usd HeroLab

JAHRES-BERICHT 2020

Wir schützen Unternehmen vor Hackern und Kriminellen.

more security. usd

INHALT

RISIKEN ERKENNEN. ZIELGERICHTET HANDELN. IT-SICHERHEITSNIVEAU ERHÖHEN.

“Auch 2020 haben wir während unserer Pentests wieder unzählige Schwachstellen in IT-Systemen und Applikationen unserer Kunden identifiziert. Daher ist es essentiell, dass Sie Ihre Risiken kennen und den Angreifern einen Schritt voraus sind.”

Matthias Göhring
Co-Head of usd HeroLab

Die Bedrohungslage durch Hackerangriffe verschärft sich unentwegt: IT-Infrastrukturen werden immer komplexer und Angreifer gehen zunehmend raffiniert und methodisch vor.

Unsere Mission „more security“ treibt uns an, stets den Überblick über das aktuelle und zukünftige IT-Sicherheitsumfeld zu wahren. Dazu gehört auch, dass die Security Analysten unseres usd HeroLabs fortwährend die Risiken unserer Kunden im Blick haben und wissen, welchen Bedrohungen Unternehmen ausgesetzt sind. Insbesondere das Krisenjahr 2020 bildete ideale Rahmenbedingungen für Cyberkriminelle: Große Unsicherheiten, beschleunigte Digitalisierung, Einsparungen und höhere Netzaktivitäten. So stiegen laut aktuellem Lagebericht des Bundesamts für Sicherheit in der Informationstechnik (BSI)[1] die Meldungen gestohlener hochsensibler personenbezogener Daten stark an. Dies zeigt, dass IT-Sicherheit gerade jetzt unabdingbar ist.

Wir sind der Überzeugung, dass technische Sicherheitsanalysen, wie sie heute in weiten Teilen durchgeführt werden, der aktuellen Bedrohungslage und den Anforderungen am Markt nicht mehr gerecht werden. Aus diesem Grund investieren wir fortlaufend in die Entwicklung der usd HeroLab Toolchain für mehr Effizienz, Transparenz und höhere Qualität. Gleichzeitig ist eine strukturierte und effiziente Einarbeitung unserer Spezialisten erforderlich, um auf gleichbleibend hohem Niveau arbeiten zu können. 2020 haben wir unser umfangreiches internes Ausbildungsprogramm, den „usd HeroLab Certified Professional“ (UCP), weiterentwickelt. Aber auch unser externes Engagement stand in diesem Jahr im Fokus. Im Rahmen unserer universitären Kooperationen haben wir IT Security in unseren „Hacker Contests“ praxisnah qualifizierten Nachwuchskräften vermittelt. Unsere zahlreichen Seminare der CST Academy haben Austausch und Wissenstransfer mit der Security Community gefördert. Gemeinsam mit anderen für mehr Sicherheit. So starten wir auch ins neue Jahr.

[1] „Die Lage der IT-Sicherheit in Deutschland 2020“, Bundesamt für Sicherheit in der Informationstechnik

TOP 5 AUFFÄLLIGSTE SCHWACHSTELLEN

Unsere Security Analysten decken immer wieder Einfallstore in Systemen und Anwendungen auf, welche erhebliche Risiken für die Unternehmenssicherheit darstellen. Einige Schwachstellen treten vermehrt in diversen IT-Systemen auf. Wir haben für Sie die fünf auffälligsten Schwachstellen zusammengetragen und übersichtlich aufbereitet – wie geht der Hacker vor? Welche Folgen resultieren für Ihr Unternehmen? Wie können Sie sich besser schützen?

Nachfolgend sprechen wir allgemeingültige Maßnahmenempfehlungen aus.
Gern unterstützen wir Sie mit individuellen Lösungen.

CROSS-SITE-SCRIPTING (XSS)

Als Cross-Site-Scripting bezeichnet man eine Kategorie von Schwachstellen, die es einem Angreifer erlauben, schadhaften JavaScript-Code in die Antworten eines Webservers einzuschleusen.

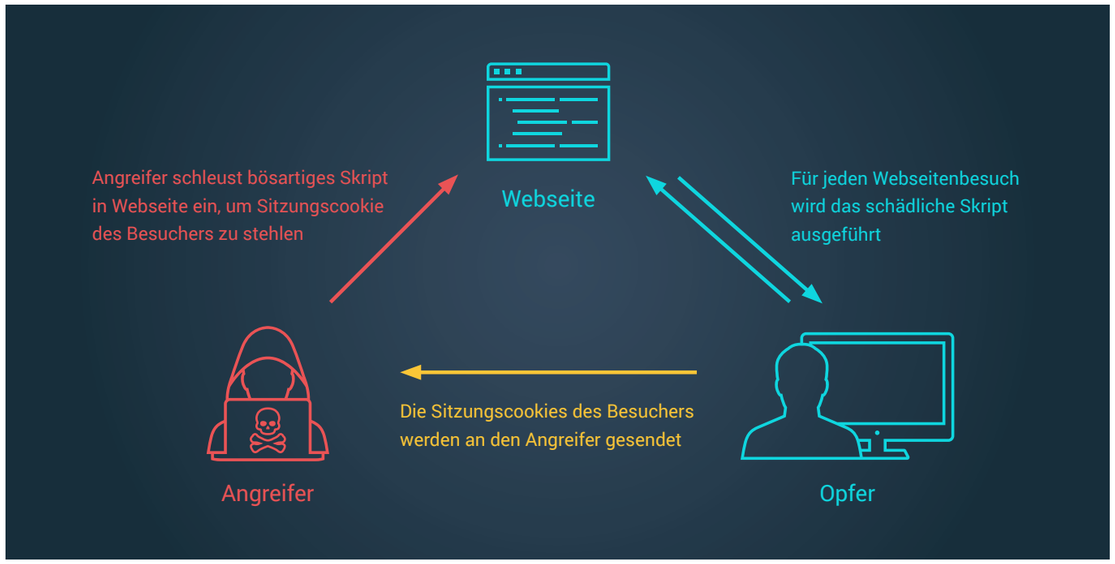
Der Webbrowser anderer Benutzer kann den vom Angreifer eingefügten JavaScript-Code dann nicht von dem legitimen Code der Anwendung unterscheiden und führt schadhafte Scripts entsprechend aus. Dies führt in der Regel dazu, dass der Angreifer die aktuelle Sitzung seines Opfers komplett übernehmen kann.

