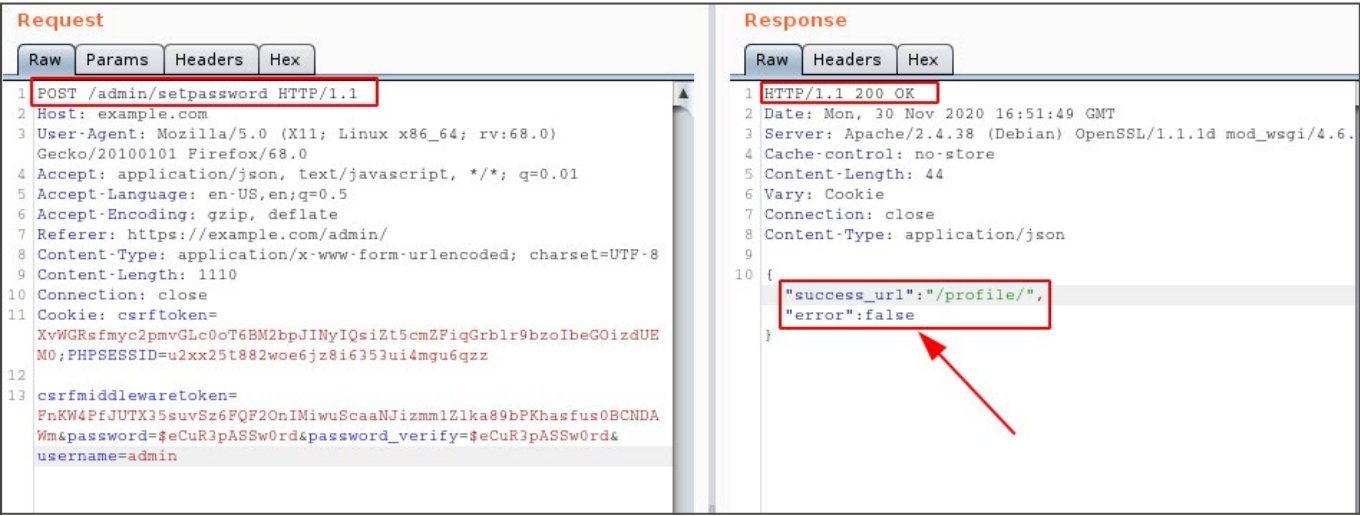
Client-side access control should never be used as the only safeguard against unauthorized access. As demonstrated above, a lack of representation within the application does not prevent an attacker from using the endpoint anyway. Only validation on the server side can prevent unauthorized use of an endpoint. This applies not only to web applications but also to local applications (thick clients).



Example



Access to the administration in the form of an HTTP GET request is prevented by the application



HTTP POST request to set a password with a non-privileged user is still executed successfully

Although a low-privileged user cannot see the administrative section of the application, he can send the request shown above to reset the password of any user. Information about the corresponding endpoint could be obtained by an attacker from internal sources, the application's JavaScript code, or simply by guessing.

