Reaktive Sicherheit

Übungsblatt 7

Balduin Binder [s6babind@uni-bonn.de] Charlotte Mädler [s6chmaed@uni-bonn.de]
Bünyamin Sarikaya [s6busari@uni-bonn.de]

Aufgabe 2

(1)

Path Traversal Verwundbarkeit (auch Directory Traversal gennant) ist eine Sicherheitslücke in einem Webserver/einer Webanwendung, wobei unerlaubterweise auf Dateien/Verzeichnisse durch Eingabe von URLs zugegriffen wird. Normalerweise kann man nicht auf Dateien eines Webservers außerhalb des Web-Verzeichnisses oder dessen Unterverzeichnisse zugreiffen. Bei dem Angriff wird durch eine manipulierte Pfadangabe aber auf Dateien außerhalb dieser Verzeichnisse zugegriffen. Dies wird meist durch Angabe von `../` bewirkt, da man sich damit im Verzeichnis eine Ebene nach oben bewegt. Mit `/` wäre es direkt zur Wurzel.

(2)

Die Vertaulichkeit wird verlezt (eventuelles auslesen von vertraulichen Daten, Passwörter, etc.)

(3)

Falls vorhanden öffnet sich die cat.jpg, falls nicht öffnet sich die Seite mit dem Schriftzug: Cat not found

(4)

Die passwd-Datei liegt in der selben Partion wie die webcat.py Datei. Deswegen kann die Path Traversal Verwundbarkeit ausgenutzt werden. Wenn webcat und passwd in unterschiedlichen Partitionen sind geht dies nicht.

(5)

`http://127.0.0.1:5000/?filename=/etc/passwd` lädt die psswd-Datei herunter. Es wird angenommen, dass sich dort das Bild befindet und als `return send_file(real_filename)` wird die passwd Datei zurückgegeben und gedownloadet.

Aufgabe 5

(1)

Hauptsächlich unterscheiden sich Programme und Prozesse darin, dass das Programm aus einer Gruppe von Anweisungen besteht, die eine bestimme Aufgabe ausführen sollen. Ein Prozess hingegen, ist ein bestimmtes Programm, das aktuell ausgeführt wird, also eine Instanz eines ausgeführten Programmes. Ein Programm kann also auch mehrere Prozesse aufrufen (1 zu n Beziehung). Ein Prozess wird deshalb als aktive Entität betrachtet, ein Programm jedoch als passiv. Dadurch entscheiden sich auch die Lebensdauern und benötigte Resourcen von einander: Ein Programm ist im Speicher bis es manuell gelöscht wird und benötigt auch nur Speicherplatzt, ein Prozess wird nach Abschluss der Aufgabe beendet, benötigt zur Ausführung jedoch zum Teil Verarbeitungs-, Speicher- oder E/A-Resourcen.

Dateiformat x64-Linux-Programme:

Executable and Linkable Format (ELF)

Ersten vier Byte des Formats:

`0x7F 45 4C 46`. Dabei steht `45 4C 46` in ASCII für ELF.

Bytes des Headers:

ELF-Header: 64 byte, das fünfte Byte definiert ob es sich um ein 32- oder 64-bit Format handelt. Dabei steht eine 1 für 32 Bit und eine 2 für 64 Bit.

(2)

Schritt 1

Sobald der Virus das erste Mal ausgeführt wird sucht er ausgehend vom Homeverzeichnis die erste noch nicht infizierte auführbare Datei. Jetzt erstellt er eine Kopie von sich selbt in einem neuen hidden-temp-file. Dann kopiert er das Host-Programm hinten dran und signiert das Ende der Datei mit einer signature `deadbeef`. Anschließend änder er den Namen des hidden-temp-files auf den originalen Namen des Hosts.

Schritt 2

Beim ersten Start des Virus wird nach dem zuvor beschriebenen Schritt 1 folgendes ausgegeben: `Ich bin ein nettes Programm!`

Schritt 3

Beim starten einer in Schritt 1 infizierten Datei passiert nun folgendes:
Schritt 1 wie oben. (Eine weitere executable wird infiziert)
Schritt 2 wird übersprungen.
Als nächstes wird nun folgender Text ausgegeben: `Finden Sie raus, wie man in 3 Tagen risikofrei 1.000 Euro verdienen kann! https://youtu.be/dQw4w9WgXcQ`
Das gestartete infizierte Programm entfernt das originale Programm aus dem infizierten Programm und startet das originale Programm als einen child-process, sodass das originale Programm weiterhin funktionsfähig

bleibt und der Nutzer keinen Verdacht schöpft. Dabei bleiben infizierte Programme infiziert und die originale Funktion wird immer als child-process gestartet.

Der Virus befällt ausschließlich ausführbare Dateien (siehe `check_elf` im Code). Er reproduziert sich durch das kopieren in andere ausführbare Dateien wodurch er immer wieder ausgeführt und reproduziert wird (Solange immer wieder befallene ausführbare Dateien gestartet werden).
Er erkennt bereits infizierte ausführbare Dateien an der Signatur `deadbeef` die er an deren Ende hinterlässt und jedes mal prüft ob diese Signatur vorhanden ist.

(3)

Durch `echo -en '\xef\xbe\xad\xde' >> virus` wird `deadbeef` an das Ende des auführbaren Datei `virus` geschrieben. Dabei aktiviert `-e` die Interpretation von backslash ecapes und `-n` verhindert dass eine newline hinzugefügt wird.
Man signiert also sein eigenes Programm als schon infiziert, damit es sich nicht noch einmal selbst infiziert. Es verhindert lediglich, die Selbstinfektion, welche das Verbreiten des Virus nur verhindert, wenn sich der Virus im ersten Schritt selbst infiziert und somit weitere Programme nur infiziert werden können wenn der Nutzer den Virus noch einmal ausführt.

(4)

Dort wo der Virus ausgeführt wird kann er weitere ausführbare Programme infizieren. Er kann nicht das Directory wechseln, da `find_clean_hostfile` immer nur in `./` (also in dem Directory wo er sich gerade befindet) ausgeführt wird, bis er dort kein noch nicht infiziertes Programm mehr findet.