Die Tatsache, dass Cross-Site-Scripting in einer Statistik zu häufigen Schwachstellen auftaucht, ist nicht wirklich überaschend. Dennoch ist es interessant zu sehen, dass trotz zunehmendem Einsatz von Frameworks und steigender Awareness bei Softwareentwicklern noch immer mehr als zwei Drittel der von uns getesteten Webapplikationen eine solche Verwundbarkeit aufwiesen.

Cross-Site-Scripting Schwachstellen werden von der usd in der Regel als kritisches Sicherheitsrisiko eingestuft, da die Vertraulichkeit und Integrität von Nutzerdaten akut gefährdet wird.

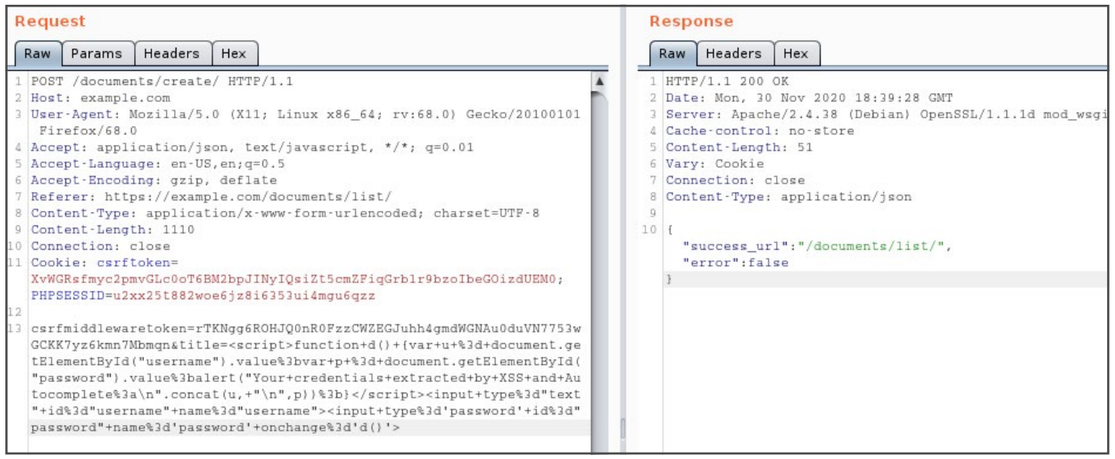
Maßnahmenempfehlung

Vom Benutzer kontrollierte Eingaben sollten immer als potentiell gefährlich betrachtet und niemals ohne ausreichende Filterung und Enkodierung innerhalb von Serverantworten eingebettet werden. Entsprechende Funktionen zu Filtering und Enkodierung von Eingaben stehen dabei in allen gängigen Programmiersprachen zur Verfügung. Der korrekte Einsatz von Frameworks sowie regelmäßige Schulungen von Entwicklern sind wichtige Maßnahmen, um Cross-Site-Scripting-Schwachstellen zu verhindern.

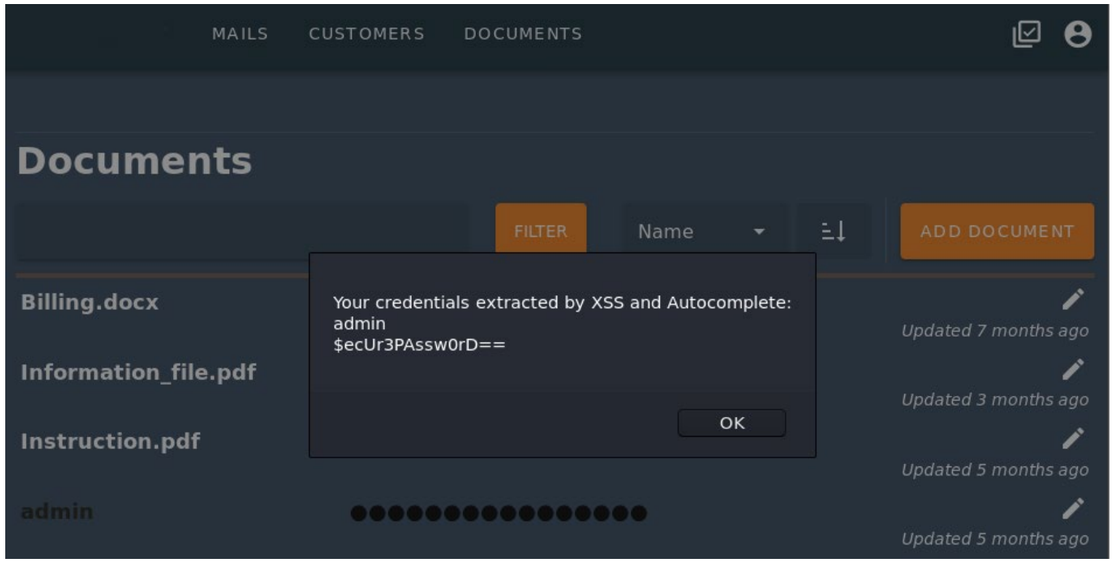


Beispiel

Wir demonstrieren einen Angriff, bei dem im Webbrowser gespeicherte Zugangsdaten vom Angreifer über JavaScript ausgelesen werden.