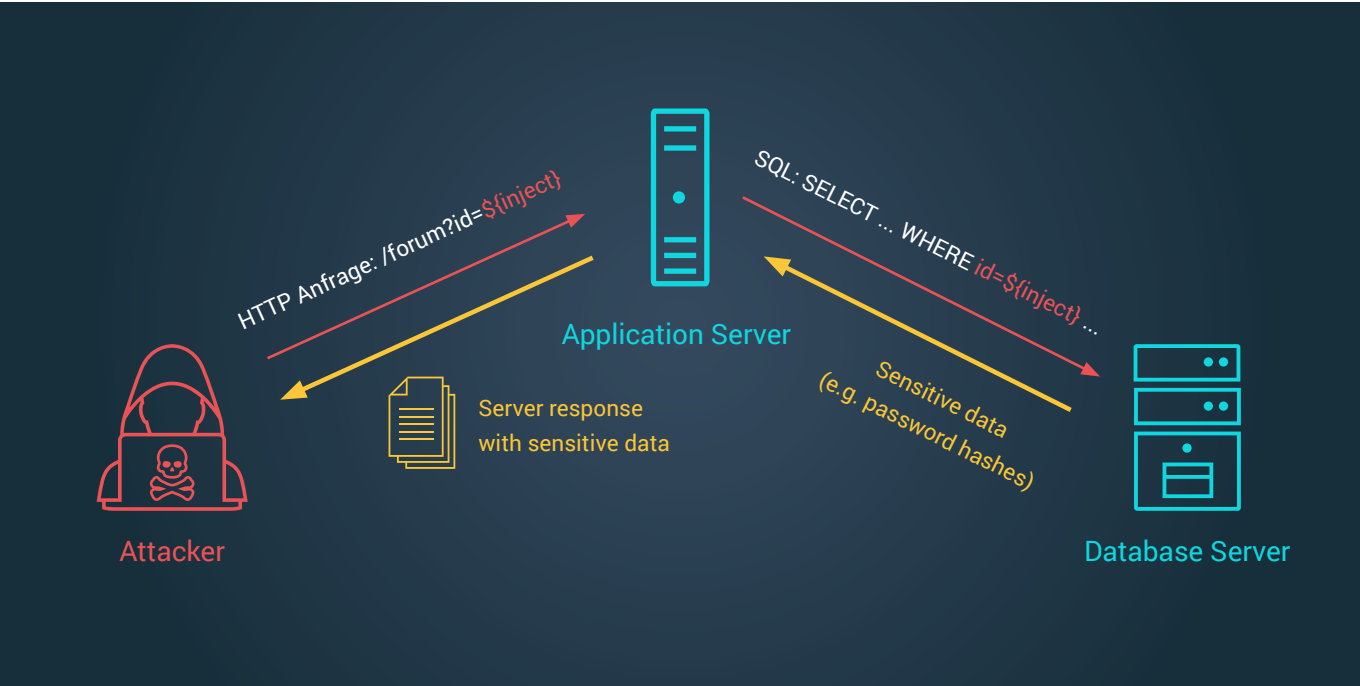
SQL INJECTION

SQL injection vulnerabilities allow an attacker to inject their own database commands into legitimate database queries. This can be used for various types of attacks. Usually, a successful attack allows full access to the application-relevant parts of the database. In many cases, it is then still possible to escalate permissions within the database or to access the server's file system. In the worst case, an SQL injection vulnerability also allows the execution of arbitrary operating system commands on the underlying server.

SQL injection vulnerabilities have always been one of the most common vulnerabilities within web applications. Despite the increased use of frameworks and rising awareness among developers, usd found SQL injection vulnerabilities within about one third of all penetration tests in 2020.

Recommended measures

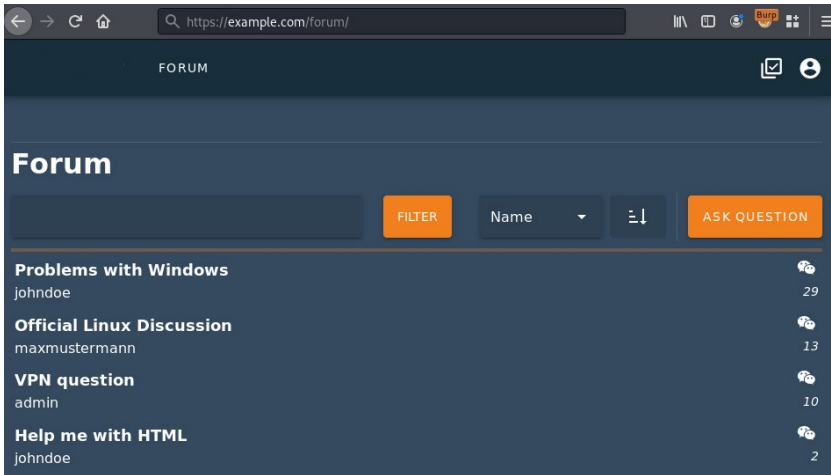
User-controlled input should always be considered potentially dangerous and should never be used within database queries without sufficient filtering and encoding. Appropriate functions for filtering input are available in all common programming languages. Furthermore, it is recommended to use „prepared statements“. With this technique, the structure of a database query is sent to the database in advance, before the data actually used for the query is inserted. The database server thus knows the structure of the query and subsequent modification by an attacker is no longer possible.



