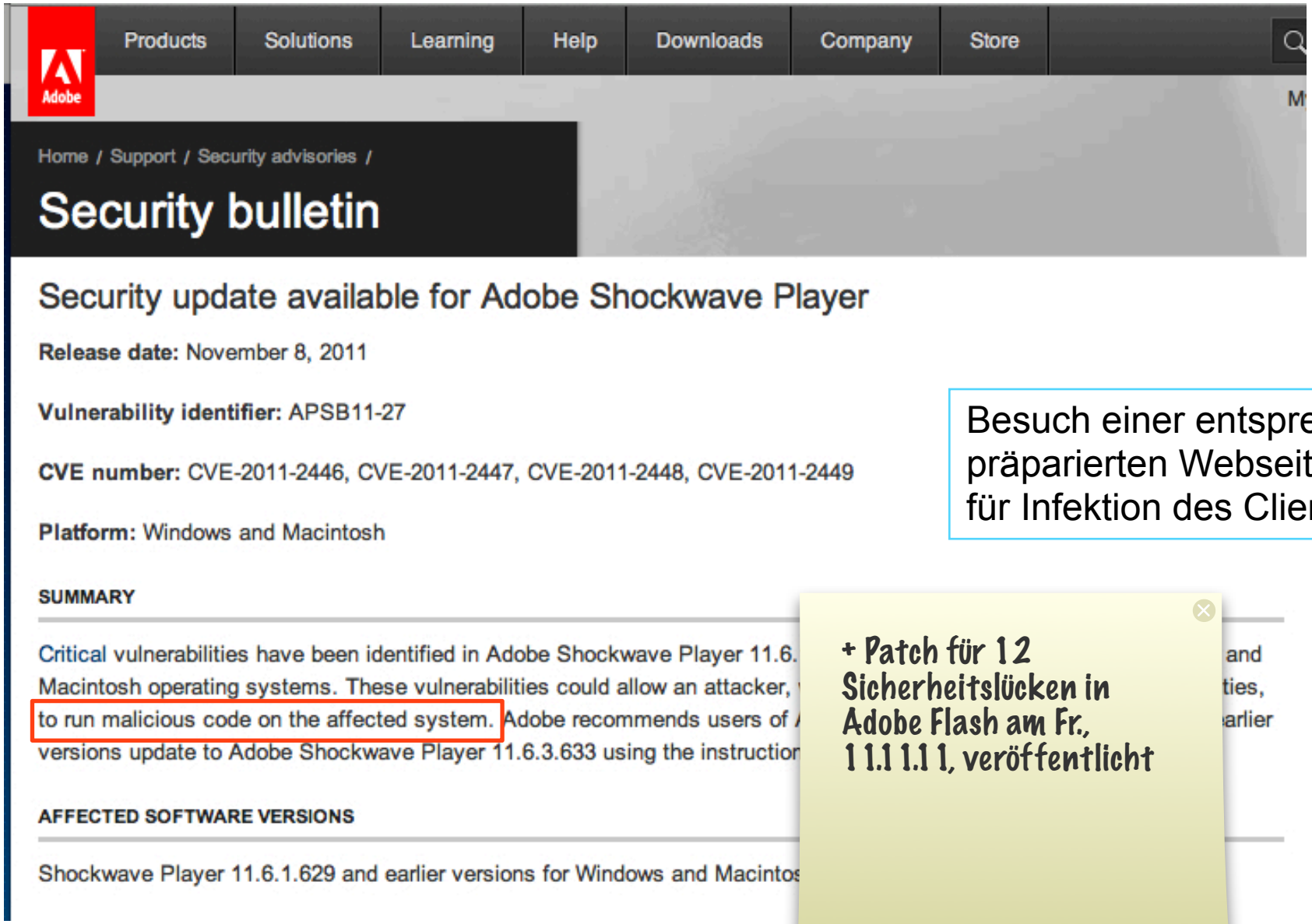
Lücke in der JavaScript-Engine von iOS [07.11.2011]

- Kombination aus trojanischem Pferd, Nachladen von Schadcode und fehlerhafter JavaScript-Implementierung.
- Demo-App von Charlie Miller für Konferenz SyScan 2011:
  - Nutzfunktionalität: Börsenticker-App für iPhone
  - Vom Apple App-Store geprüft und zugelassen
  - App-Store signiert die App, iPhone führt eigentlich nur signierten Code aus
  - App ruft nach dem Start weiteren Code von zentralem Server ab
  - Performance-optimierte neue JavaScript-Engine von iOS (Nitro) hat Implementierungsfehler
  - Trojanisches Pferd kann darüber den nachgeladenen, nicht signierten Code ausführen.
  - Demo: Shell für Fernzugriff, Auslesen des iPhone-Adressbuchs
- Folge:
  - Apple entfernt App aus dem App-Store und entzieht Miller die Entwicklerlizenz für iOS

Shockwave Player malicious code execution [08.11.11]



Besuch einer entsprechend präparierten Webseite reicht für Infektion des Clients aus.