Angreifer platziert schadhaften JavaScript-Code innerhalb einer verwundbaren Anwendung



Ein Benutzer besucht die verwundbare Seite – Seine Zugangsdaten werden extrahiert

Während die Zugangsdaten des Opfers zur besseren Sichtbarkeit dargestellt wurden, würde ein echter Angriff ohne für das Opfer sichtbare Spuren ablaufen. Die Zugangsdaten würden dann nicht dargestellt, sondern über das Netzwerk an einen vom Angreifer kontrollierten Server gesendet werden.

BROKEN ACCESS CONTROL

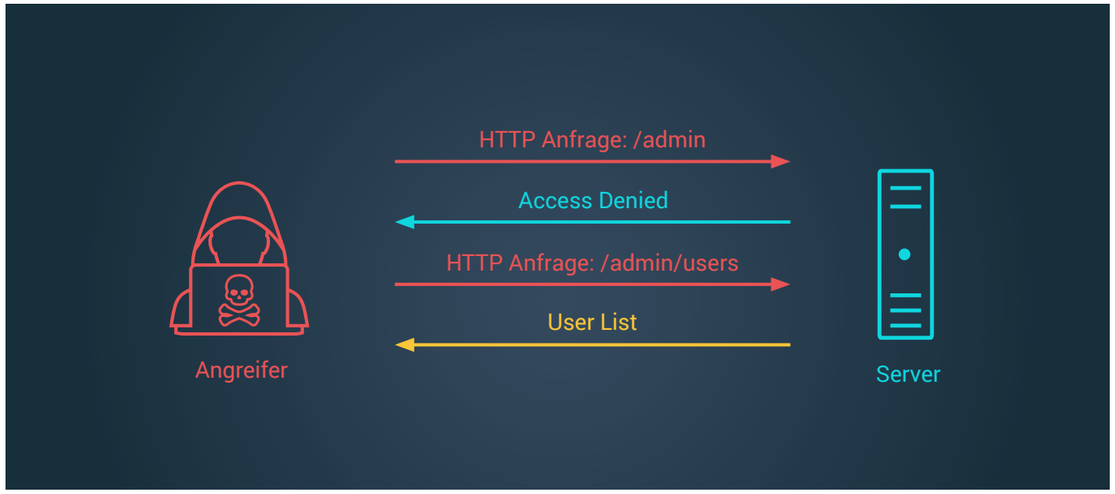
Als Broken Access Control bezeichnet man Schwachstellen, bei der Endpunkte oder Funktionalitäten in einer Anwendung nicht ausreichend durch Authentifizierungs- und Autorisierungsmechanismen geschützt sind. Angreifer können diese Endpunkte aufrufen, bzw. entsprechende Funktionalitäten verwenden, ohne dafür ausreichende Berechtigungen zu besitzen.

Mit einer Häufigkeit von 52% trat diese Schwachstelle in Jahr 2020 in mehr als jedem zweiten Pentest auf. Einer der häufigsten Gründe hierfür besteht darin, dass lediglich eine clientseitige Validierung von Anfragen verwendet wird, während auf Serverseite keine weitere Prüfung erfolgt. Im Folgenden wird dies an einer Beispielapplikation demonstriert, welche Benutzeranfragen nicht serverseitig validiert.

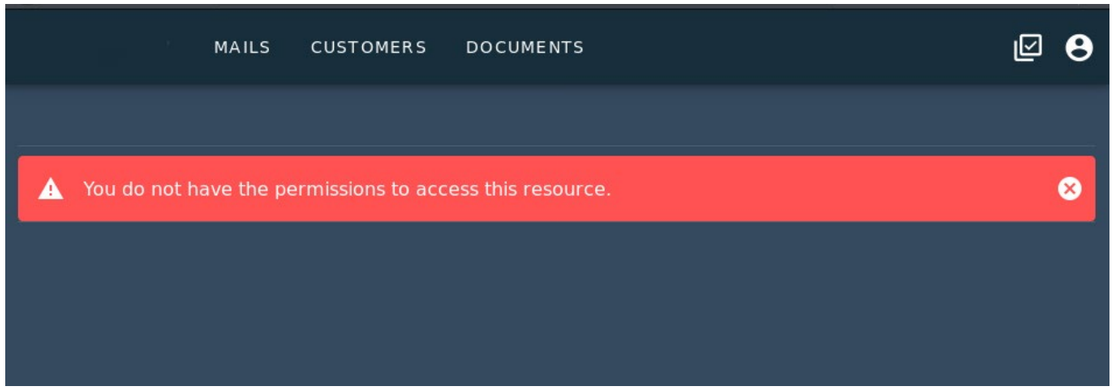
Innerhalb des administrativen Bereichs der Anwendung haben Administratoren die Möglichkeit, Passwörter von Benutzern zu setzen. Wird die entsprechende Aktion ausgeführt, wird eine HTTP-POST-Anfrage an die Anwendung gesendet, welche den entsprechenden Prozess anstößt.