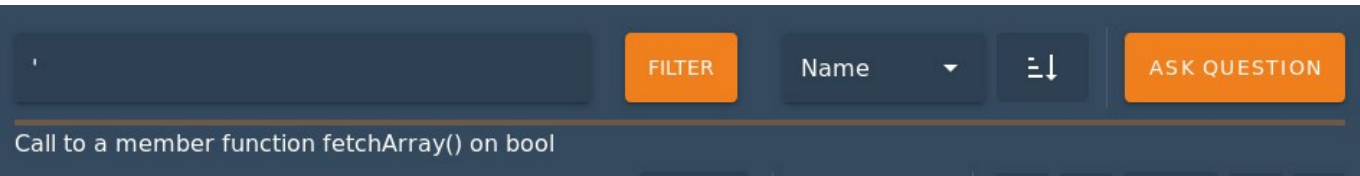
Example

The following demonstrates an SQL injection vulnerability that ultimately allows the attacker to access password hashes of registered users. The initial entry point is located inside a search for forum posts.

As can be seen on the right, forum posts can be searched for within the application using a title. The corresponding search term is transferred as HTTP GET parameter and is visible within the URL. Inserting special characters within the search query can provoke a database error, as the resulting database query has an invalid syntax.

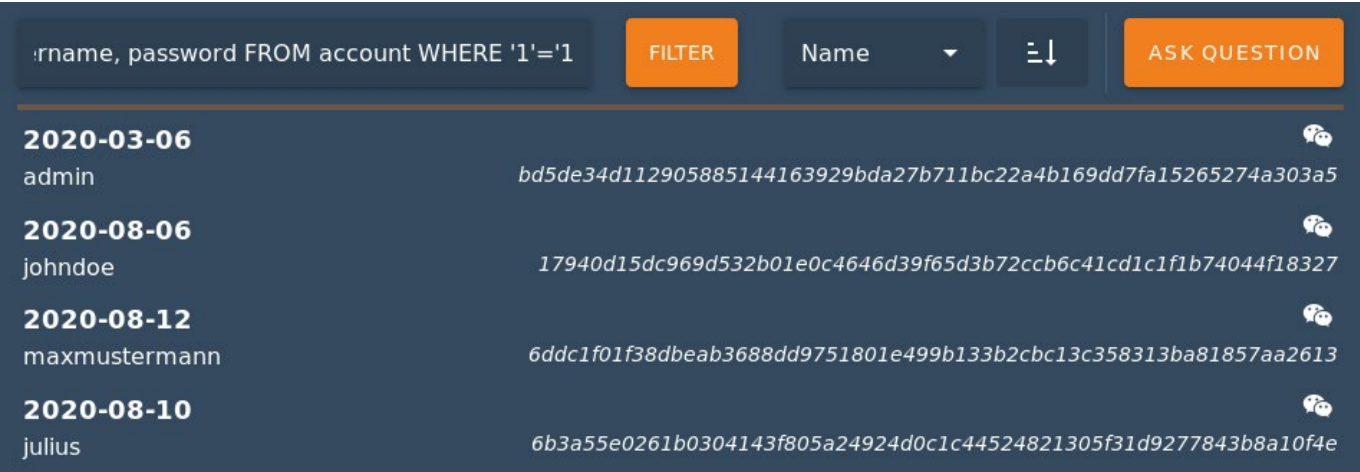


Application with a search for forum posts



Database error caused by special characters

Based on the caused database error, an attacker can now refine his attack. In the following request, the vulnerability is exploited to extract password hashes from the database:



Password hashes are extracted from the database

TRANSPORT LAYER SECURITY (TLS) 1.0

TLS is often used for authentication and encryption of Internet connections. TLS is a protocol that lies between TCP and the application and presentation layer protocols. The authenticity of the contacted server is guaranteed by a certificate and the connection between client and server is encrypted.

