SecLab

Praktikum IT-Sicherheit

- Versuchshandbuch -

Programme manipulieren

Buffer Overflow

Buffer Overflow gehört zu den häufigsten Sicherheitslücken in aktueller Software. Im Wesentlichen werden bei einem Buffer Overflow, durch Fehler im Programm, zu große Datenmengen in einen dafür zu klein reservierten Speicherbereich geschrieben. Hierdurch werden nachfolgende Informationen im Speicher überschrieben. Wie diese Technik dazu eingesetzt werden kann Programme zu manipulieren, wird in diesem Versuch gezeigt.

B.Sc. BG 24
M.Sc. AI MN 1
M.Sc. EB 10

IT-Sicherheit

Aufgabe1 – Buffer Overflow Grundlagen

a) Starten Sie Cygwin und navigieren Sie in den „BufferOverflow"- Ordner.
   - cd C:
   - cd Dokumente und Einstellungen/Administrator/Desktop/BufferOverflow
   - cd overflowbeispiel1

   Untersuchen Sie zunächst Beispiel eins und öffnen Sie mit dem Programm notepad++ den Quelltext „prog1.c". Vergegenwärtigen Sie sich, dass die Funktion „OverflowMe" eine unsichere Leseanweisung von der Console nutzt. Warum ist diese Leseanweisung unsicher? Wie kann sicher eingelesen werden?

   .....

   .....

   .....

   .....

b) Wie groß ist der Buffer (das char-Array) der gelesen werden soll?

   .....

c) Starten Sie das Programm mit einem String-Parameter von der Länge 4 über die Console „prog1.exe abcd". Beobachten Sie den Ablauf. Schauen Sie erneut in den Quelltext und vergegenwärtigen Sie sich, warum immer der else-Zweig des Programms aufgerufen wird. Warum ist dies so?

   .....

   .....

   .....

   .....

d) Überfluten Sie nun den Buffer, indem Sie einen sehr langen String, bestehen aus dem Zeichen „A" an das Programm übergeben. („prog1.exe AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA")
Kann das Programm diesen lange String einlesen?

.....

.....

.....

e) Das Überfluten provoziert die Meldung „Segmentation fault (core dumped)". Schauen Sie sich das Protokoll prog1.exe.stackdump, welches im selben Ordner wie das Programm liegt, mit notepad++ an. Welche Adresse steht im Extended Instruction Pointer (EIP)?

.....

.....

f) Vergewissern Sie sich, dass „A" dem Hex-Wert „41" entspricht, indem Sie eine neue Datei in der lediglich dem Zeichen „A" steht, mit notepad++ erstellen. Schalten Sie im notepad++ in der Werkzeugleiste auf die Hex-Sicht um (H).
Bei dem Ausführen des Programms mit dem sehr langen String wurde also fälschlicher Weise an die Stelle „41 41 41 41" im Speicher gesprungen. Warum kommt es nun zu einem Segmentation fault, wenn an die Adresse „41 41 41 41" gesprungen wird?

.....

.....

.....

.....

g) Setzen Sie nun wie folgt beschrieben diese Sprungadresse durch einen Buffer Overflow auf die Adresse des then-Zweigs welche das Programm prinzipiell nutzen darf. Nutzen Sie den Debugger „OllyDbg" und öffnen Sie das Programm „prog1.exe". Finden Sie das C-Programm in dieser Ansicht der Executable wieder? Ermitteln Sie die Speicheradresse des then-Zweigs und notieren Sie sich diese. Wie lautet die Adresse?
Kann der Angreifer ohne Kenntnis des eigentlichen Quellcodes Speicheradressen ermitteln?

.....

.....

.....

.....

h) Schreiben Sie sich eine Datei „Muster" welche den String beinhaltet, der zu dem Overflow führt (also eine Datei mit viele „A"s). Führen Sie nun das Programm aus und übergeben als Parameter den Inhalt der Datei mit dem Befehl „./prog1.exe `cat Muster`". Verringern Sie die Menge an „A"s in der Datei sukzessiv und rufen Sie jedesmal prog1.exe mit dem Muster auf. Beobachten Sie dabei die Veränderung von der Datei „prog1.exe.stackdump". Was können Sie beobachten?

.....

.....

.....

.....