Maßnahmenempfehlung

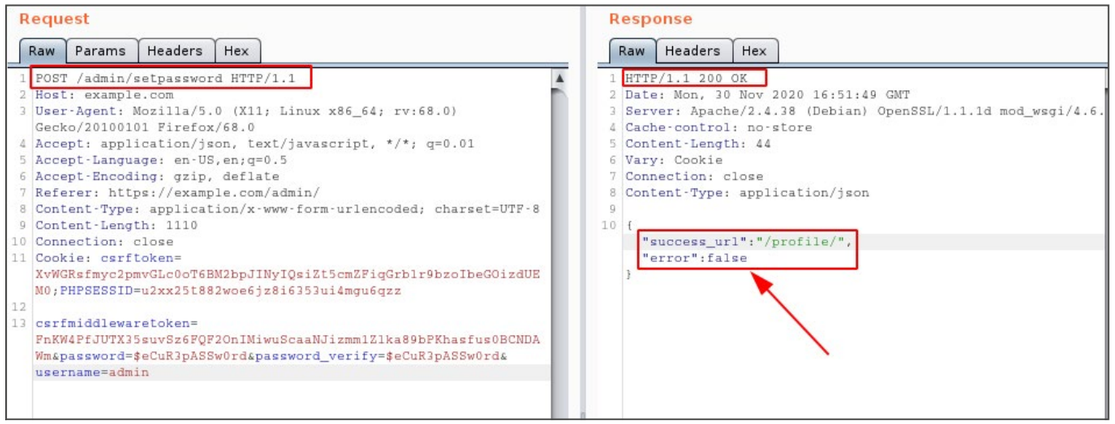
Clientseitige Zugriffskontrolle sollte nie als einzige Schutzvorkehrung gegen unbefugten Zugriff verwendet werden. Wie oben demonstriert verhindert eine fehlende Darstellung innerhalb der Applikation nicht, dass ein Angreifer den Endpunkt dennoch verwendet. Nur eine Validierung auf Serverseite kann die unbefugte Nutzung eines Endpunktes verhindern. Dies gilt neben Webanwendungen insbesondere auch für lokale Anwendungen (Thick Clients).



Beispiel



Der Zugriff auf die Administration in Form einer HTTP-GET-Anfrage wird durch die Anwendung unterbunden



HTTP-POST-Anfrage zum Setzen eines Passworts mit einem nicht privilegierten Nutzer wird dennoch erfolgreich ausgeführt

Obwohl ein niedrig privilegierter Benutzer den administrativen Bereich der Anwendung nicht angezeigt bekommt, kann er die oben gezeigte Anfrage absenden, um das Passwort eines beliebigen Benutzers zurückzusetzen. Informationen über den entsprechenden Endpunkt könnte ein Angreifer aus internen Quellen, dem JavaScript -Code der Anwendung oder durch einfaches Raten erhalten.

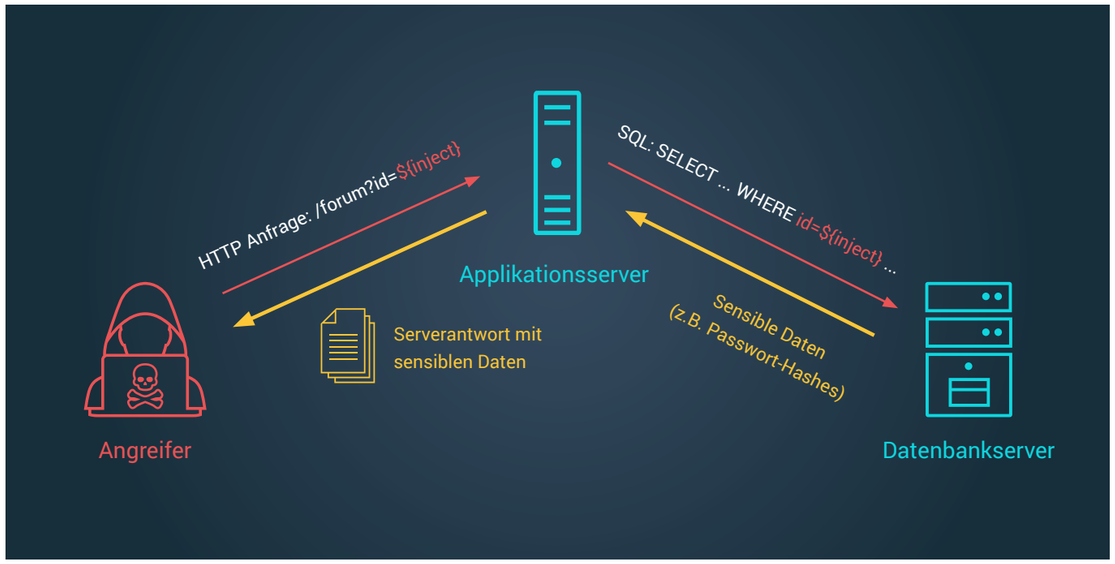
SQL INJECTION

SQL-Injection-Schwachstellen erlauben es einem Angreifer, eigene Datenbankkommandos in legitimen Datenbankanfragen einzuschleusen. Dies kann für verschiedene Arten von Angriffen genutzt werden. In der Regel erlaubt ein erfolgreicher Angriff Vollzugriff auf die applikationsrelevanten Teile der Datenbank. Oftmals ist es dann weiterhin noch möglich, Berechtigungen innerhalb der Datenbank zu eskalieren oder auf das Dateisystem des Servers zuzugreifen. Im schlimmsten Fall erlaubt eine SQL Injection-Schwachstelle auch das Ausführen von beliebigen Betriebssystemkommandos auf dem unterliegenden Server.

SQL-Injection-Schwachstellen sind seit jeher eine der am häufigsten auftretenden Schwachstellen innerhalb von Webapplikationen. Trotz der vermehrten Verwendung von Frameworks und steigender Awareness bei Entwicklern fand die usd im Jahr 2020 SQL-Injection-Schwachstellen innerhalb von rund einem Drittel aller Penetrationstests.