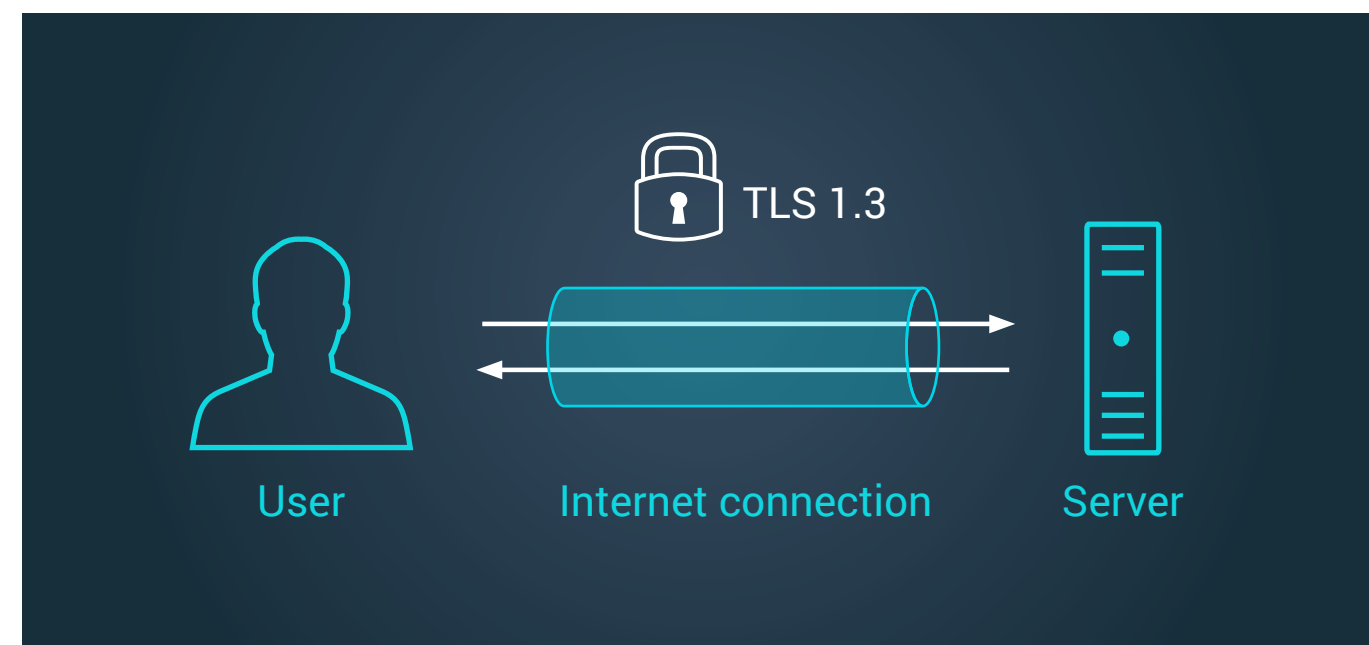
Transport Layer Security (TLS) is probably one of the most widely used encryption protocols for network communications. The encryption is separated from the actual application protocol, so that application programmers do not have to deal with the encryption layer. Only the configuration of TLS still requires manual setting and thus provides a lot of potential for vulnerabilities. Many systems still use the outdated version TLSv1.0, which has no longer been recognized as sufficiently secure by the PCI Council since 2016.