+ Patch für 12 Sicherheitslücken in Adobe Flash am Fr., 11.11.11, veröffentlicht

Angriffe auf Dienstanbieter [11./12.11.2011]

- Ausspähen von Kunden-Root-Passwörtern bei 1blu
  - vgl. Hetzner-Vorfall 10.10.2011
  - 1blu sperrt root-Accounts, die noch das Initialpasswort verwenden
  - Kunden sollen alle Passwörter ändern, auch für Kundenservicecenter und E-Mail-Zugang
- Spiele-Plattform Valve Steam gehackt
  - Erfolgreicher Angriff auf Foren-Server
  - Angreifer haben Zugriff auf Datenbank mit Kundennamen, salted Hashes von Passwörtern, Rechnungsadressen, Kauf-Historie, verschlüsselte Kreditkartendaten.
  - Offizielle Empfehlungen
    - Kreditkartenabrechnungen genau prüfen
    - Passwörter für Foren- und Download-Plattform ändern

FBI-Operation „Ghost Click“

- Sechs Festnahmen in Estland, Auslieferung nach USA beantragt
- Größtes Botnet, dessen Betreiber verhaftet wurden:
  Über 4 Mio. Rechner in über 100 Ländern
- Malware DNSChanger:
  - ändert auf infiziertem Rechner die IP-Adresse des verwendeten DNS-Servers
  - dieser liefert falsche Antworten zurück; Benutzer landet auf anderer Webseite, als er eigentlich wollte.
  - $14 Mio. Einnahmen durch Einblendung von Werbung und Umleitung auf andere Websites.
  - Malware enthält auch DHCP-Server: IP-Adressen und zu verwendender DNS-Server werden auch an andere Geräte im LAN weitergegeben.
  - Malware versucht, DSL-Router per Wörterbuch-Angriff zu knacken, um auch dort die DNS-Server-Einstellung zu modifizieren.
  - Malware hindert diverse Antiviren-Software am Signatur-Update.
- FBI betreibt vorübergehend die DNS-Server weiter.

Nachtrag: UNIX Passwort-Hashing mit crypt()

- Passwort wird nicht im Klartext in /etc/passwd gespeichert, sondern verschlüsselt (User-Passwort ist dabei sowohl Klartext als auch Passwort für die Verschlüsselung).
- Angreifer könnte verschlüsselte Werte für alle Wörterbuch-Einträge vorab berechnen und müsste nur noch vergleichen.

```c
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    char *ergebnisAA, *ergebnisxy;

    ergebnisAA = crypt("GeheimesPasswort", "AA");
    printf("Salt AA: %s\n", ergebnisAA);

    ergebnisxy = crypt("GeheimesPasswort", "xy");
    printf("Salt xy: %s\n", ergebnisxy);

    return 0;
}
```

- Deshalb: In die Verschlüsselung fließen zwei weitere Zeichen („Salt“) ein, die zufällig gewählt werden.
- Salt wird im Klartext hinterlegt.
- Angreifer müsste 4096 Werte pro Wörterbuch-Eintrag vorab berechnen.
- (Aus heutiger Sicht kein großer Aufwand mehr)

Ausgabe:

```
Salt AA: AA3w0THiFXV1A
Salt xy: xyj.4bikXtQ1o
```

Nachtrag: UNIX Passwort-Hashing mit crypt()

- Neuerer Ansatz:
  - Verschlüsselte / gehashte Passwörter in /etc/shadow ausgelagert.
  - Nur noch „root“ hat überhaupt Lesezugriff, reguläre Benutzer kommen nicht an die verschlüsselten / gehashten Passwörter heran.
  - Längeres Salt.
  - Aufwendigere Hashverfahren, z.B. SHA-512, in mehreren Runden angewandt (z.B. 1000x SHA-512).

- Konsequenzen:
  - Angreifer muss zunächst root-Rechte erlangen, um /etc/shadow auslesen zu können.
  - Angreifer muss Wörterbuch-Angriff anpassen an
    - individuelles Salt pro Benutzer bzw. pro Passwort
    - systemweit gewählten Hash-Algorithmus und Anzahl an Runden