Maßnahmenempfehlung

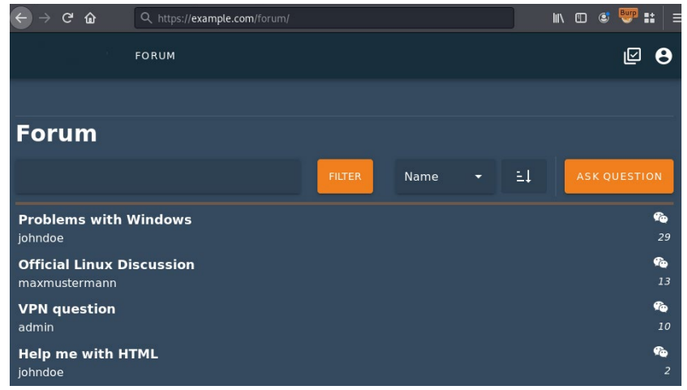
Vom Benutzer kontrollierte Eingaben sollten immer als potentiell gefährlich betrachtet und niemals ohne ausreichende Filterung und Enkodierung innerhalb von Datenbankanfragen verwendet werden. Entsprechende Funktionen zum Filtering von Eingaben stehen dabei in allen gängigen Programmiersprachen zur Verfügung. Weiterhin wird empfohlen, so genannte *prepared statements* zu verwenden. Mit dieser Technik wird die Struktur einer Datenbankanfrage schon vorab an die Datenbank gesendet, bevor die eigentlich für die Anfrage verwendeten Daten eingefügt werden. Der Datenbankserver kennt somit die Struktur der Anfrage und eine nachträgliche Veränderung durch einen Angreifer ist nicht mehr möglich.



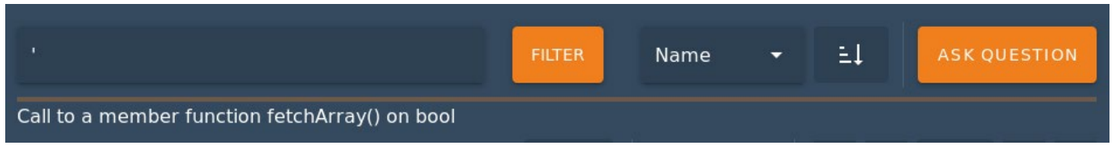
Beispiel

Im Folgenden wird eine SQL-Injection-Schwachstelle demonstriert, die dem Angreifer letztlich Zugriff auf Passwort-Hashes von registrierten Benutzern ermöglicht. Dabei befindet sich der initiale Eintrittspunkt innerhalb einer Suche für Foren-Beiträge.

Wie rechts zu sehen ist, lassen sich innerhalb der Applikation Forenbeiträge anhand eines Titels suchen. Der entsprechende Suchbegriff wird dabei als HTTP GET Parameter übertragen und ist innerhalb der URL sichtbar. Durch das Einfügen von Sonderzeichen innerhalb der Suchanfrage kann ein Datenbankfehler provoziert werden, da die resultierende Datenbankanfrage eine ungültige Syntax besitzt.

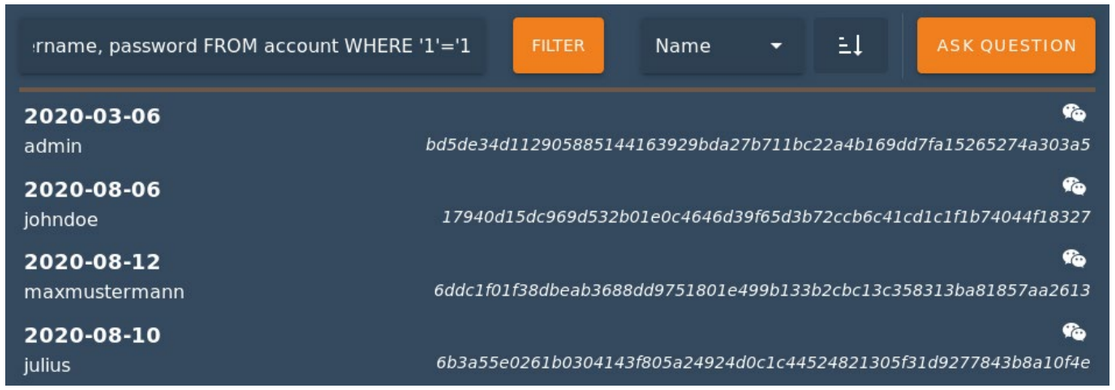


Applikation mit einer Suche für Foren-Beiträge



Von Sonderzeichen hervorgerufener Datenbankfehler

Basierend auf dem hervorgerufenen Datenbankfehler kann ein Angreifer nun seinen Angriff verfeinern. In der folgenden Anfrage wird die Schwachstelle ausgenutzt, um Passwort-Hashes aus der Datenbank zu extrahieren:



Passwort-Hashes werden aus der Datenbank extrahiert

TRANSPORT LAYER SECURITY (TLS) 1.0

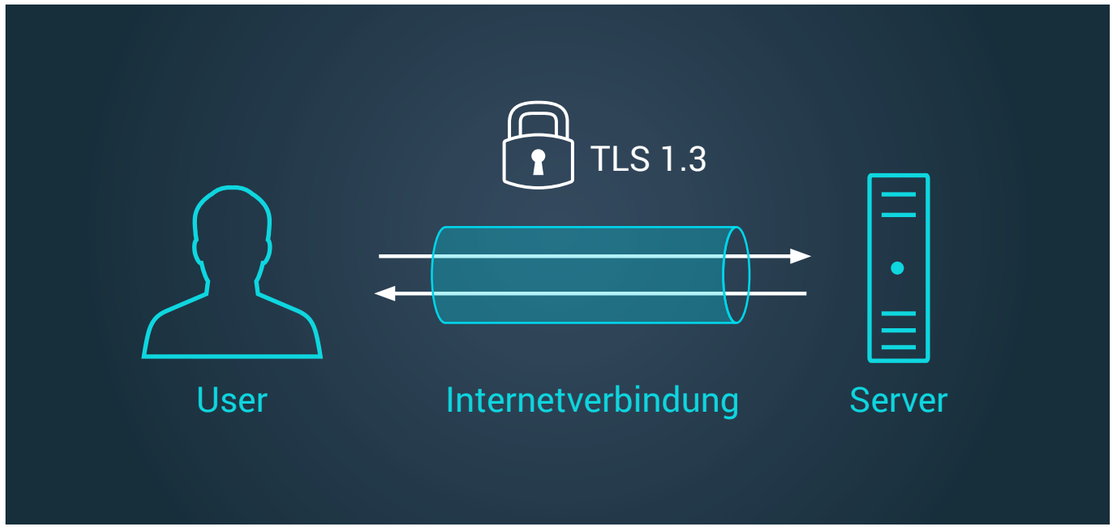
Zur Authentifizierung und Verschlüsselung von Internetverbindungen wird häufig TLS eingesetzt. TLS ist ein Protokoll, das zwischen TCP und den Protokollen der Anwendungs- und Darstellungsebene liegt. Die Authentizität des kontaktierten Servers wird durch ein Zertifikat garantiert und die Verbindung zwischen Client und Server verschlüsselt.