Vulnerabilities at the TLS level are not particularly noteworthy (for this short report, which lacks room for

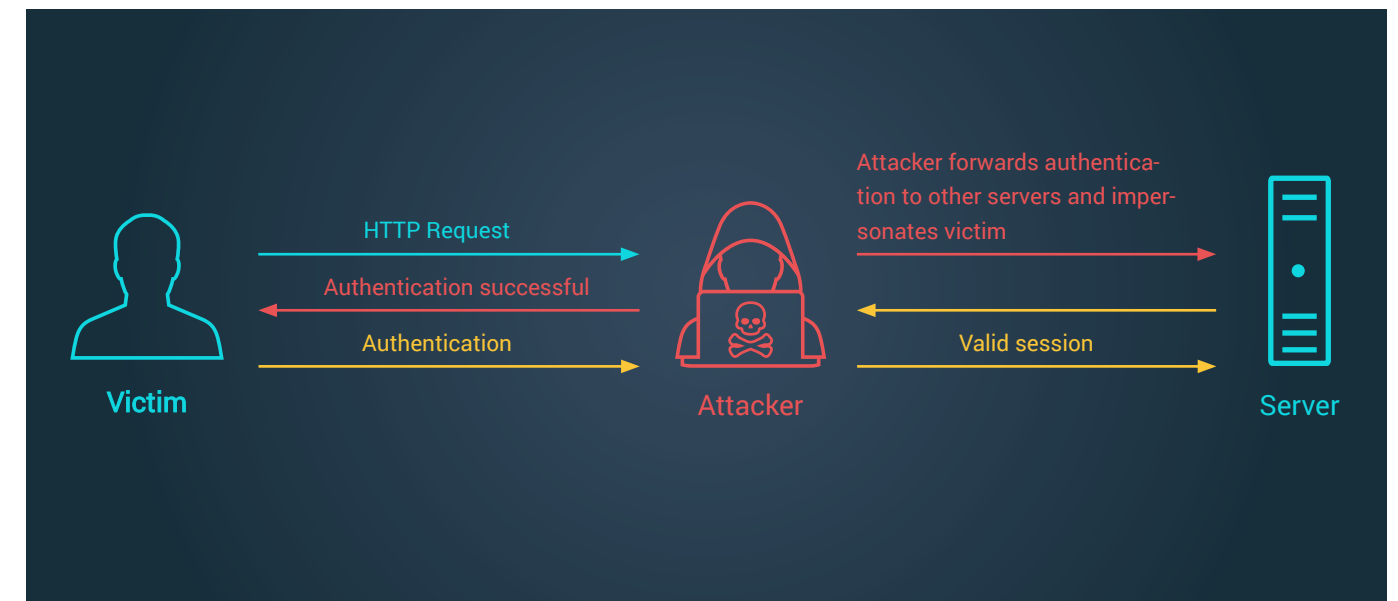
cryptographic details). The reason why this vulnerability category has nevertheless made it into our list is its outstanding frequency. In 2020, 96% of all tested systems were found to use the outdated TLS version TLSv1.0. A clear sign that vulnerabilities at the TLS level are still not taken seriously.

Recommended measures

TLS 1.0 is an outdated version of the TLS protocol with known vulnerabilities. Although concrete exploitation is difficult, there is still a security risk. In particular, PCI-relevant systems must no longer support TLS 1.0 in order to meet compliance guidelines.



SMB 1.0 & SMB SIGNING



The Server Message Block protocol (SMB) is a widespread network protocol that is mainly used to exchange files and print jobs. It plays a central role in Windows environments in particular, since remote procedure calls (RPC) can also be mapped via SMB in addition to the transfer of data.

One of the oldest versions of the SMB protocol is SMBv1, which has general security problems in addition to very well-known vulnerabilities such as EternalBlue or SMBLost. The latest version of the SMB protocol is SMBv3, which has significant security advantages over its predecessors.

SMB Signing is an additional mechanism to increase the security of the SMB protocol. Each SMB packet is signed by the sender and verified by the receiver. Attackers cannot modify signed SMB packets, which limits many attack vectors.

Within our annual statistics, both SMBv1 and missing SMB signing occurred with a frequency of about 40%.

