API security in Python (ASIP)

ID ASIP **Preis** CHF 2'250.– (exkl. MwSt.) **Dauer** 3 Tage

Zielgruppe

Python-API-Entwickler

Voraussetzungen

Allgemeine Python-Entwicklung

Kursziele

- Vertrautmachen mit grundlegenden Konzepten der Cybersicherheit
- Verständnis von API-Sicherheitsfragen
- Detaillierte Analyse der OWASP API Security Top Ten Elemente
- API-Sicherheit in den Kontext von Python stellen
- Über die niedrig hängenden Früchte hinausgehen
- Verwaltung von Schwachstellen in Komponenten von Drittanbietern
- Ansätze und Grundsätze der Eingabevalidierung

Kursinhalt

Tag 1

- Grundlagen der Cybersicherheit
  - Was ist Sicherheit?
  - Bedrohung und Risiko
  - Arten von Cybersicherheitsbedrohungen - die CIA-Triade
  - Folgen von unsicherer Software
- OWASP API-Sicherheit Top Ten
  - OWASP API-Sicherheit Top 10 2023
- API1 - Gebrochene Autorisierung auf Objektebene
  - Verwirrter Abgeordneter
  - Unsichere direkte Objektreferenz (IDOR)
  - Übung - Unsichere direkte Objektreferenz
  - Berechtigungsumgehung durch benutzergesteuerte Schlüssel
  - Fallbeispiel - Fernübernahme von Nexx Garagentoren und Alarmanlagen
  - Labor - Horizontale Genehmigung
  - Hochladen von Dateien
  - Uneingeschränkter Datei-Upload
  - Bewährte Praktiken
  - Labor - Uneingeschränkter Datei-Upload
- API2 - Fehlerhafte Authentifizierung
  - Grundlagen der Authentifizierung
  - Multi-Faktor-Authentifizierung (MFA)
  - Fallstudie - Der InfinityGauntlet-Angriff
  - Passwortlose Lösungen
  - Zeitbasierte Einmal-Passwörter (TOTP)
  - Schwachstellen bei der Authentifizierung
  - Spoofing im Internet
  - Passwortverwaltung
  - Speichern von Kontopasswörtern
  - Passwort im Transit
  - Labor - Reicht das Hashing von Passwörtern aus?
  - Wörterbuchangriffe und Brute-Forcing
  - Salzen
  - Adaptive Hash-Funktionen für die Passwortspeicherung
  - Übung - Verwendung adaptiver Hash-Funktionen in Python
  - Verwendung von Tools zum Knacken von Passwörtern
  - Passwortknacken unter Windows
  - Passwort ändern
  - Probleme bei der Passwortwiederherstellung
  - Bewährte Verfahren zur Passwortwiederherstellung
  - Übung - Schwachstelle beim Zurücksetzen des Passworts
  - Fallstudie - Übernahme eines Facebook-Kontos über einen Wiederherstellungscode
  - Fallstudie - Übernahme eines GitLab-Kontos
  - Anti-Automatisierung
  - Passwort-Politik
  - NIST-Authentifikator-Anforderungen für gespeicherte Geheimnisse
  - Passwort-Härtung
  - Verwendung von Passphrasen
  - Migration der Passwort-Datenbank
  - (Fehl-)Handhabung Keine Passwörter

Tag 2

- API3 - Gebrochene Objekt-Eigenschaftsebene-Autorisierung
  - Informationsexposition
  - Offenlegung durch extrahierte Daten und

