Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Kommunikationssysteme und Systemprogrammierung
Prof. Dr. Helmut Reiser

MNM TEAM
Munich Network Management Team

IT-Sicherheit im Wintersemester 2021/2022
Übungsblatt 4

Besprechung: Di, 23.11.2021

Achtung: Zur Bearbeitung einiger Übungsaufgaben ist es notwendig sich über den Vorlesungsinhalt hinaus, durch Internet- und Literaturrecherche mit dem Thema zu beschäftigen.

Aufgabe 11: (T) Malicious Code & SPAM-Protection

a. Zur Erkennung von Malicious Code auf einem System werden in der Regel Antiviren-Programme eingesetzt, die eine Reihe verschiedener Erkennungstechniken kombiniert verwenden. Erläutern Sie *Signatur-basierte*, *Heuristische/Anomalie-basierte* und *Emulations-basierte Erkennung* und beschreiben sie jeweils die Stärken und Schwächen des jeweiligen Ansatzes.

b. Um der Erkennung durch aktuelle Antiviren-Programme zu entgehen, werden bei der Erstellung und Programmierung polymorpher Viren verschiedene Techniken eingesetzt. Erläutern Sie die folgenden Techniken

   - Garbage instructions
   - Instruction reordering
   - Interchangeable instructions

c. Es existieren verschiedene Maßnahmen, SPAM zu erkennen und diesen herauszufiltern bzw. zu blocken. Erläutern Sie folgende Verfahren: *DNS-basierte Blacklists*, *RHSBLs* und *naive Bayes-Klassifizierung*. Gehen Sie hier zusätzlich auf rechtliche Probleme ein, die Ihnen bei Einsatz dieser Verfahren begegnen.

Aufgabe 12: (T) Endpoint Security & EDR

Die Endgeräte der Nutzer sind häufig Einstiegspunkt von Cyberangriffen auf Unternehmensnetze. Die Strategie der Endpoint Security zielt darauf ab, das zentrale Netz bereits an den einzelnen Endgeräten zu schützen, die sich am Rand des Netzes und oftmals außerhalb der Firmenfirewall befinden.

a. Welche Ansätze von Virenscannern gibt es? Welche Nachteile haben sie?

b. Welchen Ansatz verfolgen *Endpoint Security*-Systeme? Wo liegt hier der Unterschied zu Antivirenprogrammen?

c. Aus welchen Komponenten können Endpoint Security bzw. Endpoint Protection Systeme bestehen?

d. Welche Ziele verfolgt *Endpoint Detection und Response* (EDR)? Wie spielt EDR mit Endpoint Security zusammen?

e. Mit welchem Ansatz versucht *Extended EDR* (XDR) noch einen Schritt weiterzugehen?

Aufgabe 13: (T) XSS & SQL-Injection

a. In der Vorlesung wurden drei verschiedene Arten von *Cross-Site-Scripting* (XSS) vorgestellt. Welche der Varianten besitzt das höchste Bedrohungspotential?

b. Zum Schutz vor XSS existieren verschiedene Maßnahmen, welche einen Angriff verhindern oder dessen Auswirkungen verringern sollen. Beschreiben Sie kurz die folgenden Techniken:

   - Eingabevalidierung
   - Content Security Policy
   - HTTPonly

c. Beschreiben Sie wie SQL-Injection funktioniert und wie Sie eine Anfrage formulieren würden, um die Loginmaske einer Webseite zu umgehen und auf den Account **administrator** zuzugreifen, von der Sie wissen, dass die Passwörter mit folgendem SQL-Query überprüft werden:

   ```
   'SELECT uid FROM users WHERE username = "' + $username + '" AND
   password = "' + $password + '"'
   ```

   Sie können dabei die Parameter `$username` und `$password` über das Formular frei eingeben.

d. Beschreiben Sie sowohl abstrakt als auch konkret, wie Sie den Query aus der vorhergehenden Teilaufgabe verändern müssen, um SQL-Injection zu verhindern. Sie können sich bei der konkreten Beschreibung eine gängige Programmiersprache für das Web (z.B. PHP) aussuchen.

Aufgabe 14: (T) Security Code Review 101

Viele Verwundbarkeiten von Anwendungen gehen auf Unachtsamkeiten während der Programmierung zurück. Ein (unabhängig und teils automatisiert durchgeführtes) Code Review zielt darauf Security-Aspekte bereits zur Programmierzeit zu verbessern.

Nutzen Sie das OWASP *Secure Coding Dojo*, um typische Fehler bei Eingabenvalidierung, Speicheroperationen etc. zu identifizieren und Ihren eigenen Programmierstil zu verbessern!
`https://owasp.org/SecureCodingDojo/codereview101/`

Aufgabe 15: (H) Buffer-Overflow

Angreifer nutzen oftmals Schwachstellen in lokal installierten Applikationen.

a. Erläutern Sie, was bei einem Buffer-Overflow genau passiert und wie ein Angreifer diesen für einen Angriff ausnutzen könnte?

b. Beschreiben Sie den Unterschied zwischen einem klassischen Buffer-Overflow und einem return-to-libc Angriff.

c. Nennen und beschreiben Sie mindestens drei Schutzmaßnahmen, die zum Schutz vor Buffer-Overflows eingesetzt werden können.

Aufgabe 16: (H) Web-Vulnerabilites: bWAPP

Web-Applikationen stehen ihrer weiten Verbreitung wegen häufig im Fokus von Cyberattacken. Eine Web-Applikation mit beinahe allen nur erdenklichen Verwundbarkeiten finden Sie in *bWAPP* – a buggy WebApp.

bWAPP steht Ihnen zur Verfügung als

- PHP/MySQL-WebApp:
  `https://sourceforge.net/projects/bwapp/files/bWAPP/`
- VM-Image *bee-box*:
  `https://sourceforge.net/projects/bwapp/files/bee-box/`

Testen Sie Ihr Können! Wieviele der über 100 Schwachstellen können Sie ausnutzen?