Transport Layer Security (TLS) ist vermutlich eines der am meisten verbreiteten Verschlüsselungsprotokolle für Netzwerkkommunikation. Die Verschlüsselung wird hierbei von dem eigentlichen Applikationsprotokoll separiert, sodass Anwendungsprogrammierer sich mit der Verschlüsselungsebene nicht auseinandersetzen müssen. Nur die Konfiguration von TLS bedarf noch manueller Einstellung und liefert somit viel Potential für Schwachstellen. So verwenden viele Systeme noch die veraltete Version TLSv1.0, die bereits seit 2016 nicht mehr vom PCI Council als ausreichend sicher anerkannt wird.

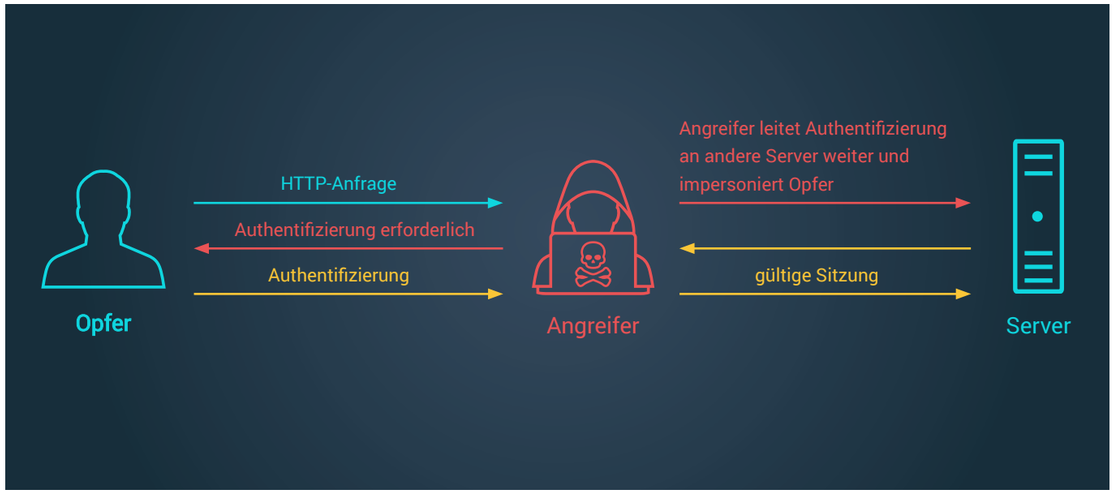
Schwachstellen auf TLS-Ebene sind (für dieses begrenzte Format, in dem keine kryptographischen Details Platz finden) nicht besonders nennenswert. Der Grund dafür, dass es diese Schwachstellen-Kategorie dennoch in unsere Auflistung geschafft hat, ist die herausragende Häufigkeit. So wurden im Jahr 2020 bei 96% aller getesteten Systeme die veraltete TLS -Version TLSv1.0 identifiziert. Ein klares Zeichen, dass Schwachstellen auf TLS-Ebene weiterhin nicht ernst genommen werden.

Maßnahmenempfehlung

TLS 1.0 ist eine veraltete Version des TLS-Protokolls mit bekannten Schwachstellen. Eine konkrete Ausnutzung ist zwar schwierig, dennoch ist ein Sicherheitsrisiko gegeben. Insbesondere PCI-relevante Systeme dürfen kein TLS 1.0 mehr unterstützen, um Compliance-Richtlinien zu erfüllen.



SMB 1.0 & SMB SIGNING



Das Server Message Block Protokoll (SMB) ist ein weit verbreitetes Netzwerkprotokoll, das hauptsächlich zum Austausch von Dateien und Druckaufträgen verwendet wird. Insbesondere in Windows-Umgebungen spielt es eine zentrale Rolle, da neben dem Transfer von Daten auch Remote Procedure Calls (RPC) über SMB abgebildet werden können.

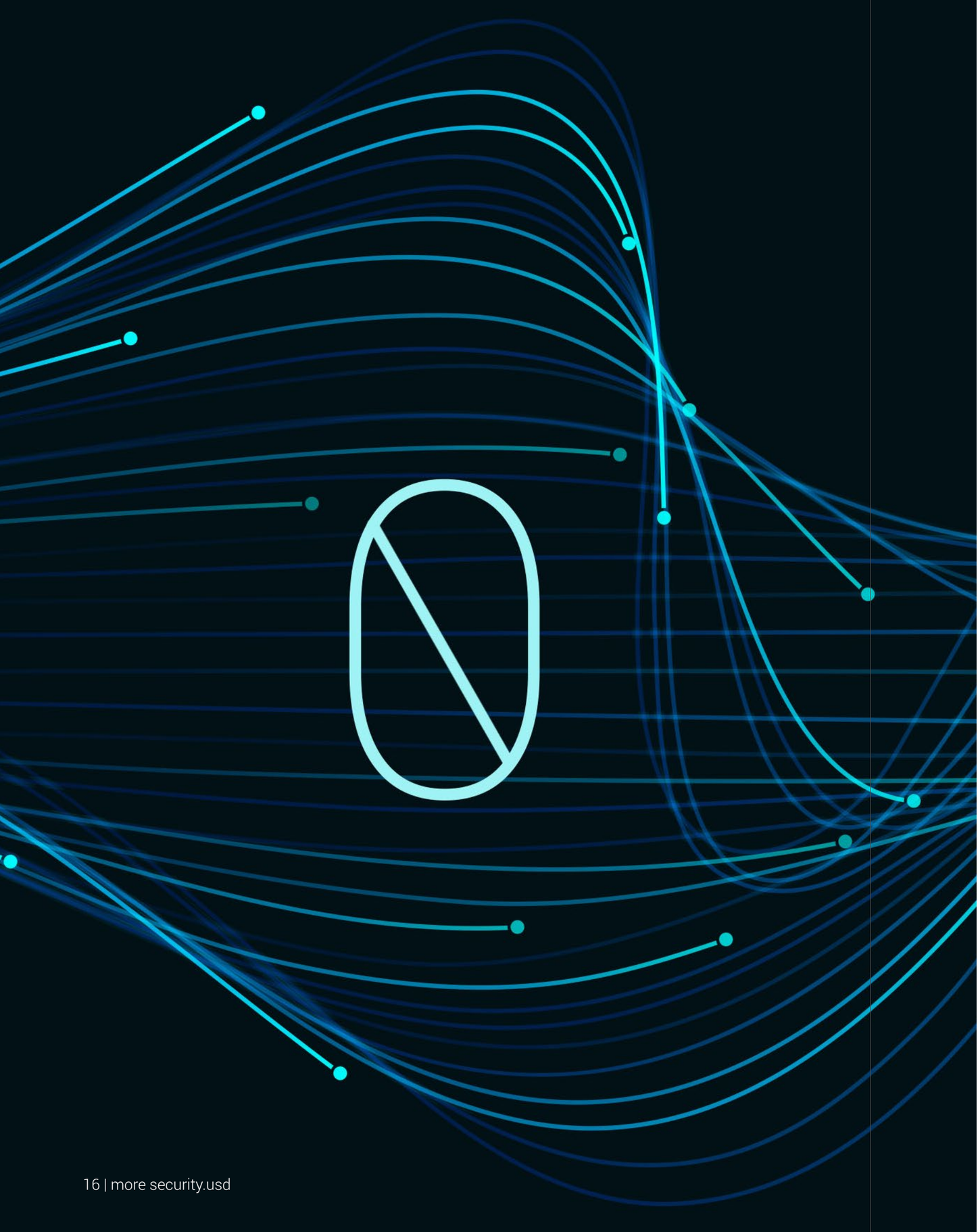
Eine der ältesten Versionen des SMB-Protokolls ist SMBv1, das neben sehr bekannten Schwachstellen wie EternalBlue oder SMBLost generelle Sicherheitsprobleme mit sich bringt. Die aktuell neuste Version des SMB-Protokolls ist SMBv3, das erhebliche Sicherheitsvorteile gegenüber seinen Vorgängerversionen besitzt.

SMB Signing ist ein zusätzlicher Mechanismus, um die Sicherheit des SMB-Protokolls zu erhöhen. Dabei wird jedes SMB Paket vom Sender signiert und vom Empfänger entsprechend verifiziert. Angreifer können signierte SMB Pakete nicht verändern, wodurch viele Angriffsvektoren eingeschränkt werden.

Innerhalb unserer Jahres-Statistik kamen sowohl SMBv1 als auch fehlendes SMB Signing mit einer Häufigkeit von ungefähr 40% vor. Ein erstaunlich hohes Resultat, wenn man bedenkt, dass die Statistik auch reine Applikationstests und Pentests auf nicht Windows-basierten Umgebungen enthält. Dabei können die Folgen einer erfolgreichen Ausnutzung kritisch sein.

Maßnahmenempfehlung

Das SMBv1 Protokoll ist stark veraltet und alle modernen Geräte unterstützen die Verwendung von neueren SMB-Versionen sowie SMB Signing. Solange Legacy-Systeme also keine Rolle spielen, sollte die Verwendung von SMBv1 vermieden und SMB Signing von Servern erzwungen werden. In Legacy-Umgebungen sollte auf ein ausreichendes Patch Level der entsprechenden Systeme geachtet werden und externer Zugriff zu den entsprechenden Netzwerken gut abgesichert sein.



TOP 3 ZERO-DAYS

Immer wieder identifizieren Security Analysten des usd HeroLab im Rahmen ihrer Arbeit bis dato unbekannte Schwachstellen in Produkten. Für diese sogenannten Zero-Day-Schwachstellen existieren bislang keine Sicherheitspatches (Updates zur Behebung der Sicherheitslücken). Der verantwortungsvolle Umgang mit gefundenen Schwachstellen hat für uns oberste Priorität. Gemäß unserer Responsible Disclosure Policy informieren wir deshalb Hersteller über von uns identifizierte Schwachstellen in Standardprodukten und veröffentlichen diese in Form von „Security Advisories“ verantwortungsvoll, nachdem der Softwarehersteller ein Update bereitgestellt hat. Im Jahr 2020 haben wir insgesamt 43 Security Advisories veröffentlicht – unsere Top Drei stellen wir Ihnen vor.

TOP 3 ZERO-DAYS

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

Advisory ID: usd-2020-0060
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

Timeline:
2020-07-16 First contact request via Github
2020-07-16 Net-SNMP v5.8 is released and fixes the vulnerability
2020-09-29 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.