Aggregation
    - Fallstudie - Strava-Datenexposition
    - Durchsickern von Systeminformationen
    - Auslaufende Systeminformationen
    - Bewährte Praktiken der Informationsexposition
    - Verwaltung von Geheimnissen
    - Fest kodierte Passwörter
    - Bewährte Praktiken
    - Labor - Hartkodiertes Passwort
    - Schutz sensibler Informationen im Speicher
    - Herausforderungen beim Schutz der Erinnerung
    - Fallstudie - Diebstahl geheimer Microsoft-Schlüssel über Dump-Dateien
- API4 - Uneingeschränkter Ressourcenverbrauch
    - Denial of Service
    - Überschwemmungen
    - Erschöpfung der Ressourcen
    - Nachhaltiges Kundenengagement
    - Unendliche Schleifen
    - Wirtschaftliche Verweigerung der Nachhaltigkeit (EDoS)
    - Fragen der algorithmischen Komplexität
    - Denial of Service mit regulären Ausdrücken (ReDoS)
    - Labor - ReDoS
    - Der Umgang mit ReDoS
    - Fallstudie - ReDoS-Schwachstellen in Python
- API5 - Gebrochene Autorisierung auf Funktionsebene
    - Autorisierung
    - Grundlagen der Zugangskontrolle
    - Arten der Zugangskontrolle
    - Fehlende oder unzulässige Genehmigung
    - Versäumnis, den URL-Zugang zu beschränken
    - Cross-Site Request Forgery (CSRF)
    - Labor - Cross-Site Request Forgery
    - Bewährte CSRF-Verfahren
    - CSRF-Verteidigung in der Tiefe
    - Übung - CSRF-Schutz mit Token
- API6 - Uneingeschränkter Zugang zu sensiblen Geschäftsflüssen
    - Sicherheit durch Design
    - Das STRIDE-Modell der Bedrohungen
    - Sichere Gestaltungsprinzipien von Saltzer und Schroeder
    - Wirtschaftlichkeit des Mechanismus
    - Ausfallsichere Standardwerte
    - Vollständige Mediation
    - Open design
    - Trennung der Privilegien
    - Geringstes Privileg
    - Am wenigsten verbreiteter Mechanismus
    - Psychologische Akzeptanz
    - Protokollierung und Überwachung
    - Grundsätze der Protokollierung und Überwachung
    - Unzureichende Protokollierung
    - Fallstudie - Klartext-Passwörter bei Facebook
    - Baumstammfälschung
    - Web-Log-Fälschung
    - Labor - Baumstammfälschung
    - Protokollfälschung - bewährte Verfahren
    - Bewährte Praktiken der Protokollierung
    - Überwachung bewährter Verfahren
- API7 - Server Side Request Forgery
    - Server-seitige Anforderungsfälschung (SSRF)
    - Fallbeispiel - SSRF in Ivanti Connect Secure
- API8 - Fehlkonfiguration der Sicherheit
    - Offenlegung von Informationen durch Fehlermeldungen
    - Informationsleck über Fehlerseiten
    - Labor - Informationsleck im Kolben
    - Fallstudie - Informationslecks durch Fehler in Apache Superset
    - Cookie-Sicherheit
    - Cookie-Attribute
    - Politik der gleichen Herkunft
    - Einfacher Antrag
    - Preflight-Anfrage
    - Ursprungsübergreifende Ressourcennutzung (CORS)
    - Labor - Demo zur Politik des gleichen Herkunftslandes
    - Konfigurieren von XML-Parsern
    - DTD und die Entitäten
    - Erweiterung der Entität
    - Angriff auf externe Entitäten (XXE)
    - Einbeziehung von Dateien mit externen Stellen
    - Server-Side Request Forgery mit externen Entitäten
    - Labor - Angriff einer externen Einheit
    - Verhinderung von XXE
    - Labor - Verbot der DTD
    - Fallstudie - XXE-Schwachstelle in Ivanti-Produkten

