Projektideen
Projekt u23

u23-Crew

22. Oktober 2015

Einleitung

Die Auflistung liefert verschiedene Ideen für die Abschlussprojekte — ihr könnt gerne auch eigene Ideen verfolgen. Die Projektpräsentation findet im Rahmen des OpenChaos am 26. November statt. Ihr seid herlich eingeladen, auch darüber hinaus an Euren Projekten weiter zu arbeiten.

Jedes Projekt sollte in Kleingruppe (2-3 Personen) bearbeitet werden — ergänzt Euch.

1 Diffie-Hellman-MITM Attacke

Beschreibung: Implementiert eine funktionierende Man-In-The-Middle-Attacke gegen den Diffie-Hellman Schlüsselaustausch.

- Variante A: Verwendet ein eigenes IRC basiertes Protokoll mit DH-Schlüsselaustausch und ein Proof-of-Concept-Angriff. Gestaltet das Projekt als Game / Challange: Spieler sehen verschlüsselte Nachrichten in einem IRC-Chat und müssen den Schlüssel knacken.
- Variante B: Implementiert einen MITM-Angriff für OTR, der nicht-verfizierte Fingerprints ausnutzt

Skills: Netzwerkprogrammierung, Scripting

Schwierigkeit: Einfach (A) / Schwierig (B)

2 WPA (TKIP) brechen

Beschreibung: M. Vanhoef und F. Piessens behaupten in „All Your Biases Belong To Us: Breaking RC4 in WPA-TKIP and TLS“ WPA (TKIP) verschlüsselte Netzwerke in unter

einen Stunde knacken zu können. Versucht den Angriff nachzustellen und implementiert passende Routinen in aircrack-ng. Koordiniert Eure Arbeit mit den Autoren der Software.

Skills: Mathematisches Verständnis, C

Schwierigkeit: Sehr schwierig

3 Visuelle Kryptohographie App für Android

Beschreibung: Visky verschlüsselt Bilder auf Basis eines One-Time-Pads und wird im Rahmen der Schüler-Crypto genutzt: Key und Bild werden auf Folien gedruckt, die dann auf einem Overheaad-Projektor zusammen gelegt werden. Das Applet ist jedoch in die Jahre gekommen. Implementiert Visky man in Android neu.

Skills: Android-Entwicklung

Schwierigkeit: Mittel

4 Angriff auf encfs

Beschreibung: Encfs wird benutzt, um Dateien innerhalb von Cloud-Dienste zu verschlüsseln. Anstelle eines Containers wird jede Datei einzeln verschlüsselt, so dass Dienste wie Dropbox oder owncloud die Dateien synchronisieren können. Im Rahmen eines Audits wurde eine Schwachstelle gefunden: Verfügt ein Angreifer über mehrere Versionen einer Datei, so kann er Rückschlüsse auf deren Inhalt ziehen. Es existiert jedoch kein Proof-Of-Concept Code. Implementiert einen Angriff auf Encfs.

Skills: Mathematisches Verständnis, Python / Ruby

Schwierigkeit: Mittel bis Schwierig

5 GnuPG Testsystems

Beschreibung: Mit Adele existierte ein Testsystem für GnuPG. Benutzer konnte verschlüsselte Nachrichten dorthin senden und bekamen eine verschlüsselte und signierte Antwort die sie verifizieren konnten. Leider ist adele nicht mehr aktiv. Implementiert Adele neu.

Skills: Scripting, Webentwicklung

Schwierigkeit: Mittel

6 GPG-Keyserver mit Identitätscheck

Beschreibung: Aktuelle Keyserver überprüfen nicht, ob der Hochlader tatsächlich unter der angegebenen E-Mail Adresse erreichbar ist. Trolle laden Schlüssel für fremde E-Mail Adressen hoch. Implementiert oder modifiziert einen Keyserver, der Absenderadressen verifiziert.

Skills: Python / Ruby, Webentwicklung

Schwierigkeit: Mittel

7 GPG: Fingerprint-Verifikation

Beschreibung: Die Überprüfung von GnuPG Fingerprints ist aufwendig: Lange Zahlenkolonnen müssen verglichen werden. Für u.a. OTR und Threma existieren einfachere Verfahren, bei denen der Gegenüber beispielsweise eine Frage beantworten muss. Setzt ein einfacheres Verfahren mit enigmail und Thunderbird um.

Skills: JavaScript, Kreativität

Schwierigkeit: Mittel

8 Kryptokampange: Praxis-Tests

Beschreibung: Überzeugt Freunde und Bekannte (> 20) auf TextSecure oder Threema umzusteigen. Gelingt es Euch? Wie viele von 20 Personen bleiben dabei? Womit habt Ihr Erfolg? Welche Probleme gab es? Wie habt ihr sie gelöst?

Skills: Sozialkompetenz, Statistik

Schwierigkeit: Einfach

9 GPG: Web-Of-Trust Visualisierung

Beschreibung: Wie stellt Ihr Euch das Web-Of-Trust vor? Was ist Euer strongset? Im Internet gibt es verschiede Projekte die das Web-Of-Trust Visualisieren — einige zeigen aber nur langweilliges HTML. Visualisiert Beziehungen zwischen GPG-Keys interaktiv mit JavaScript.

Skills: JavaScript, d3

Schwierigkeit: Einfach

10 Batch-GCD-Attacken

Beschreibung: Zwei RSA-Moduli $N_1,\ N_2$ können faktorisiert werden, wenn $gcd(N_1, N_2) \neq 1$ gilt. Das passiert, wenn die Zufallszahlen bei der Erzeugung schlecht gewählt werden. Forscher konnten so SSH-Keys, SSL-Zertifikate und elektronische Gesundheitskarten knacken. Implementiert ein Verfahren für Batch-GCD-Attacken auf SSH-Keys und testet es an OpenWRT-Routern: Typischer Weise haben günstige Heimrouter bei der Erzeugung der Schlüssel nur sehr wenig Entropie.
Beobachtet Ihr eine Änderung, wenn Die Geräte havegd verwenden?

Skills: Scripting

Schwierigkeit: Einfach bis Mittel

11 Sichere Mailinglisten

Beschreibung: Mailingliste verteilen E-Mails an viele Teilnehmer. Bei einer verschlüsselten Mailingliste sollen alle Teilnehmer unter dem gleichen GPG-Key erreichbar sein, ohne dass der private Key an alle Teilnehmer zentral erstellt und kommuniziert werden muss. Implementiere ein System, bei dem ein Elgamal-Key generiert wird, indem alle Teilnehmer einen mehrfach Diffie-Hellman Schlüsselaustausch (Kreis oder Baum) durchführen. Der Schlüssel soll dann als GPG-key bereit gestellt werden.

Bonus: Verhindere Man-In-The-Middle-Angriffe, indem jeder Mailinglisten-Teilnehmer seine Nachrichten mit GPG signiert.

Skills: JavaScript, GPG — Falls möglich, versuche die Software als Thunderbird-Plugin umzusetzen.

Schwierigkeit: Mittel