This is a surprisingly high result considering that the statistics also include application-only tests and pen-tests on non-Windows based environments. The consequences of a successful exploit can be critical.

Recommended measures

The SMBv1 protocol is heavily outdated and all modern devices support the use of newer SMB versions as well as SMB signing. So as long as legacy systems do not play a role, the use of SMBv1 should be avoided and SMB signing should be enforced by servers. In legacy environments, care should be taken to ensure a sufficient patch level of the corresponding systems, and external access to the corresponding networks should be well secured.



TOP 3 ZERO-DAYS

Time and again, security analysts at usd HeroLabs identify previously unknown vulnerabilities in products as part of their work. For these zero-day vulnerabilities, no security patches (updates to fix the security gaps) exist at the time of their discovery. Handling vulnerabilities responsibly is a top priority for us. In accordance with our Responsible Disclosure Policy, we therefore inform manufacturers about vulnerabilities we have identified in standard products and publish them responsibly in the form of „Security Advisories“ after the software manufacturer has provided an update. In 2020, we published a total of 43 Security Advisories - here is our top three.

TOP 3 ZERO-DAYS

usd-2020-0060 (CVE-2020-15861) | Net-SNMP

Advisory ID:	usd-2020-0060	Timeline:	
CVE Number:	CVE-2020-15861		
Affected Product:	Net-SNMP		
Affected Version:	5.7.3		
Vulnerability Type:	Elevation of Privileges		
Security Risk:	High		
Vendor URL:	http://www.net-snmp.org		
Vendor Status:	Fixed		
			<i>2020-07-16</i> First contact request via Github
			<i>2020-07-16</i> Net-SNMP v5.8 is released and fixes the vulnerability
			<i>2020-09-29</i> Security advisory released

Description:

On Debian based systems, the NET-SNMP daemon runs as a low privileged user account. However, in combination with the snmp-mibs-downloader package this protection can be bypassed and it is possible for this account to elevate permissions to the root user.

The Simple Network Management Protocol (SNMP) is a widely used network protocol for controlling and monitoring network devices. Since the corresponding service (SNMP daemon) needs access to a lot of system components and (per default) binds the network port 161, it usually runs as the root user. On Debian based systems, the default installation of SNMP sets up a dedicated low privileged user account (Debian-snmp), that is used to run the SNMP daemon. This adds an additional layer of security, as a compromise of the SNMP service does not directly allow root access to the targeted device.



More details and further security advisories can be found here: <https://herolab.usd.de/security-advisories>

usd-2020-0002 (CVE-2020-6581) | Nagios NRPE

Advisory ID:	usd-2020-0002	Timeline:	
CVE Number:	CVE-2020-6581		
Affected Product:	Nagios NRPE		<i>2020-01-06</i> Found by manual code review of Nagios NRPE
Affected Version:	v.3.2.1		<i>2020-01-08</i> Initial Contact
Vulnerability Type:	Insufficient Filtering of Configuration file		<i>2020-01-15</i> Nagios NRPE v4.0.0 is released
Security Risk:	Medium		<i>2020-03-04</i> security advisory released
Vendor URL:	https://www.nagios.org		
Vendor Status:	Fixed in v.4.0.0 (not verified)		

Description:

Insufficient Filtering and incorrect parsing of the configuration file may lead to command injection.

usd-2020-0016 (CVE-2020-5836) | Symantec Endpoint Protection

Advisory ID:	usd-2020-0016	Timeline:	
CVE Number:	CVE-2020-5836		
Affected Product:	Symantec Endpoint Protection		<i>2020-03-12</i> First contact request via symantec.psirt@broadcom.com
Affected Version:	14.2.2.1		<i>2020-05-05</i> Fix is released in Symantec Endpoint Protection 14.3
Vulnerability Type:	Hardlink Vulnerability		<i>2020-05-11</i> Broadcom publishes Advisory
Security Risk:	Critical		<i>2020-06-18</i> Security advisory released
Vendor URL:	https://www.broadcom.com		
Vendor Status:	Fixed		
Vendor Advisory:	https://support.broadcom.com/security-advisory/content/security-advisories/Symantec-Endpoint-Protection-Security-Update/SYMSA1762		

Description :

Hardlink attacks become more and more popular on Windows operating systems. A hardlink is just a directory entry that points to an already existing file and redirects certain file operations to the actual target. When privileged processes interact with user controlled parts of the file system, hardlinks can be used to redirect privileged file operations in order to achieve an elevation of privileges. In the most recent versions of Windows, mitigations against hardlink attacks have been implemented. These require write access to the targeted file during link creation and protect from attacks like demonstrated in the following. However, unpatched systems are still vulnerable to this type of attack.



usd HeroLab Toolchain

Since there is no recognized standard for pentests on the market, the performance of different providers can vary considerably. This makes it difficult to compare results. Furthermore, due to insufficient documentation, identified vulnerabilities are often not verifiable. For this reason, we invested heavily in our usd HeroLab toolchain in 2020. This provides our security analysts with a range of self-developed, automated tools that bundle their know-how and are unique on the market. The toolchain allows our security analysts to perform their assessments even more efficiently and comprehensively. They are left with time for targeted, manual analyses in which they can take customer-specific requirements into account. In this way, we guarantee a structured, traceable and individualized review of systems and applications.

usd HeroLab Toolchain

An insight into our most important tools.

usd ExPeRT

The heart of the usd toolchain.
This is where all information comes together.

- Project Planning & Cooperation
- Mapping of long-established internal processes
- Integrated checklists
- Status tracking & resource management
- Synchronization of data, e.g. with usd Icebreaker
- Final report & test report
- Export option of results

usd Cooperator

Burp Suite plugin to further improve web app pentests

- Assists in documenting vulnerabilities in web application pentests
- Collaborative working throughout the pentesting process
 - Real-time exchange of results
 - Assignment of subtasks to individual coworkers
 - Transparent management of project details

usd ICEBREAKER

Analysis tool for all team members & knowledge base

- Implementation of plugins & self-developed scripts
- Analysis tool for all team members & knowledge database
- Continuous development and adaptation to research results
- High automation of manual processes
- „Scanner“/ Automation Engine
- Combines the best publicly available and usd internally developed tools
- Consistently high quality level
- Automated documentation of test data

EXPERT VOICES

„ We protect companies against hackers and criminals. We ensure this with a broad range of support services, which includes sharing knowledge with the community. Because more security can only be achieved together.“

Stephan Neumann

Co-Head of usd HeroLab

„ People still underestimate existing risks far too often. A pentest uncovers gateways into your IT systems so that you can eliminate them. We help you reduce your risks“

Tobias Neitzel

usd Product Manager Pentest

WE SUPPORT YOU

YOUR CHALLENGES

Whether you are a company that wants to use applications and systems securely or a product manufacturer that wants to offer verifiably greater security

to your customers: With usd HeroLab you have one of the leading partners at your side.



- Do you have critical applications?
- Do you have complex environments?
- Are you bound by compliance requirements?
- Do you store or process sensitive data?

OUR SOLUTIONS

The goal of our security analyses is to identify vulnerabilities, point out resulting risks and show ways to improve your security. Together with you, we find the right solution for you and your goals, environments and potential risks.

Our tool-based reproducible testing process guarantees you efficiency, transparency and the highest quality. Our performance promise is more security!

- **Detect vulnerabilities in your IT infrastructure in time**
Pentests, Security Scans, Code Reviews, Cloud Security
- **Handle security vulnerabilities with structure**
Vulnerability Management, Managed Security Services & Remediation Handling
- **Be optimally prepared in case of emergency**
Incident Response and Digital Forensics
- **Benefit from the know-how of an entire community**
Bug Bounty Programs

usd AG

Frankfurter Str. 233, Forum C1
63263 Neu-Isenburg
Germany

Phone: +49 6102 8631-0
Email: contact@usd.de

www.usd.de | herolab.usd.de