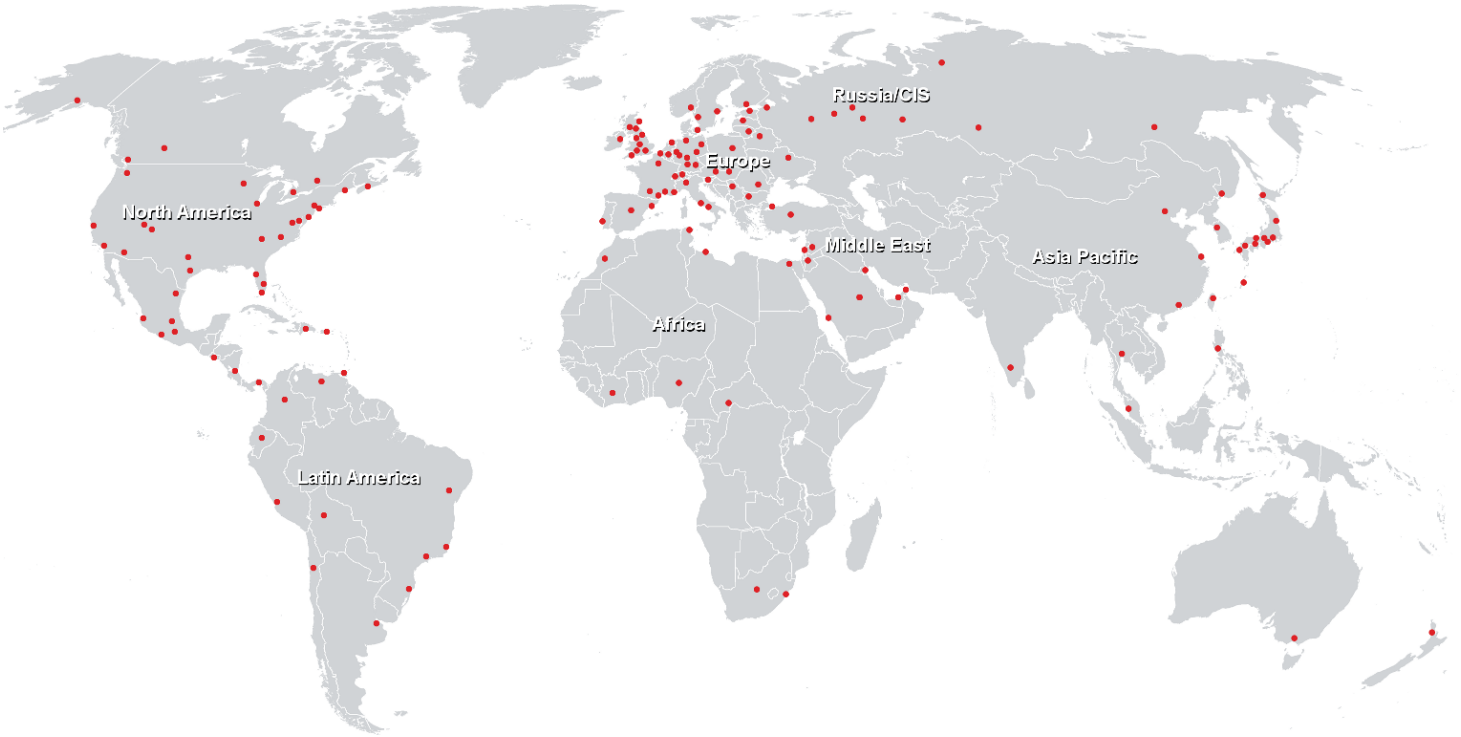
Tag 3

- API9 - Unsachgemässe Bestandsverwaltung
    - Blinde Flecken in der Dokumentation
    - Blinde Flecken im Datenfluss
    - Verwendung anfälliger Komponenten
    - Import von nicht vertrauenswürdigen Funktionen
    - Bösartige Pakete in Python
    - Fallstudie - Der Angriff auf die Lieferkette von Polyfill.io
    - Management von Schwachstellen
    - Übung - Auffinden von Schwachstellen in Komponenten von Drittanbietern
- API10 - Unsicherer Verbrauch von APIs
    - Überprüfung der Eingaben
    - Input-Validierungsprinzipien

- Denylisten und Zulassungslisten
- Fallstudie - Denylist-Fehler in urllib.parse.urlparse()
- Was zu validieren ist - die Angriffsfläche
- Wo soll validiert werden - Verteidigung in der Tiefe
- Wann validieren - Validierung vs. Umwandlung
- Bereinigung der Ausgabe
- Herausforderungen bei der Kodierung
- Unicode-Herausforderungen
- Validierung mit Regex
- Einspritzung
- Injektionsprinzipien
- Injektionsangriffe
- SQL-Einschleusung
- Grundlagen der SQL-Injektion
- Übung - SQL-Injektion
- Angriffsmethoden
- Inhaltsbasierte blinde SQL-Injektion
- Zeitbasierte blinde SQL-Injektion
- Bewährte Praktiken zur SQL-Einschleusung
- Überprüfung der Eingaben
- Parametrisierte Abfragen
- Übung - Verwendung vorbereiteter Erklärungen
- Datenbankverteidigung in der Tiefe
- Fallstudie - SQL-Injection gegen US-Flughafensicherheit
- Code-Einspritzung
- Code-Injektion über input()
- OS-Befehlsinjektion
- Übung - Befehlsinjektion
- Bewährte Praktiken zur Injektion von OS-Befehlen
- Vermeidung von Befehlseingaben mit den richtigen APIs
- Übung - Bewährte Praktiken der Befehlsinjektion
- Fallstudie - Shellshock
- Labor - Shellshock
- Fallstudie - Befehlsinjektion in Ivanti-Sicherheitsanwendungen
- Offene Umleitungen und Weiterleitungen
- Offene Umleitungen und Weiterleitungen - bewährte Verfahren
- Dateien und Datenströme
- Pfadüberquerung
- Übung - Pfadüberquerung
- Zusätzliche Herausforderungen in Windows
- Fallstudie - Dateispoofing in WinRAR
- Bewährte Verfahren zur Pfadüberquerung
- Labor - Kanonisierung von Pfaden
- Einpacken
- Grundsätze der sicheren Kodierung
- Grundsätze der robusten Programmierung von Matt Bishop
- Sichere Gestaltungsprinzipien von Saltzer und Schroeder
- Und was nun?
- Quellen zur Softwaresicherheit und weiterführende Literatur
- Python-Ressourcen

API security in Python (ASIP)

Weltweite Trainingscenter





Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3
CH-8304 Wallisellen
Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch