Portknocking und verwandte Techniken

Dr. Ralf Schlatterbeck
Open Source Consulting

Email: office@runtux.com
Web: http://www.runtux.com
Tel. +43/650/621 40 17

Inhalt

Anwendungen / Prinzipien

- Zusätzlicher Schutz von SSH
  oder für andere Protokolle (SSL)
- z. B. Fernwartungszugang
- Unsichtbar: Kommunikation ist one-way
- Ports sind normalerweise geschlossen
- können durch „Anklopfen“ geöffnet werden
- → Weitere Verteidigungslinie gegen 0-day exploit

Motivation

- Komplexe Security-Protokolle sind Angriffsziel
  → SSH-Exploits, Fehler in SSL, . . .
- Kryptographische Methoden sind aufwändig
  → Gefahr von Denial of Service (DOS)
- Daher *ergänzen* durch weitere, billigere Verteidigungslinien
- lightweight *one-way-signalling* [BHI+02]
- Wichtig: Nicht *Ersatz* für weitere Authentifizierungsmethoden!

Concealment and Authentication

"... significant benefit in having multiple, progressively stronger, layers of security, rather than attempting to have a single 'perfect' security layer"

- service should be *hidden* ... while still providing service to authorized parties
- credentials should be easy to validate, yet difficult to forge
- Full-strength application-specific security mechanisms are still used ... [for] end-to-end authentication [BHI+02]

Concealment and Authentication (2)

Drei vorgeschlagene Methoden:

- Spread-Spectrum TCP: Key im Header bei einer Sequenz von TCP SYN Paketen (z.B. dest. port number oder initial sequence number)
- Tailgate TCP: Einzelnes Paket (auch UDP) enthält Schlüssel und öffnet Firewall für nachkommenden Verbindungsaufbau
- Option-Keyed TCP: Schlüssel ist in einem IP oder TCP Option Feld.

Port-Knocking

- Idee von Martin Krzywinski [Krz03a, Krz03b]
- Eine Variante von Spread-Spectrum TCP aus [BHI+02]
- Sequenz von Paketen auf Ports an einem Server
- Verbindungsversuche landen in einer Log-Datei
- Script sucht nach definierten Sequenzen im Log
- ... und „öffnet die Tür" bei erkannter Sequenz
- Viiiele Implementierungen [0ld04, Mee04, Vin04, War04] ...

Nachteile von einfachem Port-Knocking

- Replay Angriffe möglich
- Replay öffnet auch für andere IP-Adresse (keine Authentication!)
- Große Firewall-Logs: Kein rate-limit möglich
- Spezielles Client-Side Programm nötig um Knocking-Sequenz zu erzeugen

Verbesserungen von Port-Knocking

- Authentifizierung durch Kryptographie: trapdoor2 [KW05] oder Firewall Knock Operator [Ras04]
- One-Time Schlüssel in CÖK [Wor04]
- Eigener Server: Benötigt keine grossen Logs
  - Listen-only UDP
  - Packet capturing

Cryptographic One-Time Knocking: CÖK

CÖK [Wor04] verwendet schon viele diese Methoden

- One-Time Schlüssel
- Packet capturing
- braucht aber speziellen Client
- Eigene Kommandos zum Öffnen und Schliessen der Firewall

Nachteile von „verbessertem“ Port-Knocking

- Große Firewall-Logs: Kein rate-limit möglich bei Logfile-basierten Methoden
- Spezielles Client-Side Programm nötig um Knocking-Sequenz zu erzeugen
- Einige kryptographische Methoden haben neue Nachteile:
  - Heavy-Weight: Denial of Service möglich
  - Single Point of Failure: SSH und andere kryptographische Methoden verwenden die gleichen Libraries.

Application Piggy-Backing

- Beiläufig erwähnt von CÖK-Autor David Worth [Wor04]
- Versteckter Authentication Server verwendet Packet Capturing (libpcap oder Kernel-Level Socketfilter)
- Lauscht auf bekannten Ports auf welchen auch ein Service laufen kann
- Alternativ: UDP Listen-only server
- Checkt Pakete auf einen Authentifizierungsversuch

Application Piggy-Backing

- Verwendung von one-time passwords (OTP): RFC 2289 S/Key oder OPIE
- Beispiele:
  - TFTP-Protokoll: Im Dateinamen wird OTP übertragen
  - Domain-Name Service: Übertragen des OTP im Request

Setzen wir das Puzzle zusammen

- Light-Weight One time passwords
- Kein spezielles Client-Side Programm sondern
  - DNS-Client (host, nslookup, Web-Browser (!))
  - TFTP Client
  - ...
- Automatisches Schliessen von Ports (expire von Filter-Regeln) mit netfilter-Modul iptree – Idee von Clifford Wolf [Wol05]

Zusammenfassung

- One-Time Port-Knocking mit Application Piggy-Backing löst beschriebene Probleme
- Einfach und lightweight
- Kein spezieller Client nötig
- kein großes Log-Aufkommen
- *Aber* kann nicht vor Man-in-the-middle Angriffen schützen – aber dafür gibt es SSH.

Literatur

[0ld04] 0ldW0lf. Toctoc. Software, 2004.

[BHI+02] Paul Barham, Steven Hand, Rebecca Isaacs, Paul Jardetzky, Richard Mortier, and Timothy Roscoe. Techniques for lightweight concealment and authentication in IP networks. Technical Report IRB-TR-02-009, Intel Research, July 2002.

[Krz03a] Martin Krzywinski. Howto: Port knocking. *Linux Journal online*, June 2003.

Literatur

[Krz03b] Martin Krzywinski. Port knocking: Network authentication across closed ports. *Sys Admin Magazine*, 12(6):12–17, June 2003.

[KW05] Andreas Krennmair and Clifford Wolf. trap-door2. Linbit open source software, 2005.

[Mee04] James Meehan. pasmal. Software, 2004.

[Ras04] Michael Rash. Combining port knocking and passive os fingerprinting with fwknop. *;login:*, 29(6):19–25, 2004.

Literatur

[Vin04] Judd Vinet. knockd. Software, 2004.

[War04] Bruce Ward. The doorman. Software, 2004.

[Wol05] Clifford Wolf. ipset type “expire”. Netfilter mailinglist posting, 2005.

[Wor04] David Worth. CÖK – cryptographic one-time knocking: Port knocking done „better“. Slides for Blackhat-US, 2004.