Mehr Details und weitere Security Advisories finden Sie hier: https://herolab.usd.de/security-advisories

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

Advisory ID: usd-2020-0002
CVE Number: CVE-2020-6581
Affected Product: Nagios NRPE
Affected Version: v.3.2.1
Vulnerability Type: Insufficient Filtering of Configuration file
Security Risk: Medium
Vendor URL: https://www.nagios.org
Vendor Status: Fixed in v.4.0.0 (not verified)

Timeline:
2020-01-06 Found by manual code review of Nagios NRPE
2020-01-08 Initial Contact
2020-01-15 Nagios NRPE v4.0.0 is released
2020-03-04 security advisory released

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
CVE Number: CVE-2020-5836
Affected Product: Symantec Endpoint Protection
Affected Version: 14.2.2.1
Vulnerability Type: Hardlink Vulnerability
Security Risk: Critical
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

Timeline:
2020-03-12 First contact request via symantec.psirt@broadcom.com
2020-05-05 Fix is released in Symantec Endpoint Protection 14.3
2020-05-11 Broadcom publishes Advisory
2020-06-18 Security advisory released

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

Da es am Markt keinen anerkannten Standard für Pentests gibt, können sich die Leistungen verschiedener Anbieter erheblich unterscheiden. Dadurch lassen sich Ergebnisse kaum miteinander vergleichen. Durch unzureichende Dokumentation ist außerdem die Nachweisbarkeit identifizierter Schwachstellen häufig nicht gegeben. Daher investierten wir 2020 verstärkt in unsere usd HeroLab Toolchain. Dadurch stehen unseren Security Analysten eine Reihe von selbstentwickelten, automatisierten Tools zur Verfügung, die ihr Know-how bündeln und im Marktvergleich einzigartig sind. Die Toolchain erlaubt unseren Security Analysten, ihre Prüfungen noch effizienter und umfassender durchzuführen. Ihnen bleibt Zeit für gezielte, manuelle Analysen, bei denen sie kundenspezifische Anforderungen berücksichtigen können. Somit garantieren wir eine strukturierte, nachvollziehbare und individuelle Überprüfung von Systemen und Anwendungen.

usd HeroLab Toolchain

Wir geben einen Einblick in die wichtigsten Tools.

usd ExPeRT

Das Herzstück der usd HeroLab Toolchain.
Hier laufen alle Informationen zusammen.

- Projektplanung & Zusammenarbeit
- Abbildung langjährig bewährter interner Prozesse
- Integrierte Checklisten
- Statusverfolgung & Ressourcenmanagement
- Synchronisation von Daten, z.B. mit usd Icebreaker
- Abschluss- & Prüfbericht
- Exportmöglichkeit der Ergebnisse

usd Cooperator

Burp-Suite Plugin zur weiteren Verbesserung
von Pentests von Webapplikationen

- Unterstützt bei der Dokumentation von Schwachstellen bei Pentests von Webanwendungen
- Kollaboratives Zusammenarbeiten während des gesamten Pentests
  - Austausch von Ergebnissen in Echtzeit
  - Zuweisung von Teilbereichen an einzelne Kollegen
  - Transparente Verwaltung von Projektdetails

usd ICEBREAKER

Analysewerkzeug für alle Teammitglieder
& Wissensdatenbank

- Implementierung von Plugins & selbst entwickelten Skripten
- Analysewerkzeug für alle Teammitglieder & Wissensdatenbank
- Stetige Weiterentwicklung und Anpassung an Forschungsergebnisse
- Hohe Automatisierung manueller Prozesse
- „Scanner“/ Automation Engine
- Verbindet die besten öffentlich vorhandenen und usd intern entwickelten Tools
- Gleichbleibend hohes Qualitätsniveau
- Automatisierte Dokumentation von Prüfinhalten

EXPERTENSTIMMEN

„Wir schützen Unternehmen vor Hackern und Kriminellen. Dafür sorgen wir mit einem breiten Unterstützungsangebot bis hin zum Wissensaustausch in der Community. Denn mehr Sicherheit gelingt nur zusammen.“

Stephan Neumann
Co-Head of usd HeroLab

„Die vorhandenen Risiken werden immer noch viel zu häufig unterschätzt. Ein Pentest deckt Einfallstore in Ihren IT-Systemen auf, sodass Sie diese beseitigen können. Wir helfen Ihnen, Ihr Risiko zu reduzieren.“

Tobias Neitzel
usd Leistungsverantwortlicher Pentest

WIR UNTERSTÜTZEN SIE

IHRE HERAUSFORDERUNGEN

Ob Sie als Unternehmen Anwendungen und Systeme sicher einsetzen oder als Produkthersteller nachweislich mehr Sicherheit für Ihre Kunden anbieten möchten: Mit dem usd HeroLab haben Sie einen der führenden Partner an Ihrer Seite.

- Sie haben kritische Applikationen?
- Sie haben komplexe Umgebungen?
- Sie sind an Compliance Anforderungen gebunden?
- Sie speichern oder verarbeiten sensible Daten?

UNSERE LÖSUNGEN

Das Ziel unserer Sicherheitsanalysen ist es, Schwachstellen zu identifizieren, daraus resultierende Risiken zu benennen und Wege aufzuzeigen, Ihre Sicherheit zu verbessern. Gemeinsam mit Ihnen finden wir die für Sie passende Lösung und sprechen über Ziele, Umgebungen und potentielle Risiken. Unser Toolbasiertes reproduzierbares Prüfverfahren garantiert Ihnen Effizienz, Transparenz und höchste Qualität. Unser Leistungsversprechen ist mehr Sicherheit!

- **Schwachstellen in Ihrer IT-Infrastruktur rechtzeitig erkennen**
  Pentests, Security Scans, Code Reviews, Cloud Security
- **Sicherheitslücken strukturiert bearbeiten**
  Vulnerability Management, Managed Security Services & Remediation Handling
- **Im Ernstfall optimal vorbereitet sein**
  Incident Response und digitale Forensik
- **Nutzen Sie das Know-how einer ganzen Community**
  Bug-Bounty-Programme

usd AG
Frankfurter Str. 233, Haus C1
63263 Neu-Isenburg

Telefon: +49 6102 8631-0
Mail: kontakt@usd.de

www.usd.de | herolab.usd.de