.....

i) Wiederholen Sei diesen Vorgang bis der EIP auf „00 00 41 41"" steht. Sie haben nun die Position der Sprungadresse (EIP) ermittelt und können diese anpassen.
Fügen Sie in die ermittelte Sprungadresse des then-Zweigs ein. Dazu benutzen Sie die Hex-Ansicht des Programms notepad++ und die angepasste Datei Muster. Passen Sie die letzten Zeichen gemäß der Sprungadresse an (Beachten Sie, dass die Reihenfolge der Bits wegen der Big-Endian-Formatierung umgekehrt ist). Die Sprungadresse müsste nun auf „d4 10 40 00"
Ihr Muster sieht nun in Hex ungefähr so aus: „41 41 41 41 … 41 41 d4 10 40 00".
Übergeben Sie nun das Muster wie gehabt an das Programm „prog1.exe".
Was stellen Sie fest?

.....

.....

.....

.....

j) Das zweite Beispiel „overflow2" ist mehr kommentiert. Schauen Sie sich hier den Quelltext „prog2.c" an. Betrachten Sie die „prog2.exe" mit OllyDBG. Vergleichen Sie die Dateien „exploit1" und „exploit2". Was ist der Unterschied zwischen beiden? Warum wird in beiden Fällen die Funktion „myFunction" ausgeführt.

.....

.....

.....

.....

.....

.....

k) Sie haben gesehen, wie ein Angreifer eine Schwachstelle in einem Programm ausnutzen und Sprungadressen ändern kann. Wie würden Sie nun Schadcode einführen und aufrufen? Bitte nur ein Konzept aufzeigen.

.....

.....

.....

.....

.....

.....

Aufgabe 2 – Metasploit

Nutzen Sie die virtuellen Maschinen „Windows XP Opfer" und „ubuntu10desktop".

Benutername für das ubuntu-System ist „user" und Passwort ist auch „user".

a) Erstellen Sie ihr erstes Exploit nach folgender Anweisung. Starten Sie das Metasploit-Frameword (MSF), indem Sie ein Terminal öffnen und in den Ordner „Desktop/framework-3.1" wechseln. Rufen Sie dort die MSF-Executable „./msfconsole" auf. Lassen Sie sich alle Exploits mit dem Befehl „show exploits" anzeigen. Welche Exploits gibt es für den Microsoft Internet Explorer?

.....

.....

.....

b) Benutzen Sie einen Bufferoverflow im Internet Explorer 6 mittels „use windows/browser/ms06_001_wmf_setabortproc". Dieser Exploit bietet die Möglichkeit verschiedene Payloads auszuführen. Lassen Sie sich alle Payloads mit „show payloads" anzeigen. Nennen Sie 3 interessante Payloads und beschreiben Sie deren Funktion anhand der Beschreibung.

.....

.....

.....

.....

.....

.....

c) Nutzen Sie den Payload „windows/exec" mittels dem Befehl „set PAYLOAD windows/exec". Dieser Befehl erlaubt es Ihnen Befehle auf der Console des Opfers auszuführen. Es soll nun über die Console des Opfer der Taschenrechner des

Opfers ausgeführt werden. Dies funktioniert, indem Sie die Anweisung mit dem Befehl „set CMD calc" setzen. Das MSF ist zum Testen eines Systems gedacht. Damit ein Overflow in einem Webbrowser getestet werden kann stellt MSF uns nun für dieses Exploit einen Webserver zur Verfügung. Mittels dem Befehl „set URIPATH ieTest" setzen Sie eine URI unter dem das Exploit für das Opfer zur Verfügung steht. Lassen Sie sich das Exploit mit „show options" anzeigen. Welche Konfiguration haben Sie gemacht?

.....

.....

.....

.....

d) Nun sind alle notwendigen Informationen eingetragen. Starten Sie das Exploit mittels dem Befehl „exploit". Ermitteln Sie nun die IP-Adresse von dem System ubuntu10desktop. Das Exploit steht unter dieser IP-Adresse am Port 8080 im Ordner ieTest zur Verfügung.

Öffnen Sie nun die Opfermaschine „Windows XP Opfer". Öffnen Sie den Internet Explorer 6 und besuchen Sie das ubuntu10desktop-System. Geben Sie dazu im Browser die Adresse: „http://<IP-Adresse von ubuntu>:8080/ieTest".

Was beobachten Sie auf der Opfermaschine?

.....

.....

.....

.....

.....